



# Sintassi dello script e funzioni grafiche

Qlik Sense®

November 2025

Copyright © 1993-aaaa QlikTech International AB. Tutti i diritti riservati.



---

|                                                                              |           |
|------------------------------------------------------------------------------|-----------|
| <b>1 Cos'è Qlik Sense?</b>                                                   | <b>16</b> |
| 1.1 Operazioni che è possibile eseguire in Qlik Sense                        | 16        |
| 1.2 Funzionamento di Qlik Sense                                              | 16        |
| Il modello delle app                                                         | 16        |
| L'esperienza associativa                                                     | 16        |
| Collaborazione e mobilità                                                    | 16        |
| 1.3 Come è possibile distribuire Qlik Sense?                                 | 17        |
| Qlik Sense Desktop                                                           | 17        |
| Qlik Sense Enterprise                                                        | 17        |
| 1.4 Come amministrare e gestire un sito Qlik Sense                           | 17        |
| 1.5 Estensione di Qlik Sense e adattamento in base alle esigenze dell'utente | 17        |
| Creazione di estensioni e mashup                                             | 17        |
| Creazione di client                                                          | 17        |
| Creazione di strumenti server                                                | 17        |
| Connessione ad altre sorgenti dati                                           | 17        |
| <b>2 Panoramica sulla sintassi dello script</b>                              | <b>18</b> |
| 2.1 Introduzione alla sintassi dello script                                  | 18        |
| 2.2 Che cos'è la metasintassi Backus-Naur Form?                              | 18        |
| <b>3 Istruzioni e parole chiave dello script</b>                             | <b>20</b> |
| 3.1 Istruzioni di controllo dello script                                     | 20        |
| Prospetto delle istruzioni di controllo dello script                         | 20        |
| Call                                                                         | 22        |
| Cicli                                                                        | 24        |
| End                                                                          | 25        |
| Exit                                                                         | 25        |
| Exit script                                                                  | 25        |
| For..next                                                                    | 25        |
| For each..next                                                               | 27        |
| If..then..elseif..else..end if                                               | 30        |
| Next                                                                         | 31        |
| Sub..end sub                                                                 | 31        |
| Switch..case..default..end switch                                            | 33        |
| To                                                                           | 34        |
| 3.2 Prefissi dello script                                                    | 34        |
| Prospetto dei prefissi dello script                                          | 34        |
| Add                                                                          | 39        |
| Buffer                                                                       | 40        |
| Concatenate                                                                  | 42        |
| Crosstable                                                                   | 47        |
| First                                                                        | 57        |
| Generic                                                                      | 59        |
| Hierarchy                                                                    | 66        |
| HierarchyBelongsTo                                                           | 68        |
| Inner                                                                        | 70        |
| IntervalMatch                                                                | 71        |
| Join                                                                         | 74        |
| Keep                                                                         | 84        |

---

|                                                        |     |
|--------------------------------------------------------|-----|
| Left .....                                             | 85  |
| Mapping .....                                          | 87  |
| Unisci .....                                           | 88  |
| NoConcatenate .....                                    | 93  |
| Only .....                                             | 102 |
| Outer .....                                            | 102 |
| Caricamento parziale .....                             | 103 |
| Replace .....                                          | 107 |
| Right .....                                            | 108 |
| Sample .....                                           | 110 |
| Semantic .....                                         | 113 |
| Unless .....                                           | 117 |
| When .....                                             | 122 |
| 3.3 Istruzioni regolari dello script .....             | 129 |
| Prospetto delle istruzioni regolari dello script ..... | 129 |
| Alias .....                                            | 136 |
| AutoNumber .....                                       | 137 |
| Binary .....                                           | 140 |
| Comment field .....                                    | 142 |
| Comment table .....                                    | 143 |
| Connect .....                                          | 143 |
| Declare .....                                          | 145 |
| Derive .....                                           | 148 |
| Direct Query .....                                     | 149 |
| Directory .....                                        | 154 |
| Disconnect .....                                       | 155 |
| Drop .....                                             | 156 |
| Drop table .....                                       | 157 |
| Execute .....                                          | 158 |
| Field/Fields .....                                     | 159 |
| FlushLog .....                                         | 160 |
| Force .....                                            | 160 |
| From .....                                             | 162 |
| Load .....                                             | 163 |
| Let .....                                              | 185 |
| Loosen Table .....                                     | 185 |
| Map .....                                              | 186 |
| NullAsNull .....                                       | 187 |
| NullAsValue .....                                      | 187 |
| Qualify .....                                          | 188 |
| Rem .....                                              | 189 |
| Rename .....                                           | 190 |
| Search .....                                           | 192 |
| Section .....                                          | 193 |
| Select .....                                           | 193 |
| Set .....                                              | 196 |
| Sleep .....                                            | 196 |
| SQL .....                                              | 196 |

---

|                                                                         |            |
|-------------------------------------------------------------------------|------------|
| SQLColumns .....                                                        | 197        |
| SQLTables .....                                                         | 198        |
| SQLTypes .....                                                          | 198        |
| Star .....                                                              | 200        |
| Store .....                                                             | 201        |
| Table/Tables .....                                                      | 209        |
| Tag .....                                                               | 209        |
| Trace .....                                                             | 210        |
| Unmap .....                                                             | 210        |
| Unqualify .....                                                         | 211        |
| Untag .....                                                             | 211        |
| 3.4 Directory di lavoro .....                                           | 212        |
| Directory di lavoro Qlik Sense Desktop .....                            | 213        |
| Directory di lavoro Qlik Sense .....                                    | 213        |
| <b>4 Utilizzo delle variabili nell'editor caricamento dati .....</b>    | <b>214</b> |
| 4.1 Panoramica .....                                                    | 214        |
| 4.2 Definizione di una variabile .....                                  | 214        |
| Denominazione delle variabili .....                                     | 215        |
| 4.3 Eliminazione di una variabile .....                                 | 215        |
| 4.4 Caricamento di un valore della variabile come valore di campo ..... | 215        |
| 4.5 Calcolo della variabile .....                                       | 215        |
| 4.6 Variabili di sistema .....                                          | 216        |
| Prospetto delle variabili di sistema .....                              | 217        |
| CreateSearchIndexOnReload .....                                         | 219        |
| HidePrefix .....                                                        | 220        |
| HideSuffix .....                                                        | 220        |
| Include .....                                                           | 221        |
| OpenUrlTimeout .....                                                    | 222        |
| StripComments .....                                                     | 222        |
| Verbatim .....                                                          | 223        |
| 4.7 Variabili di gestione del valore .....                              | 223        |
| Prospetto delle variabili di gestione del valore .....                  | 223        |
| NullDisplay .....                                                       | 224        |
| NullInterpret .....                                                     | 224        |
| NullValue .....                                                         | 224        |
| OtherSymbol .....                                                       | 224        |
| 4.8 Variabili di interpretazione numerica .....                         | 225        |
| Formattazione valuta .....                                              | 225        |
| Formattazione numero .....                                              | 226        |
| Formattazione dell'ora .....                                            | 226        |
| BrokenWeeks .....                                                       | 227        |
| DateFormat .....                                                        | 229        |
| DayNames .....                                                          | 235        |
| DecimalSep .....                                                        | 239        |
| FirstWeekDay .....                                                      | 242        |
| LongDayNames .....                                                      | 246        |
| LongMonthNames .....                                                    | 249        |

---

|                                                                                                                 |            |
|-----------------------------------------------------------------------------------------------------------------|------------|
| MoneyDecimalSep .....                                                                                           | 253        |
| MoneyFormat .....                                                                                               | 257        |
| MoneyThousandSep .....                                                                                          | 261        |
| MonthNames .....                                                                                                | 265        |
| NumericalAbbreviation .....                                                                                     | 271        |
| ReferenceDay .....                                                                                              | 272        |
| ThousandSep .....                                                                                               | 277        |
| TimeFormat .....                                                                                                | 283        |
| TimestampFormat .....                                                                                           | 283        |
| 4.9 Variabili di Direct Discovery .....                                                                         | 286        |
| Variabili di sistema di Direct Discovery .....                                                                  | 286        |
| Variabili di unione di query Teradata .....                                                                     | 288        |
| Direct DiscoveryVariabili di carattere .....                                                                    | 288        |
| Variabili di interpretazione numerica Direct Discovery .....                                                    | 289        |
| 4.10 Variabili di errore .....                                                                                  | 290        |
| Prospetto delle variabili di errore .....                                                                       | 290        |
| ErrorMode .....                                                                                                 | 291        |
| ScriptError .....                                                                                               | 291        |
| ScriptErrorCount .....                                                                                          | 293        |
| ScriptErrorList .....                                                                                           | 293        |
| <b>5 Espressioni nello script .....</b>                                                                         | <b>294</b> |
| <b>6 Espressioni del grafico .....</b>                                                                          | <b>295</b> |
| 6.1 Definizione dell'ambito di aggregazione .....                                                               | 295        |
| Esempio: Qualificatore TOTAL .....                                                                              | 296        |
| Esempio: analisi di gruppo .....                                                                                | 297        |
| Esempio: qualificatore TOTAL e analisi di gruppo (identificatore {1}) .....                                     | 297        |
| Esempio: Qualificatore TOTAL e analisi di gruppo (confronto dei dati rispetto a un valore di riferimento) ..... | 298        |
| Esempio: Qualificatore TOTAL con eccezione e analisi di gruppo .....                                            | 299        |
| Dati utilizzati negli esempi di questa pagina: .....                                                            | 300        |
| 6.2 Analisi di gruppo .....                                                                                     | 300        |
| Espressioni set .....                                                                                           | 300        |
| Esempi .....                                                                                                    | 302        |
| Set naturali .....                                                                                              | 302        |
| Identificatori set .....                                                                                        | 304        |
| Operatori set .....                                                                                             | 305        |
| Modificatori set .....                                                                                          | 306        |
| Flag di conservazione del set vuoto .....                                                                       | 329        |
| Espressioni set interne ed esterne .....                                                                        | 331        |
| Espressioni set esterne ed espansioni con il simbolo del dollaro .....                                          | 335        |
| Tutorial - Creazione di un'espressione set .....                                                                | 338        |
| Sintassi per le espressioni set .....                                                                           | 347        |
| 6.3 Sintassi generale per le espressioni grafiche .....                                                         | 347        |
| 6.4 Sintassi generale per le aggregazioni .....                                                                 | 348        |
| <b>7 Operatori .....</b>                                                                                        | <b>349</b> |
| 7.1 Operatori bit a bit .....                                                                                   | 349        |
| 7.2 Operatori logici .....                                                                                      | 350        |

|                                                                                          |            |
|------------------------------------------------------------------------------------------|------------|
| 7.3 Operatori numerici .....                                                             | 350        |
| 7.4 Operatori relazionali .....                                                          | 351        |
| 7.5 Operatori su stringa .....                                                           | 353        |
| & .....                                                                                  | 353        |
| like .....                                                                               | 353        |
| <b>8 Funzioni per script e grafici .....</b>                                             | <b>354</b> |
| 8.1 Connessioni di analisi per estensioni lato server (SSE, Server-Side Extension) ..... | 354        |
| 8.2 Funzioni di aggregazione .....                                                       | 354        |
| Utilizzo delle funzioni di aggregazione in uno script di caricamento dei dati .....      | 355        |
| Utilizzo delle funzioni di aggregazione nelle espressioni grafiche .....                 | 355        |
| Come vengono calcolate le aggregazioni .....                                             | 355        |
| Aggregazione dei campi chiave .....                                                      | 355        |
| Funzioni di aggregazione di base .....                                                   | 357        |
| Funzioni di aggregazione contatore .....                                                 | 393        |
| Funzioni di aggregazione finanziaria .....                                               | 421        |
| Funzioni di aggregazione statistica .....                                                | 450        |
| Funzioni di test statistici .....                                                        | 521        |
| Funzioni di aggregazione delle stringhe .....                                            | 589        |
| Funzioni di dimensione sintetica .....                                                   | 602        |
| Aggregazioni nidificate .....                                                            | 605        |
| 8.3 Aggr - funzione per grafici .....                                                    | 605        |
| Esempi: espressioni del grafico che utilizzano Aggr .....                                | 608        |
| 8.4 Funzioni colore .....                                                                | 612        |
| Funzioni colori predefiniti .....                                                        | 614        |
| ARGB .....                                                                               | 615        |
| RGB .....                                                                                | 620        |
| HSL .....                                                                                | 626        |
| 8.5 Funzioni condizionali .....                                                          | 631        |
| Panoramica sulle funzioni condizionali .....                                             | 631        |
| alt .....                                                                                | 632        |
| class .....                                                                              | 640        |
| coalesce .....                                                                           | 645        |
| if .....                                                                                 | 649        |
| match .....                                                                              | 656        |
| mixmatch .....                                                                           | 664        |
| pick .....                                                                               | 672        |
| wildmatch .....                                                                          | 677        |
| 8.6 Funzioni di conteggio .....                                                          | 685        |
| Prospetto delle funzioni di conteggio .....                                              | 685        |
| autonumber .....                                                                         | 686        |
| autonumberhash128 .....                                                                  | 688        |
| autonumberhash256 .....                                                                  | 690        |
| IterNo .....                                                                             | 692        |
| RecNo .....                                                                              | 693        |
| RowNo .....                                                                              | 695        |
| RowNo - funzione per grafici .....                                                       | 696        |
| 8.7 Funzioni data e ora .....                                                            | 698        |

---

|                                           |      |
|-------------------------------------------|------|
| Prospetto delle funzioni data e ora ..... | 699  |
| addmonths .....                           | 708  |
| addyears .....                            | 717  |
| age .....                                 | 725  |
| converttolocaltime .....                  | 727  |
| day .....                                 | 730  |
| dayend .....                              | 737  |
| daylightsaving .....                      | 745  |
| dayname .....                             | 745  |
| daynumberofquarter .....                  | 747  |
| daynumberofyear .....                     | 754  |
| daystart .....                            | 760  |
| firstworkdate .....                       | 768  |
| GMT .....                                 | 769  |
| hour .....                                | 773  |
| inday .....                               | 777  |
| indaytotime .....                         | 785  |
| inlunarweek .....                         | 795  |
| inlunarweektodate .....                   | 807  |
| inmonth .....                             | 819  |
| inmonths .....                            | 827  |
| inmonthstodate .....                      | 841  |
| inmonthtodate .....                       | 854  |
| inquarter .....                           | 864  |
| inquartertodate .....                     | 877  |
| inweek .....                              | 890  |
| inweektodate .....                        | 907  |
| inyear .....                              | 921  |
| inyeartodate .....                        | 934  |
| lastworkdate .....                        | 947  |
| localtime .....                           | 956  |
| lunarweekend .....                        | 960  |
| lunarweekname .....                       | 972  |
| lunarweekstart .....                      | 984  |
| makedate .....                            | 996  |
| maketime .....                            | 1003 |
| makeweekdate .....                        | 1010 |
| minute .....                              | 1019 |
| month .....                               | 1024 |
| monthend .....                            | 1031 |
| monthname .....                           | 1040 |
| monthsend .....                           | 1048 |
| monthsname .....                          | 1061 |
| monthsstart .....                         | 1075 |
| monthstart .....                          | 1088 |
| networkdays .....                         | 1097 |
| now .....                                 | 1107 |
| quarterend .....                          | 1114 |

---

|                                                   |      |
|---------------------------------------------------|------|
| quartername .....                                 | 1128 |
| quarterstart .....                                | 1140 |
| second .....                                      | 1151 |
| setdateyear .....                                 | 1157 |
| setdateyearmonth .....                            | 1159 |
| timezone .....                                    | 1161 |
| today .....                                       | 1161 |
| UTC .....                                         | 1167 |
| week .....                                        | 1167 |
| weekday .....                                     | 1184 |
| weekend .....                                     | 1193 |
| weekname .....                                    | 1205 |
| weekstart .....                                   | 1220 |
| weekyear .....                                    | 1232 |
| year .....                                        | 1242 |
| yearend .....                                     | 1249 |
| yearname .....                                    | 1261 |
| yearstart .....                                   | 1274 |
| yeartodate .....                                  | 1286 |
| 8.8 Funzioni esponenziali e logaritmiche .....    | 1301 |
| 8.9 Funzioni di campo .....                       | 1302 |
| Funzioni di conteggio .....                       | 1303 |
| Funzioni di campo e di selezione .....            | 1303 |
| GetAlternativeCount - funzione per grafici .....  | 1304 |
| GetCurrentSelections - funzione per grafici ..... | 1310 |
| GetExcludedCount - funzione per grafici .....     | 1315 |
| GetFieldSelections - funzione per grafici .....   | 1320 |
| GetObjectDimension - funzione per grafici .....   | 1324 |
| GetObjectField - funzione per grafici .....       | 1328 |
| GetObjectMeasure - funzione per grafici .....     | 1332 |
| GetPossibleCount - funzione per grafici .....     | 1334 |
| GetSelectedCount - funzione per grafici .....     | 1338 |
| GetStateCounts - funzione per grafici .....       | 1343 |
| 8.10 Funzioni di file .....                       | 1348 |
| Prospetto delle funzioni di file .....            | 1348 |
| Attribute .....                                   | 1350 |
| ConnectionString .....                            | 1358 |
| FileName .....                                    | 1359 |
| FileDir .....                                     | 1359 |
| FileExtension .....                               | 1360 |
| FileName .....                                    | 1360 |
| FilePath .....                                    | 1360 |
| FileSize .....                                    | 1361 |
| FileTime .....                                    | 1362 |
| GetFolderPath .....                               | 1363 |
| QvdCreateTime .....                               | 1364 |
| QvdFieldName .....                                | 1365 |
| QvdNoOfFields .....                               | 1365 |

---

|                                                     |      |
|-----------------------------------------------------|------|
| QvdNoOfRecords .....                                | 1366 |
| QvdTableName .....                                  | 1367 |
| 8.11 Funzioni finanziarie .....                     | 1368 |
| Panoramica sulle funzioni finanziarie .....         | 1369 |
| BlackAndSchole .....                                | 1369 |
| FV .....                                            | 1370 |
| nPer .....                                          | 1371 |
| Pmt .....                                           | 1372 |
| PV .....                                            | 1373 |
| Rate .....                                          | 1374 |
| 8.12 Funzioni di formattazione .....                | 1375 |
| Panoramica sulle funzioni di formattazione .....    | 1375 |
| ApplyCodepage .....                                 | 1376 |
| Date .....                                          | 1378 |
| Dual .....                                          | 1381 |
| Interval .....                                      | 1387 |
| Money .....                                         | 1390 |
| Num .....                                           | 1395 |
| Time .....                                          | 1401 |
| Timestamp .....                                     | 1405 |
| 8.13 Funzioni numeriche generiche .....             | 1410 |
| Panoramica delle funzioni numeriche generiche ..... | 1410 |
| Funzioni di combinazione e permutazione .....       | 1411 |
| Funzioni modulo .....                               | 1412 |
| Funzioni di parità .....                            | 1412 |
| Funzioni di arrotondamento .....                    | 1412 |
| BitCount .....                                      | 1413 |
| Ceil .....                                          | 1413 |
| Combin .....                                        | 1414 |
| Div .....                                           | 1415 |
| Even .....                                          | 1415 |
| Fabs .....                                          | 1416 |
| Fact .....                                          | 1416 |
| Floor .....                                         | 1417 |
| Fmod .....                                          | 1418 |
| Frac .....                                          | 1419 |
| Mod .....                                           | 1420 |
| Odd .....                                           | 1420 |
| Permut .....                                        | 1421 |
| Round .....                                         | 1421 |
| Sign .....                                          | 1423 |
| 8.14 Funzioni geospaziali .....                     | 1423 |
| Panoramica delle funzioni geospaziali .....         | 1424 |
| GeoAggrGeometry .....                               | 1425 |
| GeoBoundingBox .....                                | 1426 |
| GeoCountVertex .....                                | 1427 |
| GeoGetBoundingBox .....                             | 1427 |
| GeoGetPolygonCenter .....                           | 1428 |

---

---

|                                                                  |      |
|------------------------------------------------------------------|------|
| GeoInvProjectGeometry .....                                      | 1429 |
| GeoMakePoint .....                                               | 1429 |
| GeoProject .....                                                 | 1430 |
| GeoProjectGeometry .....                                         | 1431 |
| GeoReduceGeometry .....                                          | 1431 |
| 8.15 Funzioni di interpretazione .....                           | 1433 |
| Prospetto delle funzioni di interpretazione .....                | 1433 |
| Date# .....                                                      | 1434 |
| Interval# .....                                                  | 1438 |
| Money# .....                                                     | 1441 |
| Num# .....                                                       | 1446 |
| Text .....                                                       | 1450 |
| Time# .....                                                      | 1453 |
| Timestamp# .....                                                 | 1457 |
| 8.16 Funzioni intra-record .....                                 | 1461 |
| Funzioni di riga .....                                           | 1461 |
| Funzioni di colonna .....                                        | 1462 |
| Funzioni di campo .....                                          | 1462 |
| Funzioni tabella pivot .....                                     | 1463 |
| Funzioni intra-record nello script di caricamento dei dati ..... | 1464 |
| Above - funzione per grafici .....                               | 1464 |
| Below - funzione per grafici .....                               | 1478 |
| Bottom - funzione per grafici .....                              | 1488 |
| Column - funzione per grafici .....                              | 1495 |
| Dimensionality - funzione per grafici .....                      | 1500 |
| Exists .....                                                     | 1504 |
| FieldIndex .....                                                 | 1508 |
| FieldValue .....                                                 | 1510 |
| FieldValueCount .....                                            | 1511 |
| LookUp .....                                                     | 1513 |
| NoOfRows .....                                                   | 1515 |
| Peek .....                                                       | 1517 |
| Previous .....                                                   | 1525 |
| Top - funzione per grafici .....                                 | 1526 |
| SecondaryDimensionality - funzione per grafici .....             | 1533 |
| After - funzione per grafici .....                               | 1537 |
| Before - funzione per grafici .....                              | 1542 |
| First - funzione per grafici .....                               | 1546 |
| Last - funzione per grafici .....                                | 1552 |
| ColumnNo - funzione per grafici .....                            | 1558 |
| NoOfColumns - funzione per grafici .....                         | 1562 |
| 8.17 Funzioni logiche .....                                      | 1565 |
| 8.18 Funzioni di mapping .....                                   | 1566 |
| Panoramica sulle funzioni di mapping .....                       | 1566 |
| ApplyMap .....                                                   | 1567 |
| MapSubstring .....                                               | 1568 |
| 8.19 Funzioni matematiche .....                                  | 1570 |
| 8.20 Funzioni di NULL .....                                      | 1571 |

---

---

|                                                                    |      |
|--------------------------------------------------------------------|------|
| Panoramica sulle funzioni di NULL .....                            | 1571 |
| EmptyIsNull .....                                                  | 1571 |
| IsNull .....                                                       | 1576 |
| NULL .....                                                         | 1580 |
| 8.21 Funzioni di scala .....                                       | 1584 |
| Funzioni di scala di base .....                                    | 1584 |
| Funzioni di scala di conteggio .....                               | 1585 |
| Funzioni di scala statistiche .....                                | 1585 |
| Funzioni di scala finanziarie .....                                | 1586 |
| RangeAvg .....                                                     | 1587 |
| RangeCorrel .....                                                  | 1589 |
| RangeCount .....                                                   | 1591 |
| RangeFractile .....                                                | 1594 |
| RangeIRR .....                                                     | 1596 |
| RangeKurtosis .....                                                | 1597 |
| RangeMax .....                                                     | 1598 |
| RangeMaxString .....                                               | 1600 |
| RangeMin .....                                                     | 1602 |
| RangeMinString .....                                               | 1604 |
| RangeMissingCount .....                                            | 1605 |
| RangeMode .....                                                    | 1607 |
| RangeNPV .....                                                     | 1609 |
| RangeNullCount .....                                               | 1610 |
| RangeNumericCount .....                                            | 1612 |
| RangeOnly .....                                                    | 1613 |
| RangeSkew .....                                                    | 1614 |
| RangeStdev .....                                                   | 1615 |
| RangeSum .....                                                     | 1617 |
| RangeTextCount .....                                               | 1619 |
| RangeXIRR .....                                                    | 1621 |
| RangeXNPV .....                                                    | 1622 |
| 8.22 Funzioni relazionali .....                                    | 1624 |
| Funzioni di classificazione .....                                  | 1625 |
| Funzioni di raggruppamento .....                                   | 1625 |
| Funzioni di scomposizione serie temporale .....                    | 1626 |
| Rank - funzione per grafici .....                                  | 1627 |
| HRank - funzione per grafici .....                                 | 1632 |
| "Ottimizzazione con k-means: un esempio del mondo reale" .....     | 1634 |
| KMeans2D - funzione per grafici .....                              | 1643 |
| KMeansND - funzione per grafici .....                              | 1658 |
| KMeansCentroid2D - funzione per grafici .....                      | 1673 |
| KMeansCentroidND - funzione per grafici .....                      | 1674 |
| STL_Trend - funzione per grafici .....                             | 1675 |
| STL_Seasonal - funzione per grafici .....                          | 1677 |
| STL_Residual - funzione per grafici .....                          | 1679 |
| Tutorial - Scomposizione delle serie temporali in Qlik Sense ..... | 1681 |
| 8.23 Funzioni di distribuzione statistica .....                    | 1686 |
| Panoramica sulle funzioni di distribuzione statistica .....        | 1686 |

---

|                                                                                                          |      |
|----------------------------------------------------------------------------------------------------------|------|
| BetaDensity .....                                                                                        | 1689 |
| BetaDist .....                                                                                           | 1689 |
| BetaInv .....                                                                                            | 1689 |
| BinomDist .....                                                                                          | 1690 |
| BinomFrequency .....                                                                                     | 1690 |
| BinomInv .....                                                                                           | 1691 |
| ChiDensity .....                                                                                         | 1691 |
| ChiDist .....                                                                                            | 1692 |
| ChiInv .....                                                                                             | 1692 |
| FDensity .....                                                                                           | 1693 |
| FDist .....                                                                                              | 1693 |
| FInv .....                                                                                               | 1694 |
| GammaDensity .....                                                                                       | 1695 |
| GammaDist .....                                                                                          | 1695 |
| Gammalnv .....                                                                                           | 1696 |
| NormDist .....                                                                                           | 1696 |
| NormInv .....                                                                                            | 1697 |
| PoissonDist .....                                                                                        | 1698 |
| PoissonFrequency .....                                                                                   | 1698 |
| PoissonInv .....                                                                                         | 1698 |
| TDensity .....                                                                                           | 1699 |
| TDist .....                                                                                              | 1699 |
| TInv .....                                                                                               | 1700 |
| 8.24 Funzioni di stringa .....                                                                           | 1701 |
| Panoramica sulle funzioni di stringa .....                                                               | 1701 |
| Utilizzo delle espressioni regolari negli script di caricamento e nelle espressioni dei<br>grafici ..... | 1706 |
| Capitalize .....                                                                                         | 1708 |
| Chr .....                                                                                                | 1711 |
| CountRegEx .....                                                                                         | 1713 |
| Evaluate .....                                                                                           | 1719 |
| ExtractRegEx .....                                                                                       | 1722 |
| ExtractRegExGroup .....                                                                                  | 1728 |
| FindOneOf .....                                                                                          | 1738 |
| Hash128 .....                                                                                            | 1742 |
| Hash160 .....                                                                                            | 1746 |
| Hash256 .....                                                                                            | 1751 |
| Index .....                                                                                              | 1756 |
| IndexRegEx .....                                                                                         | 1760 |
| IndexRegExGroup .....                                                                                    | 1763 |
| IsJson .....                                                                                             | 1769 |
| IsRegEx .....                                                                                            | 1772 |
| JsonArray .....                                                                                          | 1776 |
| JsonGet .....                                                                                            | 1777 |
| JsonObject .....                                                                                         | 1779 |
| JsonSet .....                                                                                            | 1780 |
| JsonSetEx .....                                                                                          | 1782 |
| KeepChar .....                                                                                           | 1783 |

|                                                                                             |             |
|---------------------------------------------------------------------------------------------|-------------|
| Left .....                                                                                  | 1786        |
| Len .....                                                                                   | 1791        |
| LevenshteinDist .....                                                                       | 1795        |
| Lower .....                                                                                 | 1799        |
| LTrim .....                                                                                 | 1802        |
| MatchRegEx .....                                                                            | 1806        |
| Mid .....                                                                                   | 1810        |
| Ord .....                                                                                   | 1814        |
| PurgeChar .....                                                                             | 1817        |
| Repeat .....                                                                                | 1820        |
| Replace .....                                                                               | 1824        |
| ReplaceRegEx .....                                                                          | 1827        |
| ReplaceRegExGroup .....                                                                     | 1833        |
| Right .....                                                                                 | 1839        |
| RTrim .....                                                                                 | 1842        |
| SubField .....                                                                              | 1846        |
| SubFieldRegEx .....                                                                         | 1851        |
| SubStringCount .....                                                                        | 1855        |
| TextBetween .....                                                                           | 1861        |
| Trim .....                                                                                  | 1865        |
| Upper .....                                                                                 | 1869        |
| 8.25 Funzioni di sistema .....                                                              | 1873        |
| Prospetto delle funzioni di sistema .....                                                   | 1873        |
| CalcDim - funzione per grafici .....                                                        | 1876        |
| EngineVersion .....                                                                         | 1882        |
| Funzioni di sistema .....                                                                   | 1883        |
| GetSysAttr .....                                                                            | 1886        |
| GetUserAttr .....                                                                           | 1887        |
| InObject - funzione per grafici .....                                                       | 1887        |
| IsPartialReload .....                                                                       | 1892        |
| ObjectId - funzione per grafici .....                                                       | 1892        |
| ProductVersion .....                                                                        | 1895        |
| StateName - funzione per grafici .....                                                      | 1896        |
| 8.26 Funzioni di tabella .....                                                              | 1896        |
| Panoramica sulle funzioni di tabella .....                                                  | 1896        |
| FieldName .....                                                                             | 1898        |
| FieldNumber .....                                                                           | 1899        |
| NoOfFields .....                                                                            | 1899        |
| NoOfRows .....                                                                              | 1900        |
| 8.27 Funzioni trigonometriche e iperboliche .....                                           | 1900        |
| 8.28 Funzioni Window .....                                                                  | 1902        |
| Window .....                                                                                | 1903        |
| WRank .....                                                                                 | 1912        |
| <b>9 Restrizione dell'accesso al file system .....</b>                                      | <b>1919</b> |
| 9.1 Aspetti relativi alla sicurezza quando si effettua la connessione alle connessioni dati |             |
| ODBC e OLE DB basate su file .....                                                          | 1919        |
| 9.2 Limitazioni nella modalità standard .....                                               | 1919        |

---

|                                                                                |             |
|--------------------------------------------------------------------------------|-------------|
| Variabili di sistema .....                                                     | 1920        |
| Istruzioni di script regolari .....                                            | 1921        |
| Istruzioni di controllo dello script .....                                     | 1923        |
| Funzioni file .....                                                            | 1923        |
| Funzioni di sistema .....                                                      | 1925        |
| 9.3 Disabilitazione della modalità standard .....                              | 1926        |
| Qlik Sense .....                                                               | 1926        |
| Qlik Sense Desktop .....                                                       | 1926        |
| <b>10 Scripting a livello di grafico .....</b>                                 | <b>1927</b> |
| 10.1 Istruzione di controllo .....                                             | 1927        |
| Panoramica istruzioni di controllo modificatore grafico .....                  | 1927        |
| Call .....                                                                     | 1929        |
| Do..ciclo .....                                                                | 1930        |
| End .....                                                                      | 1931        |
| Exit .....                                                                     | 1931        |
| Exit script .....                                                              | 1931        |
| For..next .....                                                                | 1932        |
| For each..next .....                                                           | 1933        |
| If..then..elseif..else..end if .....                                           | 1936        |
| Next .....                                                                     | 1937        |
| Sub..end sub .....                                                             | 1937        |
| Switch..case..default..end switch .....                                        | 1939        |
| To .....                                                                       | 1940        |
| 10.2 Prefissi .....                                                            | 1940        |
| Panoramica prefissi modificatore grafico .....                                 | 1940        |
| Add .....                                                                      | 1940        |
| Replace .....                                                                  | 1941        |
| 10.3 Istruzioni regolari .....                                                 | 1941        |
| Panoramica istruzioni regolari modificatore grafico .....                      | 1941        |
| Load .....                                                                     | 1942        |
| Let .....                                                                      | 1948        |
| Set .....                                                                      | 1949        |
| Put .....                                                                      | 1949        |
| HCValue .....                                                                  | 1950        |
| <b>11 Funzioni e istruzioni di QlikView non supportate in Qlik Sense .....</b> | <b>1952</b> |
| 11.1 Istruzioni di script non supportate in Qlik Sense .....                   | 1952        |
| 11.2 Funzioni non supportate in Qlik Sense .....                               | 1952        |
| 11.3 Prefissi non supportati in Qlik Sense .....                               | 1952        |
| <b>12 Funzioni e istruzioni non consigliate in Qlik Sense .....</b>            | <b>1953</b> |
| 12.1 Istruzioni di script non consigliate in Qlik Sense .....                  | 1953        |
| 12.2 Parametri dell'istruzione di script non consigliati in Qlik Sense .....   | 1953        |
| 12.3 Funzioni non consigliate in Qlik Sense .....                              | 1955        |
| Qualificatore ALL .....                                                        | 1955        |

---

# 1 Cos'è Qlik Sense?

Qlik Sense è una piattaforma per l'analisi dei dati. Con Qlik Sense è possibile analizzare i dati ed effettuare rilevazioni dati per proprio conto. È quindi possibile condividere conoscenze e analizzare i dati in gruppi e tra più organizzazioni. Qlik Sense consente di porre domande e trovare risposte oltre che seguire i propri percorsi personali per giungere alle proprie conclusioni. Qlik Sense consente agli utenti e ai loro colleghi di prendere decisioni in modo collaborativo.

## 1.1 Operazioni che è possibile eseguire in Qlik Sense

La maggior parte dei prodotti di BI (Business Intelligence) consente di rispondere a domande che possono essere formulate in anticipo. Ma come è possibile gestire le domande di follow-up, ossia quelle che possono sorgere dopo che un utente ha letto un report o ha esaminato una visualizzazione? Grazie all'esperienza associativa di Qlik Sense, è possibile rispondere a una domanda dopo l'altra completando i propri percorsi personali per giungere alle proprie conclusioni. Con Qlik Sense, è possibile esplorare liberamente i dati con pochi clic, apprendere in ogni fase del processo e individuare i passi successivi in base a quanto rilevato in precedenza.

## 1.2 Funzionamento di Qlik Sense

Qlik Sense genera immediatamente viste di informazioni per l'utente. Qlik Sense non richiede report statici e predefiniti, né impone la dipendenza da altri utenti: è sufficiente un clic per apprendere. Ogni volta che si fa clic, Qlik Sense risponde immediatamente, aggiornando ogni visualizzazione e vista di Qlik Sense nell'app con serie di dati appena calcolati e visualizzazioni specifiche per le proprie selezioni.

### Il modello delle app

Anziché distribuire e gestire applicazioni aziendali complesse, è possibile creare le proprie app Qlik Sense che sarà possibile riutilizzare, modificare e condividere con altri utenti. Il modello app consente di chiedere e rispondere da soli alla domanda successiva senza la necessità di ricorrere a un esperto per creare un nuovo report o una nuova visualizzazione.

### L'esperienza associativa

Qlik Sense gestisce in modo automatico tutte le relazioni tra i dati e presenta le informazioni mediante una visione metaforica basata sui colori **green/white/gray**. Le selezioni vengono evidenziate in verde, i dati associati vengono visualizzati in bianco e i dati esclusi (non associati) vengono visualizzati in grigio. Questo feedback immediato consente di formulare nuove domande oltre che di continuare a esplorare e ad acquisire nuove conoscenze.

### Collaborazione e mobilità

Qlik Sense consente inoltre di collaborare con i colleghi indipendentemente dal momento e dal luogo in cui si trovano. Tutte le funzionalità di Qlik Sense, compresa l'esperienza associativa e la collaborazione, sono disponibili sui dispositivi mobili. Con Qlik Sense, è possibile porre domande, quindi reperire risposte anche per le domande di follow-up, con i colleghi, indipendentemente dall'ubicazione.

### 1.3 Come è possibile distribuire Qlik Sense?

Qlik Sense può essere distribuito in due versioni, Qlik Sense Desktop e Qlik Sense Enterprise.

#### Qlik Sense Desktop

Questa è una versione per utenti singoli di facile installazione, che viene generalmente installata su un computer locale.

#### Qlik Sense Enterprise

Questa versione viene utilizzata per distribuire i siti Qlik Sense. Un sito è una raccolta di uno o più computer server connessi a un repository logico comune o a un nodo centrale.

### 1.4 Come amministrare e gestire un sito Qlik Sense

Con Console di gestione Qlik, è possibile configurare, gestire e monitorare i siti Qlik Sense in modo semplice e intuitivo. È possibile gestire licenze, regole di accesso e sicurezza, configurare nodi e connessioni di sorgenti dati oltre che sincronizzare il contenuto e gli utenti tra numerose attività e risorse.

### 1.5 Estensione di Qlik Sense e adattamento in base alle esigenze dell'utente

Qlik Sense fornisce API e SDK flessibili per consentire agli utenti di sviluppare le proprie estensioni, quindi di adattare e integrare Qlik Sense per diversi scopi tra cui:

#### Creazione di estensioni e mashup

È possibile effettuare lo sviluppo Web utilizzando JavaScript per creare estensioni che rappresentano una visualizzazione personalizzata nelle app Qlik Sense oppure utilizzare API di mashup per creare siti Web con il contenuto Qlik Sense.

#### Creazione di client

È possibile distribuire client in .NET e incorporare oggetti Qlik Sense nelle proprie applicazioni. È inoltre possibile creare client nativi in qualsiasi linguaggio di programmazione in grado di gestire la comunicazione WebSocket mediante l'utilizzo del protocollo client di Qlik Sense.

#### Creazione di strumenti server

Con le API del servizio e della directory utente, è possibile creare i propri strumenti personalizzati per amministrare e gestire i siti Qlik Sense.

#### Connessione ad altre sorgenti dati

È possibile creare connettori Qlik Sense per recuperare i dati da sorgenti dati personalizzate.

## 2 Panoramica sulla sintassi dello script

### 2.1 Introduzione alla sintassi dello script

In uno script vengono definiti il nome della sorgente dati, i nomi delle tabelle e i nomi dei campi inclusi nella logica. Vengono definiti inoltre i campi con diritti di accesso specificati nello script. Uno script è costituito da un certo numero di istruzioni che vengono eseguite consecutivamente.

La sintassi dello script e la sintassi della riga di comando di Qlik Sense vengono descritte in una notazione denominata metasintassi Backus-Naur Form o codice BNF.

Le prime righe di codice vengono generate automaticamente quando si crea un nuovo file Qlik Sense. I valori predefiniti di queste variabili di interpretazione numerica derivano dalle impostazioni internazionali del sistema operativo.

Lo script è costituito da numerose istruzioni dello script e parole chiave che vengono eseguite consecutivamente. Tutte le istruzioni dello script devono terminare con un punto e virgola, ";".

È possibile utilizzare espressioni e funzioni nelle istruzioni **LOAD** per trasformare i dati caricati.

Per i file tabella contenenti virgole, tabulazioni o punti e virgola come delimitatori, è possibile utilizzare un'istruzione **LOAD**. Per impostazione predefinita, un'istruzione **LOAD** carica tutti i campi del file.

È possibile accedere ai database generici utilizzando i connettori del database ODBC o OLE DB. In questo caso, vengono usate le istruzioni SQL standard. La sintassi SQL accettata varia a seconda dei diversi driver ODBC.

Inoltre, è possibile accedere ad altre sorgenti dati utilizzando i connettori personalizzati.

### 2.2 Che cos'è la metasintassi Backus-Naur Form?

La sintassi dello script e la sintassi della riga di comando di Qlik Sense vengono descritte in una notazione denominata metasintassi Backus-Naur Form o codice BNF.

Nella tabella indicata di seguito viene fornito un elenco di simboli utilizzati nel codice BNF, con una descrizione di come vengono interpretati:

Simboli

| Simbolo | Descrizione                                                                                   |
|---------|-----------------------------------------------------------------------------------------------|
|         | Operatore OR logico: il simbolo può essere utilizzato da entrambi i lati.                     |
| ()      | Parentesi che stabiliscono la precedenza: vengono utilizzate per strutturare la sintassi BNF. |
| []      | Parentesi quadre: gli elementi racchiusi sono opzionali.                                      |
| { }     | Parentesi graffe: gli elementi racchiusi possono essere ripetuti zero o più volte.            |

## 2 Panoramica sulla sintassi dello script

---

| Simbolo     | Descrizione                                                                                                                                                                                        |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Simbolo     | Categoria sintattica nonterminale: può essere suddivisa ulteriormente in altri simboli. Ad esempio, composizione dei simboli precedenti, altri simboli nonterminali, stringhe di testo e così via. |
| ::=         | Contrassegna l'inizio di un blocco che definisce un simbolo.                                                                                                                                       |
| <b>LOAD</b> | Simbolo terminale che consiste in una stringa di testo. Deve essere scritto nello script così come è visualizzato.                                                                                 |

Tutti i simboli terminali sono stampati con carattere **bold face**. Ad esempio, "(" va interpretata come parentesi che stabilisce una precedenza, mentre "(" va interpretata come carattere che deve apparire nello script.

### Esempio:

La descrizione dell'istruzione alias è:

```
alias fieldname as aliasname { , fieldname as aliasname }
```

Questo deve essere interpretato come la stringa testo "alias", seguita da un nome campo arbitrario, seguito dalla stringa di testo "as", seguita da un nome alias arbitrario. È possibile inserire qualsiasi numero di combinazioni aggiuntive di "fieldname as alias", separate da virgole.

Le seguenti istruzioni sono corrette:

```
alias a as first;
```

```
alias a as first, b as second;
```

```
alias a as first, b as second, c as third;
```

Le seguenti istruzioni non sono corrette:

```
alias a as first b as second;
```

```
alias a as first { , b as second };
```

### 3 Istruzioni e parole chiave dello script

Lo script di Qlik Sense è costituito da una serie di istruzioni. Un'istruzione può essere un'istruzione di script regolare o un'istruzione di controllo dello script. Alcune istruzioni possono essere precedute da prefissi.

Le istruzioni regolari vengono generalmente utilizzate per la manipolazione dei dati. Queste istruzioni possono essere scritte su un qualsiasi numero di righe nello script e devono sempre terminare con un punto e virgola, ";".

In genere, le istruzioni di controllo vengono utilizzate per controllare il flusso di esecuzione dello script. Ogni clausola di un'istruzione di controllo deve essere mantenuta in una singola riga dello script e può terminare con un punto e virgola oppure con un fine riga.

I prefissi possono essere applicati alle istruzioni regolari pertinenti, ma mai a istruzioni di controllo. I prefissi **when** e **unless** possono comunque essere utilizzati come suffissi per alcune specifiche clausole di istruzioni di controllo.

Nel seguente sottocapitolo, è riportato un elenco in ordine alfabetico di tutte le istruzioni di script, le istruzioni di controllo e di tutti i prefissi.

Tutte le parole chiave dello script possono essere immesse con qualsiasi combinazione di caratteri maiuscoli e minuscoli. I nomi dei campi e delle variabili utilizzati nelle istruzioni possono essere immessi indipendentemente dal formato del carattere.

#### 3.1 Istruzioni di controllo dello script

Lo script di Qlik Sense è costituito da una serie di istruzioni. Un'istruzione può essere un'istruzione di script regolare o un'istruzione di controllo dello script.

In genere, le istruzioni di controllo vengono utilizzate per controllare il flusso di esecuzione dello script. Ogni clausola di un'istruzione di controllo deve essere inserita in una singola riga nello script e può terminare con un punto e virgola o con un carattere di fine riga.

I prefissi non vengono mai utilizzati nelle istruzioni di controllo, con l'eccezione dei prefissi **when** e **unless**, che possono essere utilizzati con alcune istruzioni di controllo specifiche.

Tutte le parole chiave dello script possono essere immesse con qualsiasi combinazione di caratteri maiuscoli e minuscoli.

#### Prospetto delle istruzioni di controllo dello script

Ciascuna funzione viene descritta ulteriormente dopo la panoramica. È inoltre possibile fare clic sul nome della funzione nella sintassi per accedere immediatamente ai dettagli per tale funzione specifica.

##### Call

L'istruzione di controllo **call** consente di chiamare una subroutine che deve essere definita da un'istruzione **sub** precedente.

```
Call name ( [ paramlist ] )
```

### Do..loop

L'istruzione di controllo **do..loop** è un costrutto per la ripetizione di script che esegue una o più istruzioni finché non incontra una condizione logica.

```
Cicli [ ( while | until ) condition ] [statements]  
[exit do [ ( when | unless ) condition ] [statements]  
loop [ ( while | until ) condition ]
```

### Exit script

Questa istruzione di controllo interrompe l'esecuzione dello script. Può essere inserita in un punto qualsiasi dello script.

```
Exit script [ (when | unless) condition ]
```

### For each ..next

L'istruzione di controllo **for each..next** è un costrutto per la ripetizione di script che esegue una o più istruzioni per ogni valore in un elenco le cui voci sono separate da virgole. Le istruzioni incluse nel ciclo fra **for** e **next** verranno eseguite per ogni valore nell'elenco.

```
For each..next var in list
```

```
[statements]
```

```
[exit for [ ( when | unless ) condition ]
```

```
[statements]
```

```
next [var]
```

### For..next

L'istruzione di controllo **for..next** è un costrutto per la ripetizione di script con un contatore. Le istruzioni all'interno del ciclo incluso tra **for** e **next** verranno eseguite per ogni valore del contatore in base ai limiti inferiore e superiore specificati.

```
For..next counter = expr1 to expr2 [ stepexpr3 ]
```

```
[statements]
```

```
[exit for [ ( when | unless ) condition ]
```

```
[statements]
```

```
Next [counter]
```

### If..then

L'istruzione di controllo **if..then** è un costrutto per la selezione di script che forza l'esecuzione dello script su percorsi diversi in base a una o più condizioni logiche.



Poiché **if..then** è un'istruzione di controllo e come tale termina con un punto e virgola o con un carattere di fine riga, ciascuna delle quattro possibili clausole corrispondenti (**if..then**, **elseif..then**, **else** e **end if**) deve essere contenuta in una sola riga.

```
If..then..elseif..else..end if condition then
```

```
[ statements ]
```

```
{ elseif condition then
```

```
[ statements ] }
```

```
[ else
```

```
[ statements ] ]
```

```
end if
```

#### Sub

L'istruzione di controllo **sub..end sub** definisce una subroutine che può essere richiamata da un'istruzione **call**.

```
Sub..end sub name [ ( paramlist ) ] statements end sub
```

#### Switch

L'istruzione di controllo **switch** è un costrutto per la selezione di script che forza l'esecuzione dello script su percorsi diversi, in base al valore di un'espressione.

```
Switch..case..default..end switch expression {case valuelist [ statements ]}  
[default statements] end switch
```

#### Call

L'istruzione di controllo **call** consente di chiamare una subroutine che deve essere definita da un'istruzione **sub** precedente.

#### Sintassi:

```
Call name ( [ paramlist ] )
```

#### Argomenti:

##### Argomenti

| Argomento | Descrizione               |
|-----------|---------------------------|
| name      | Il nome della subroutine. |

### 3 Istruzioni e parole chiave dello script

| Argomento | Descrizione                                                                                                                                                                    |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| paramlist | Un elenco separato da virgole di parametri effettivi da inviare alla subroutine. Ogni voce dell'elenco può essere un nome di campo, una variabile o un'espressione arbitraria. |

La subroutine chiamata da un'istruzione **call** deve essere definita da un'istruzione **sub** rilevata precedentemente durante l'esecuzione dello script.

I parametri vengono copiati nella subroutine e, se il parametro nell'istruzione **call** è una variabile e non un'espressione, verranno copiati nuovamente all'uscita dalla subroutine.

#### Limiti:

- Poiché **call** è un'istruzione di controllo e come tale termina con un punto e virgola o con un carattere di fine riga, non deve superare un limite di riga.
- Quando si definisce una routine secondaria con `sub . . end sub` all'interno di un'istruzione di controllo, ad esempio `if . . then`, è possibile richiamare la routine secondaria solo dall'interno della stessa istruzione di controllo.

#### Esempio:

In questo esempio sono mostrati tutti i file correlati a Qlik all'interno di una cartella e delle relative sottocartelle e le informazioni dei file vengono memorizzate in una tabella. Si presuppone che sia stata creata una connessione dati alla cartella denominata Apps.

La subroutine DoDir viene chiamata con il riferimento alla cartella, 'lib://Apps', come parametro. All'interno della subroutine, è presente una chiamata ricorrente `call doDir (Dir)`, che indica alla funzione di ricercare in modo ricorrente i file nelle sottocartelle.

```
sub DoDir (Root)
  For Each Ext in 'qvw', 'qvo', 'qvs', 'qvt', 'qvd', 'qvc', 'qvf'
    For Each File in filelist (Root&'\'&Ext)
      LOAD
        '$(File)' as Name,
        FileSize( '$(File)' ) as Size,
        FileTime( '$(File)' ) as FileTime
      autogenerate 1;
    Next File
  Next Ext
  For Each Dir in dirlist (Root&'\'&*)
    call DoDir (Dir)
  Next Dir
End Sub

call DoDir ('lib://Apps')
```

### Cicli

L'istruzione di controllo **do..loop** è un costrutto per la ripetizione di script che esegue una o più istruzioni finché non incontra una condizione logica.

#### Sintassi:

```
Do [ ( while | until ) condition ] [statements]
[exit do [ ( when | unless ) condition ] [statements]
loop[ ( while | until ) condition ]
```



Poiché **do..loop** è un'istruzione di controllo che termina con un punto e virgola o con un carattere di fine riga, ciascuna delle tre possibili clausole corrispondenti (**do**, **exit do** e **loop**) deve essere contenuta in una sola riga.

#### Argomenti:

##### Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                       |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| condition     | Un'espressione logica che restituisce un valore True o False.                                                                                                                                                                                                                                                                     |
| statements    | Qualsiasi gruppo di una o più istruzioni dello script di Qlik Sense.                                                                                                                                                                                                                                                              |
| while / until | La clausola condizionale <b>while</b> o <b>until</b> deve comparire una sola volta in ciascuna istruzione <b>do..loop</b> , dopo <b>do</b> o dopo <b>loop</b> . Ogni espressione condition verrà interpretata solo al primo rilevamento, ma verrà valutata ogni volta che sarà rilevata nel ciclo.                                |
| exit do       | Se all'interno del ciclo è presente una clausola <b>exit do</b> , l'esecuzione dello script verrà trasferita alla prima istruzione dopo la clausola <b>loop</b> indicando quindi la fine del ciclo. Una clausola <b>exit do</b> può essere resa condizionale dall'utilizzo opzionale di un suffisso <b>when</b> o <b>unless</b> . |

#### Esempio:

```
// LOAD files file1.csv..file9.csv

Set a=1;

Do while a<10

LOAD * from file$(a).csv;

Let a=a+1;

Loop
```

### End

La parola chiave dello script **End** viene utilizzata per chiudere le clausole **If**, **Sub** e **Switch**.

### Exit

La parola chiave dello script **Exit** fa parte dell'istruzione **Exit Script**, ma può essere utilizzata anche per uscire dalle clausole **Do**, **For** o **Sub**.

### Exit script

Questa istruzione di controllo interrompe l'esecuzione dello script. Può essere inserita in un punto qualsiasi dello script.

#### Sintassi:

```
Exit Script [ (when | unless) condition ]
```

Poiché **exit script** è un'istruzione di controllo e come tale termina con un punto e virgola o con un carattere di fine riga, non deve superare un limite di riga.

#### Argomenti:

Argomenti

| Argomento     | Descrizione                                                                                                                        |
|---------------|------------------------------------------------------------------------------------------------------------------------------------|
| condition     | Un'espressione logica che restituisce un valore True o False.                                                                      |
| when / unless | Un'istruzione <b>exit script</b> può essere resa condizionale dall'utilizzo opzionale della clausola <b>when</b> o <b>unless</b> . |

#### Esempi:

```
//Exit script  
Exit Script;
```

```
//Exit script when a condition is fulfilled  
Exit Script when a=1
```

### For..next

L'istruzione di controllo **for..next** è un costrutto per la ripetizione di script con un contatore. Le istruzioni all'interno del ciclo incluso tra **for** e **next** verranno eseguite per ogni valore del contatore in base ai limiti inferiore e superiore specificati.

#### Sintassi:

```
For counter = expr1 to expr2 [ step expr3 ]
```

```
[statements]
```

### 3 Istruzioni e parole chiave dello script

```
[exit for [ ( when | unless ) condition ]
```

```
[statements]
```

```
Next [counter]
```

Le espressioni *expr1*, *expr2* ed *expr3* vengono valutate solo la prima volta che il ciclo viene eseguito. Il valore della variabile di conteggio può essere modificato dalle istruzioni all'interno del ciclo, tuttavia l'utilizzo di questa procedura di programmazione non è consigliato.

Se all'interno del ciclo è presente una clausola **exit for**, l'esecuzione dello script verrà trasferita alla prima istruzione dopo la clausola **next** indicando quindi la fine del ciclo. Una clausola **exit for** può essere resa condizionale dall'utilizzo opzionale di un suffisso **when** o **unless**.



*Poiché **for..next** è un'istruzione di controllo che termina con un punto e virgola o con un carattere di fine riga, ciascuna delle tre possibili clausole corrispondenti (**for..to..step**, **exit for** e **next**) deve essere contenuta in una sola riga.*

#### Argomenti:

##### Argomenti

| Argomento  | Descrizione                                                                                                                                                               |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| counter    | Un nome di variabile. Se <i>counter</i> viene specificato dopo <b>next</b> , deve avere lo stesso nome di variabile rilevato dopo l'istruzione <b>for</b> corrispondente. |
| expr1      | Un'espressione che determina il primo valore della variabile <i>counter</i> per cui deve essere eseguito il ciclo.                                                        |
| expr2      | Un'espressione che determina l'ultimo valore della variabile <i>counter</i> per cui deve essere eseguito il ciclo.                                                        |
| expr3      | Un'espressione che determina il valore che indica l'incremento della variabile <i>counter</i> ogni volta che il ciclo è stato eseguito.                                   |
| condition  | Un'espressione logica che restituisce un valore True o False.                                                                                                             |
| statements | Qualsiasi gruppo di una o più istruzioni dello script di Qlik Sense.                                                                                                      |

#### Example 1: Caricamento di una sequenza di file

```
// LOAD files file1.csv..file9.csv  
  
for a=1 to 9  
    LOAD * from file$(a).csv;  
next
```

### Example 2: Caricamento di un numero casuale di file

In questo esempio si presuppone l'utilizzo dei file di dati *x1.csv*, *x3.csv*, *x5.csv*, *x7.csv* e *x9.csv*. Il caricamento viene interrotto in un punto casuale mediante la condizione `if rand() < 0.5 then`.

```
for counter=1 to 9 step 2
    set filename=x$(counter).csv;

    if rand() < 0.5 then
        exit for unless counter=1
    end if

    LOAD a,b from $(filename);
next
```

### For each..next

L'istruzione di controllo **for each..next** è un costrutto per la ripetizione di script che esegue una o più istruzioni per ogni valore in un elenco le cui voci sono separate da virgole. Le istruzioni incluse nel ciclo fra **for** e **next** verranno eseguite per ogni valore nell'elenco.

#### Sintassi:

Una sintassi speciale consente di generare elenchi contenenti nomi di file e di directory nella directory attuale.

```
for each var in list
```

```
[statements]
```

```
[exit for [ ( when | unless ) condition ]
```

```
[statements]
```

```
next [var]
```

#### Argomenti:

##### Argomenti

| Argomento | Descrizione                                                                                                                                                                                                                                                     |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| var       | Il nome di una variabile di script che acquisisce un nuovo valore dall'elenco a ogni esecuzione del ciclo. Se <b>var</b> viene specificato dopo <b>next</b> , deve avere lo stesso nome di variabile rilevato dopo l'istruzione <b>for each</b> corrispondente. |

Il valore della variabile **var** può essere modificato dalle istruzioni all'interno del ciclo, tuttavia l'utilizzo di questa procedura di programmazione non è consigliato.

### 3 Istruzioni e parole chiave dello script

Se all'interno del ciclo è presente una clausola **exit for**, l'esecuzione dello script verrà trasferita alla prima istruzione dopo la clausola **next** indicando quindi la fine del ciclo. Una clausola **exit for** può essere resa condizionale dall'utilizzo opzionale di un suffisso **when** o **unless**.



*Poiché **for each..next** è un'istruzione di controllo che termina con un punto e virgola o con un carattere di fine riga, ciascuna delle tre possibili clausole corrispondenti (**for each**, **exit for** e **next**) deve essere contenuta in una sola riga.*

#### Sintassi:

```
list := item { , item }
```

```
item := constant | (expression) | filelist mask | dirlist mask |  
fieldvaluelist mask
```

#### Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                    |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| constant      | Qualsiasi numero o stringa. Tenere presente che una stringa inserita direttamente nello script deve essere racchiusa tra virgolette singole. Se la stringa non viene racchiusa tra virgolette singole, verrà interpretata come una variabile, pertanto verrà utilizzato il valore della variabile. Non è necessario che i numeri siano racchiusi tra virgolette singole.                       |
| expression    | Un'espressione arbitraria.                                                                                                                                                                                                                                                                                                                                                                     |
| mask          | Una maschera di un nome di file o di cartella che può includere un carattere qualsiasi di nome di file valido, così come i caratteri speciali standard, quali * e ?.<br><br>È possibile utilizzare percorsi di file assoluti o percorsi lib://.                                                                                                                                                |
| condition     | Un'espressione logica che restituisce un valore True o False.                                                                                                                                                                                                                                                                                                                                  |
| statements    | Qualsiasi gruppo di una o più istruzioni dello script di Qlik Sense.                                                                                                                                                                                                                                                                                                                           |
| filelist mask | Questa sintassi restituisce un elenco con valori separati da virgole di tutti i file presenti nella directory attuale che presentano una corrispondenza con la maschera del nome di file.<br><div> <i>Questo argomento supporta esclusivamente le connessioni alla libreria in modalità standard.</i></div> |

### 3 Istruzioni e parole chiave dello script

| Argomento           | Descrizione                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| dirlist mask        | <p>Questa sintassi restituisce un elenco con valori separati da virgole di tutte le cartelle incluse nella cartella attuale che presentano una corrispondenza con la maschera del nome di file.</p> <div> <i>Questo argomento supporta esclusivamente le connessioni alla libreria in modalità standard.</i></div> |
| fieldvaluelist mask | <p>Questa sintassi ripete i valori di un campo già caricato in Qlik Sense.</p>                                                                                                                                                                                                                                                                                                                      |



*Qlik Connettori a provider di archiviazione Web e altre connessioni DataFiles non supportano le maschere di filtro che utilizzano caratteri speciali (\* e ?).*

#### Example 1: Caricamento di un elenco di file

```
// LOAD the files 1.csv, 3.csv, 7.csv and xyz.csv
for each a in 1,3,7,'xyz'
    LOAD * from file$(a).csv;
next
```

#### Example 2: Creazione di un elenco di file sul disco

In questo esempio viene caricato un elenco di tutti i file correlati a Qlik Sense in una cartella.

```
sub DoDir (Root)
    for each Ext in 'qvw', 'qva', 'qvo', 'qvs', 'qvc', 'qvf', 'qvd'

        for each File in filelist (Root&'/*.' &Ext)

            LOAD
                '$(File)' as Name,
                FileSize( '$(File)' ) as Size,
                FileTime( '$(File)' ) as FileTime
            autogenerate 1;

        next File

    next Ext
    for each Dir in dirlist (Root&'/*' )

        call DoDir (Dir)

    next Dir
end sub

call DoDir ('lib://DataFiles')
```

### Example 3: Ripetizione dei valori di un campo

In questo esempio viene ripetuto l'elenco di valori caricati di FIELD e viene generato un nuovo campo NEWFIELD. Per ciascun valore di FIELD, verranno creati due record NEWFIELD.

```
load * inline [  
FIELD  
one  
two  
three  
];  
  
FOR Each a in FieldValueList('FIELD')  
LOAD '$(a)' & '-' & RecNo() as NEWFIELD AutoGenerate 2;  
NEXT a
```

La tabella risultante avrà l'aspetto seguente:

Example table

| NEWFIELD |
|----------|
| one-1    |
| one-2    |
| two-1    |
| two-2    |
| three-1  |
| three-2  |

### If..then..elseif..else..end if

L'istruzione di controllo **if..then** è un costrutto per la selezione di script che forza l'esecuzione dello script su percorsi diversi in base a una o più condizioni logiche.

In genere, le istruzioni di controllo vengono utilizzate per controllare il flusso di esecuzione dello script. In un'espressione del grafico, utilizzare invece la funzione condizionale **if**.

#### Sintassi:

```
If condition then
```

```
[ statements ]
```

```
{ elseif condition then
```

```
[ statements ] }
```

```
[ else
```

```
[ statements ] ]
```

```
end if
```

Poiché **if..then** è un'istruzione di controllo e come tale termina con un punto e virgola o con un carattere di fine riga, ciascuna delle quattro possibili clausole corrispondenti (**if..then**, **elseif..then**, **else** e **end if**) deve essere contenuta in una sola riga.

### Argomenti:

#### Argomenti

| Argomento  | Descrizione                                                          |
|------------|----------------------------------------------------------------------|
| condition  | Un'espressione logica che può restituire un valore True o False.     |
| statements | Qualsiasi gruppo di una o più istruzioni dello script di Qlik Sense. |

### Example 1:

```
if a=1 then
    LOAD * from abc.csv;

    SQL SELECT e, f, g from tab1;
end if
```

### Example 2:

```
if a=1 then; drop table xyz; end if;
```

### Example 3:

```
if x>0 then
    LOAD * from pos.csv;
elseif x<0 then
    LOAD * from neg.csv;
else
    LOAD * from zero.txt;
end if
```

## Next

La parola chiave dello script **Next** consente di chiudere i loop **For**.

## Sub..end sub

L'istruzione di controllo **sub..end sub** definisce una subroutine che può essere richiamata da un'istruzione **call**.

#### Sintassi:

```
Sub name [ ( paramlist ) ] statements end sub
```

Gli argomenti vengono copiati nella subroutine e, se i relativi parametri reali nell'istruzione **call** corrispondono a un nome di variabile, vengono copiati nuovamente quando si chiude la subroutine.

Se una subroutine presenta più parametri formali di quelli effettivi passati da un'istruzione **call**, i parametri extra vengono inizializzati su NULL e possono essere utilizzati come variabili locali all'interno della subroutine.

#### Argomenti:

Argomenti

| Argomento  | Descrizione                                                                                                                                                               |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| name       | Il nome della subroutine.                                                                                                                                                 |
| paramlist  | Un elenco separato da virgole di nomi di variabili per i parametri formali della subroutine. Può essere utilizzato come qualsiasi variabile all'interno della subroutine. |
| statements | Qualsiasi gruppo di una o più istruzioni dello script di Qlik Sense.                                                                                                      |

#### Limiti:

- Poiché **sub** è un'istruzione di controllo che termina con un punto e virgola o con un carattere di fine riga, ciascuna delle due clausole corrispondenti (**sub** e **end sub**) deve essere contenuta in una sola riga.
- Quando si definisce una routine secondaria con `sub . . end sub` all'interno di un'istruzione di controllo, ad esempio `if . . then`, è possibile richiamare la routine secondaria solo dall'interno della stessa istruzione di controllo.

#### Example 1:

```
Sub INCR (I,J)
```

```
I = I + 1
```

```
Exit Sub when I < 10
```

```
J = J + 1
```

```
End Sub
```

```
Call INCR (X,Y)
```

#### Example 2: - trasferimento parametri

```
Sub ParTrans (A,B,C)
```

```
A=A+1
```

```
B=B+1
```

```
C=C+1
```

```
End Sub
```

```
A=1
```

```
X=1
```

```
C=1
```

```
Call ParTrans (A, (X+1)*2)
```

Dall'esempio precedente risulta che localmente, all'interno della subroutine, A verrà inizializzato su 1, B verrà inizializzato su 4 e C verrà inizializzato su NULL.

Quando si chiude la subroutine, la variabile globale A otterrà 2 come valore (ricopiato dalla subroutine). Il secondo parametro reale "(X+1)\*2" non verrà ricopiato dato che non si tratta di una variabile. Infine, la variabile globale C non verrà influenzata dalla chiamata della subroutine.

### Switch..case..default..end switch

L'istruzione di controllo **switch** è un costrutto per la selezione di script che forza l'esecuzione dello script su percorsi diversi, in base al valore di un'espressione.

#### Sintassi:

```
Switch expression {case valuelist [ statements ]} [default statements] end switch
```



Poiché **switch** è un'istruzione di controllo e come tale termina con un punto e virgola o con un carattere di fine riga, ciascuna delle quattro possibili clausole corrispondenti (**switch**, **case**, **default** e **end switch**) deve essere contenuta in una sola riga.

#### Argomenti:

##### Argomenti

| Argomento  | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| expression | Un'espressione arbitraria.                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| valuelist  | Un elenco separato da virgole dei valori con i quali viene confrontato il valore dell'espressione. L'esecuzione dello script continua con le istruzioni del primo gruppo in cui il valore di valuelist è pari al valore nell'espressione. Ciascun valore in valuelist può essere un'espressione arbitraria. Se non viene individuata alcuna corrispondenza in nessuna clausola <b>case</b> , vengono eseguite le eventuali istruzioni della clausola <b>default</b> . |
| statements | Qualsiasi gruppo di una o più istruzioni dello script di Qlik Sense.                                                                                                                                                                                                                                                                                                                                                                                                  |

### Esempio:

```
Switch I

Case 1

LOAD '$(I): CASE 1' as case autogenerate 1;

Case 2

LOAD '$(I): CASE 2' as case autogenerate 1;

Default

LOAD '$(I): DEFAULT' as case autogenerate 1;

End Switch
```

### To

La parola chiave dello script **To** viene utilizzata in diverse istruzioni dello script.

## 3.2 Prefissi dello script

I prefissi possono essere applicati alle istruzioni regolari pertinenti, ma mai a istruzioni di controllo. I prefissi **when** e **unless** possono comunque essere utilizzati come suffissi per alcune specifiche clausole di istruzioni di controllo.

Tutte le parole chiave dello script possono essere immesse con qualsiasi combinazione di caratteri maiuscoli e minuscoli. I nomi dei campi e delle variabili utilizzati nelle istruzioni possono essere immessi indipendentemente dal formato del carattere.

### Prospetto dei prefissi dello script

Ciascuna funzione viene descritta ulteriormente dopo la panoramica. È inoltre possibile fare clic sul nome della funzione nella sintassi per accedere immediatamente ai dettagli per tale funzione specifica.

#### Add

Il prefisso **Add** può essere aggiunto a qualsiasi istruzione **LOAD** o **SELECT** nello script per specificare che dovrebbe aggiungere record a un'altra tabella. Specifica anche che questa istruzione dovrebbe essere eseguita in un ricaricamento parziale. Il prefisso **Add** può essere usato anche in un'istruzione **Map**.

```
Add [only] [Concatenate[(tablename )]] (loadstatement | selectstatement)
```

```
Add [ Only ] mapstatement
```

---

## 3 Istruzioni e parole chiave dello script

### Buffer

È possibile creare e gestire automaticamente i file QVD mediante il prefisso **buffer**. Questo prefisso può essere utilizzato in quasi tutte le istruzioni **LOAD** e **SELECT** di uno script e indica che i file QVD vengono utilizzati per memorizzare nella cache/nel buffer il risultato dell'istruzione.

```
Buffer[(option [ , option])] ( loadstatement | selectstatement )
```

```
option::= incremental | stale [after] amount [(days | hours)]
```

### Concatenate

Se due tabelle da concatenare contengono gruppi differenti di campi, è tuttavia possibile imporre la concatenazione di due tabelle utilizzando il prefisso **Concatenate**.

```
Concatenate[ (tablename ) ] ( loadstatement | selectstatement )
```

### Crosstable

Il prefisso di caricamento **crosstable** viene utilizzato per trasporre i dati strutturati in "tabella incrociata" o in "tabella pivot". I dati strutturati in questo modo si trovano comunemente quando si lavora con le sorgenti dei fogli di calcolo. L'output e lo scopo del prefisso di caricamento **crosstable** è di trasporre tali strutture nell'equivalente di una normale tabella in un formato a colonne, poiché questa struttura è generalmente più adatta per l'analisi in Qlik Sense.

```
Crosstable (attribute field name, data field name [ , n ] ) ( loadstatement | selectstatement )
```

### First

Il prefisso **First** aggiunto a un'istruzione **LOAD** o **SELECT (SQL)** viene utilizzato per caricare un numero di record massimo impostato dalla tabella di origine dei dati.

```
First n( loadstatement | selectstatement )
```

### Generic

Il prefisso di caricamento **Generic** consente la conversione dei dati modellati su entità-attributo-valore (EAV) in una struttura di tabella relazionale tradizionale e normalizzata. La modellazione EAV in alternativa è denominata "modellazione di dati generici" o "schema aperto".

```
Generic ( loadstatement | selectstatement )
```

### Hierarchy

Il prefisso **hierarchy** viene utilizzato per trasformare una tabella gerarchica padre-figlio in una tabella utile in un modello dati Qlik Sense. Può essere inserito prima di un'istruzione **LOAD** o **SELECT** e utilizzerà i risultati dell'istruzione di caricamento come input per la trasformazione della tabella.

```
Hierarchy (NodeID, ParentID, NodeName, [ParentName], [PathSource],  
[PathName], [PathDelimiter], [Depth])(loadstatement | selectstatement)
```

### HierarchyBelongsTo

Questo prefisso viene utilizzato per trasformare una tabella gerarchica padre-figlio in una tabella utile in un modello dati Qlik Sense. Può essere inserito prima di un'istruzione **LOAD** o **SELECT** e utilizzerà i risultati dell'istruzione di caricamento come input per la trasformazione della tabella.

```
HierarchyBelongsTo (NodeID, ParentID, NodeName, AncestorID, AncestorName, [DepthDiff]) (loadstatement | selectstatement)
```

### Inner

I prefissi **join** e **keep** possono essere preceduti dal prefisso **inner**.

Se viene inserito prima di **join**, specifica che occorre utilizzare un'unione interna. La tabella risultante contiene solo le combinazioni di valori di campo estratte dalle tabelle di dati non elaborati dove i valori di campo di collegamento vengono rappresentati in entrambe le tabelle. Se utilizzato prima di **keep**, specifica che entrambe le tabelle di dati non elaborati devono essere ridotte alla loro intersezione comune prima di essere memorizzate in Qlik Sense.

```
Inner ( Join | Keep ) [ (tablename) ] (loadstatement | selectstatement )
```

### IntervalMatch

Il prefisso **IntervalMatch** consente di creare una tabella che corrisponde sia ai valori numerici discreti su uno o più intervalli numerici che, in modo opzionale, ai valori di una o più chiavi aggiuntive.

```
IntervalMatch (matchfield) (loadstatement | selectstatement )
```

```
IntervalMatch (matchfield, keyfield1 [ , keyfield2, ... keyfield5 ] )  
(loadstatement | selectstatement )
```

### Join

Il prefisso **join** unisce la tabella caricata a una tabella denominata esistente oppure all'ultima tabella di dati creata in precedenza.

```
[Inner | Outer | Left | Right ] Join [ (tablename) ] ( loadstatement |  
selectstatement )
```

### Keep

Il prefisso **keep** è simile al prefisso **join**. Analogamente al prefisso **join**, confronta la tabella caricata con una tabella denominata esistente o con l'ultima tabella dati creata in precedenza, tuttavia, invece di unire la tabella caricata alla tabella esistente, riduce una o entrambe le due tabelle prima che vengano memorizzate in Qlik Sense, in base all'intersezione dei dati della tabella. Il confronto effettuato equivale a un'unione naturale effettuata su tutti i campi comuni, ad esempio nello stesso modo di un'unione corrispondente. In ogni modo, le due tabelle non vengono unite e verranno conservate in Qlik Sense come due tabelle denominate separatamente.

```
(Inner | Left | Right) Keep [(tablename) ] ( loadstatement | selectstatement  
)
```

### Left

I prefissi **Join** e **Keep** possono essere preceduti dal prefisso **left**.

## 3 Istruzioni e parole chiave dello script

Se viene inserito prima di **join**, specifica che occorre utilizzare un'unione sinistra. La tabella risultante conterrà solo le combinazioni di valori di campo estratte dalle tabelle di dati non elaborati, dove i valori di campo di collegamento vengono rappresentati nella prima tabella. Se viene utilizzato prima di **keep**, specifica che la seconda tabella di dati non elaborati deve essere ridotta alla sua intersezione comune con la prima tabella prima di essere memorizzata in Qlik Sense.

```
Left ( Join | Keep ) [ (tablename) ] (loadstatement | selectstatement )
```

### Mapping

Il prefisso **mapping** consente di creare una tabella di mapping che può essere utilizzata, ad esempio, per sostituire i valori di campo e i nomi di campo durante l'esecuzione dello script.

```
Mapping ( loadstatement | selectstatement )
```

### Merge

Il prefisso **Merge** può essere aggiunto a qualsiasi istruzione **LOAD** o **SELECT** nello script per specificare che dovrebbe aggiungere record a un'altra tabella. Specifica anche che questa istruzione dovrebbe essere eseguita in un ricaricamento parziale.

```
Unisci [only] [(SequenceNoField [, SequenceNoVar])] On ListOfKeys  
[Concatenate [(TableName)]] (loadstatement | selectstatement)
```

### NoConcatenate

Con il prefisso **NoConcatenate**, due tabelle che vengono caricate con gruppi di campo identici verranno considerate come due tabelle interne separate, invece di venire concatenate automaticamente.

```
NoConcatenate ( loadstatement | selectstatement )
```

### Outer

Il prefisso **Join** esplicito può essere preceduto dal prefisso **Outer** per specificare un'unione esterna. In un'unione esterna vengono generate tutte le combinazioni tra le due tabelle. La tabella risultante conterrà quindi le combinazioni di valori di campo provenienti dalle tabelle di dati non elaborati dove i valori di campo di collegamento vengono rappresentati in una o entrambe le tabelle. La parola chiave **Outer** è facoltativa ed è il tipo di unione predefinito utilizzato quando non viene specificato un prefisso di unione.

```
Outer Join [ (tablename) ] (loadstatement | selectstatement )
```

### Partial reload

Un ricaricamento completo inizia sempre eliminando tutte le tabelle nel modello dati esistente, dopodiché esegue lo script di caricamento.

Un [Caricamento parziale \(page 103\)](#) non effettuerà tale operazione. Mantiene invece tutte le tabelle nel modello dati ed esegue solo le istruzioni **Load** e **Select** precedute da un prefisso **Aggiungi**, **Unisci** o **Sostituisci**. Altre tabelle di dati non sono interessate dal comando. L'argomento **solo** indica che l'istruzione deve essere eseguita solo durante i caricamenti parziali e deve essere ignorata durante i caricamenti completi. La tabella seguente riepiloga l'esecuzione dell'istruzione per i ricaricamenti parziali e completi.

### 3 Istruzioni e parole chiave dello script

---

#### Replace

Il prefisso **Replace** può essere aggiunto a qualsiasi istruzione **LOAD** o **SELECT** nello script per specificare che la tabella caricata dovrebbe sostituire un'altra tabella. Specifica anche che questa istruzione dovrebbe essere eseguita in un ricaricamento parziale. Il prefisso **Replace** può essere usato anche in un'istruzione **Map**.

```
Replace [only] [Concatenate[ (tablename) ]] (loadstatement | selectstatement)
```

```
Replace [only] mapstatement
```

#### Right

I prefissi **Join** e **Keep** possono essere preceduti dal prefisso **right**.

Se viene inserito prima di **join**, specifica che occorre utilizzare un'unione destra. La tabella risultante contiene solo le combinazioni di valori di campo estratte dalle tabelle di dati non elaborati, dove i valori di campo di collegamento vengono rappresentati nella seconda tabella. Se viene utilizzato prima di **keep**, specifica che la prima tabella di dati non elaborati deve essere ridotta alla sua intersezione comune con la seconda tabella prima di essere memorizzata in Qlik Sense.

```
Right (Join | Keep) [(tablename)] (loadstatement | selectstatement )
```

#### Sample

Il prefisso **sample** aggiunto a un'istruzione **LOAD** o **SELECT** viene utilizzato per caricare un campione casuale di record dalla sorgente dati.

```
Sample p ( loadstatement | selectstatement )
```

#### Semantic

È possibile caricare le tabelle che contengono relazioni fra i record utilizzando il prefisso **semantic**. Si può trattare ad esempio di auto-riferimenti all'interno di una tabella, in cui un record punta a un altro, come ad esempio padre, appartenenza o predecessore.

```
Semantic ( loadstatement | selectstatement)
```

#### Unless

Il prefisso e suffisso **unless** viene utilizzato per creare una clausola condizionale che determina se valutare o meno un'istruzione oppure una clausola exit. Può essere considerato come un'alternativa compatta all'istruzione completa **if..end if**.

```
(Unless condition statement | exitstatement Unless condition )
```

#### When

Il prefisso e suffisso **when** viene utilizzato per creare una clausola condizionale che determina se eseguire o meno un'istruzione oppure una clausola exit. Può essere considerato come un'alternativa compatta all'istruzione completa **if..end if**.

```
( When condition statement | exitstatement when condition )
```

### Add

Il prefisso **Add** può essere aggiunto a qualsiasi istruzione **LOAD** o **SELECT** nello script per specificare che dovrebbe aggiungere record a un'altra tabella. Specifica anche che questa istruzione dovrebbe essere eseguita in un ricaricamento parziale. Il prefisso **Add** può essere usato anche in un'istruzione **Map**.



*Affinché il ricaricamento parziale funzioni correttamente, aprire l'app con i dati prima di attivare un ricaricamento parziale.*

Eseguire un ricaricamento parziale usando il pulsante **Ricarica**. È anche possibile utilizzare Qlik Engine JSON API.

#### Sintassi:

```
Add [only] [Concatenate [(tablename)]] (loadstatement | selectstatement)
```

```
Add [only] mapstatement
```

Durante un caricamento normale (non parziale), la costruzione **Add LOAD** funzionerà come normale istruzione **LOAD**. I record verranno generati e archiviati in una tabella.

Se viene utilizzato il prefisso **Concatenate**, o se esiste una tabella con lo stesso set di campi, i record verranno aggiunti alla tabella esistente rilevante. Altrimenti, la costruzione **Add LOAD** creerà una nuova tabella.

Un caricamento parziale otterrà lo stesso risultato. L'unica differenza è che la costruzione **Add LOAD** non creerà mai una nuova tabella. Esiste sempre una tabella pertinente dall'esecuzione script precedente a cui aggiungere i record.

Non viene eseguito alcun controllo di duplicati. Pertanto, un'istruzione che utilizza il prefisso **Add** spesso includerà un qualificatore distinto o una clausola dove a protezione dei duplicati.

L'istruzione **Add Map...Using** determina l'esecuzione del mapping anche durante l'esecuzione parziale dello script.

#### Argomenti:

##### Argomenti

| Argomento | Descrizione                                                                                                                                                                             |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| only      | Un qualificatore opzionale che denota che l'istruzione dovrebbe essere eseguita solo durante i caricamenti parziali. Deve essere ignorata durante i caricamenti normali (non parziali). |

## 3 Istruzioni e parole chiave dello script

Esempi e risultati:

| Esempio                                                                                                                                        | Risultato                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tab1:<br><br>LOAD Name, Number<br>FROM Persons.csv;<br><br>Add LOAD Name,<br>Number FROM<br>NewPersons.csv;                                    | <p>Durante il ricaricamento normale, i dati vengono caricati dal file <i>Persons.csv</i> e memorizzati nella tabella di Qlik Sense Tab1. I dati in <i>NewPersons.csv</i> vengono quindi concatenati alla stessa tabella Qlik Sense.</p> <p>Durante il ricaricamento parziale, i dati vengono caricati dal file <i>NewPersons.csv</i> e aggiunti alla tabella di Qlik Sense Tab1. Non viene eseguito alcun controllo di duplicati.</p>                                                                                                                                                                                                                            |
| Tab1:<br><br>SQL SELECT Name,<br>Number FROM<br>Persons.csv;<br><br>Add LOAD Name,<br>Number FROM<br>NewPersons.csv where<br>not exists(Name); | <p>Viene eseguito un controllo di duplicati mediante la verifica dell'esistenza di Name nei dati della tabella caricati in precedenza.</p> <p>Durante il ricaricamento normale, i dati vengono caricati dal file <i>Persons.csv</i> e memorizzati nella tabella di Qlik Sense Tab1. I dati in <i>NewPersons.csv</i> vengono quindi concatenati alla stessa tabella Qlik Sense.</p> <p>Durante il ricaricamento parziale, i dati vengono caricati dal file <i>NewPersons.csv</i>, che viene aggiunto alla tabella Qlik Sense Tab1. Viene eseguito un controllo di duplicati che verifica l'esistenza di Name nei dati della tabella precedentemente caricata.</p> |
| Tab1:<br><br>LOAD Name, Number<br>FROM Persons.csv;<br><br>Add Only LOAD Name,<br>Number FROM<br>NewPersons.csv where<br>not exists(Name);     | <p>Durante il ricaricamento normale, i dati vengono caricati dal file <i>Persons.csv</i> e memorizzati nella tabella di Qlik Sense Tab1. L'istruzione che carica <i>NewPersons.csv</i> viene ignorata.</p> <p>Durante il ricaricamento parziale, i dati vengono caricati dal file <i>NewPersons.csv</i>, che viene aggiunto alla tabella Qlik Sense Tab1. Viene eseguito un controllo di duplicati che verifica l'esistenza di Name nei dati della tabella precedentemente caricata.</p>                                                                                                                                                                         |

### Buffer

È possibile creare e gestire automaticamente i file QVD mediante il prefisso **buffer**. Questo prefisso può essere utilizzato in quasi tutte le istruzioni **LOAD** e **SELECT** di uno script e indica che i file QVD vengono utilizzati per memorizzare nella cache/nel buffer il risultato dell'istruzione.

Sintassi:

```
Buffer [(option [ , option])] ( loadstatement | selectstatement )  
option::= incremental | stale [after] amount [(days | hours)]
```

Se nessuna opzione viene utilizzata, il buffer QVD creato alla prima esecuzione dello script verrà utilizzato indefinitamente.

Il file del buffer viene salvato nella sottocartella *Buffers* solitamente in *C:\ProgramData\Qlik\Sense\Engine\Buffers* (installazione sul server) o *C:\Utenti\{utente}\Documenti\Qlik\Sense\Buffers* (Qlik Sense Desktop).

### 3 Istruzioni e parole chiave dello script

Il nome del file QVD è un nome calcolato, un hash esadecimale a 160 bit di tutta l'istruzione **LOAD** o **SELECT** successiva e delle altre informazioni discriminanti. Questo significa che il buffer QVD verrà invalidato da qualsiasi modifica apportata all'istruzione **LOAD** o **SELECT** seguente.

I buffer QVD vengono normalmente rimossi quando non esistono più riferimenti a essi durante l'intera esecuzione di uno script nell'app che li ha creati oppure quando l'app che li ha creati non esiste più.

#### Argomenti:

| Argomenti                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|--------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Argomento                                  | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| incrementale                               | <p>L'opzione incremental consente di leggere solo parte di un file sottostante. La dimensione precedente del file viene salvata nell'intestazione XML del file QVD. Queste informazioni risultano particolarmente utili con i file di registro. Tutti i record caricati nell'occasione precedente vengono letti dal file QVD, mentre i nuovi record seguenti vengono letti dalla sorgente originale, quindi viene creato un file QVD aggiornato.</p> <p>L'opzione incremental può essere utilizzata solo con le istruzioni <b>LOAD</b> e i file di testo. Il carico incrementale non può essere utilizzato quando i precedenti dati vengono cambiati o eliminati.</p>                                                                                                                                                                         |
| obsoleto [dopo] numero di [(giorni   ore)] | <p>amount è un numero che specifica l'intervallo di tempo. Possono essere utilizzati valori decimali. Se omessa, verrà utilizzata l'unità di misura giorni.</p> <p>In genere, l'opzione stale after viene utilizzata con sorgenti DB i cui dati originali non dispongono di alcun indicatore temporale semplice. In alternativa, è possibile specificare per quanto tempo conservare lo snapshot QVD. Una clausola stale after dichiara semplicemente l'intervallo di tempo a partire dalla creazione del buffer QVD, trascorso il quale non verrà più considerato valido. Prima di quel tempo, il buffer QVD verrà utilizzato come sorgente dei dati e, trascorso l'intervallo specificato, verrà utilizzata la sorgente dati iniziale. Il file del buffer QVD verrà aggiornato automaticamente, quindi avrà inizio un nuovo intervallo.</p> |

#### Limiti:

Esistono numerose limitazioni, la più importante delle quali stabilisce che dovrà esistere necessariamente un'istruzione **LOAD** o **SELECT** per file alla base di qualsiasi istruzione complessa.

#### Example 1:

```
Buffer SELECT * from MyTable;
```

#### Example 2:

```
Buffer (stale after 7 days) SELECT * from MyTable;
```

### Example 3:

```
Buffer (incremental) LOAD * from MyLog.log;
```

## Concatenate

**Concatenate** è un prefisso di caricamento dello script che consente di aggiungere un set di dati a una tabella in memoria già esistente. Viene spesso utilizzato per aggiungere diversi set di dati transazionali a un'unica tabella dei fatti centrale, o per costruire insiemi di dati di riferimento comuni di un tipo specifico che provengono da più fonti. La sua funzionalità è simile a quella dell'operatore UNION di SQL.

La tabella risultante da un'operazione **concatenate** conterrà il set di dati originale con le nuove righe di dati aggiunte in fondo alla tabella. Le tabelle di origine e di destinazione possono presentare campi diversi. Se i campi sono diversi, la tabella risultante sarà ampliata per rappresentare il risultato combinato di tutti i campi presenti sia nella tabella di origine che in quella di destinazione.

### Sintassi:

```
Concatenate [ (tablename ) ] ( loadstatement | selectstatement )
```

#### Argomenti

| Argomento                     | Descrizione                                                                                                                                                                                                                                                                                                                   |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| tablename                     | Il nome di una tabella esistente. La tabella denominata sarà l'obiettivo dell'operazione <b>concatenate</b> e tutti i record di dati caricati saranno aggiunti a quella tabella. Se il parametro <b>tablename</b> non viene utilizzato, la tabella di destinazione sarà l'ultima tabella caricata prima di questa istruzione. |
| loadstatement/selectstatement | L'argomento <b>loadstatement/selectstatement</b> che segue l'argomento <b>tablename</b> sarà concatenato alla tabella specificata.                                                                                                                                                                                            |

## Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione **SET DateFormat** nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

Esempio di funzione

| Esempio                                     | Risultato                                                                                                                                                                                                                                                                           |
|---------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Concatenate<br>(Transactions)<br>Load ... ; | I dati caricati nell'istruzione di caricamento sotto il prefisso concatenate verranno aggiunti alla tabella esistente in memoria denominata Transactions (supponendo che una tabella denominata Transactions sia stata caricata prima di questo punto nello script di caricamento). |

### Esempio 1 - Aggiunta di più set di dati a una tabella di destinazione con il prefisso di caricamento Concatenate

Script di caricamento e risultati

#### Panoramica

In questo esempio verranno caricati due script in ordine sequenziale.

- Il primo script di caricamento contiene un set di dati iniziale con date e importi che viene inviato a una tabella denominata Transactions.
- Il secondo script di caricamento contiene:
  - Un secondo set di dati che viene aggiunto al set di dati iniziale utilizzando il prefisso concatenate. Questo set di dati contiene un campo aggiuntivo, type, che non è presente nel set di dati iniziale.
  - Il prefisso Concatenate.

Aprire l'editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

#### Primo script di caricamento

```
Transactions:  
Load * Inline [
```

```
id, date, amount  
3750, 08/30/2018, 23.56  
3751, 09/07/2018, 556.31  
3752, 09/16/2018, 5.75  
3753, 09/22/2018, 125.00  
3754, 09/22/2018, 484.21  
3756, 09/22/2018, 59.18  
3757, 09/23/2018, 177.42  
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- date

- amount

Tabella dei risultati del primo script di caricamento

| id   | date       | importo |
|------|------------|---------|
| 3750 | 08/30/2018 | 23.56   |
| 3751 | 09/07/2018 | 556.31  |
| 3752 | 09/16/2018 | 5.75    |
| 3753 | 09/22/2018 | 125.00  |
| 3754 | 09/22/2018 | 484.21  |
| 3756 | 09/22/2018 | 59.18   |
| 3757 | 09/23/2018 | 177.42  |

La tabella mostra il set di dati iniziale.

#### Secondo script di caricamento

Aprire l'editor caricamento dati e aggiungere lo script di caricamento sotto.

```
Concatenate(Transactions)
Load * Inline [
id, date, amount, type
3758, 10/01/2018, 164.27, Internal
3759, 10/03/2018, 384.00, External
3760, 10/06/2018, 25.82, Internal
3761, 10/09/2018, 312.00, Internal
3762, 10/15/2018, 4.56, Internal
3763, 10/16/2018, 90.24, Internal
3764, 10/18/2018, 19.32, External
];
```

#### Risultati

Caricare i dati e passare al foglio. Creare questo campo come una dimensione.

- type

Tabella dei risultati del secondo script di caricamento

| id   | date       | importo | tipo |
|------|------------|---------|------|
| 3750 | 08/30/2018 | 23.56   | -    |
| 3751 | 09/07/2018 | 556.31  | -    |
| 3752 | 09/16/2018 | 5.75    | -    |

| id   | date       | importo | tipo    |
|------|------------|---------|---------|
| 3753 | 09/22/2018 | 125.00  | -       |
| 3754 | 09/22/2018 | 484.21  | -       |
| 3756 | 09/22/2018 | 59.18   | -       |
| 3757 | 09/23/2018 | 177.42  | -       |
| 3758 | 10/01/2018 | 164.27  | Interne |
| 3759 | 10/03/2018 | 384.00  | Esterni |
| 3760 | 10/06/2018 | 25.82   | Interne |
| 3761 | 10/09/2018 | 312.00  | Interne |
| 3762 | 10/15/2018 | 4.56    | Interne |
| 3763 | 10/16/2018 | 90.24   | Interne |
| 3764 | 10/18/2018 | 19.32   | Esterni |

Si notino i valori null nel campo type per i primi sette record caricati in cui type non era stato definito.

#### Esempio 2 - Aggiunta di più set di dati a una tabella di destinazione utilizzando la concatenazione implicita

Script di caricamento e risultati

##### Panoramica

Un caso tipico di applicazione implicita dei dati si ha quando si caricano diversi file di dati strutturati in modo identico e si desidera aggiungerli tutti a una tabella di destinazione.

Ad esempio, utilizzando wildcards nei nomi dei file con una sintassi come:

```
myTable:
Load * from [myFile_*.qvd] (qvd);
```

o in cicli che utilizzano costrutti come:

```
for each file in filelist('myFile_*.qvd')

myTable:
Load * from [$(file)] (qvd);

next file
```



*La concatenazione implicita avverrà tra due tabelle caricate con campi di nome identico, anche se non sono definite una dopo l'altra nello script. Ciò può portare all'aggiunta involontaria di dati alle tabelle. Se non si vuole che una tabella secondaria con campi identici venga aggiunta in questo modo, utilizzare il prefisso `load noconcatenate`. Rinominare la tabella con un tag nome alternativo non è sufficiente a impedire che si verifichi una concatenazione implicita. Per ulteriori informazioni, vedere [NoConcatenate \(page 93\)](#).*

In questo esempio verranno caricati due script in ordine sequenziale.

- Il primo script di caricamento contiene un set di dati iniziale con quattro campi che viene inviato a una tabella denominata `Transactions`.
- Il secondo script di caricamento contiene un set di dati con gli stessi campi del primo set di dati.

Aprire l'editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

### Primo script di caricamento

```
Transactions:
Load * Inline [
id, date, amount, type
3758, 10/01/2018, 164.27, Internal
3759, 10/03/2018, 384.00, External
3760, 10/06/2018, 25.82, Internal
3761, 10/09/2018, 312.00, Internal
3762, 10/15/2018, 4.56, Internal
3763, 10/16/2018, 90.24, Internal
3764, 10/18/2018, 19.32, External
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- date
- amount
- type

Tabella dei risultati del primo script di caricamento

| id   | date       | tipo    | importo |
|------|------------|---------|---------|
| 3758 | 10/01/2018 | Interne | 164.27  |
| 3759 | 10/03/2018 | Esterni | 384.00  |
| 3760 | 10/06/2018 | Interne | 25.82   |
| 3761 | 10/09/2018 | Interne | 312.00  |
| 3762 | 10/15/2018 | Interne | 4.56    |

### 3 Istruzioni e parole chiave dello script

---

| id   | date       | tipo    | importo |
|------|------------|---------|---------|
| 3763 | 10/16/2018 | Interne | 90.24   |
| 3764 | 10/18/2018 | Esterni | 19.32   |

La tabella mostra il set di dati iniziale.

#### Secondo script di caricamento

Aprire l'editor caricamento dati e aggiungere lo script di caricamento sotto.

```
Load * Inline [  
id, date, amount, type  
3765, 11/03/2018, 129.40, Internal  
3766, 11/05/2018, 638.50, External  
];
```

#### Risultati

Caricare i dati e passare al foglio.

Tabella dei risultati del secondo script di caricamento

| id   | date       | tipo    | importo |
|------|------------|---------|---------|
| 3758 | 10/01/2018 | Interne | 164.27  |
| 3759 | 10/03/2018 | Esterni | 384.00  |
| 3760 | 10/06/2018 | Interne | 25.82   |
| 3761 | 10/09/2018 | Interne | 312.00  |
| 3762 | 10/15/2018 | Interne | 4.56    |
| 3763 | 10/16/2018 | Interne | 90.24   |
| 3764 | 10/18/2018 | Esterni | 19.32   |
| 3765 | 11/03/2018 | Interne | 129.40  |
| 3766 | 11/05/2018 | Esterni | 638.50  |

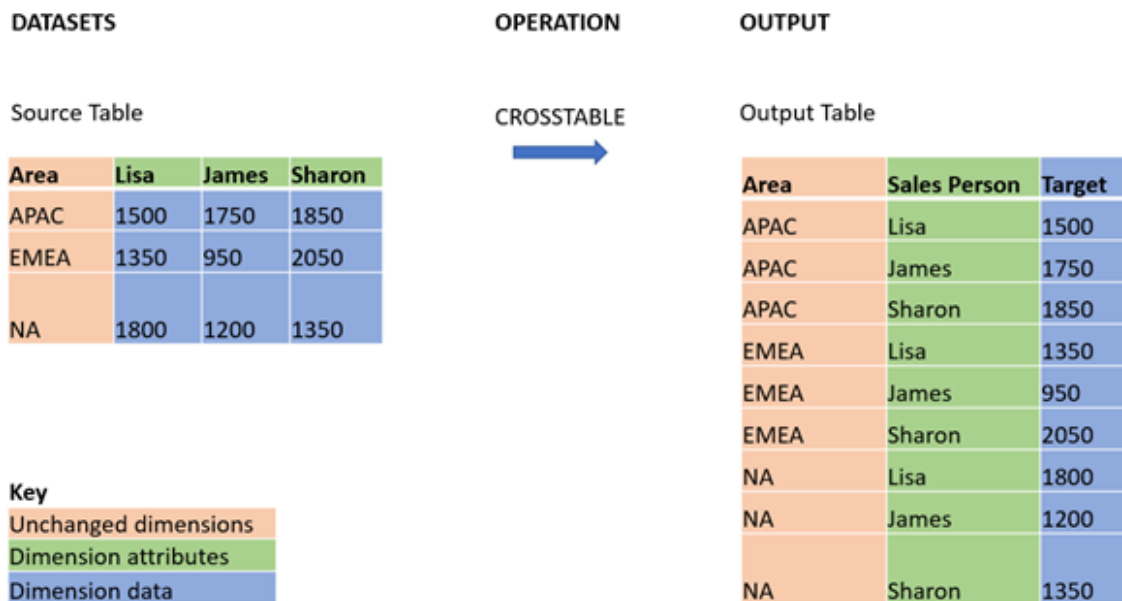
Il secondo set di dati è stato concatenato implicitamente al set di dati iniziale, poiché i campi sono identici.

#### Crosstable

Il prefisso di caricamento **crosstable** viene utilizzato per trasporre i dati strutturati in "tabella incrociata" o in "tabella pivot". I dati strutturati in questo modo si trovano comunemente quando si lavora con le sorgenti dei fogli di calcolo. L'output e lo scopo del prefisso di caricamento **crosstable** è di trasporre tali strutture nell'equivalente di una normale tabella in un formato a colonne, poiché questa struttura è generalmente più adatta per l'analisi in Qlik Sense.

### 3 Istruzioni e parole chiave dello script

Esempio di dati strutturati come tabella incrociata e struttura equivalente dopo una trasformazione crosstable



Sintassi:

```
crosstable (attribute field name, data field name [ , n ] ) ( loadstatement | selectstatement )
```

Argomenti

| Argomento            | Descrizione                                                                                                                                                                         |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| attribute field name | Il nome del campo di output desiderato che descrive la dimensione orientata orizzontalmente da trasporre (la riga di intestazione).                                                 |
| data field name      | Il nome del campo di output desiderato che descrive i dati orientati orizzontalmente della dimensione da trasporre (ma matrice dei valori di dati sotto la riga dell'intestazione). |
| n                    | Il numero di campi qualificatori, o le dimensioni invariate, che precedono la tabella da trasformare in un formato generico. Il valore predefinito è 1.                             |

Questa funzione di script è correlata alle funzioni seguenti:

Funzioni correlate

| Funzione                          | Interazione                                                                                                                                                                                                                                     |
|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">Generic (page 59)</a> | Un prefisso del carico di trasformazione che prende un set di dati strutturati entity-attribute-value e lo trasforma nella struttura di una normale tabella relazionale, separando ogni attributo rilevato in un nuovo campo o colonna di dati. |

### Esempio 1 - Trasformazione dei dati di vendita pivot (semplice)

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere il primo script di caricamento di seguito in una nuova scheda.

Il primo script di caricamento contiene un set di dati a cui verrà successivamente applicato il prefisso dello script `crosstable`, con la sezione che applica `crosstable` con commenti. Ciò significa che la sintassi dei commenti è stata utilizzata per disabilitare questa sezione nello script di caricamento.

Il secondo script di caricamento è uguale al primo, ma con l'applicazione di `crosstable` senza commenti (abilitato rimuovendo la sintassi del commento). Gli script vengono mostrati in questo modo per evidenziare il valore di questa funzione di script nella trasformazione dei dati.

#### Primo script di caricamento (funzione non applicata)

```
tmpData:
//Crosstable (MonthText, Sales)
Load * inline [
Product, Jan 2021, Feb 2021, Mar 2021, Apr 2021, May 2021, Jun 2021
A, 100, 98, 103, 63, 108, 82
B, 284, 279, 297, 305, 294, 292
C, 50, 53, 50, 54, 49, 51];

//Final:
//Load Product,
//Date(Date#(MonthText,'MMM YYYY'),'MMM YYYY') as Month,
//Sales

//Resident tmpData;

//Drop Table tmpData;
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- Product
- Jan 2021
- Feb 2021
- Mar 2021
- Apr 2021
- May 2021
- Jun 2021

### 3 Istruzioni e parole chiave dello script

Tabella dei risultati

| Prodotto | Gen 2021 | Feb 2021 | Mar 2021 | Apr 2021 | May 2021 | Giu 2021 |
|----------|----------|----------|----------|----------|----------|----------|
| A        | 100      | 98       | 103      | 63       | 108      | 82       |
| B        | 284      | 279      | 297      | 305      | 294      | 292      |
| C        | 50       | 53       | 50       | 54       | 49       | 51       |

Questo script consente la creazione di una tabella incrociata con una colonna per ogni mese e una riga per prodotto. Nel suo formato attuale, questi dati non sono facili da analizzare. Sarebbe preferibile disporre di tutti i numeri in un campo e tutti i mesi in un altro, in una tabella a tre colonne. La sezione successiva spiega come eseguire questa trasformazione nella tabella incrociata.

#### Secondo script di caricamento (funzione non applicata)

Cancellare i commenti nello script rimuovendo //. Lo script di caricamento dovrebbe avere l'aspetto seguente:

```
tmpData:
Crosstable (MonthText, Sales)
Load * inline [
Product, Jan 2021, Feb 2021, Mar 2021, Apr 2021, May 2021, Jun 2021
A, 100, 98, 103, 63, 108, 82
B, 284, 279, 297, 305, 294, 292
C, 50, 53, 50, 54, 49, 51];
```

```
Final:
Load Product,
Date(Date#(MonthText, 'MMM YYYY'), 'MMM YYYY') as Month,
Sales
```

```
Resident tmpData;
```

```
Drop Table tmpData;
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- Product
- Month
- Sales

Tabella dei risultati

| Prodotto | Mese     | Vendite |
|----------|----------|---------|
| A        | Jan 2021 | 100     |

| Prodotto | Mese     | Vendite |
|----------|----------|---------|
| A        | Feb 2021 | 98      |
| A        | Mar 2021 | 103     |
| A        | Apr 2021 | 63      |
| A        | May 2021 | 108     |
| A        | Jun 2021 | 82      |
| B        | Jan 2021 | 284     |
| B        | Feb 2021 | 279     |
| B        | Mar 2021 | 297     |
| B        | Apr 2021 | 305     |
| B        | May 2021 | 294     |
| B        | Jun 2021 | 292     |
| C        | Jan 2021 | 50      |
| C        | Feb 2021 | 53      |
| C        | Mar 2021 | 50      |
| C        | Apr 2021 | 54      |
| C        | May 2021 | 49      |
| C        | Jun 2021 | 51      |

Una volta applicato il prefisso dello script, la tabella incrociata viene trasformata in una tabella lineare con una colonna per il valore `Month` e un'altra per `sales`. Ciò migliora la leggibilità dei dati.

#### Esempio 2 - Trasformazione dei dati sugli obiettivi di vendita pivot in una tabella con struttura verticale (livello intermedio)

Script di caricamento ed espressione del grafico

##### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella denominata `Destinazione`.
- Il prefisso di caricamento `crosstable`, che traspone i nomi degli agenti dedicati alle vendite pivot in un campo a sé stante con l'etichetta `sales Person`.
- I dati degli obiettivi di vendita associati, sono strutturati in un campo denominato `Target`.

### Script di caricamento

```
SalesTargets:
CROSTABLE([Sales Person],Target,1)
LOAD
*
INLINE [
Area, Lisa, James, Sharon
APAC, 1500, 1750, 1850
EMEA, 1350, 950, 2050
NA, 1800, 1200, 1350
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- Area
- Sales Person

Aggiungere questa misura:

```
=Sum(Target)
```

Tabella dei risultati

| Area | Venditore | =SOMMA(Target) |
|------|-----------|----------------|
| APAC | James     | 1750           |
| APAC | Lisa      | 1.500          |
| APAC | Sharon    | 1850           |
| EMEA | James     | 950            |
| EMEA | Lisa      | 1350           |
| EMEA | Sharon    | 2050           |
| N/D  | James     | 1200           |
| N/D  | Lisa      | 1800           |
| N/D  | Sharon    | 1350           |

Se si desidera replicare la visualizzazione dei dati come tabella pivot di input, è possibile creare una tabella pivot equivalente in un foglio.

### Procedere come indicato di seguito:

1. Copiare e incollare la tabella appena creata nel foglio.
2. Trascinare l'oggetto grafico **Tabella pivot** sopra la copia della tabella appena creata. Selezionare **Converti**.

### 3 Istruzioni e parole chiave dello script

---

3. Fare clic su **✓ Termina modifica**.
4. Trascinare il campo `sa1es Person` dalla casella della colonna verticale alla casella della colonna orizzontale.

La tabella seguente mostra i dati nella forma della tabella iniziale, come è visualizzato in Qlik Sense:

Tabella dei risultati originale, come mostrato in  
Qlik Sense

| Area   | Venditore | =SOMMA(Target) |
|--------|-----------|----------------|
| Totali | -         | 13800          |
| APAC   | James     | 1750           |
| APAC   | Lisa      | 1.500          |
| APAC   | Sharon    | 1850           |
| EMEA   | James     | 950            |
| EMEA   | Lisa      | 1350           |
| EMEA   | Sharon    | 2050           |
| N/D    | James     | 1200           |
| N/D    | Lisa      | 1800           |
| N/D    | Sharon    | 1350           |

La tabella pivot equivalente è simile a quella seguente, con la colonna per il nome di ogni dipendente addetto alle vendite contenuta all'interno della riga più grande per `sa1es Person`:

Tabella pivot equivalente con il campo `sa1es Person` ruotato orizzontalmente

| Area | James | Lisa  | Sharon |
|------|-------|-------|--------|
| APAC | 1750  | 1.500 | 1850   |
| EMEA | 950   | 1350  | 2050   |
| N/D  | 1350  | 1350  | 1350   |

### 3 Istruzioni e parole chiave dello script

Esempio di dati visualizzati come una tabella e una tabella pivot equivalente con il campo `Sales Person` ruotato orizzontalmente

| Table  |  |              | Pivot table |  |  |       |
|--------|--|--------------|-------------|--|--|-------|
| Area   |  | Sales Person |             |  |  |       |
| Totals |  |              |             |  |  | 13800 |
| APAC   |  | James        |             |  |  | 1750  |
| APAC   |  | Lisa         |             |  |  | 1500  |
| APAC   |  | Sharon       |             |  |  | 1850  |
| EMEA   |  | James        |             |  |  | 950   |
| EMEA   |  | Lisa         |             |  |  | 1350  |
| EMEA   |  | Sharon       |             |  |  | 2050  |
| NA     |  | James        |             |  |  | 1200  |
| NA     |  | Lisa         |             |  |  | 1800  |
| NA     |  | Sharon       |             |  |  | 1350  |

| Area |  | Sales Person |      |        |  |
|------|--|--------------|------|--------|--|
|      |  | James        | Lisa | Sharon |  |
| APAC |  | 1750         | 1500 | 1850   |  |
| EMEA |  | 950          | 1350 | 2050   |  |
| NA   |  | 1200         | 1800 | 1350   |  |

### Esempio 3 - Trasformazione dei dati e degli obiettivi di vendita pivot in una tabella con struttura verticale (avanzata)

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che rappresenta i dati sulle vendite e sugli obiettivi, organizzati per area e mese dell'anno. Questo viene caricato in una tabella denominata `SalesAndTargets`.
- Il prefisso di caricamento `crosstable`. Questo viene utilizzato per annullare il pivot della dimensione `Month Year` in un campo dedicato, nonché per trasporre la matrice delle vendite e degli importi `target` in un campo dedicato chiamato `Amount`.
- Conversione del campo `Month Year` da testo in una data corretta, utilizzando la funzione di conversione `text-to-date date#`. Il campo `Month Year` convertito in data viene unito nuovamente alla tabella `SalesAndTarget` tramite un prefisso di caricamento `join`.

#### Script di caricamento

`SalesAndTargets:`

```
CROSSTABLE(MonthYearAsText,Amount,2)
```

```
LOAD
```

```
*
```

```
INLINE [
```

| Area | Type   | Jan-22 | Feb-22 | Mar-22 | Apr-22 | May-22 | Jun-22 | Jul-22 | Aug-22 | Sep-22 | Oct-22 | Nov-22 | Dec-22 |
|------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| APAC | Target | 425    | 425    | 425    | 425    | 425    | 425    | 425    | 425    | 425    | 425    | 425    | 425    |
| APAC | Actual | 435    | 434    | 397    | 404    | 458    | 447    | 413    | 458    | 385    | 421    | 448    | 397    |

### 3 Istruzioni e parole chiave dello script

```
EMEA Target 362.5 362.5 362.5 362.5 362.5 362.5 362.5 362.5 362.5 362.5 362.5 362.5 362.5
EMEA Actual 363.5 359.5 337.5 361.5 341.5 337.5 379.5 352.5 327.5 337.5 360.5 334.5
NA Target 375 375 375 375 375 375 375 375 375 375 375 375 375
NA Actual 378 415 363 356 403 343 401 365 393 340 360 405
```

```
] (delimiter is '\t');
```

```
tmp:
```

```
LOAD DISTINCT MonthYearAsText,date#(MonthYearAsText,'MMM-YY') AS [Month Year]
```

```
RESIDENT SalesAndTargets;
```

```
JOIN (SalesAndTargets)
```

```
LOAD * RESIDENT tmp;
```

```
DROP TABLE tmp;
```

```
DROP FIELD MonthYearAsText;
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- Area
- Month Year

Creare le seguenti misure, con l'etichetta Actual:

```
=Sum({<Type={'Actual'}>} Amount)
```

Inoltre, creare questa misura, con l'etichetta Target:

```
=Sum({<Type={'Target'}>} Amount)
```

Tabella dei risultati (ritagliata)

| Area | Anno mese | Effettivo | Destinazione |
|------|-----------|-----------|--------------|
| APAC | Gen-22    | 435       | 425          |
| APAC | Feb-22    | 434       | 425          |
| APAC | Mar-22    | 397       | 425          |
| APAC | Apr-22    | 404       | 425          |
| APAC | Mag-22    | 458       | 425          |
| APAC | Giu-22    | 447       | 425          |
| APAC | Lug-22    | 413       | 425          |
| APAC | Ago-22    | 458       | 425          |
| APAC | Set-22    | 385       | 425          |
| APAC | Ott-22    | 421       | 425          |
| APAC | Nov-22    | 448       | 425          |

### 3 Istruzioni e parole chiave dello script

| Area | Anno mese | Effettivo | Destinazione |
|------|-----------|-----------|--------------|
| APAC | Dic-22    | 397       | 425          |
| EMEA | Gen-22    | 363.5     | 362,5        |
| EMEA | Feb-22    | 359.5     | 362,5        |

Se si desidera replicare la visualizzazione dei dati come tabella pivot di input, è possibile creare una tabella pivot equivalente in un foglio.

**Procedere come indicato di seguito:**

1. Copiare e incollare la tabella appena creata nel foglio.
2. Trascinare l'oggetto grafico **Tabella pivot** sopra la copia della tabella appena creata. Selezionare **Converti**.
3. Fare clic su **✓ Termina modifica**.
4. Trascinare il campo **Month Year** dalla casella della colonna verticale alla casella della colonna orizzontale.
5. Trascinare l'elemento **values** dalla casella della colonna verticale alla casella della colonna orizzontale.

La tabella seguente mostra i dati nella forma della tabella iniziale, come è visualizzato in Qlik Sense:

Tabella dei risultati originale (ritagliata), come mostrata in  
Qlik Sense

| Area   | Anno mese | Effettivo | Destinazione |
|--------|-----------|-----------|--------------|
| Totali | -         | 13812     | 13950        |
| APAC   | Gen-22    | 435       | 425          |
| APAC   | Feb-22    | 434       | 425          |
| APAC   | Mar-22    | 397       | 425          |
| APAC   | Apr-22    | 404       | 425          |
| APAC   | Mag-22    | 458       | 425          |
| APAC   | Giu-22    | 447       | 425          |
| APAC   | Lug-22    | 413       | 425          |
| APAC   | Ago-22    | 458       | 425          |
| APAC   | Set-22    | 385       | 425          |
| APAC   | Ott-22    | 421       | 425          |
| APAC   | Nov-22    | 448       | 425          |
| APAC   | Dic-22    | 397       | 425          |

### 3 Istruzioni e parole chiave dello script

| Area | Anno mese | Effettivo | Destinazione |
|------|-----------|-----------|--------------|
| EMEA | Gen-22    | 363.5     | 362,5        |
| EMEA | Feb-22    | 359.5     | 362,5        |

La tabella pivot equivalente è simile a quella seguente, con la colonna per ogni mese dell'anno contenuta all'interno della riga più grande per Month Year:

Tabella pivot (ritagliata) equivalente con il campo Month Year ruotato orizzontalmente

| Area (valori)       | Ge n-22 | Feb-22 | Mar-22 | Apr-22 | Ma g-22 | Giu-22 | Lug-22 | Ago-22 | Set-22 | Ott-22 | Nov-22 | Dic-22 |
|---------------------|---------|--------|--------|--------|---------|--------|--------|--------|--------|--------|--------|--------|
| APAC - Effettivo    | 435     | 434    | 397    | 404    | 458     | 447    | 413    | 458    | 385    | 421    | 448    | 397    |
| APAC - Destinazione | 425     | 425    | 425    | 425    | 425     | 425    | 425    | 425    | 425    | 425    | 425    | 425    |
| EMEA - Effettivo    | 363.5   | 359.5  | 337.5  | 361.5  | 341.5   | 337.5  | 379.5  | 352.5  | 327.5  | 337.5  | 360.5  | 334.5  |
| EMEA - Destinazione | 362,5   | 362,5  | 362,5  | 362,5  | 362,5   | 362,5  | 362,5  | 362,5  | 362,5  | 362,5  | 362,5  | 362,5  |
| N/D - Effettivo     | 378     | 415    | 363    | 356    | 403     | 343    | 401    | 365    | 393    | 340    | 360    | 405    |
| N/D - Destinazione  | 375     | 375    | 375    | 375    | 375     | 375    | 375    | 375    | 375    | 375    | 375    | 375    |

Esempio di dati visualizzati come una tabella e una tabella pivot equivalente con il campo Month Year ruotato orizzontalmente

| Table  |   |            |   | Pivot table |        |            |        |        |        |        |        |        |        |        |        |        |
|--------|---|------------|---|-------------|--------|------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Area   | Q | Month Year | Q | Actual      | Target | Month Year |        |        |        |        |        |        |        |        |        |        |
| Totals |   |            |   | 13812       | 13950  |            |        |        |        |        |        |        |        |        |        |        |
| APAC   |   | Jan-22     |   | 435         | 425    | Jan-22     | Feb-22 | Mar-22 | Apr-22 | May-22 | Jun-22 | Jul-22 | Aug-22 | Sep-22 | Oct-22 | Nov-22 |
| APAC   |   | Feb-22     |   | 434         | 425    | Jan-22     | Feb-22 | Mar-22 | Apr-22 | May-22 | Jun-22 | Jul-22 | Aug-22 | Sep-22 | Oct-22 | Nov-22 |
| APAC   |   | Mar-22     |   | 397         | 425    | Jan-22     | Feb-22 | Mar-22 | Apr-22 | May-22 | Jun-22 | Jul-22 | Aug-22 | Sep-22 | Oct-22 | Nov-22 |
| APAC   |   | Apr-22     |   | 404         | 425    | Jan-22     | Feb-22 | Mar-22 | Apr-22 | May-22 | Jun-22 | Jul-22 | Aug-22 | Sep-22 | Oct-22 | Nov-22 |
| APAC   |   | May-22     |   | 458         | 425    | Jan-22     | Feb-22 | Mar-22 | Apr-22 | May-22 | Jun-22 | Jul-22 | Aug-22 | Sep-22 | Oct-22 | Nov-22 |
| APAC   |   | Jun-22     |   | 447         | 425    | Jan-22     | Feb-22 | Mar-22 | Apr-22 | May-22 | Jun-22 | Jul-22 | Aug-22 | Sep-22 | Oct-22 | Nov-22 |
| APAC   |   | Jul-22     |   | 413         | 425    | Jan-22     | Feb-22 | Mar-22 | Apr-22 | May-22 | Jun-22 | Jul-22 | Aug-22 | Sep-22 | Oct-22 | Nov-22 |
| APAC   |   | Aug-22     |   | 458         | 425    | Jan-22     | Feb-22 | Mar-22 | Apr-22 | May-22 | Jun-22 | Jul-22 | Aug-22 | Sep-22 | Oct-22 | Nov-22 |
| APAC   |   | Sep-22     |   | 385         | 425    | Jan-22     | Feb-22 | Mar-22 | Apr-22 | May-22 | Jun-22 | Jul-22 | Aug-22 | Sep-22 | Oct-22 | Nov-22 |
| APAC   |   | Oct-22     |   | 421         | 425    | Jan-22     | Feb-22 | Mar-22 | Apr-22 | May-22 | Jun-22 | Jul-22 | Aug-22 | Sep-22 | Oct-22 | Nov-22 |
| APAC   |   | Nov-22     |   | 448         | 425    | Jan-22     | Feb-22 | Mar-22 | Apr-22 | May-22 | Jun-22 | Jul-22 | Aug-22 | Sep-22 | Oct-22 | Nov-22 |

## First

Il prefisso `First` aggiunto a un'istruzione `LOAD` o `SELECT` (SQL) viene utilizzato per caricare un numero di record massimo impostato dalla tabella di origine dei dati. Un tipico caso d'uso per l'utilizzo del prefisso `First` è quando si desidera recuperare un piccolo sottoinsieme di record da una fase di caricamento dei dati di grandi dimensioni e/o lenta. Non appena il numero di

## 3 Istruzioni e parole chiave dello script

record definito "n" viene caricato, la fase di caricamento termina prematuramente e il resto dell'esecuzione dello script continua normalmente.

### Sintassi:

```
First n ( loadstatement | selectstatement )
```

| Argomenti                       |                                                                                                                                                                       |
|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Argomento                       | Descrizione                                                                                                                                                           |
| n                               | Un'espressione arbitraria che restituisce un numero intero indicante il numero massimo di record da leggere. È possibile anche racchiudere n tra parentesi: (n).      |
| loadstatement   selectstatement | Il valore load statement/select statement che segue l'argomento n definirà la tabella specificata che deve essere caricata con il numero massimo di record impostato. |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione SET DateFormat nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

| Esempi di funzioni                  |                                                                             |
|-------------------------------------|-----------------------------------------------------------------------------|
| Esempio                             | Risultato                                                                   |
| FIRST 10 LOAD * from abc.csv;       | Questo esempio recupererà le prime dieci righe da un file Excel.            |
| FIRST (1) SQL SELECT * from orders; | Questo esempio recupererà la prima riga selezionata dal set di dati orders. |

### Esempio: Caricamento delle prime cinque righe

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati delle prime due settimane del 2020.
- La variabile `First` che indica all'applicazione di caricare solo i primi cinque record.

### Script di caricamento

```
Sales:
FIRST 5
LOAD
*
Inline [
date,sales
01/01/2020,6000
01/02/2020,3000
01/03/2020,6000
01/04/2020,8000
01/05/2020,5000
01/06/2020,7000
01/07/2020,3000
01/08/2020,5000
01/09/2020,9000
01/10/2020,5000
01/11/2020,7000
01/12/2020,7000
01/13/2020,7000
01/14/2020,7000
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere `date` come campo e `sum(sales)` come misura.

Tabella dei risultati

| Date       | sum(sales) |
|------------|------------|
| 01/01/2020 | 6000       |
| 01/02/2020 | 3000       |
| 01/03/2020 | 6000       |
| 01/04/2020 | 8000       |
| 01/05/2020 | 5000       |

Lo script carica solo i primi cinque record della tabella `sales`.

## Generic

Il prefisso di caricamento **Generic** consente la conversione dei dati modellati su entità-attributo-valore (EAV) in una struttura di tabella relazionale tradizionale e normalizzata. La modellazione EAV in alternativa è denominata "modellazione di dati generici" o "schema aperto".

### 3 Istruzioni e parole chiave dello script

Esempio di dati EAV modellati e tabella relazionale denormalizzata equivalente

| Product ID | Attribute | Value        |
|------------|-----------|--------------|
| 13         | Status    | Discontinued |
| 13         | Colour    | Brown        |
| 20         | Colour    | White        |
| 13         | Size      | 13-15        |
| 20         | Size      | 16-18        |



| Product ID | Status       | Colour | Size  |
|------------|--------------|--------|-------|
| 13         | Discontinued | Brown  | 13-15 |
| 20         |              | White  | 16-18 |

Esempio di dati EAV modellati e tabella relazionale denormalizzata equivalente

| Product ID | Attribute | Value        |
|------------|-----------|--------------|
| 13         | Status    | Discontinued |
| 13         | Colour    | Brown        |
| 20         | Colour    | White        |
| 13         | Size      | 13-15        |
| 20         | Size      | 16-18        |



| Product ID | Status       |
|------------|--------------|
| 13         | Discontinued |

| Product ID | Colour |
|------------|--------|
| 13         | Brown  |
| 20         | White  |

| Product ID | Size  |
|------------|-------|
| 13         | 13-15 |
| 20         | 16-18 |

Sebbene sia tecnicamente possibile caricare e analizzare i dati modellati EAV in Qlik, è spesso più semplice lavorare con una struttura di dati relazionale tradizionale equivalente.

**Sintassi:**

```
Generic( loadstatement | selectstatement )
```

I seguenti argomenti possono aiutarti a lavorare con questa funzione:

Argomenti correlati

| Argomento                                     | Descrizione                                                                                                                                                                                                                                                                                                                           |
|-----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">Crosstable (page 47)</a>          | Il prefisso di caricamento <code>Crosstable</code> trasforma i dati orientati orizzontalmente in dati orientati verticalmente. Da una prospettiva puramente funzionale, esegue la trasformazione opposta al prefisso di caricamento <code>Generic</code> , sebbene i prefissi servano in genere per casi d'uso completamente diversi. |
| <b>Database generici in Gestione dei dati</b> | I modelli di dati strutturati EAV sono ulteriormente descritti qui.                                                                                                                                                                                                                                                                   |

### Esempio 1 - Trasformazione di dati strutturati EAV con il prefisso di caricamento generico

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor di caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento contiene un set di dati che viene caricato in una tabella denominata `Transactions`. Il set di dati include un campo `data`. Viene utilizzata la definizione predefinita `MonthNames`.

#### Script di caricamento

```
Products:
Generic
Load * inline [
Product ID, Attribute, value
13, Status, Discontinued
13, Color, Brown
20, Color, white
13, Size, 13-15
20, Size, 16-18
2, Status, Discontinued
5, Color, Brown
2, Color, white
44, Color, Brown
45, Size, 16-18
45, Color, Brown
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: `color`.

Aggiungere questa misura:

```
=Count([Product ID])
```

Ora è possibile controllare il numero di prodotti per colore.

Tabella dei risultati

| Colore  | =Conteggio([ID prodotto]) |
|---------|---------------------------|
| Marrone | 4                         |
| Bianco  | 2                         |

### 3 Istruzioni e parole chiave dello script

Si noti la forma del modello dati, in cui ogni attributo è stato suddiviso in una tabella separata denominata in base al contrassegno della tabella di destinazione originale `Product`. Ogni tabella ha l'attributo come suffisso. Un esempio è `Product.Color`. I record di output dell'Attributo prodotto risultanti sono associati per `Product ID`.

*Rappresentazione dei risultati del sistema di visualizzazione modello dati*

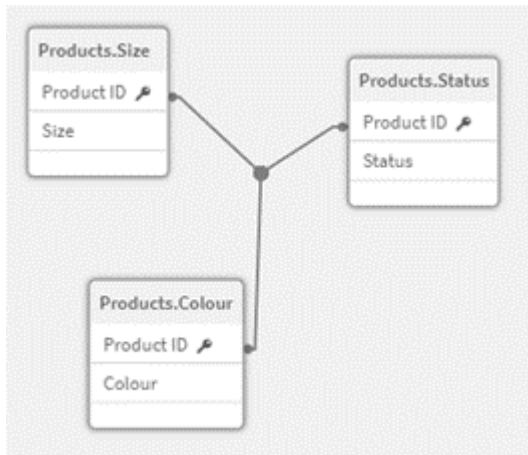


Tabella dei record  
risultante: `Products.Status`

| ID prodotto | Stato      |
|-------------|------------|
| 13          | Interrotto |
| 2           | Interrotto |

Tabella dei record  
risultante: `Products.Size`

| ID prodotto | Dimensioni |
|-------------|------------|
| 13          | 13-15      |
| 20          | 16-18      |
| 45          | 16-18      |

Tabella dei record  
risultante: `Products.Color`

| ID prodotto | Colore  |
|-------------|---------|
| 13          | Marrone |
| 5           | Marrone |
| 44          | Marrone |
| 45          | Marrone |

| ID prodotto | Colore |
|-------------|--------|
| 20          | Bianco |
| 2           | Bianco |

### Esempio 2 - Analisi dei dati strutturati EAV senza il prefisso di caricamento generico

Script di caricamento ed espressione del grafico

#### Panoramica

Questo esempio mostra come analizzare i dati strutturati EAV nella loro forma originale.

Aprire l'editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento contiene un set di dati che viene caricato in una tabella denominata `Products` in una struttura EAV.

In questo esempio, i prodotti continuano ad essere contati per attributo colore. Per analizzare i dati strutturati in questo modo, è necessario applicare un filtro a livello di espressione per i prodotti con il valore `Attributo color`.

Inoltre, i singoli attributi non sono disponibili per la selezione come dimensioni o campi, rendendo più difficile determinare come creare visualizzazioni efficaci.

#### Script di caricamento

```
Products:
Load * Inline
[
Product ID, Attribute, Value
13, Status, Discontinued
13, Color, Brown
20, Color, White
13, Size, 13-15
20, Size, 16-18
2, Status, Discontinued
5, Color, Brown
2, Color, White
44, Color, Brown
45, Size, 16-18
45, Color, Brown
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: `Value`.

### 3 Istruzioni e parole chiave dello script

---

Creare la seguente misura:

```
=Count({<Attribute={'Color'}>} [Product ID])
```

Ora è possibile controllare il numero di prodotti per colore.

Tabella dei record risultante: Products.Status

| Valore  | =Conteggio({<Attributo={'Colore'}>} [ID prodotto]) |
|---------|----------------------------------------------------|
| Marrone | 4                                                  |
| Bianco  | 2                                                  |

#### Esempio 3 - Denormalizzazione delle tabelle di output risultanti da un caricamento generico (avanzato)

Script di caricamento ed espressione del grafico

##### Panoramica

In questo esempio, viene mostrato come la struttura dati normalizzata prodotta dal prefisso di caricamento `Generic` può essere denormalizzata nuovamente in una tabella dimensionale `Product` consolidata. Questa è una tecnica di modellazione avanzata che può essere impiegata come parte dell'ottimizzazione delle prestazioni del modello dati.

Aprire l'editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

##### Script di caricamento

Products:

```
Generic
Load * inline [
Product ID, Attribute, Value
13, Status, Discontinued
13, Color, Brown
20, Color, white
13, Size, 13-15
20, Size, 16-18
2, Status, Discontinued
5, Color, Brown
2, Color, white
44, Color, Brown
45, Size, 16-18
45, Color, Brown
];
```

```
RENAME TABLE Products.Color TO Products;
```

```
OUTER JOIN (Products)
LOAD * RESIDENT Products.Size;
```

### 3 Istruzioni e parole chiave dello script

---

```
OUTER JOIN (Products)
LOAD * RESIDENT Products.Status;
DROP TABLES Products.Size,Products.Status;
```

#### Risultati

Aprire il sistema di visualizzazione modello dati e annotare la forma del modello di dati risultante. È presente una sola tabella denormalizzata. Questa è una combinazione delle tre tabelle di output intermedie: `Products.Size`, `Products.Status` e `Products.Color`.

Modello di dati  
interni risultante

| Prodotti    |
|-------------|
| ID prodotto |
| Stato       |
| Colore      |
| Dimensioni  |

Tabella dei record risultante: Products

| ID prodotto | Stato      | Colore  | Dimensioni |
|-------------|------------|---------|------------|
| 13          | Interrotto | Marrone | 13-15      |
| 20          | -          | Bianco  | 16-18      |
| 2           | Interrotto | Bianco  | -          |
| 5           | -          | Marrone | -          |
| 44          | -          | Marrone | -          |
| 45          | -          | Marrone | 16-18      |

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: `color`.

Aggiungere questa misura:

```
=Count([Product ID])
```

Tabella dei risultati

| Colore  | =Conteggio([ID Prodotto]) |
|---------|---------------------------|
| Marrone | 4                         |
| Bianco  | 2                         |

### Hierarchy

Il prefisso **hierarchy** viene utilizzato per trasformare una tabella gerarchica padre-figlio in una tabella utile in un modello dati Qlik Sense. Può essere inserito prima di un'istruzione **LOAD** o **SELECT** e utilizzerà i risultati dell'istruzione di caricamento come input per la trasformazione della tabella.

Il prefisso crea una tabella di nodi espansi che, in generale, presenta lo stesso numero di record della tabella di input, ma dove ogni livello all'interno della gerarchia viene memorizzato in un campo separato. Il campo del percorso può essere utilizzato in una struttura ad albero.

#### Sintassi:

```
Hierarchy (NodeID, ParentID, NodeName, [ParentName, [PathSource, [PathName, [PathDelimiter, Depth]]]]) (loadstatement | selectstatement)
```

La tabella di input deve essere una tabella di nodi adiacenti. Le tabelle di nodi adiacenti sono tabelle in cui ogni record corrisponde a un nodo e presenta un campo contenente un riferimento al nodo padre. In questa tabella il nodo è salvato solamente su un record, anche se può presentare un qualsiasi numero di figli. Ovviamente la tabella può contenere campi aggiuntivi che descrivono gli attributi dei nodi.

Il prefisso crea una tabella di nodi espansi che, in generale, presenta lo stesso numero di record della tabella di input, ma dove ogni livello all'interno della gerarchia viene memorizzato in un campo separato. Il campo del percorso può essere utilizzato in una struttura ad albero.

In generale, la tabella di input presenta esattamente un record per nodo; in questi casi, la tabella di output contiene lo stesso numero di record. Tuttavia, a volte esistono nodi con più padri, ad esempio un nodo è rappresentato da più record nella tabella di input. In questo caso, la tabella di output potrà avere più record della tabella di input.

Tutti i nodi con un ID padre non presente nella colonna dell'ID del nodo (inclusi i nodi con ID padre mancanti) verranno considerati nodi radice. Inoltre, verranno caricati solo i nodi con una connessione, diretta o indiretta, al nodo radice, in modo da evitare riferimenti circolari.

È anche possibile creare campi aggiuntivi contenenti il nome del nodo padre, il percorso del nodo e la relativa profondità.

#### Argomenti:

##### Argomenti

| Argomento | Descrizione                                                                                               |
|-----------|-----------------------------------------------------------------------------------------------------------|
| NodeID    | Il nome del campo contenente l'ID del nodo. Questo campo deve esistere nella tabella di input.            |
| ParentID  | Il nome del campo contenente l'ID nodo del nodo padre. Questo campo deve esistere nella tabella di input. |

### 3 Istruzioni e parole chiave dello script

| Argomento     | Descrizione                                                                                                                                                                                          |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NodeName      | Il nome del campo contenente il nome del nodo. Questo campo deve esistere nella tabella di input.                                                                                                    |
| ParentName    | Una stringa utilizzata per assegnare un nome al nuovo campo <b>ParentName</b> . Se omessa, il campo non verrà creato.                                                                                |
| ParentSource  | Il nome del campo contenente il nome del nodo utilizzato per creare il percorso del nodo. È un parametro opzionale. Se omesso, verrà utilizzato <b>NodeName</b> .                                    |
| PathName      | Una stringa utilizzata per assegnare un nome al nuovo campo <b>Path</b> , che contiene il percorso dalla radice al nodo. È un parametro opzionale. Se omessa, il campo non verrà creato.             |
| PathDelimiter | Una stringa utilizzata come delimitatore nel nuovo campo <b>Path</b> . È un parametro opzionale. Se omesso, sarà utilizzato il simbolo '/                                                            |
| Depth         | Una stringa utilizzata per assegnare un nome al nuovo campo <b>Depth</b> , il quale contiene la profondità del nodo nella gerarchia. È un parametro opzionale. Se omessa, il campo non verrà creato. |

#### Esempio:

```
Hierarchy(NodeID, ParentID, NodeName, ParentName, NodeName, PathName, '\', Depth) LOAD *  
inline [
```

```
NodeID, ParentID, NodeName
```

```
1, 4, London
```

```
2, 3, Munich
```

```
3, 5, Germany
```

```
4, 5, UK
```

```
5, , Europe
```

```
];
```

| Node ID | ParentID | NodeName | NodeName1 | NodeName2 | NodeName3 | ParentName | PathName              | Depth |
|---------|----------|----------|-----------|-----------|-----------|------------|-----------------------|-------|
| 1       | 4        | London   | Europe    | UK        | London    | UK         | Europe\UK\London      | 3     |
| 2       | 3        | Munich   | Europe    | Germany   | Munich    | Germany    | Europe\Germany\Munich | 3     |
| 3       | 5        | German   | Europe    | Germany   | -         | Europe     | Europe\Germany        | 2     |

|   |   |        |        |    |   |        |           |   |
|---|---|--------|--------|----|---|--------|-----------|---|
|   |   | y      |        |    |   |        |           |   |
| 4 | 5 | UK     | Europe | UK | - | Europe | Europe\UK | 2 |
| 5 |   | Europe | Europe | -  | - | -      | Europe    | 1 |

### HierarchyBelongsTo

Questo prefisso viene utilizzato per trasformare una tabella gerarchica padre-figlio in una tabella utile in un modello dati Qlik Sense. Può essere inserito prima di un'istruzione **LOAD** o **SELECT** e utilizzerà i risultati dell'istruzione di caricamento come input per la trasformazione della tabella.

Il prefisso consente di creare una tabella contenente tutte le relazioni padre-figlio della gerarchia. I campi padre possono essere quindi utilizzati per selezionare intere sezioni di tale gerarchia. Nella maggior parte dei casi, la tabella di output contiene più record per nodo.

#### Sintassi:

```
HierarchyBelongsTo (NodeID, ParentID, NodeName, AncestorID, AncestorName,  
[DepthDiff]) (loadstatement | selectstatement)
```

La tabella di input deve essere una tabella di nodi adiacenti. Le tabelle di nodi adiacenti sono tabelle in cui ogni record corrisponde a un nodo e presenta un campo contenente un riferimento al nodo padre. In questa tabella il nodo è salvato solamente su un record, anche se può presentare un qualsiasi numero di figli. Ovviamente la tabella può contenere campi aggiuntivi che descrivono gli attributi dei nodi.

Il prefisso consente di creare una tabella contenente tutte le relazioni padre-figlio della gerarchia. I campi padre possono essere quindi utilizzati per selezionare intere sezioni di tale gerarchia. Nella maggior parte dei casi, la tabella di output contiene più record per nodo.

Può essere creato un campo aggiuntivo contenente la differenza di profondità dei nodi.

#### Argomenti:

##### Argomenti

| Argomento  | Descrizione                                                                                               |
|------------|-----------------------------------------------------------------------------------------------------------|
| NodeID     | Il nome del campo contenente l'ID del nodo. Questo campo deve esistere nella tabella di input.            |
| ParentID   | Il nome del campo contenente l'ID nodo del nodo padre. Questo campo deve esistere nella tabella di input. |
| NodeName   | Il nome del campo contenente il nome del nodo. Questo campo deve esistere nella tabella di input.         |
| AncestorID | Una stringa utilizzata per assegnare un nome al nuovo campo ID padre, contenente l'ID del nodo padre.     |

### 3 Istruzioni e parole chiave dello script

| Argomento    | Descrizione                                                                                                                                                                                                              |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AncestorName | Una stringa utilizzata per assegnare un nome al nuovo campo padre, contenente il nome del nodo padre.                                                                                                                    |
| DepthDiff    | Una stringa utilizzata per assegnare un nome al nuovo campo <b>DepthDiff</b> , contenente la profondità del nodo nella gerarchia relativa al nodo padre. È un parametro opzionale. Se omessa, il campo non verrà creato. |

#### Esempio:

```
HierarchyBelongsTo (NodeID, AncestorID, NodeName, AncestorID, AncestorName, DepthDiff) LOAD *  
inline [
```

```
NodeID, AncestorID, NodeName
```

```
1, 4, London
```

```
2, 3, Munich
```

```
3, 5, Germany
```

```
4, 5, UK
```

```
5, , Europe
```

```
];
```

#### Results

| NodeID | AncestorID | NodeName | AncestorName | DepthDiff |
|--------|------------|----------|--------------|-----------|
| 1      | 1          | London   | London       | 0         |
| 1      | 4          | London   | UK           | 1         |
| 1      | 5          | London   | Europe       | 2         |
| 2      | 2          | Munich   | Munich       | 0         |
| 2      | 3          | Munich   | Germany      | 1         |
| 2      | 5          | Munich   | Europe       | 2         |
| 3      | 3          | Germany  | Germany      | 0         |
| 3      | 5          | Germany  | Europe       | 1         |
| 4      | 4          | UK       | UK           | 0         |
| 4      | 5          | UK       | Europe       | 1         |
| 5      | 5          | Europe   | Europe       | 0         |

### Inner

I prefissi **join** e **keep** possono essere preceduti dal prefisso **inner**. Se viene inserito prima di **join**, specifica che occorre utilizzare un'unione interna. La tabella risultante contiene solo le combinazioni di valori di campo estratte dalle tabelle di dati non elaborati dove i valori di campo di collegamento vengono rappresentati in entrambe le tabelle. Se utilizzato prima di **keep**, specifica che entrambe le tabelle di dati non elaborati devono essere ridotte alla loro intersezione comune prima di essere memorizzate in Qlik Sense.

#### Sintassi:

```
Inner ( Join | Keep ) [ (tablename) ] (loadstatement | selectstatement )
```

#### Argomenti:

##### Argomenti

| Argomento                           | Descrizione                                                       |
|-------------------------------------|-------------------------------------------------------------------|
| tablename                           | La tabella denominata da confrontare con la tabella caricata.     |
| loadstatementoppure selectstatement | L'istruzione <b>LOAD</b> o <b>SELECT</b> per la tabella caricata. |

#### Esempio

##### Script di caricamento

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

Table1:

```
Load * inline [
Column1, Column2
A, B
1, aa
2, cc
3, ee ];
```

Table2:

```
Inner Join Load * inline [
Column1, Column3
A, C
1, xx
4, yy ];
```

#### Risultato

##### Tabella risultante

| Colonna1 | Colonna2 | Colonna3 |
|----------|----------|----------|
| A        | B        | C        |
| 1        | aa       | xx       |

### Spiegazione

Questo esempio dimostra l'output Inner Join dove vengono uniti solo i valori presenti sia nella prima (sinistra) che nella seconda (destra) tabella.

### IntervalMatch

Il prefisso **IntervalMatch** consente di creare una tabella che corrisponde sia ai valori numerici discreti su uno o più intervalli numerici che, in modo opzionale, ai valori di una o più chiavi aggiuntive.

#### Sintassi:

```
IntervalMatch (matchfield) (loadstatement | selectstatement )
```

```
IntervalMatch (matchfield, keyfield1 [ , keyfield2, ... keyfield5 ] )  
(loadstatement | selectstatement )
```

Il prefisso **IntervalMatch** deve essere inserito prima di un'istruzione **LOAD** o **SELECT** che carica gli intervalli. Il campo che contiene i punti dati discreti (Time nell'esempio seguente) e chiavi aggiuntive deve essere già stato caricato in Qlik Sense prima dell'istruzione con il prefisso **IntervalMatch**. Il prefisso non è in grado di leggere questo campo dalla tabella del database, pertanto trasforma la tabella caricata degli intervalli e delle chiavi in una tabella contenente una colonna aggiuntiva: i punti dati numerici discreti. Inoltre, espande il numero di record in modo che la nuova tabella disponga di un record per ogni combinazione possibile di punti dati discreti, intervallo e valore dei campi chiave.

Gli intervalli possono sovrapporsi e i valori discreti saranno collegati a tutti gli intervalli corrispondenti.

Quando il prefisso **IntervalMatch** viene esteso con i campi chiave, consente di creare una tabella che corrisponde sia ai valori numerici discreti presenti su uno o più intervalli numerici sia ai valori di una o più chiavi aggiuntive.

Per evitare che i limiti degli intervalli non definiti vengano ignorati, potrebbe essere necessario consentire il mapping dei valori NULL sugli altri campi che costituiscono i limiti inferiore e superiore dell'intervallo. Questa operazione può essere eseguita dall'istruzione **NullAsValue** o da un test esplicito che sostituisce i valori NULL con un valore numerico prima o dopo qualsiasi dei punti dati numerici discreti.

#### Argomenti:

##### Argomenti

| Argomento  | Descrizione                                                                    |
|------------|--------------------------------------------------------------------------------|
| matchfield | Il campo contenente i valori numerici discreti da collegare agli intervalli.   |
| keyfield   | I campi contenenti gli attributi aggiuntivi da associare nella trasformazione. |

### 3 Istruzioni e parole chiave dello script

| Argomento                          | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| loadstatementor<br>selectstatement | Il risultato deve essere una tabella in cui il primo campo contiene il limite inferiore di ciascun intervallo, il secondo campo contiene il limite superiore di ciascun intervallo e, nel caso di utilizzo di una corrispondenza chiave, il terzo campo e quelli successivi contengono gli elementi keyfield presenti nell'istruzione <b>IntervalMatch</b> . Gli intervalli sono sempre chiusi, ossia i punti di fine sono inclusi nell'intervallo. I limiti non numerici fanno in modo che l'intervallo venga ignorato (non definito). |

#### Example 1:

Nelle due tabelle seguenti, la prima tabella contiene un elenco di eventi discreti, mentre la seconda definisce l'ora di inizio e l'ora di fine relative alla produzione di ordini differenti. Utilizzando il prefisso **IntervalMatch**, è possibile eseguire il collegamento logico delle due tabelle in modo da poter individuare, ad esempio, gli ordini che hanno subito interruzioni e gli ordini elaborati in base a turni specifici.

```
EventLog:
LOAD * Inline [
Time, Event, Comment
00:00, 0, Start of shift 1
01:18, 1, Line stop
02:23, 2, Line restart 50%
04:15, 3, Line speed 100%
08:00, 4, Start of shift 2
11:43, 5, End of production
];
```

```
OrderLog:
LOAD * INLINE [
Start, End, Order
01:00, 03:35, A
02:30, 07:58, B
03:04, 10:27, C
07:23, 11:43, D
];
```

```
//Link the field Time to the time intervals defined by the fields Start and End.
Inner Join IntervalMatch ( Time )
LOAD Start, End
Resident OrderLog;
```

La tabella **OrderLog** contiene ora una colonna aggiuntiva: *Time*. Anche il numero di record risulta espanso.

Table with additional column

| Time  | Start | End | Order |
|-------|-------|-----|-------|
| 00:00 | -     | -   | -     |

### 3 Istruzioni e parole chiave dello script

---

| Time  | Start | End   | Order |
|-------|-------|-------|-------|
| 01:18 | 01:00 | 03:35 | A     |
| 02:23 | 01:00 | 03:35 | A     |
| 04:15 | 02:30 | 07:58 | B     |
| 04:15 | 03:04 | 10:27 | C     |
| 08:00 | 03:04 | 10:27 | C     |
| 08:00 | 07:23 | 11:43 | D     |
| 11:43 | 07:23 | 11:43 | D     |

#### Example 2: (mediante keyfield)

Lo stesso esempio illustrato in precedenza, con l'aggiunta di *ProductionLine* come campo chiave.

EventLog:

```
LOAD * Inline [  
  
Time, Event, Comment, ProductionLine  
  
00:00, 0, Start of shift 1, P1  
  
01:00, 0, Start of shift 1, P2  
  
01:18, 1, Line stop, P1  
  
02:23, 2, Line restart 50%, P1  
  
04:15, 3, Line speed 100%, P1  
  
08:00, 4, Start of shift 2, P1  
  
09:00, 4, Start of shift 2, P2  
  
11:43, 5, End of production, P1  
  
11:43, 5, End of production, P2  
  
];
```

OrderLog:

```
LOAD * INLINE [  
  
Start, End, Order, ProductionLine  
  
01:00, 03:35, A, P1
```

### 3 Istruzioni e parole chiave dello script

---

```
02:30, 07:58, B, P1
```

```
03:04, 10:27, C, P1
```

```
07:23, 11:43, D, P2
```

```
];
```

```
//Link the field Time to the time intervals defined by the fields Start and End and match the values
```

```
// to the key ProductionLine.
```

```
Inner Join
```

```
IntervalMatch ( Time, ProductionLine )
```

```
LOAD Start, End, ProductionLine
```

```
Resident OrderLog;
```

È ora possibile creare una tabella come quella seguente:

Tablebox example

| ProductionLine | Time  | Event | Comment           | Order | Start | End   |
|----------------|-------|-------|-------------------|-------|-------|-------|
| P1             | 00:00 | 0     | Start of shift 1  | -     | -     | -     |
| P2             | 01:00 | 0     | Start of shift 1  | -     | -     | -     |
| P1             | 01:18 | 1     | Line stop         | A     | 01:00 | 03:35 |
| P1             | 02:23 | 2     | Line restart 50%  | A     | 01:00 | 03:35 |
| P1             | 04:15 | 3     | Line speed 100%   | B     | 02:30 | 07:58 |
| P1             | 04:15 | 3     | Line speed 100%   | C     | 03:04 | 10:27 |
| P1             | 08:00 | 4     | Start of shift 2  | C     | 03:04 | 10:27 |
| P2             | 09:00 | 4     | Start of shift 2  | D     | 07:23 | 11:43 |
| P1             | 11:43 | 5     | End of production | -     | -     | -     |
| P2             | 11:43 | 5     | End of production | D     | 07:23 | 11:43 |

## Join

Il prefisso **join** unisce la tabella caricata a una tabella denominata esistente oppure all'ultima tabella di dati creata in precedenza.

L'effetto dell'unione dei dati è quello di estendere la tabella di destinazione di un insieme aggiuntivo di campi o attributi, vale a dire quelli che non sono già presenti nella tabella di destinazione. Qualsiasi nome di campo comune tra il set di dati di origine e la tabella di destinazione viene utilizzato per determinare come associare i nuovi record in ingresso. Questa viene comunemente indicata come "unione naturale". Un'operazione Join Qlik può fare in modo che la tabella di destinazione risultante abbia più o meno record di quelli con cui è stata creata, a seconda dell'unicità dell'associazione join e del tipo di operazione join utilizzato.

Esistono quattro tipi di join:

### **Unione esterna a sinistra (Left outer join)**

Le operazioni Left join sono il tipo più utilizzato. Ad esempio, se si dispone di un set di dati per una transazione e si desidera combinarlo con un set di dati di riferimento, in genere si utilizza `Left Join`. Per prima cosa è necessario caricare la tabella delle transazioni, quindi caricare il set di dati di riferimento e unirli tramite un prefisso `Left Join` alla tabella delle transazioni già caricata. `Left Join` consente di mantenere tutte le transazioni così come sono e di aggiungere i campi dati di riferimento supplementari in cui viene trovata una corrispondenza.

### **Unione interna (Inner join)**

Quando si dispone di due set di dati in cui sono importanti solo i risultati in cui è presente un'associazione di corrispondenza, considerare l'utilizzo di `Inner Join`. Ciò consente di eliminare tutti i record sia dai dati di origine caricati che dalla tabella di destinazione se non viene trovata alcuna corrispondenza. Di conseguenza, l'operazione può la tabella di destinazione con meno record rispetto a prima che venisse completata l'operazione join.

### **Unione esterna (Outer join)**

Quando è necessario conservare sia i record di destinazione che tutti i record in entrata, utilizzare `outer join`. Se non viene trovata alcuna corrispondenza, ogni set di record viene comunque conservato, mentre i campi dal lato opposto del JOIN non verranno popolati (null).

Se la parola chiave del tipo viene omessa, il tipo di join predefinito è un outer join.

### **Unione esterna a destra (Right outer join)**

Questo tipo di join mantiene tutti i record che stanno per essere caricati, riducendo i record nella tabella di destinazione tramite l'operazione di join solo ai record in cui è presente una corrispondenza di associazione nei record in entrata. Questo è un tipo di join di nicchia che viene talvolta utilizzato come mezzo per ridurre una tabella di record già caricata precedentemente in un sottoinsieme richiesto.

### 3 Istruzioni e parole chiave dello script

Esempi di set di risultati da diversi tipi di operazioni join

| DATASETS                                                                                                                                                                   | OPERATION                                                                                           | OUTPUT                                                                                                                                                                            |          |              |        |             |                                                                                                     |                                                                                                                                                                                                                                        |          |             |           |        |              |     |        |             |  |       |  |           |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|--------------|--------|-------------|-----------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|-------------|-----------|--------|--------------|-----|--------|-------------|--|-------|--|-----------|
| <p>Target Table</p> <table><tr><th>Trade ID</th><th>Asset Class</th></tr><tr><td>101533</td><td>Fixed Income</td></tr><tr><td>606601</td><td>Commodities</td></tr></table> | Trade ID                                                                                            | Asset Class                                                                                                                                                                       | 101533   | Fixed Income | 606601 | Commodities | <p>LEFT JOIN</p>   | <table><tr><th>Trade ID</th><th>Asset Class</th><th></th></tr><tr><td>101533</td><td>Fixed Income</td><td>LSE</td></tr><tr><td>606601</td><td>Commodities</td><td></td></tr></table>                                                   | Trade ID | Asset Class |           | 101533 | Fixed Income | LSE | 606601 | Commodities |  |       |  |           |
| Trade ID                                                                                                                                                                   | Asset Class                                                                                         |                                                                                                                                                                                   |          |              |        |             |                                                                                                     |                                                                                                                                                                                                                                        |          |             |           |        |              |     |        |             |  |       |  |           |
| 101533                                                                                                                                                                     | Fixed Income                                                                                        |                                                                                                                                                                                   |          |              |        |             |                                                                                                     |                                                                                                                                                                                                                                        |          |             |           |        |              |     |        |             |  |       |  |           |
| 606601                                                                                                                                                                     | Commodities                                                                                         |                                                                                                                                                                                   |          |              |        |             |                                                                                                     |                                                                                                                                                                                                                                        |          |             |           |        |              |     |        |             |  |       |  |           |
| Trade ID                                                                                                                                                                   | Asset Class                                                                                         |                                                                                                                                                                                   |          |              |        |             |                                                                                                     |                                                                                                                                                                                                                                        |          |             |           |        |              |     |        |             |  |       |  |           |
| 101533                                                                                                                                                                     | Fixed Income                                                                                        | LSE                                                                                                                                                                               |          |              |        |             |                                                                                                     |                                                                                                                                                                                                                                        |          |             |           |        |              |     |        |             |  |       |  |           |
| 606601                                                                                                                                                                     | Commodities                                                                                         |                                                                                                                                                                                   |          |              |        |             |                                                                                                     |                                                                                                                                                                                                                                        |          |             |           |        |              |     |        |             |  |       |  |           |
|                                                                                                                                                                            | <p>INNER JOIN</p>  | <table><tr><th>Trade ID</th><th>Asset Class</th><th></th></tr><tr><td>101533</td><td>Fixed Income</td><td>LSE</td></tr></table>                                                   | Trade ID | Asset Class  |        | 101533      | Fixed Income                                                                                        | LSE                                                                                                                                                                                                                                    |          |             |           |        |              |     |        |             |  |       |  |           |
| Trade ID                                                                                                                                                                   | Asset Class                                                                                         |                                                                                                                                                                                   |          |              |        |             |                                                                                                     |                                                                                                                                                                                                                                        |          |             |           |        |              |     |        |             |  |       |  |           |
| 101533                                                                                                                                                                     | Fixed Income                                                                                        | LSE                                                                                                                                                                               |          |              |        |             |                                                                                                     |                                                                                                                                                                                                                                        |          |             |           |        |              |     |        |             |  |       |  |           |
| <p>Incoming Dataset</p> <table><tr><th>Trade ID</th><th>Exchange</th></tr><tr><td>101533</td><td>LSE</td></tr><tr><td>79052</td><td>Hong Kong</td></tr></table>            | Trade ID                                                                                            | Exchange                                                                                                                                                                          | 101533   | LSE          | 79052  | Hong Kong   | <p>OUTER JOIN</p>  | <table><tr><th>Trade ID</th><th>Asset Class</th><th></th></tr><tr><td>101533</td><td>Fixed Income</td><td>LSE</td></tr><tr><td>606601</td><td>Commodities</td><td></td></tr><tr><td>79052</td><td></td><td>Hong Kong</td></tr></table> | Trade ID | Asset Class |           | 101533 | Fixed Income | LSE | 606601 | Commodities |  | 79052 |  | Hong Kong |
| Trade ID                                                                                                                                                                   | Exchange                                                                                            |                                                                                                                                                                                   |          |              |        |             |                                                                                                     |                                                                                                                                                                                                                                        |          |             |           |        |              |     |        |             |  |       |  |           |
| 101533                                                                                                                                                                     | LSE                                                                                                 |                                                                                                                                                                                   |          |              |        |             |                                                                                                     |                                                                                                                                                                                                                                        |          |             |           |        |              |     |        |             |  |       |  |           |
| 79052                                                                                                                                                                      | Hong Kong                                                                                           |                                                                                                                                                                                   |          |              |        |             |                                                                                                     |                                                                                                                                                                                                                                        |          |             |           |        |              |     |        |             |  |       |  |           |
| Trade ID                                                                                                                                                                   | Asset Class                                                                                         |                                                                                                                                                                                   |          |              |        |             |                                                                                                     |                                                                                                                                                                                                                                        |          |             |           |        |              |     |        |             |  |       |  |           |
| 101533                                                                                                                                                                     | Fixed Income                                                                                        | LSE                                                                                                                                                                               |          |              |        |             |                                                                                                     |                                                                                                                                                                                                                                        |          |             |           |        |              |     |        |             |  |       |  |           |
| 606601                                                                                                                                                                     | Commodities                                                                                         |                                                                                                                                                                                   |          |              |        |             |                                                                                                     |                                                                                                                                                                                                                                        |          |             |           |        |              |     |        |             |  |       |  |           |
| 79052                                                                                                                                                                      |                                                                                                     | Hong Kong                                                                                                                                                                         |          |              |        |             |                                                                                                     |                                                                                                                                                                                                                                        |          |             |           |        |              |     |        |             |  |       |  |           |
|                                                                                                                                                                            | <p>RIGHT JOIN</p>  | <table><tr><th>Trade ID</th><th>Asset Class</th><th></th></tr><tr><td>101533</td><td>Fixed Income</td><td>LSE</td></tr><tr><td>79052</td><td></td><td>Hong Kong</td></tr></table> | Trade ID | Asset Class  |        | 101533      | Fixed Income                                                                                        | LSE                                                                                                                                                                                                                                    | 79052    |             | Hong Kong |        |              |     |        |             |  |       |  |           |
| Trade ID                                                                                                                                                                   | Asset Class                                                                                         |                                                                                                                                                                                   |          |              |        |             |                                                                                                     |                                                                                                                                                                                                                                        |          |             |           |        |              |     |        |             |  |       |  |           |
| 101533                                                                                                                                                                     | Fixed Income                                                                                        | LSE                                                                                                                                                                               |          |              |        |             |                                                                                                     |                                                                                                                                                                                                                                        |          |             |           |        |              |     |        |             |  |       |  |           |
| 79052                                                                                                                                                                      |                                                                                                     | Hong Kong                                                                                                                                                                         |          |              |        |             |                                                                                                     |                                                                                                                                                                                                                                        |          |             |           |        |              |     |        |             |  |       |  |           |



Se non ci sono nomi di campo in comune tra l'origine e la destinazione di un'operazione join, questa darà come risultato un prodotto cartesiano di tutte le righe, che viene definito "cross join".

Esempio di set di risultati da un'operazione di "cross join".

| DATASETS                                                                                                                                                                                             | OPERATION       | OUTPUT        |                 |        |     |      |        |     |      |                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                             |          |               |        |                 |      |        |     |      |     |      |        |     |      |     |      |        |     |      |     |      |        |     |      |     |      |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|---------------|-----------------|--------|-----|------|--------|-----|------|------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------|--------|-----------------|------|--------|-----|------|-----|------|--------|-----|------|-----|------|--------|-----|------|-----|------|--------|-----|------|-----|------|
| <p>Target Table</p> <table><tr><th>Trade ID</th><th>Base Currency</th><th>Amount</th></tr><tr><td>101533</td><td>EUR</td><td>1250</td></tr><tr><td>606601</td><td>EUR</td><td>1650</td></tr></table> | Trade ID        | Base Currency | Amount          | 101533 | EUR | 1250 | 606601 | EUR | 1650 | <p>JOIN (any type)</p>  | <table><tr><th>Trade ID</th><th>Base Currency</th><th>Amount</th><th>Target Currency</th><th>Rate</th></tr><tr><td>101533</td><td>EUR</td><td>1250</td><td>USD</td><td>1.08</td></tr><tr><td>101533</td><td>EUR</td><td>1250</td><td>GBP</td><td>0.84</td></tr><tr><td>606601</td><td>EUR</td><td>1650</td><td>USD</td><td>1.08</td></tr><tr><td>606601</td><td>EUR</td><td>1650</td><td>GBP</td><td>0.84</td></tr></table> | Trade ID | Base Currency | Amount | Target Currency | Rate | 101533 | EUR | 1250 | USD | 1.08 | 101533 | EUR | 1250 | GBP | 0.84 | 606601 | EUR | 1650 | USD | 1.08 | 606601 | EUR | 1650 | GBP | 0.84 |
| Trade ID                                                                                                                                                                                             | Base Currency   | Amount        |                 |        |     |      |        |     |      |                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                             |          |               |        |                 |      |        |     |      |     |      |        |     |      |     |      |        |     |      |     |      |        |     |      |     |      |
| 101533                                                                                                                                                                                               | EUR             | 1250          |                 |        |     |      |        |     |      |                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                             |          |               |        |                 |      |        |     |      |     |      |        |     |      |     |      |        |     |      |     |      |        |     |      |     |      |
| 606601                                                                                                                                                                                               | EUR             | 1650          |                 |        |     |      |        |     |      |                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                             |          |               |        |                 |      |        |     |      |     |      |        |     |      |     |      |        |     |      |     |      |        |     |      |     |      |
| Trade ID                                                                                                                                                                                             | Base Currency   | Amount        | Target Currency | Rate   |     |      |        |     |      |                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                             |          |               |        |                 |      |        |     |      |     |      |        |     |      |     |      |        |     |      |     |      |        |     |      |     |      |
| 101533                                                                                                                                                                                               | EUR             | 1250          | USD             | 1.08   |     |      |        |     |      |                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                             |          |               |        |                 |      |        |     |      |     |      |        |     |      |     |      |        |     |      |     |      |        |     |      |     |      |
| 101533                                                                                                                                                                                               | EUR             | 1250          | GBP             | 0.84   |     |      |        |     |      |                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                             |          |               |        |                 |      |        |     |      |     |      |        |     |      |     |      |        |     |      |     |      |        |     |      |     |      |
| 606601                                                                                                                                                                                               | EUR             | 1650          | USD             | 1.08   |     |      |        |     |      |                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                             |          |               |        |                 |      |        |     |      |     |      |        |     |      |     |      |        |     |      |     |      |        |     |      |     |      |
| 606601                                                                                                                                                                                               | EUR             | 1650          | GBP             | 0.84   |     |      |        |     |      |                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                             |          |               |        |                 |      |        |     |      |     |      |        |     |      |     |      |        |     |      |     |      |        |     |      |     |      |
| <p>Incoming Dataset</p> <table><tr><th>Target Currency</th><th>Rate</th></tr><tr><td>USD</td><td>1.08</td></tr><tr><td>GBP</td><td>0.84</td></tr></table>                                            | Target Currency | Rate          | USD             | 1.08   | GBP | 0.84 |        |     |      |                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                             |          |               |        |                 |      |        |     |      |     |      |        |     |      |     |      |        |     |      |     |      |        |     |      |     |      |
| Target Currency                                                                                                                                                                                      | Rate            |               |                 |        |     |      |        |     |      |                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                             |          |               |        |                 |      |        |     |      |     |      |        |     |      |     |      |        |     |      |     |      |        |     |      |     |      |
| USD                                                                                                                                                                                                  | 1.08            |               |                 |        |     |      |        |     |      |                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                             |          |               |        |                 |      |        |     |      |     |      |        |     |      |     |      |        |     |      |     |      |        |     |      |     |      |
| GBP                                                                                                                                                                                                  | 0.84            |               |                 |        |     |      |        |     |      |                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                             |          |               |        |                 |      |        |     |      |     |      |        |     |      |     |      |        |     |      |     |      |        |     |      |     |      |

Sintassi:

```
[inner | outer | left | right ]Join [ (tablename ) ] ( loadstatement | selectstatement )
```

### 3 Istruzioni e parole chiave dello script

#### Argomenti

| Argomento                           | Descrizione                                                       |
|-------------------------------------|-------------------------------------------------------------------|
| tablename                           | La tabella denominata da confrontare con la tabella caricata.     |
| loadstatementoppure selectstatement | L'istruzione <b>LOAD</b> o <b>SELECT</b> per la tabella caricata. |

I seguenti argomenti possono aiutarti a lavorare con questa funzione:

#### Argomenti correlati

| Argomento                                                                  | Descrizione                                                                                                                                                                                                                                             |
|----------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Combinazione di tabelle con Join e Keep</b> in <i>Gestione dei dati</i> | Questo argomento fornisce un'ulteriore spiegazione dei concetti di "unione" e "mantenimento" dei set di dati.                                                                                                                                           |
| <a href="#">Keep (page 84)</a>                                             | Il prefisso di caricamento <code>keep</code> è simile al prefisso <code>join</code> , ma non combina i set di dati di origine e di destinazione. Invece, ritaglia ogni set di dati in base al tipo di operazione adottata (inner, outer, left o right). |

### Esempio 1 - Unione esterna a sinistra: miglioramento dei dati di una tabella di destinazione con un set di dati di riferimento

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che rappresenta i record delle modifiche, che viene caricato in una tabella denominata `changes`. Questa include il campo chiave "Status ID".
- Un secondo set di dati che rappresenta gli stati di modifica, che viene caricato e combinato con i record di modifica originali unendoli con un prefisso di caricamento `join` a sinistra.

Questa operazione di left join assicura che i record di modifica rimangano intatti durante l'aggiunta di attributi di stato in cui viene trovata una corrispondenza nei record di stato in entrata in base a un ID di stato comune.

#### Script di caricamento

Changes:

Load \* inline [

| Change ID | Status ID  | Scheduled Start Date | Scheduled End Date | Business Impact |
|-----------|------------|----------------------|--------------------|-----------------|
| 10030 4   | 19/01/2022 | 23/02/2022           | None               |                 |
| 10015 3   | 04/01/2022 | 15/02/2022           | Low                |                 |

### 3 Istruzioni e parole chiave dello script

```
10103 1      02/04/2022      29/05/2022      Medium
10185 2      23/06/2022      08/09/2022      None
10323 1      08/11/2022      26/11/2022      High
10326 2      11/11/2022      05/12/2022      None
10138 2      07/05/2022      03/08/2022      None
10031 3      20/01/2022      25/03/2022      Low
10040 1      29/01/2022      22/04/2022      None
10134 1      03/05/2022      08/07/2022      Low
10334 2      19/11/2022      06/02/2023      Low
10220 2      28/07/2022      06/09/2022      None
10264 1      10/09/2022      17/10/2022      Medium
10116 1      15/04/2022      24/04/2022      None
10187 2      25/06/2022      24/08/2022      Low
```

```
] (delimiter is '\t');
```

Status:

Left Join (Changes)

Load \* inline [

Status ID Status Sub Status

1 Open Not Started

2 Open Started

3 Closed Completed

4 Closed Cancelled

5 Closed Obsolete

```
] (delimiter is '\t');
```

#### Risultati

Aprire il sistema di visualizzazione del modello dati e annotare la forma del modello di dati. È presente una sola tabella denormalizzata. Questa comprende una combinazione di tutti i record di modifica originali, con gli attributi di stato corrispondenti uniti a ciascun record di modifica.

Modello di dati interni  
risultante

| Modifiche                  |
|----------------------------|
| Cambia ID                  |
| ID stato                   |
| Data di inizio programmata |
| Data di fine programmata   |
| Impatto aziendale          |
| Stato                      |
| Stato secondario           |

Se si espande la finestra di anteprima nel sistema di visualizzazione modello dati, viene visualizzata una parte di questo set di risultati completo organizzato in una tabella:

### 3 Istruzioni e parole chiave dello script

Anteprima della tabella Modifiche nel sistema di visualizzazione modello dati

| Cambia ID | ID stato | Data di inizio programmata | Data di fine programmata | Impatto aziendale | Stato  | Stato secondario      |
|-----------|----------|----------------------------|--------------------------|-------------------|--------|-----------------------|
| 10030     | 4        | 19/01/2022                 | 23/02/2022               | Nessuno           | Chiuso | Annullato             |
| 10031     | 3        | 20/01/2022                 | 25/03/2022               | Scarso            | Chiuso | Operazione completata |
| 10015     | 3        | 04/01/2022                 | 15/02/2022               | Scarso            | Chiuso | Operazione completata |
| 10103     | 1        | 02/04/2022                 | 29/05/2022               | Medio             | Aperto | Non avviato           |
| 10116     | 1        | 15/04/2022                 | 24/04/2022               | Nessuno           | Apri   | Non avviato           |
| 10134     | 1        | 03/05/2022                 | 08/07/2022               | Scarso            | Apri   | Non avviato           |
| 10264     | 1        | 10/09/2022                 | 17/10/2022               | Medio             | Aperto | Non avviato           |
| 10040     | 1        | 29/01/2022                 | 22/04/2022               | Nessuno           | Apri   | Non avviato           |
| 10323     | 1        | 08/11/2022                 | 26/11/2022               | Elevati           | Aperto | Non avviato           |
| 10187     | 2        | 25/06/2022                 | 24/08/2022               | Scarso            | Aperto | Avviato               |
| 10185     | 2        | 23/06/2022                 | 08/09/2022               | Nessuno           | Aperto | Avviato               |
| 10220     | 2        | 28/07/2022                 | 06/09/2022               | Nessuno           | Aperto | Avviato               |
| 10326     | 2        | 11/11/2022                 | 05/12/2022               | Nessuno           | Aperto | Avviato               |
| 10138     | 2        | 07/05/2022                 | 03/08/2022               | Nessuno           | Aperto | Avviato               |
| 10334     | 2        | 19/11/2022                 | 06/02/2023               | Scarso            | Aperto | Avviato               |

Dato che la quinta riga nella tabella di stato (ID stato: "5", Stato: "Chiuso", Stato secondario: "Obsoleto") non corrisponde a nessuno dei record nella tabella delle modifiche, le informazioni in questa riga non appaiono nel set di risultati soprastante.

Tornare all'Editor caricamento dati. Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: status.

Aggiungere questa misura:

=Count([Change ID])

Ora è possibile verificare il numero di modifiche per stato.

Tabella dei risultati

| Stato  | =Count([Change ID]) |
|--------|---------------------|
| Aperto | 12                  |
| Chiuso | 3                   |

### Esempio 2 - Unione interna: combinazione solo dei record corrispondenti

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che rappresenta i record delle modifiche, che viene caricato in una tabella denominata `Changes`.
- Un secondo set di dati che rappresenta i record delle modifiche originate dal sistema di origine `JIRA`. Questo viene caricato e combinato con i record originali unendoli con un prefisso di caricamento `Inner Join`.

`Inner Join` garantisce che vengano mantenuti solo i cinque record delle modifiche che si trovano in entrambi i set di dati.

#### Script di caricamento

`Changes:`

`Load * inline [`

| Change ID | Status ID  | Scheduled Start Date | Scheduled End Date | Business Impact |
|-----------|------------|----------------------|--------------------|-----------------|
| 10030 4   | 19/01/2022 | 23/02/2022           | None               |                 |
| 10015 3   | 04/01/2022 | 15/02/2022           | Low                |                 |
| 10103 1   | 02/04/2022 | 29/05/2022           | Medium             |                 |
| 10185 2   | 23/06/2022 | 08/09/2022           | None               |                 |
| 10323 1   | 08/11/2022 | 26/11/2022           | High               |                 |
| 10326 2   | 11/11/2022 | 05/12/2022           | None               |                 |
| 10138 2   | 07/05/2022 | 03/08/2022           | None               |                 |
| 10031 3   | 20/01/2022 | 25/03/2022           | Low                |                 |
| 10040 1   | 29/01/2022 | 22/04/2022           | None               |                 |
| 10134 1   | 03/05/2022 | 08/07/2022           | Low                |                 |
| 10334 2   | 19/11/2022 | 06/02/2023           | Low                |                 |
| 10220 2   | 28/07/2022 | 06/09/2022           | None               |                 |
| 10264 1   | 10/09/2022 | 17/10/2022           | Medium             |                 |
| 10116 1   | 15/04/2022 | 24/04/2022           | None               |                 |
| 10187 2   | 25/06/2022 | 24/08/2022           | Low                |                 |

`] (delimiter is '\t');`

`JIRA_changes:`

`Inner Join (Changes)`

`Load`

`[Ticket ID] AS [Change ID],`

`[Source System]`

`inline`

`[`

| Ticket ID | Source System |
|-----------|---------------|
|-----------|---------------|

|       |      |
|-------|------|
| 10000 | JIRA |
|-------|------|

|       |      |
|-------|------|
| 10030 | JIRA |
|-------|------|

|       |      |
|-------|------|
| 10323 | JIRA |
|-------|------|

### 3 Istruzioni e parole chiave dello script

---

```
10134 JIRA
10334 JIRA
10220 JIRA
20000 TFS
] (delimiter is '\t');
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- Source System
- Change ID
- Business Impact

Ora è possibile esaminare i cinque record risultanti. La tabella risultante da Inner Join includerà solo i record con informazioni corrispondenti in entrambi i set di dati.

Tabella dei risultati

| Sistema di origine | Cambia ID | Impatto aziendale |
|--------------------|-----------|-------------------|
| JIRA               | 10030     | Nessuno           |
| JIRA               | 10134     | Scarso            |
| JIRA               | 10220     | Nessuno           |
| JIRA               | 10323     | Elevati           |
| JIRA               | 10334     | Bassa             |

### Esempio 3 - Unione interna: combinazione di set di record sovrapposti

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che rappresenta i record delle modifiche, che viene caricato in una tabella denominata Changes.
- Un secondo set di dati che rappresenta i record delle modifiche originate dal sistema di origine JIRA, che viene caricato e combinato con i record originali unendoli con un prefisso di caricamento outer Join.

Ciò garantisce che tutti i record delle modifiche sovrapposte di entrambi i set di dati vengano mantenuti.

### Script di caricamento

```
// 8 Change records
```

```
Changes:
```

```
Load * inline [
```

| Change ID | Status ID | Scheduled Start Date | Scheduled End Date | Business Impact |
|-----------|-----------|----------------------|--------------------|-----------------|
| 10030     | 4         | 19/01/2022           | 23/02/2022         | None            |
| 10015     | 3         | 04/01/2022           | 15/02/2022         | Low             |
| 10138     | 2         | 07/05/2022           | 03/08/2022         | None            |
| 10031     | 3         | 20/01/2022           | 25/03/2022         | Low             |
| 10040     | 1         | 29/01/2022           | 22/04/2022         | None            |
| 10134     | 1         | 03/05/2022           | 08/07/2022         | Low             |
| 10334     | 2         | 19/11/2022           | 06/02/2023         | Low             |
| 10220     | 2         | 28/07/2022           | 06/09/2022         | None            |

```
] (delimiter is '\t');
```

```
// 6 Change records
```

```
JIRA_changes:
```

```
Outer Join (Changes)
```

```
Load
```

```
    [Ticket ID] AS [Change ID],
```

```
    [Source System]
```

```
inline
```

```
[
```

| Ticket ID | Source System |
|-----------|---------------|
| 10030     | JIRA          |
| 10323     | JIRA          |
| 10134     | JIRA          |
| 10334     | JIRA          |
| 10220     | JIRA          |
| 10597     | JIRA          |

```
] (delimiter is '\t');
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- Source System
- Change ID
- Business Impact

Ora è possibile esaminare i 10 record risultanti.

Tabella dei risultati

| Sistema di origine | Cambia ID | Impatto aziendale |
|--------------------|-----------|-------------------|
| JIRA               | 10030     | Nessuno           |
| JIRA               | 10134     | Scarso            |
| JIRA               | 10220     | Nessuno           |

| Sistema di origine | Cambia ID | Impatto aziendale |
|--------------------|-----------|-------------------|
| JIRA               | 10323     | -                 |
| JIRA               | 10334     | Scarso            |
| JIRA               | 10597     | -                 |
| -                  | 10015     | Scarso            |
| -                  | 10031     | Scarso            |
| -                  | 10040     | Nessuno           |
| -                  | 10138     | Nessuno           |

#### Esempio 4 - Unione a destra: semplificazione di una tabella di destinazione da un set di dati master secondario

Script di caricamento e risultati

##### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che rappresenta i record delle modifiche, che viene caricato in una tabella denominata `Changes`.
- Un secondo set di dati che rappresenta i record di modifica provenienti dal sistema di origine `Teamwork`. Questo viene caricato e combinato con i record originali unendolo con un prefisso di caricamento `Right Join`.

Ciò garantisce che vengano mantenuti solo i record delle modifiche `Teamwork`, senza perdere alcun record `Teamwork` se la tabella di destinazione non ha nessuna corrispondenza `change ID`.

##### Script di caricamento

`Changes:`

```
Load * inline [
Change ID      Status ID      Scheduled Start Date      Scheduled End Date      Business Impact
10030 4        19/01/2022              23/02/2022              None
10015 3        04/01/2022              15/02/2022              Low
10103 1        02/04/2022              29/05/2022              Medium
10185 2        23/06/2022              08/09/2022              None
10323 1        08/11/2022              26/11/2022              High
10326 2        11/11/2022              05/12/2022              None
10138 2        07/05/2022              03/08/2022              None
10031 3        20/01/2022              25/03/2022              Low
10040 1        29/01/2022              22/04/2022              None
10134 1        03/05/2022              08/07/2022              Low
10334 2        19/11/2022              06/02/2023              Low
```

### 3 Istruzioni e parole chiave dello script

```
10220 2      28/07/2022    06/09/2022    None
10264 1      10/09/2022    17/10/2022    Medium
10116 1      15/04/2022    24/04/2022    None
10187 2      25/06/2022    24/08/2022    Low
] (delimiter is '\t');
```

```
Teamwork_changes:
Right Join (Changes)
Load
    [Ticket ID] AS [Change ID],
    [Source System]
inline
[
Ticket ID      Source System
10040 Teamwork
10015 Teamwork
10103 Teamwork
10031 Teamwork
50231 Teamwork
] (delimiter is '\t');
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- Source System
- Change ID
- Business Impact

Ora è possibile esaminare i cinque record risultanti.

Tabella dei risultati

| Sistema di origine | Cambia ID | Impatto aziendale |
|--------------------|-----------|-------------------|
| Lavoro in team     | 10015     | Scarso            |
| Lavoro in team     | 10031     | Scarso            |
| Lavoro in team     | 10040     | Nessuno           |
| Lavoro in team     | 10103     | Medio             |
| Lavoro in team     | 50231     | -                 |

#### Keep

Il prefisso **keep** è simile al prefisso **join**. Analogamente al prefisso **join**, confronta la tabella caricata con una tabella denominata esistente o con l'ultima tabella dati creata in precedenza, tuttavia, invece di unire la tabella caricata alla tabella esistente, riduce una o entrambe le due tabelle prima che vengano memorizzate in Qlik Sense, in base all'intersezione dei dati della tabella. Il confronto effettuato equivale a un'unione naturale effettuata su tutti i campi comuni, ad esempio nello stesso modo di un'unione corrispondente. In ogni modo, le due tabelle non vengono unite e verranno conservate in Qlik Sense come due tabelle denominate separatamente.

### Sintassi:

```
(inner | left | right) keep [(tablename ) ]( loadstatement | selectstatement )
```

Il prefisso **keep** deve essere preceduto da uno dei prefissi seguenti: **inner**, **left** o **right**.

Il prefisso esplicito **join** nel linguaggio di script di Qlik Sense consente di eseguire un'unione completa delle due tabelle. Il risultato è una sola tabella. In alcuni casi tale operazione di unione produce tabelle di dimensioni notevoli. Una delle principali funzioni di Qlik Sense è la capacità di generare associazioni tra più tabelle invece di unirle, un'operazione che riduce notevolmente l'utilizzo della memoria, aumenta le prestazioni e offre una notevole flessibilità. In generale, si sconsiglia di utilizzare operazioni di unione esplicite negli script di Qlik Sense. La funzionalità **keep** è stata studiata proprio per ridurre le situazioni in cui occorre utilizzare operazioni di unione esplicite.

### Argomenti:

| Argomenti                           |                                                                   |
|-------------------------------------|-------------------------------------------------------------------|
| Argomento                           | Descrizione                                                       |
| tablename                           | La tabella denominata da confrontare con la tabella caricata.     |
| loadstatementoppure selectstatement | L'istruzione <b>LOAD</b> o <b>SELECT</b> per la tabella caricata. |

### Esempio:

```
Inner Keep LOAD * from abc.csv;

Left Keep SELECT * from table1;

tab1:

LOAD * from file1.csv;

tab2:

LOAD * from file2.csv;

.. .. .

Left Keep (tab1) LOAD * from file3.csv;
```

## Left

I prefissi **Join** e **Keep** possono essere preceduti dal prefisso **left**.

Se viene inserito prima di **join**, specifica che occorre utilizzare un'unione sinistra. La tabella risultante conterrà solo le combinazioni di valori di campo estratte dalle tabelle di dati non elaborati, dove i valori di

### 3 Istruzioni e parole chiave dello script

campo di collegamento vengono rappresentati nella prima tabella. Se viene utilizzato prima di **keep**, specifica che la seconda tabella di dati non elaborati deve essere ridotta alla sua intersezione comune con la prima tabella prima di essere memorizzata in Qlik Sense.



*Si stava cercando la funzione di stringa con lo stesso nome? Vedere: [Left \(page 1786\)](#)*

#### Sintassi:

```
Left ( Join | Keep ) [ (tablename) ] (loadstatement | selectstatement)
```

#### Argomenti:

##### Argomenti

| Argomento                           | Descrizione                                                       |
|-------------------------------------|-------------------------------------------------------------------|
| tablename                           | La tabella denominata da confrontare con la tabella caricata.     |
| loadstatementoppure selectstatement | L'istruzione <b>LOAD</b> o <b>SELECT</b> per la tabella caricata. |

#### Esempio

##### Script di caricamento

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

Table1:

```
Load * inline [
Column1, Column2
A, B
1, aa
2, cc
3, ee ];
```

Table2:

```
Left Join Load * inline [
Column1, Column3
A, C
1, xx
4, yy ];
```

#### Risultato

##### Tabella risultante

| Colonna1 | Colonna2 | Colonna3 |
|----------|----------|----------|
| A        | B        | C        |
| 1        | aa       | xx       |
| 2        | cc       | -        |
| 3        | ee       | -        |

### Spiegazione

Questo esempio dimostra l'output Left Join dove vengono uniti solo i valori presenti nella prima tabella (sinistra).

### Mapping

Il prefisso **mapping** consente di creare una tabella di mapping che può essere utilizzata, ad esempio, per sostituire i valori di campo e i nomi di campo durante l'esecuzione dello script.

#### Sintassi:

```
Mapping( loadstatement | selectstatement )
```

Il prefisso **mapping** può essere inserito prima di un'istruzione **LOAD** o **SELECT** e consente di memorizzare i risultati dell'istruzione di caricamento come tabella di mapping. Il mapping rappresenta un metodo efficiente per la sostituzione dei valori di campo durante l'esecuzione dello script, ad esempio la sostituzione di US, U.S. o America con USA. Una tabella di mapping è composta da due colonne: la prima contiene dei valori di confronto, mentre la seconda contiene i valori di mapping desiderati. Le tabelle di mapping vengono salvate temporaneamente in memoria e vengono eliminate automaticamente una volta eseguito lo script.

È possibile accedere al contenuto della tabella di mapping utilizzando l'istruzione **Map ... Using**, l'istruzione **Rename Field**, la funzione **Applymap()** o la funzione **Mapsubstring()**.

#### Esempio:

In questo esempio viene caricato un elenco del personale addetto alle vendite con un codice paese che ne identifica il paese di residenza. Viene utilizzata una tabella per il mapping di un codice paese a un paese al fine di per sostituire il codice paese con il relativo nome. Nella tabella di mapping vengono definiti solo tre paesi, mentre gli altri codici paese vengono mappati a 'Rest of the world'.

```
// Load mapping table of country codes:
map1:
mapping LOAD *
inline [
CCode, Country
Sw, Sweden
Dk, Denmark
No, Norway
] ;
// Load list of salesmen, mapping country code to country
// If the country code is not in the mapping table, put Rest of the world
Salespersons:
LOAD *,
ApplyMap('map1', CCode,'Rest of the world') As Country
inline [
CCode, Salesperson
Sw, John
Sw, Mary
```

### 3 Istruzioni e parole chiave dello script

```
Sw, Per  
Dk, Preben  
Dk, Olle  
No, Ole  
Sf, Risttu] ;  
// We don't need the CCode anymore  
Drop Field 'CCode';  
La tabella risultante avrà l'aspetto seguente:
```

Mapping table

| Salesperson | Country           |
|-------------|-------------------|
| John        | Sweden            |
| Mary        | Sweden            |
| Per         | Sweden            |
| Preben      | Denmark           |
| Olle        | Denmark           |
| Ole         | Norway            |
| Risttu      | Rest of the world |

## Unisci

Il prefisso **Merge** può essere aggiunto a qualsiasi istruzione **LOAD** o **SELECT** nello script per specificare che dovrebbe aggiungere record a un'altra tabella. Specifica anche che questa istruzione dovrebbe essere eseguita in un ricaricamento parziale.

Il caso tipico di utilizzo riguarda il caricamento di un registro dei cambiamenti che si desidera utilizzare per applicare inserts, updates e deletes a una tabella esistente.



*Affinché il ricaricamento parziale funzioni correttamente, aprire l'app con i dati prima di attivare un ricaricamento parziale.*

Eseguire un ricaricamento parziale usando il pulsante **Ricarica**. È anche possibile utilizzare Qlik Engine JSON API.

#### Sintassi:

```
Merge [only] [(SequenceNoField [, SequenceNoVar])] On ListOfKeys [Concatenate  
[(TableName)]] (loadstatement | selectstatement)
```

### Argomenti:

Argomenti

| Argomento       | Descrizione                                                                                                                                                                                      |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| only            | Un qualificatore opzionale che denota che l'istruzione dovrebbe essere eseguita solo durante i caricamenti parziali. L'istruzione viene ignorata durante i ricaricamenti normali (non parziali). |
| SequenceNoField | Il nome del campo contenente una data e ora o un numero di sequenza che definisce l'ordine delle operazioni.                                                                                     |
| SequenceNoVar   | Il nome della variabile che viene assegnata al valore massimo per SequenceNoField della tabella unita.                                                                                           |
| ListOfKeys      | Un elenco separato da virgole di nomi di campo che specifica la chiave primaria.                                                                                                                 |
| Operation       | Il primo campo dell'istruzione LOAD deve contenere l'operazione come una stringa di testo: sono accettati anche 'Insert', 'Update' o 'Delete'. 'i', 'u' e 'd'.                                   |

### Funzionalità generale

Durante un caricamento normale (non parziale), la costruzione **LOAD Unisci** funziona come una normale istruzione **Load** ma con la funzionalità aggiuntiva di rimuovere i record vecchi e obsoleti e i record contrassegnati per l'eliminazione. Il primo campo dell'istruzione **Load** deve presentare informazioni sull'operazione: Insert, Update o Delete.

Per ciascun record caricato, l'identificatore record verrà confrontato con i record precedentemente caricati e verrà conservato solo il record più recente (in base al numero di sequenza). Se il record più recente è contrassegnato con Delete, non ne verrà conservato nessuno.

### Tabella di destinazione

Quale tabella modificare è determinato dal set di campi. Se una tabella con lo stesso set di campi (tranne il primo campo; l'operazione) esiste già, sarà questa la tabella da modificare. In alternativa, un prefisso **Concatena** può essere utilizzato per specificare la tabella. Se la tabella di destinazione non viene determinata, il risultato della costruzione **Unisci LOAD** viene archiviato in una nuova tabella.

Se viene utilizzato il prefisso Concatena, la tabella risultante ha un insieme di campi corrispondenti all'unione della tabella esistente e l'input per unire. Pertanto, la tabella di destinazione può avere più campi rispetto al registro delle modifiche utilizzato come input per l'unione.

Un ricaricamento parziale consente di ottenere lo stesso risultato di un ricaricamento completo. Una differenza è che un ricaricamento parziale raramente crea una nuova tabella. A meno che non sia stata utilizzata la clausola **Solo**, esiste sempre una tabella di destinazione con lo stesso set di campi dell'esecuzione precedente dello script.

### Numero di sequenza

Se il registro delle modifiche caricato è di tipo accumulato, ovvero contiene modifiche già caricate, il parametro `SequenceNoVar` può essere utilizzato in una clausola **Where** per limitare la quantità di dati di input. La costruzione **Unisci LOAD** può quindi essere realizzata per caricare soltanto i record in cui il campo `SequenceNoField` è superiore a `SequenceNoVar`. Al completamento, la costruzione **Unisci LOAD** assegna un nuovo valore a `SequenceNoVar` con il valore massimo visto nel campo `SequenceNoField`.

### Operazioni

**Unisci LOAD** può avere meno campi rispetto alla tabella di destinazione. Le operazioni diverse trattano i campi mancanti in modo diverso:

**Inserisci:** i campi mancanti in **Unisci LOAD**, ma esistenti nella tabella di destinazione, aggiungono un valore NULL nella tabella di destinazione.

**Elimina:** i campi mancanti non influenzano il risultato. I record rilevanti vengono eliminati comunque.

**Aggiorna:** i campi elencati in **Unisci LOAD** sono aggiornati nella tabella di destinazione. I campi mancanti non vengono modificati. Ciò significa che le due istruzioni seguenti non sono identiche:

- Merge on Key Concatenate Load 'U' as Operation, Key, F1, Null() as F2 From ...;
- Merge on Key Concatenate Load 'U' as Operation, Key, F1 From ...;

La prima istruzione aggiorna i record elencati e cambia F2 in NULL. La seconda non cambia F2, ma lascia i valori nella tabella di destinazione.

### Esempi

#### Esempio 1: unione semplice con una tabella specificata

In questo esempio, una tabella inline denominata `Persons` è caricata con tre righe. **Unisci** quindi modifica la tabella nel modo seguente:

- Aggiunge la riga, *Mary*, 4.
- Elimina la riga, *Steven*, 3.
- Assegna il numero 5 a *Jake*.

La variabile `LastChangeDate` è impostata al valore massimo nella colonna `ChangeDate` dopo l'esecuzione di **Unisci**.

### Script di caricamento

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

```
Set DateFormat='D/M/YYYY';
Persons:
load * inline [
Name, Number
Jake, 3
Jill, 2
Steven, 3
```

### 3 Istruzioni e parole chiave dello script

---

```
];
```

```
Merge (ChangeDate, LastChangeDate) on Name Concatenate(Persons)
```

```
LOAD * inline [
```

```
Operation, ChangeDate, Name, Number
```

```
Insert, 1/1/2021, Mary, 4
```

```
Delete, 1/1/2021, Steven,
```

```
Update, 2/1/2021, Jake, 5
```

```
];
```

#### Risultato

Prima di **Unisci LOAD**, la tabella risultante appare come segue:

Resulting table

| Name   | Number |
|--------|--------|
| Jake   | 3      |
| Jill   | 2      |
| Steven | 3      |

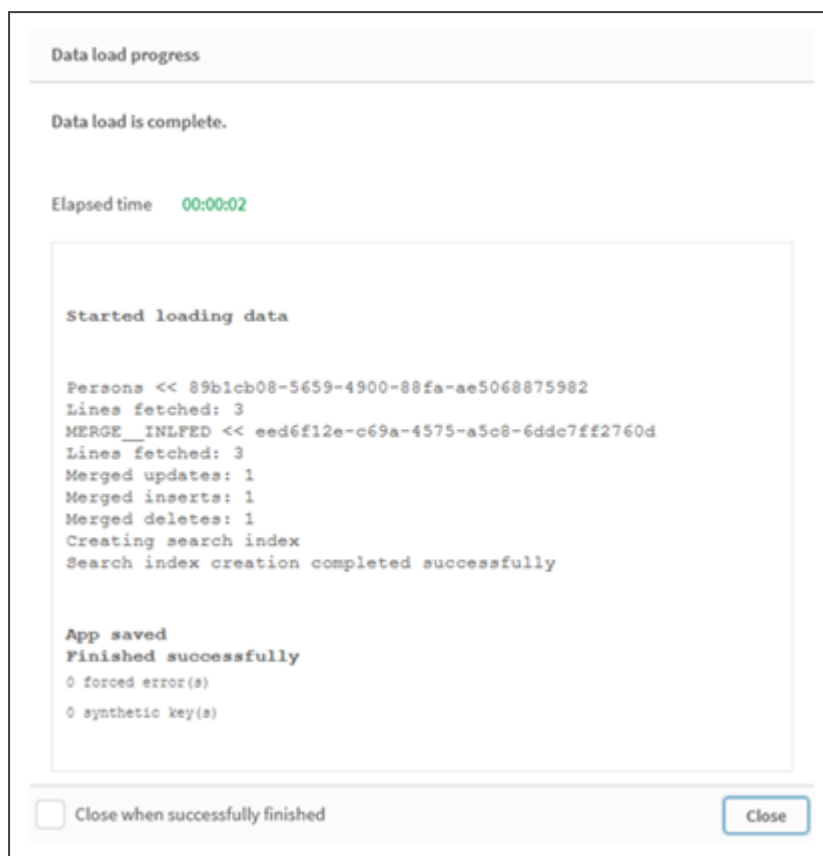
A seguito del comando **Unisci LOAD**, la tabella appare come segue:

Resulting table

| ChangeDate | Name | Number |
|------------|------|--------|
| 2/1/2021   | Jake | 5      |
| -          | Jill | 2      |
| 1/1/2021   | Mary | 4      |

Quando i dati vengono caricati, la finestra di dialogo **Avanzamento caricamento dati** mostra le operazioni che vengono eseguite:

*Finestra di dialogo Avanzamento caricamento dati*



### Esempio 2: script di caricamento dei dati con campi mancanti

In questo esempio, vengono caricati gli stessi dati di cui sopra, ma ora con un ID per ciascuna persona.

**Unisci** cambia la tabella nel modo seguente:

- Aggiunge la riga, *Mary*, 4.
- Elimina la riga, *Steven*, 3.
- Assegna il numero 5 a *Jake*.
- Assegna il numero 6 a *Jill*.

### Script di caricamento

Qui utilizziamo due istruzioni **Unisci LOAD**, una per 'Inserisci' ed 'Elimina' e una per 'Aggiorna'.

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

```
Set DateFormat='D/M/YYYY';
Persons:
Load * Inline [
PersonID, Name, Number
1, Jake, 3
2, Jill, 2
3, Steven, 3
];
```

### 3 Istruzioni e parole chiave dello script

```
Merge (ChangeDate, LastChangeDate) on PersonID Concatenate(Persons)
Load * Inline [
Operation, ChangeDate, PersonID, Name, Number
Insert, 1/1/2021, 4, Mary, 4
Delete, 1/1/2021, 3, Steven,
];
```

```
Merge (ChangeDate, LastChangeDate) on PersonID Concatenate(Persons)
Load * Inline [
Operation, ChangeDate, PersonID, Number
Update, 2/1/2021, 1, 5
Update, 3/1/2021, 2, 6
];
```

#### Risultato

Seguendo le istruzioni **Unisci LOAD**, la tabella avrà il seguente aspetto:

Resulting table

| PersonID | ChangeDate | Name | Number |
|----------|------------|------|--------|
| 1        | 2/1/2021   | Jake | 5      |
| 2        | 3/1/2021   | Jill | 6      |
| 4        | 1/1/2021   | Mary | 4      |

Notare che la seconda istruzione **Unisci** non include il campo **Nome**, e come conseguenza, i nomi non sono stati cambiati.

#### Esempio 3: script di caricamento dei dati - ricaricamento parziale usando un'istruzione Where con ChangeDate

Nell'esempio seguente, l'argomento **Solo** specifica che il comando **Unisci** viene eseguito solo durante un ricaricamento parziale. Gli aggiornamenti vengono filtrati in base al valore LastChangeDate precedentemente acquisito. Al termine dell'operazione **Unisci**, alla variabile LastChangeDate è assegnato il valore massimo della colonna ChangeDate elaborato durante l'unione

#### Script di caricamento

```
Merge Only (ChangeDate, LastChangeDate) on Name Concatenate(Persons)
LOAD Operation, ChangeDate, Name, Number
from [lib://ChangeFilesFolder/BulkChangesInPersonsTable.csv] (txt)
where ChangeDate >='$(LastChangeDate)';
```

#### NoConcatenate

Con il prefisso **NoConcatenate**, due tabelle che vengono caricate con gruppi di campo identici verranno considerate come due tabelle interne separate, invece di venire concatenate automaticamente.

#### Sintassi:

```
NoConcatenate ( loadstatement | selectstatement )
```

### 3 Istruzioni e parole chiave dello script

Per impostazione predefinita, se viene caricata una tabella che contiene un numero identico di campi e nomi di campi corrispondenti a una tabella caricata in precedenza nello script, queste due tabelle verranno concatenate automaticamente da Qlik Sense. Questo accade anche se la seconda tabella ha un nome diverso.

Tuttavia, se il prefisso dello script `NoConcatenate` viene incluso prima dell'istruzione `LOAD` o `SELECT` della seconda tabella, le due tabelle verranno caricate separatamente.

Un caso d'uso tipico per `NoConcatenate` è quello in cui è necessario creare una copia temporanea di una tabella per eseguire alcune trasformazioni temporanee su tale copia, conservando una copia dei dati originali. `NoConcatenate` assicura che sia possibile creare tale copia senza aggiungerla implicitamente alla tabella di origine.

#### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

Esempio di funzione

| Esempio                                                                                            | Risultato                                                                                                                                                                     |
|----------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Source: LOAD A,B from<br>file1.csv;<br>CopyOfSource:<br>NoConcatenate LOAD A,B<br>resident Source; | Viene caricata una tabella con A e B come misure. Una seconda tabella con gli stessi campi viene caricata separatamente utilizzando la variabile <code>NoConcatenate</code> . |

#### Esempio 1 - Concatenazione implicita

Script di caricamento e risultati

##### Panoramica

In questo esempio verranno aggiunti due script di caricamento in ordine sequenziale.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati iniziale con date e importi che viene inviato a una tabella denominata `Transactions`.

### Primo script di caricamento

```
Transactions:
LOAD
*
Inline [
id, date, amount
1, 08/30/2018, 23.56
2, 09/07/2018, 556.31
3, 09/16/2018, 5.75
4, 09/22/2018, 125.00
5, 09/22/2018, 484.21
6, 09/22/2018, 59.18
7, 09/23/2018, 177.42
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `id`
- `date`
- `amount`

Prima tabella dei risultati

| id | date       | importo |
|----|------------|---------|
| 1  | 08/30/2018 | 23.56   |
| 2  | 09/07/2018 | 556.31  |
| 3  | 09/16/2018 | 5.75    |
| 4  | 09/22/2018 | 125.00  |
| 5  | 09/22/2018 | 484.21  |
| 6  | 09/22/2018 | 59.18   |
| 7  | 09/23/2018 | 177.42  |

### Secondo script di caricamento

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un secondo set di dati con campi identici viene inviato a una tabella denominata `sa1es`.

```
sa1es:
LOAD
*
Inline [
```

```
id, date, amount
8, 10/01/2018, 164.27
9, 10/03/2018, 384.00
10, 10/06/2018, 25.82
11, 10/09/2018, 312.00
12, 10/15/2018, 4.56
13, 10/16/2018, 90.24
14, 10/18/2018, 19.32
];
```

#### Risultati

Caricare i dati e passare alla tabella.

Seconda tabella dei risultati

| id | date       | importo |
|----|------------|---------|
| 1  | 08/30/2018 | 23.56   |
| 2  | 09/07/2018 | 556.31  |
| 3  | 09/16/2018 | 5.75    |
| 4  | 09/22/2018 | 125.00  |
| 5  | 09/22/2018 | 484.21  |
| 6  | 09/22/2018 | 59.18   |
| 7  | 09/23/2018 | 177.42  |
| 8  | 10/01/2018 | 164.27  |
| 9  | 10/03/2018 | 384.00  |
| 10 | 10/06/2018 | 25.82   |
| 11 | 10/09/2018 | 312.00  |
| 12 | 10/15/2018 | 4.56    |
| 13 | 10/16/2018 | 90.24   |
| 14 | 10/18/2018 | 19.32   |

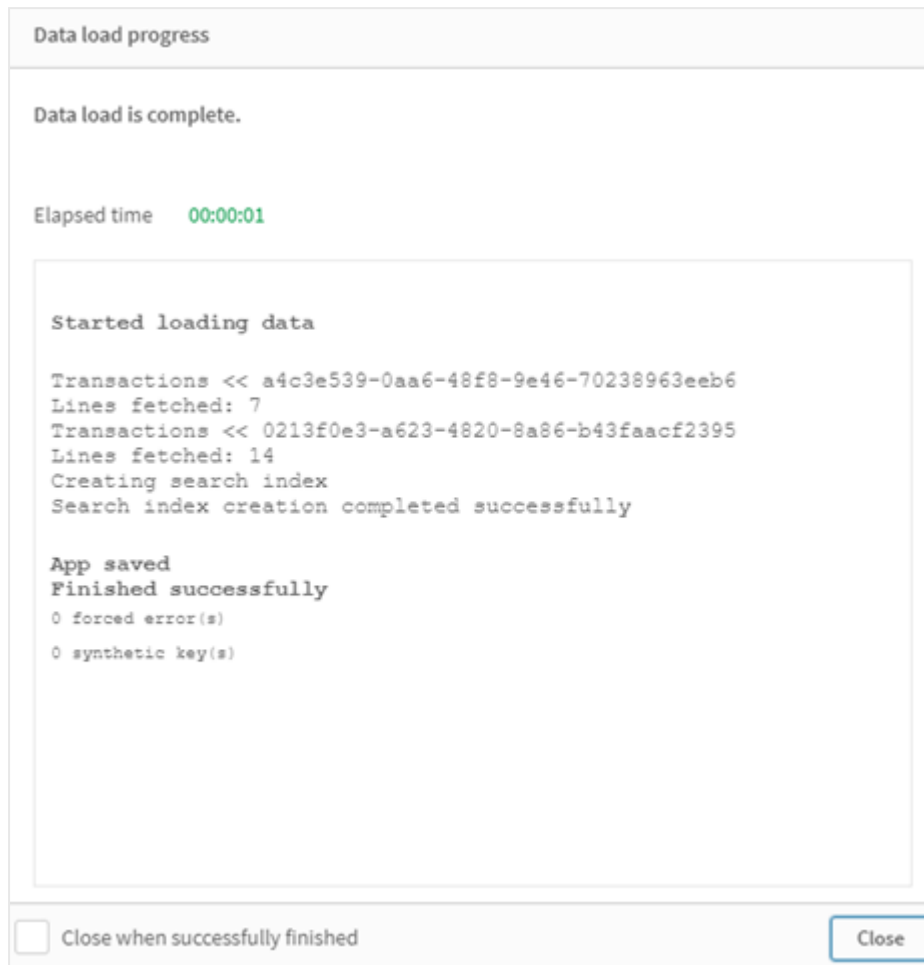
Quando viene eseguito lo script, la tabella `sa1es` viene concatenata implicitamente alla tabella `Transactions` esistente, poiché i due set di dati condividono un numero identico di campi, con nomi di campo identici. Questo accade nonostante il secondo tag di nome della tabella tenti di dare un nome al set di risultati `'sa1es'`.

Si può vedere che il set di dati Vendite è concatenato implicitamente osservando il registro di

**Avanzamento del caricamento dei dati.**

### 3 Istruzioni e parole chiave dello script

Registro di avanzamento del caricamento dei dati che mostra la concatenazione implicita dei dati delle transazioni.



#### Esempio 2 - Scenario d'uso

Script di caricamento e risultati

##### Panoramica

In questo scenario d'uso si ha:

- Un set di dati sulle transazioni con:
  - id
  - date
  - importo (in GBP)
- Una tabella delle valute con:
  - Tassi di conversione da USD a GBP
- Un secondo set di dati sulle transazioni con:
  - id

### 3 Istruzioni e parole chiave dello script

---

- date
- importo (in USD)

Verranno caricati cinque script in ordine sequenziale.

- Il primo script di caricamento contiene un set di dati iniziale con date e importi in GBP che viene inviato a una tabella denominata `Transactions`.
- Il secondo script di caricamento contiene:
  - Un secondo set di dati con date e importi in USD viene inviato a una tabella denominata `Transactions_in_USD`.
  - Il prefisso `noconcatenate` che viene posto prima dell'istruzione `LOAD` del set di dati `Transactions_in_USD` per evitare la concatenazione implicita.
- Il terzo script di caricamento contiene il prefisso `join` che verrà utilizzato per creare un tasso di cambio tra GBP e USD nella tabella `Transactions_in_USD`.
- Il quarto script di caricamento contiene il prefisso `concatenate` che aggiungerà `Transactions_in_USD` alla tabella iniziale `Transactions`.
- Il quinto script di caricamento contiene l'istruzione `drop table` che rimuove la tabella `Transactions_in_USD` i cui dati sono stati concatenati alla tabella `Transactions`.

#### Primo script di caricamento

`Transactions:`

```
Load * Inline [
id, date, amount
1, 12/30/2018, 23.56
2, 12/07/2018, 556.31
3, 12/16/2018, 5.75
4, 12/22/2018, 125.00
5, 12/22/2018, 484.21
6, 12/22/2018, 59.18
7, 12/23/2018, 177.42
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- date
- amount

Risultati del primo script di  
caricamento

| id | date       | importo |
|----|------------|---------|
| 1  | 12/30/2018 | 23.56   |

### 3 Istruzioni e parole chiave dello script

---

| id | date       | importo |
|----|------------|---------|
| 2  | 12/07/2018 | 556.31  |
| 3  | 12/16/2018 | 5.75    |
| 4  | 12/22/2018 | 125.00  |
| 5  | 12/22/2018 | 484.21  |
| 6  | 12/22/2018 | 59.18   |
| 7  | 12/23/2018 | 177.42  |

La tabella mostra il set di dati iniziale con importi in GBP.

#### Secondo script di caricamento

```
Transactions_in_USD:
NoConcatenate
Load * Inline [
id, date, amount
8, 01/01/2019, 164.27
9, 01/03/2019, 384.00
10, 01/06/2019, 25.82
11, 01/09/2019, 312.00
12, 01/15/2019, 4.56
13, 01/16/2019, 90.24
14, 01/18/2019, 19.32
];
```

#### Risultati

Caricare i dati e passare alla tabella.

Risultati del secondo script di  
caricamento

| id | date       | importo |
|----|------------|---------|
| 1  | 12/30/2018 | 23.56   |
| 2  | 12/07/2018 | 556.31  |
| 3  | 12/16/2018 | 5.75    |
| 4  | 12/22/2018 | 125.00  |
| 5  | 12/22/2018 | 484.21  |
| 6  | 12/22/2018 | 59.18   |
| 7  | 12/23/2018 | 177.42  |
| 8  | 01/01/2019 | 164.27  |
| 9  | 01/03/2019 | 384.00  |

| id | date       | importo |
|----|------------|---------|
| 10 | 01/06/2019 | 25.82   |
| 11 | 01/09/2019 | 312.00  |
| 12 | 01/15/2019 | 4.56    |
| 13 | 01/16/2019 | 90.24   |
| 14 | 01/18/2019 | 19.32   |

Si vedrà che il secondo set di dati della tabella Transactions\_in\_USD è stato aggiunto.

#### Terzo script di caricamento

Questo script di caricamento aggiunge alla tabella Transactions\_in\_USD un tasso di cambio da USD a GBP.

```
Join (Transactions_in_USD)
Load * Inline [
rate
0.7
];
```

#### Risultati

Caricare i dati e accedere al sistema di visualizzazione del modello dati. Selezionare la tabella Transactions\_in\_USD per vedere che ogni record esistente ha un valore di 0,7 nel campo "tasso".

#### Quarto script di caricamento

Utilizzando il caricamento residente, questo script di caricamento concatenerà la tabella Transactions\_in\_USD alla tabella Transactions dopo aver convertito gli importi in USD.

```
Concatenate (Transactions)
LOAD
id,
date,
amount * rate as amount
Resident Transactions_in_USD;
```

#### Risultati

Caricare i dati e passare alla tabella. Dalle righe otto a quattordici vengono visualizzate nuove voci con importi in GBP.

Risultati del quarto script di  
caricamento

| id | date       | importo |
|----|------------|---------|
| 1  | 12/30/2018 | 23.56   |

### 3 Istruzioni e parole chiave dello script

---

| id | date       | importo |
|----|------------|---------|
| 2  | 12/07/2018 | 556.31  |
| 3  | 12/16/2018 | 5.75    |
| 4  | 12/22/2018 | 125.00  |
| 5  | 12/22/2018 | 484.21  |
| 6  | 12/22/2018 | 59.18   |
| 7  | 12/23/2018 | 177.42  |
| 8  | 01/01/2019 | 114.989 |
| 8  | 01/01/2019 | 164.27  |
| 9  | 01/03/2019 | 268.80  |
| 9  | 01/03/2019 | 384.00  |
| 10 | 01/06/2019 | 18.074  |
| 10 | 01/06/2019 | 25.82   |
| 11 | 01/09/2019 | 218.40  |
| 11 | 01/09/2019 | 312.00  |
| 12 | 01/15/2019 | 3.192   |
| 12 | 01/15/2019 | 4.56    |
| 13 | 01/16/2019 | 63.168  |
| 13 | 01/16/2019 | 90.24   |
| 14 | 01/18/2019 | 13.524  |
| 14 | 01/18/2019 | 19.32   |

#### Quinto script di caricamento

Questo script di caricamento eliminerà le voci duplicate dalla tabella dei risultati del quarto script di caricamento, lasciando solo le voci con importi in GBP.

```
drop tables Transactions_in_USD;
```

#### Risultati

Caricare i dati e passare alla tabella.

Risultati del quinto script di  
caricamento

| id | date       | importo |
|----|------------|---------|
| 1  | 12/30/2018 | 23.56   |

| id | date       | importo |
|----|------------|---------|
| 2  | 12/07/2018 | 556.31  |
| 3  | 12/16/2018 | 5.75    |
| 4  | 12/22/2018 | 125.00  |
| 5  | 12/22/2018 | 484.21  |
| 6  | 12/22/2018 | 59.18   |
| 7  | 12/23/2018 | 177.42  |
| 8  | 01/01/2019 | 114.989 |
| 9  | 01/03/2019 | 268.80  |
| 10 | 01/06/2019 | 18.074  |
| 11 | 01/09/2019 | 218.40  |
| 12 | 01/15/2019 | 3.192   |
| 13 | 01/16/2019 | 63.168  |
| 14 | 01/18/2019 | 13.524  |

Dopo aver caricato il quinto script di caricamento, la tabella dei risultati mostra tutte le quattordici transazioni esistenti in entrambi i set di dati di transazioni; tuttavia, le transazioni 8-14 sono state convertite in GBP.

Se si rimuove il prefisso `NoConcatenate` utilizzato prima di `Transactions_in_USD` nel secondo script di caricamento, lo script non riuscirà e mostrerà l'errore: "Tabella 'Transactions\_in\_USD' non trovata". Questo perché la tabella `Transactions_in_USD` sarebbe stata concatenata automaticamente alla tabella originale `Transactions`.

## Only

La parola chiave di script **Only** viene utilizzata come funzione di aggregazione o come parte della sintassi nei prefissi di ricaricamento parziale **Add**, **Replace** e **Merge**.

## Outer

Il prefisso **Join** esplicito può essere preceduto dal prefisso **Outer** per specificare un'unione esterna. In un'unione esterna vengono generate tutte le combinazioni tra le due tabelle. La tabella risultante conterrà quindi le combinazioni di valori di campo provenienti dalle tabelle di dati non elaborati dove i valori di campo di collegamento vengono rappresentati in una o entrambe le tabelle. La parola chiave **Outer** è facoltativa ed è il tipo di unione predefinito utilizzato quando non viene specificato un prefisso di unione.

### Sintassi:

```
Outer Join [ (tablename) ] (loadstatement | selectstatement )
```

### Argomenti:

| Argomenti                           |                                                                   |
|-------------------------------------|-------------------------------------------------------------------|
| Argomento                           | Descrizione                                                       |
| tablename                           | La tabella denominata da confrontare con la tabella caricata.     |
| loadstatementoppure selectstatement | L'istruzione <b>LOAD</b> o <b>SELECT</b> per la tabella caricata. |

### Esempio

#### Script di caricamento

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

Table1:

```
Load * inline [
Column1, Column2
A, B
1, aa
2, cc
3, ee ];
```

Table2:

```
Outer Join Load * inline [
Column1, Column3
A, C
1, xx
4, yy ];
```

Tabella risultante

| Colonna1 | Colonna2 | Colonna3 |
|----------|----------|----------|
| A        | B        | C        |
| 1        | aa       | xx       |
| 2        | cc       | -        |
| 3        | ee       | -        |
| 4        | -        | yy       |

### Spiegazione

In questo esempio, le due tabelle, Table1 e Table2, sono fuse in una singola tabella etichettata Table1. In casi come questo, il prefisso **outer** è spesso utilizzato per unire diverse tabelle in una singola tabella per eseguire aggregazioni sui valori di una singola tabella.

### Caricamento parziale

Un ricaricamento completo inizia sempre eliminando tutte le tabelle nel modello dati esistente, dopodiché esegue lo script di caricamento.

### 3 Istruzioni e parole chiave dello script

Con un caricamento parziale questo non è possibile. Mantiene invece tutte le tabelle nel modello dati ed esegue solo le istruzioni **Load** e **Select** precedute da un prefisso **Aggiungi**, **Unisci** o **Sostituisci**. Altre tabelle di dati non sono interessate dal comando. L'argomento **solo** indica che l'istruzione deve essere eseguita solo durante i caricamenti parziali e deve essere ignorata durante i caricamenti completi. La tabella seguente riepiloga l'esecuzione dell'istruzione per i ricaricamenti parziali e completi.

| Istruzione                               | Ricaricamento completo          | Caricamento parziale            |
|------------------------------------------|---------------------------------|---------------------------------|
| Load ...                                 | Verrà eseguita l'istruzione     | Non verrà eseguita l'istruzione |
| Aggiungi/Sostituisci/Unisci LOAD...      | Verrà eseguita l'istruzione     | Verrà eseguita l'istruzione     |
| Aggiungi/Sostituisci/Unisci solo LOAD... | Non verrà eseguita l'istruzione | Verrà eseguita l'istruzione     |

I ricaricamenti parziali comportano svariati benefici rispetto ai ricaricamenti completi:

- Sono più rapidi, perché solo i dati modificati di recente devono essere caricati. Con grandi set di dati la differenza è considerevole.
- Viene consumata meno memoria, visto che vengono caricati meno dati.
- Aumenta l'affidabilità, dato che le query ai dati sorgente vengono eseguite più rapidamente, riducendo il rischio di problemi di rete.



*Affinché il ricaricamento parziale funzioni correttamente, aprire l'app con i dati prima di attivare un ricaricamento parziale.*

Eseguire un ricaricamento parziale usando il pulsante **Ricarica**. È anche possibile utilizzare Qlik Engine JSON API.

#### Limitazioni

Un ricaricamento parziale non verrà completato correttamente se vi sono comandi con riferimento alle tabelle che sono esistite durante il ricaricamento completo ma non durante il ricaricamento parziale.

#### Esempio

##### Comandi di esempio

```
LEFT JOIN(<Table_removed_after_full_reload>)  
CONCATENATE(<Table_removed_after_full_reload>)
```

Dove <Table\_removed\_after\_full\_reload> è una tabella che esisteva a table nel ricaricamento completo, ma che non è presente in quello parziale.

#### Soluzione

Come soluzione il comando può essere racchiuso dalla seguente istruzione if:

### 3 Istruzioni e parole chiave dello script

---

```
IF NOT IsPartialReload() THEN ... ENDIF.
```

Un ricaricamento parziale può rimuovere valori dai dati. Tuttavia, ciò non verrà riflesso nell'elenco dei valori distinti, che rappresenta una tabella mantenuta internamente. Pertanto, dopo un ricaricamento parziale, l'elenco conterrà tutti i valori distinti esistenti nel campo dall'ultimo ricaricamento completo, che potrebbero essere superiori a quanto attualmente esistente dopo il ricaricamento parziale. Ciò influisce sull'output delle funzioni FieldValueCount() e FieldValue(). Il valore FieldValueCount() potrebbe potenzialmente restituire un numero superiore al numero corrente di valori di campo.

#### Esempio

#### Esempio 1

#### Script di caricamento

Aggiungere lo script di esempio alla propria app ed eseguire un ricaricamento parziale. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

T1:

```
Add only Load distinct recno()+10 as Num autogenerate 10;
```

#### Risultato

Resulting table

| Num | Count(Num) |
|-----|------------|
| 11  | 1          |
| 12  | 1          |
| 13  | 1          |
| 14  | 1          |
| 15  | 1          |
| 16  | 1          |
| 17  | 1          |
| 18  | 1          |
| 19  | 1          |
| 20  | 1          |

#### Spiegazione

L'istruzione viene eseguita solo durante un ricaricamento parziale. Se il prefisso "distinto" viene omissso, il conteggio del campo **Num** aumenterà con ciascun ricaricamento parziale successivo.

### Esempio 2

#### Script di caricamento

Aggiungere lo script di esempio alla propria app. Eseguire un ricaricamento completo e visualizzare il risultato. Quindi, eseguire un ricaricamento parziale e visualizzare il risultato. Per visualizzare i risultati, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

T1:

```
Load recno() as ID, recno() as Value autogenerated 10;
```

T1:

```
Replace only Load recno() as ID, repeat(recno(),3) as Value autogenerated 10;
```

#### Risultato

Output table after full reload

| ID | Value |
|----|-------|
| 1  | 1     |
| 2  | 2     |
| 3  | 3     |
| 4  | 4     |
| 5  | 5     |
| 6  | 6     |
| 7  | 7     |
| 8  | 8     |
| 9  | 9     |
| 10 | 10    |

Output table after partial reload

| ID | Value |
|----|-------|
| 1  | 111   |
| 2  | 222   |
| 3  | 333   |
| 4  | 444   |
| 5  | 555   |
| 6  | 666   |
| 7  | 777   |

| ID | Value  |
|----|--------|
| 8  | 888    |
| 9  | 999    |
| 10 | 101010 |

### Spiegazione

La prima tabella viene caricata durante un ricaricamento completo, mentre la seconda tabella sostituisce semplicemente la prima tabella durante un ricaricamento parziale.

## Replace

La parola chiave di script **Replace** viene utilizzata come funzione stringa o come prefisso nel ricaricamento parziale.

### Replace

Il prefisso **Replace** può essere aggiunto a qualsiasi istruzione **LOAD** o **SELECT** nello script per specificare che la tabella caricata dovrebbe sostituire un'altra tabella. Specifica anche che questa istruzione dovrebbe essere eseguita in un ricaricamento parziale. Il prefisso **Replace** può essere usato anche in un'istruzione **Map**.



*Affinché il ricaricamento parziale funzioni correttamente, aprire l'app con i dati prima di attivare un ricaricamento parziale.*

Eseguire un ricaricamento parziale usando il pulsante **Ricarica**. È anche possibile utilizzare Qlik Engine JSON API.

### Sintassi:

```
Replace [only] [Concatenate[(tablename)]] (loadstatement | selectstatement)
```

```
Replace [only] mapstatement
```

Durante un caricamento normale (non parziale), la costruzione **Replace LOAD** funzionerà come una normale istruzione **LOAD** ma sarà preceduta da un **Drop Table**. Verrà prima eliminata la vecchia tabella, quindi i record verranno generati e archiviati come una nuova tabella.

Se viene utilizzato il prefisso **Concatenate**, o se esiste una tabella con lo stesso set di campi, sarà questa la tabella pertinente da eliminare. Altrimenti, non vi sarà alcuna tabella da eliminare e la costruzione **Replace LOAD** sarà identica a un normale **LOAD**.

Un caricamento parziale otterrà lo stesso risultato. L'unica differenza è che c'è sempre una tabella da eliminare dalla precedente esecuzione script. La costruzione **Replace LOAD** eliminerà sempre prima la vecchia tabella, per poi creare la nuova.

### 3 Istruzioni e parole chiave dello script

L'istruzione **Replace Map...Using** determina l'esecuzione del mapping anche durante l'esecuzione parziale dello script.

#### Argomenti:

##### Argomenti

| Argomento | Descrizione                                                                                                                                                                             |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| only      | Un qualificatore opzionale che denota che l'istruzione dovrebbe essere eseguita solo durante i caricamenti parziali. Deve essere ignorata durante i caricamenti normali (non parziali). |

#### Esempi e risultati:

| Esempio                                                                                       | Risultato                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tab1:<br>Replace LOAD<br>* from<br>File1.csv;                                                 | Durante le operazioni di ricaricamento normale e parziale dei dati, la tabella Tab1 di Qlik Sense viene inizialmente rimossa. Successivamente, i nuovi dati vengono caricati dal file File1.csv e memorizzati nella tabella Tab1.                                                                                                                                                                                                                               |
| Tab1:<br>Replace only<br>LOAD * from<br>File1.csv;                                            | Durante le operazioni di ricaricamento normale, questa istruzione viene ignorata.<br><br>Durante il ricaricamento parziale, ogni tabella di Qlik Sense precedentemente denominata Tab1 viene inizialmente rimossa. Successivamente, i nuovi dati vengono caricati dal file File1.csv e memorizzati nella tabella Tab1.                                                                                                                                          |
| Tab1:<br>LOAD a,b,c<br>from<br>File1.csv;<br>Replace LOAD<br>a,b,c from<br>File2.csv;         | Durante il ricaricamento normale il file File1.csv viene inizialmente letto nella tabella di Qlik Sense Tab1, ma successivamente viene immediatamente rimosso e sostituito dai nuovi dati caricati dal file File2.csv. Tutti i dati del file File1.csv vengono persi.<br><br>Durante il caricamento parziale l'intera tabella di Qlik Sense Tab1 viene inizialmente rimossa. Successivamente, la tabella viene sostituita dai dati caricati dal file File2.csv. |
| Tab1:<br>LOAD a,b,c<br>from<br>File1.csv;<br>Replace only<br>LOAD a,b,c<br>from<br>File2.csv; | Durante il ricaricamento normale, i dati vengono caricati dal file File1.csv e memorizzati nella tabella di Qlik Sense Tab1. Il file File2.csv viene ignorato.<br><br>Durante il ricaricamento parziale l'intera tabella di Qlik Sense Tab1 viene inizialmente rimossa. Successivamente, la tabella viene sostituita dai dati caricati dal file File2.csv. Tutti i dati del file File1.csv vengono persi.                                                       |

## Right

I prefissi **Join** e **Keep** possono essere preceduti dal prefisso **right**.

Se viene inserito prima di **join**, specifica che occorre utilizzare un'unione destra. La tabella risultante contiene solo le combinazioni di valori di campo estratte dalle tabelle di dati non elaborati, dove i valori di campo di collegamento vengono rappresentati nella seconda tabella. Se viene utilizzato prima di **keep**, specifica che la prima tabella di dati non elaborati deve essere ridotta alla sua intersezione comune con la seconda tabella prima di essere memorizzata in Qlik Sense.



Si stava cercando la funzione di stringa con lo stesso nome? Vedere: [Right \(page 1839\)](#)

### Sintassi:

```
Right (Join | Keep) [(tablename)] (loadstatement | selectstatement )
```

### Argomenti:

| Argomenti                           |                                                                   |
|-------------------------------------|-------------------------------------------------------------------|
| Argomento                           | Descrizione                                                       |
| tablename                           | La tabella denominata da confrontare con la tabella caricata.     |
| loadstatementoppure selectstatement | L'istruzione <b>LOAD</b> o <b>SELECT</b> per la tabella caricata. |

### Esempio

#### Script di caricamento

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

Table1:

```
Load * inline [
Column1, Column2
A, B
1, aa
2, cc
3, ee ];
```

Table2:

```
Right Join Load * inline [
Column1, Column3
A, C
1, xx
4, yy ];
```

### Risultato

| Tabella risultante |          |          |
|--------------------|----------|----------|
| Colonna1           | Colonna2 | Colonna3 |
| A                  | B        | C        |
| 1                  | aa       | xx       |
| 4                  | -        | yy       |

### Spiegazione

Questo esempio dimostra l'output di Right Join dove vengono uniti solo i valori presenti nella seconda tabella (destra).

### Sample

Il prefisso **sample** aggiunto a un'istruzione **LOAD** o **SELECT** viene utilizzato per caricare un campione casuale di record dalla sorgente dati.

#### Sintassi:

```
Sample p ( loadstatement | selectstatement )
```

L'espressione che viene valutata non definisce la percentuale di record del set di dati che verranno caricati nell'applicazione Qlik Sense, ma la probabilità che ogni record letto venga caricato nell'applicazione. In altre parole, specificare un valore  $p = 0.5$  non significa che verrà caricato il 50% del numero totale di record, ma piuttosto che per ogni record ci sarà il 50% di possibilità che venga caricato nell'applicazione Qlik Sense.

#### Argomenti

| Argomento | Descrizione                                                                                                                                                                                                                                               |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| p         | Un'espressione arbitraria che restituisce un numero maggiore di 0 e minore o uguale a 1. Il numero indica la probabilità di lettura di un determinato record.<br><br>Anche se verranno letti tutti i record, solo alcuni verranno caricati in Qlik Sense. |

### Casi di utilizzo

Un campione è utile quando si desidera campionare i dati provenienti da una tabella di grandi dimensioni, per capire la natura dei dati, la distribuzione o il contenuto dei campi. Dato che porta un sottoinsieme di dati, il caricamento dei dati è più veloce e consente di testare più rapidamente gli script. A differenza di `First`, la funzione `sample` riporta i dati dell'intera tabella, invece di limitarsi alle prime righe. In alcuni casi, ciò può fornire una rappresentazione più accurata dei dati.

Gli esempi seguenti mostrano due possibili usi del prefisso script `sample`:

```
sample 0.15 SQL SELECT * from Longtable;
```

```
sample(0.15) LOAD * from Longtab.csv;
```

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - Campione da una tabella in linea

Script di caricamento e risultati

#### Panoramica

In questo esempio, lo script carica un campione di dati da un set di dati contenente sette record in una tabella denominata `Transactions` da una tabella in linea.

#### Script di caricamento

```
Transactions:
SAMPLE 0.3
LOAD
*
Inline [
id, date, amount
1, 08/30/2018, 23.56
2, 09/07/2018, 556.31
3, 09/16/2018, 5.75
4, 09/22/2018, 125.00
5, 09/22/2018, 484.21
6, 09/22/2018, 59.18
7, 09/23/2018, 177.42
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `id`
- `amount`

Aggiungere la seguente misura:

```
=sum(amount)8
```

| Tabella dei risultati |            |             |
|-----------------------|------------|-------------|
| id                    | date       | sum(amount) |
| 2                     | 09/07/2018 | 556,31      |
| 4                     | 09/22/2018 | 125         |
| 1                     | 30/08/2018 | 23,56       |
| 3                     | 16/09/2018 | 5,75        |

Nell'iterazione del caricamento utilizzata in questo esempio, sono stati letti tutti e sette i record, ma solo quattro sono stati caricati nella tabella dei dati. Qualsiasi nuova esecuzione di caricamento potrebbe risultare in un numero diverso e in un diverso set di record caricati nell'applicazione.

### Esempio 2 - Esempio di tabella generata automaticamente

Script di caricamento e risultati

#### Panoramica

In questo esempio, utilizzando Autogenerate, viene creato un set di dati di 100 record con i campi date, id, e amount. Tuttavia, viene utilizzato il prefisso sample, con un valore di 0,1.

#### Script di caricamento

```
SampleData:
Sample 0.1
LOAD
RecNo() AS id,
MakeDate(2013, Ceil(Rand() * 12), Ceil(Rand() * 29)) as date,
Rand() * 1000 AS amount
```

```
Autogenerate(100);
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- amount

Aggiungere la seguente misura:

Tabella dei risultati

| id  | date      | sum(amount) |
|-----|-----------|-------------|
| 48  | 9/28/2013 | 763         |
| 20  | 5/15/2013 | 752         |
| 19  | 11/8/2013 | 657         |
| 25  | 3/24/2013 | 522         |
| 27  | 8/23/2013 | 389         |
| 81  | 6/1/2013  | 53          |
| 100 | 8/15/2013 | 17          |

Nell'iterazione del caricamento utilizzata in questo esempio, sono stati caricati sette record dal set di dati creato. Di nuovo, qualsiasi nuova esecuzione di caricamento potrebbe risultare in un numero diverso e in un diverso set di record caricati nell'applicazione.

### Semantic

Il prefisso dell'istruzione di caricamento di `semantic` crea un tipo speciale di campo che può essere utilizzato in Qlik Sense per connettere e gestire dati relazionali, come strutture ad albero, dati strutturati genitore-figlio autoreferenziali e/o dati che possono essere descritti come un grafico.

Si noti che il carico `semantic` può funzionare in modo simile ai prefissi [Hierarchy \(page 66\)](#) e [HierarchyBelongsTo \(page 68\)](#). Tutti e tre i prefissi possono essere utilizzati come elementi costitutivi in efficaci soluzioni front-end per l'attraversamento dei dati relazionali.

#### Sintassi:

```
Semantic( loadstatement | selectstatement)
```

Un carico semantico prevede un input largo esattamente tre o quattro campi con una definizione rigorosa di ciò che rappresenta ciascun campo ordinato, come mostrato nella tabella seguente:

Campi di carico semantico

| Nome campo | Descrizione del campo                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1° campo:  | questo tag è una rappresentazione del primo di due oggetti tra i quali esiste una relazione.                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 2° campo:  | questo tag verrà utilizzato per descrivere la relazione "forward" tra il primo e il secondo oggetto. Se il primo è un oggetto figlio e il secondo è un oggetto padre, è possibile creare una scheda di relazione che indichi "padre" o "padre di", per fare in modo di seguire dall'oggetto figlio all'oggetto padre.                                                                                                                                                                                                                           |
| 3° campo:  | questo tag è una rappresentazione del primo di due oggetti tra i quali esiste una relazione.                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 4° campo:  | questo campo è facoltativo. Questo tag descrive le relazioni "forward" o "inverse" tra il primo e il secondo oggetto. Se il primo è un oggetto figlio e il secondo è un oggetto padre, una scheda di relazione può indicare "figlio" o "figlio di", per fare in modo di seguire dall'oggetto padre all'oggetto figlio. Se non si aggiunge un quarto campo, il secondo tag di campo verrà utilizzato per descrivere la relazione in entrambe le direzioni. In tal caso, viene aggiunto automaticamente un simbolo di freccia come parte del tag. |

Il codice seguente è un esempio del prefisso `semantic`.

```
semantic  
load  
Object,  
'Parent' AS Relationship,  
NeighbouringObject AS Object,  
'Child' AS Relationship  
from graphdata.csv;
```



*È consentito ed è una pratica comune etichettare il terzo campo come il primo campo. Questo crea una ricerca autoreferenziale, in modo che si possa seguire gli oggetti verso gli oggetti correlati una fase alla volta. Se il 3° campo non ha lo stesso nome, il risultato finale sarà una semplice ricerca da uno o più oggetti ai suoi vicini relazionali diretti, con un output di scarsa utilità pratica.*

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

#### Funzioni correlate

##### Funzioni

[Hierarchy \(page 66\)](#)

[HierarchyBelongsTo \(page 68\)](#)

##### Interazione

Il prefisso di caricamento `Hierarchy` viene utilizzato per dividere e organizzare i nodi in strutture di dati padre-figlio e altre strutture di dati simili a grafici e trasformarli in tabelle.

Il prefisso di caricamento `HierarchyBelongsTo` viene utilizzato per individuare e organizzare i dati predecessori dei dati padre-figlio e altre strutture di dati simili a grafici e trasformarli in tabelle.

### Esempio: creazione di un campo speciale per collegare le relazioni utilizzando il prefisso semantico

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che rappresenta i record della relazione geografica, che viene caricato in una tabella denominata `GeographyTree`.

### 3 Istruzioni e parole chiave dello script

---

- Ogni voce ha un ID all'inizio della riga e un ParentID alla fine della riga.
- Il prefisso semantic aggiungerà un campo di comportamento speciale con l'etichetta Relation.

#### Script di caricamento

GeographyTree:

```
LOAD
    ID,
    Geography,
    if(ParentID='',null(),ParentID) AS ParentID
```

```
INLINE [
ID,Geography,ParentID
1,world
2,Europe,1
3,Asia,1
4,North America,1
5,South America,1
6,UK,2
7,Germany,2
8,Sweden,2
9,South Korea,3
10,North Korea,3
11,China,3
12,London,6
13,Birmingham,6
];
```

SemanticTable:

```
Semantic Load
    ID as ID,
    'Parent' as Relation,
    ParentID as ID,
    'Child' as Relation
resident GeographyTree;
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni.

- Id
- Geography

Quindi, creare una casella di filtro con Relation come dimensione. Fare clic su **Termina modifica**.

Tabella dei risultati

| Id | Geografia |
|----|-----------|
| 1  | Mondo     |
| 2  | Europa    |

### 3 Istruzioni e parole chiave dello script

---

| Id | Geografia      |
|----|----------------|
| 3  | Asia           |
| 4  | Nord America   |
| 5  | America Latina |
| 6  | GB             |
| 7  | Germany        |
| 8  | Svezia         |
| 9  | Corea del Sud  |
| 10 | Corea del Nord |
| 11 | Cina           |
| 12 | Londra         |
| 13 | Birmingham     |

Casella di filtro

#### Relazione

Figlio

Padre

Fare clic su **Europa** dalla dimensione geography nella tabella e fare clic su **Figlio** dalla dimensione relation nella casella di filtro. Notare il risultato atteso nella tabella:

Tabella dei risultati che  
mostra i risultati "figli"  
dell'Europa

| Id | Geografia |
|----|-----------|
| 6  | GB        |
| 7  | Germany   |
| 8  | Svezia    |

Facendo nuovamente clic su **Figlio**, verranno mostrati i luoghi "figlio" del Regno Unito, un gradino più in basso.

Tabella dei risultati che  
mostra i risultati "figli" del  
Regno Unito

| Id | Geografia  |
|----|------------|
| 12 | Londra     |
| 13 | Birmingham |

### Unless

Il prefisso e suffisso **unless** viene utilizzato per creare una clausola condizionale che determina se valutare o meno un'istruzione oppure una clausola exit. Può essere considerato come un'alternativa compatta all'istruzione completa **if..end if**.

#### Sintassi:

```
(Unless condition statement | exitstatement Unless condition )
```

L'istruzione **statement** o **exitstatement** verrà eseguita solo se **condition** restituisce False.

Il prefisso **unless** può essere utilizzato con istruzioni che presentano già almeno un'altra istruzione, inclusi i prefissi aggiuntivi **when** o **unless**.

#### Argomenti

| Argomento     | Descrizione                                                                                               |
|---------------|-----------------------------------------------------------------------------------------------------------|
| condition     | Un'espressione logica che restituisce un valore True o False.                                             |
| statement     | Qualsiasi istruzione dello script di Qlik Sense, ad eccezione delle istruzioni di controllo.              |
| exitstatement | Una clausola <b>exit for</b> , <b>exit do</b> o <b>exit sub</b> oppure un'istruzione <b>exit script</b> . |

### Casi di utilizzo

L'istruzione **unless** restituisce un risultato booleano. In genere, questo tipo di funzione viene utilizzata come condizione quando l'utente desidera caricare o escludere parti dello script in modo condizionato.

Le righe seguenti mostrano tre esempi di utilizzo della funzione **unless**:

```
exit script unless A=1;
```

```
unless A=1 LOAD * from myfile.csv;
```

```
unless A=1 when B=2 drop table Tab1;
```

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione **SET DateFormat** nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - Prefisso Unless

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- La creazione della variabile A, alla quale viene assegnato il valore 1.
- Un set di dati che viene caricato in una tabella denominata Transazioni, a meno che la variabile A = 2.

#### Script di caricamento

```
LET A = 1;

UNLESS A = 2

Transactions:
LOAD
*
Inline [
id, date, amount
1, 08/30/2018, 23.56
2, 09/07/2018, 556.31
3, 09/16/2018, 5.75
4, 09/22/2018, 125.00
5, 09/22/2018, 484.21
6, 09/22/2018, 59.18
7, 09/23/2018, 177.42
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- date
- amount

Tabella dei risultati

| id | date       | importo |
|----|------------|---------|
| 1  | 30/08/2018 | 23,56   |
| 2  | 09/07/2018 | 556,31  |
| 3  | 16/09/2018 | 5,75    |

| id | date       | importo |
|----|------------|---------|
| 4  | 22/09/2018 | 125,00  |
| 5  | 22/09/2018 | 484,21  |
| 6  | 22/09/2018 | 59,18   |
| 7  | 09/23/2018 | 177.42  |

Poiché alla variabile A viene assegnato il valore 1 all'inizio dello script, la condizione che segue il prefisso `unless` viene valutata, restituendo un risultato di `FALSE`. Di conseguenza, lo script continua a eseguire l'istruzione `Load`. Nella tabella dei risultati sono visibili tutti i record della tabella `Transactions`.

Se il valore di questa variabile è impostato su 2, non verranno caricati dati nel modello dati.

### Esempio 2 - Suffisso Unless

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor di caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento dati inizia caricando un set di dati iniziale in una tabella denominata `Transactions`. Lo script viene quindi terminato se non vi sono meno di 10 record nella tabella `Transactions`.

Se questa condizione non porta alla chiusura dello script, un altro set di transazioni viene concatenato nella tabella `Transactions` e il processo viene ripetuto.

#### Script di caricamento

`Transactions:`

`LOAD`

`*`

`Inline [`

`id, date, amount`

`1, 08/30/2018, 23.56`

`2, 09/07/2018, 556.31`

`3, 09/16/2018, 5.75`

`4, 09/22/2018, 125.00`

`5, 09/22/2018, 484.21`

`6, 09/22/2018, 59.18`

`7, 09/23/2018, 177.42`

`];`

`exit script unless NoOfRows('Transactions') < 10 ;`

`Concatenate`

`LOAD`

`*`

`Inline [`

### 3 Istruzioni e parole chiave dello script

---

```
id, date, amount
8, 10/01/2018, 164.27
9, 10/03/2018, 384.00
10, 10/06/2018, 25.82
11, 10/09/2018, 312.00
12, 10/15/2018, 4.56
13, 10/16/2018, 90.24
14, 10/18/2018, 19.32
];
```

```
exit script unless NoOfRows('Transactions') < 10 ;
```

Concatenate

LOAD

\*

Inline [

```
id, date, amount
15, 10/01/2018, 164.27
16, 10/03/2018, 384.00
17, 10/06/2018, 25.82
18, 10/09/2018, 312.00
19, 10/15/2018, 4.56
20, 10/16/2018, 90.24
21, 10/18/2018, 19.32
];
```

```
exit script unless NoOfRows('Transactions') < 10 ;
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- date
- amount

Tabella dei risultati

| id | date       | importo |
|----|------------|---------|
| 1  | 30/08/2018 | 23,56   |
| 2  | 09/07/2018 | 556,31  |
| 3  | 16/09/2018 | 5,75    |
| 4  | 22/09/2018 | 125,00  |
| 5  | 22/09/2018 | 484,21  |
| 6  | 22/09/2018 | 59,18   |
| 7  | 09/23/2018 | 177.42  |
| 8  | 10/01/2018 | 164.27  |

| id | date       | importo |
|----|------------|---------|
| 9  | 10/03/2018 | 384.00  |
| 10 | 10/06/2018 | 25.82   |
| 11 | 10/09/2018 | 312.00  |
| 12 | 10/15/2018 | 4.56    |
| 13 | 10/16/2018 | 90.24   |
| 14 | 10/18/2018 | 19.32   |

Sono presenti sette record in ciascuno dei tre set di dati dello script di caricamento.

Il primo set di dati (con le transazioni da id 1 a 7) viene caricato nell'applicazione. La condizione `unless` valuta se nella tabella `Transactions` ci sono meno di 10 righe. La valutazione è pari a `TRUE`, quindi il secondo set di dati (con le transazioni id da 8 a 14) viene caricato nell'applicazione. La seconda condizione `unless` viene valutata se nella tabella `Transactions` sono presenti meno di 10 record. La valutazione è pari a `FALSE`, quindi lo script termina.

### Esempio 3 - Prefissi Unless multipli

Script di caricamento e risultati

#### Panoramica

Aprire l'editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

In questo esempio, un set di dati contenente una transazione viene creato come tabella chiamata `Transactions`. Viene quindi attivato un ciclo "for", in cui vengono valutate due istruzioni "unless" nidificate:

1. A meno che non ci siano più di 100 record nella tabella `Transactions`
2. A meno che il numero di record nella tabella `Transactions` non sia un multiplo di 6

Se queste condizioni sono `FALSE`, vengono generati altri sette record e concatenati alla tabella `Transactions` esistente. Questo processo viene ripetuto finché una delle due transazioni non restituisce un valore di `TRUE`.

#### Script di caricamento

```
Transactions:
Load
    0 as id
Autogenerate 1;

For i = 1 to 100
    unless NoOfRows('Transactions') > 100 unless mod(NoOfRows('Transactions'),6) = 0
        Concatenate
            Load
                if(isnull(Peek(id)),1,peek(id)+1) as id
```

```
Autogenerate 7;  
next i
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: id.

Tabella dei risultati

| id               |
|------------------|
| 0                |
| 1                |
| 2                |
| 3                |
| 4                |
| 5                |
| +30 righe in più |

Le istruzioni nidificate **unless** che si verificano nel ciclo **'for'** valutano quanto segue:

1. Sono presenti più di 100 righe nella tabella `Transactions`?
2. Il numero totale di record nella tabella `Transactions` è un multiplo di 6?

Ogni volta che entrambe le istruzioni **unless** restituiscono un valore di `FALSE`, vengono generati altri sette record e concatenati alla tabella `Transactions` esistente.

Queste istruzioni restituiscono un valore di `FALSE` cinque volte, a questo punto è presente un totale di 36 righe di dati nella tabella `Transactions`.

Dopodiché, la seconda istruzione **unless** restituisce un valore di `TRUE`, pertanto l'istruzione di caricamento successiva non verrà più eseguita.

## When

Il prefisso e suffisso **when** viene utilizzato per creare una clausola condizionale che determina se eseguire o meno un'istruzione oppure una clausola `exit`. Può essere considerato come un'alternativa compatta all'istruzione completa **if..end if**.

### Sintassi:

```
(when condition statement | exitstatement when condition )
```

**Tipo di dati restituiti:** Boolean

In Qlik Sense, il valore booleano vero è rappresentato da -1 e il valore falso è rappresentato da 0.

### 3 Istruzioni e parole chiave dello script

Le istruzioni **statement** o **exitstatement** vengono eseguite solamente se la condizione viene valutata essere TRUE.

Il prefisso **when** può essere utilizzato con istruzioni che presentano già almeno un'altra istruzione, inclusi i prefissi addizionali **when** o **unless**.

#### Casi di utilizzo

L'istruzione **when** restituisce un risultato booleano. In genere, questo tipo di funzione viene utilizzata come condizione quando l'utente desidera caricare o escludere parti di uno script.

#### Argomenti

| Argomento     | Descrizione                                                                                               |
|---------------|-----------------------------------------------------------------------------------------------------------|
| condition     | Un'espressione logica che restituisce un valore TRUE o FALSE                                              |
| statement     | Qualsiasi istruzione dello script di Qlik Sense, ad eccezione delle istruzioni di controllo.              |
| exitstatement | Una clausola <b>exit for</b> , <b>exit do</b> o <b>exit sub</b> oppure un'istruzione <b>exit script</b> . |

#### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione **SET DateFormat** nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

#### Esempi di funzioni

| Esempio                                           | Risultato                                                                                                                                                    |
|---------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>exit script when A=1;</code>                | Quando l'istruzione <code>A=1</code> viene valutata come TRUE, lo script si interrompe.                                                                      |
| <code>when A=1 LOAD * from myfile.csv;</code>     | Quando l'istruzione <code>A=1</code> viene valutata come TRUE, viene caricato <code>myfile.csv</code> .                                                      |
| <code>when A=1 unless B=2 drop table Tab1;</code> | Quando l'istruzione <code>A=1</code> viene valutata come TRUE e se <code>B=2</code> viene valutato come FALSE, la tabella <code>Tab1</code> viene eliminata. |

### Esempio 1 - Prefisso When

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati con date e importi che viene inviato a una tabella denominata 'Transactions'.
- L'istruzione `Let` che afferma che la variabile `A` è stata creata e ha il valore di 1.
- La condizione `when` che fornisce la condizione che se `A` è uguale a 1, allora lo script continuerà a essere caricato.

#### Script di caricamento

```
LET A = 1;

WHEN A = 1

Transactions:
LOAD
*
Inline [
id, date, amount
1, 08/30/2018, 23.56
2, 09/07/2018, 556.31
3, 09/16/2018, 5.75
4, 09/22/2018, 125.00
5, 09/22/2018, 484.21
6, 09/22/2018, 59.18
7, 09/23/2018, 177.42
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `id`
- `date`
- `amount`

Tabella dei risultati

| id | date       | importo |
|----|------------|---------|
| 1  | 30/08/2018 | 23,56   |
| 2  | 09/07/2018 | 556,31  |

| id | date       | importo |
|----|------------|---------|
| 3  | 16/09/2018 | 5,75    |
| 4  | 22/09/2018 | 125,00  |
| 5  | 22/09/2018 | 484,21  |
| 6  | 22/09/2018 | 59,18   |
| 7  | 09/23/2018 | 177.42  |

Poiché alla variabile `A` viene assegnato il valore di 1 all'inizio dello script, la condizione che segue il prefisso `when` viene valutata e restituisce un risultato di `TRUE`. Poiché il risultato è `TRUE`, lo script continua a eseguire l'istruzione `LOAD`. È possibile visualizzare tutti i record della tabella dei risultati.

Se il valore di questa variabile fosse impostato su un valore diverso da 1, non verrebbero caricati dati nel modello dati.

### Esempio 2 - Suffisso When

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Tre set di dati con date e importi che vengono inviati a una tabella denominata 'Transactions'.
  - Il primo set di dati contiene le transazioni 1-7.
  - Il secondo set di dati contiene le transazioni 8-14.
  - Il terzo set di dati contiene le transazioni 15-21.
- Una condizione `when` che determina se la tabella 'Transactions' contiene più di dieci righe. Se una qualsiasi delle istruzioni `when` viene valutata come `TRUE`, lo script di caricamento si interrompe. Questa condizione è posta alla fine di ciascuno dei tre set di dati.

#### Script di caricamento

```
Transactions:
LOAD
*
Inline [
id, date, amount
1, 08/30/2018, 23.56
2, 09/07/2018, 556.31
3, 09/16/2018, 5.75
4, 09/22/2018, 125.00
5, 09/22/2018, 484.21
6, 09/22/2018, 59.18
7, 09/23/2018, 177.42
];
```

### 3 Istruzioni e parole chiave dello script

---

```
exit script when NoOfRows('Transactions') > 10 ;
```

Concatenate

LOAD

\*

Inline [

id, date, amount

8, 10/01/2018, 164.27

9, 10/03/2018, 384.00

10, 10/06/2018, 25.82

11, 10/09/2018, 312.00

12, 10/15/2018, 4.56

13, 10/16/2018, 90.24

14, 10/18/2018, 19.32

];

```
exit script when NoOfRows('Transactions') > 10 ;
```

Concatenate

LOAD

\*

Inline [

id, date, amount

15, 10/01/2018, 164.27

16, 10/03/2018, 384.00

17, 10/06/2018, 25.82

18, 10/09/2018, 312.00

19, 10/15/2018, 4.56

20, 10/16/2018, 90.24

21, 10/18/2018, 19.32

];

```
exit script when NoOfRows('Transactions') > 10 ;
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- date
- amount

Tabella dei risultati

| id | date       | importo |
|----|------------|---------|
| 1  | 30/08/2018 | 23,56   |
| 2  | 09/07/2018 | 556,31  |
| 3  | 16/09/2018 | 5,75    |

| id | date       | importo |
|----|------------|---------|
| 4  | 22/09/2018 | 125,00  |
| 5  | 22/09/2018 | 484,21  |
| 6  | 22/09/2018 | 59,18   |
| 7  | 09/23/2018 | 177.42  |
| 8  | 10/01/2018 | 164.27  |
| 9  | 10/03/2018 | 384.00  |
| 10 | 10/06/2018 | 25.82   |
| 11 | 10/09/2018 | 312.00  |
| 12 | 10/15/2018 | 4.56    |
| 13 | 10/16/2018 | 90.24   |
| 14 | 10/18/2018 | 19.32   |

Ciascuno dei tre set di dati contiene sette transazioni. Il primo set di dati contiene le transazioni 1-7 e viene caricato nell'applicazione. La condizione `when` che segue l'istruzione di caricamento viene valutata come `FALSE` perché nella tabella `'Transactions'` sono presenti meno di dieci righe. Lo script di caricamento continua con il set di dati successivo.

Il secondo set di dati contiene le transazioni 8-14 e viene caricato nell'applicazione. La seconda condizione `when` viene valutata come `TRUE` perché ci sono più di dieci righe nella tabella `'Transactions'`. Pertanto, lo script termina.

### Esempio 3 - Prefissi When multipli

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente una singola transazione viene creato come tabella chiamata `'Transactions'`.
- Un ciclo `For` che viene attivato contiene due condizioni nidificate `when` che valutano se:
  1. Vi sono meno di 100 record nella tabella `'Transactions'`.
  2. Il numero di record nella tabella `'Transactions'` non sia un multiplo di 6

#### Script di caricamento

```
RowsCheck = NoOfRows('Transactions') < 100 or mod(NoOfRows('Transactions'),6) <> 0;  
Transactions:  
Load
```

### 3 Istruzioni e parole chiave dello script

---

```
0 as id
Autogenerate 1;
For i = 1 to 100
    when(RowsCheck)
        Concatenate
        Load
            if(isnull(Peek(id)),1,peek(id)+1) as id
        Autogenerate 7;
next i
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- id

La tabella dei risultati mostra solo i primi cinque ID di transazione, ma lo script di caricamento crea 36 righe e termina una volta soddisfatta la condizione when.

Tabella dei  
risultati

| id               |
|------------------|
| 0                |
| 1                |
| 2                |
| 3                |
| 4                |
| 5                |
| +30 righe in più |

Le condizioni when nidificate nel ciclo For valutano le seguenti domande:

- Sono presenti meno di 100 righe nella tabella 'Transactions'?
- Il numero totale di record nella tabella 'Transactions' non è un multiplo di sei?

Ogni volta che entrambe le condizioni when restituiscono un valore TRUE, vengono generati altri sette record e concatenati alla tabella 'Transactions' esistente.

Le condizioni when restituiscono un valore TRUE per cinque volte. A questo punto, nella tabella 'Transactions' vi è un totale di 36 righe di dati.

Quando vengono create 36 righe di dati nella tabella 'Transactions', la seconda istruzione when restituisce il valore FALSE, pertanto l'istruzione LOAD successiva non viene più eseguita.

### 3.3 Istruzioni regolari dello script

Le istruzioni regolari vengono generalmente utilizzate per la manipolazione dei dati. Queste istruzioni possono essere scritte su un qualsiasi numero di righe nello script e devono sempre terminare con un punto e virgola, ";".

Tutte le parole chiave dello script possono essere immesse con qualsiasi combinazione di caratteri maiuscoli e minuscoli. I nomi dei campi e delle variabili utilizzati nelle istruzioni possono essere immessi indipendentemente dal formato del carattere.

#### Prospetto delle istruzioni regolari dello script

Ciascuna funzione viene descritta ulteriormente dopo la panoramica. È inoltre possibile fare clic sul nome della funzione nella sintassi per accedere immediatamente ai dettagli per tale funzione specifica.

##### Alias

L'istruzione **alias** viene utilizzata per impostare un alias in base al quale verrà rinominato un campo quando questo ricorrerà nello script che segue.

```
Alias fieldname as aliasname {,fieldname as aliasname}
```

##### Autonumber

Questa istruzione crea un valore intero univoco per ciascun valore calcolato distinto in un campo rilevato durante l'esecuzione dello script.

```
AutoNumber fields [Using namespace] ]
```

##### Binary

L'istruzione **binary** consente di caricare i dati da un altro documento QlikView, inclusi i dati in Section Access.

```
Binary [path] filename
```

##### comment

Fornisce un modo per visualizzare i commenti dei campi (metadati) da database e fogli di calcolo. I nomi di campo non presenti nell'app verranno ignorati. In caso di più ricorrenze di uno stesso nome di campo, verrà applicato l'ultimo valore utilizzato.

```
Comment field *fieldlist using mapname
```

```
Comment field fieldname with comment
```

##### comment table

Fornisce un modo per visualizzare i commenti delle tabelle (metadati) da database o fogli di calcolo.

```
Comment table tablelist using mapname
```

```
Comment table tablename with comment
```

### Connect



*Questa funzionalità non è disponibile in Qlik Sense SaaS.*

L'istruzione **CONNECT** consente di definire l'accesso di Qlik Sense a un database generico mediante l'interfaccia OLE DB/ODBC. Per ODBC, occorre innanzitutto specificare la sorgente dati utilizzando l'amministratore ODBC.

```
ODBC Connect TO connect-string [ ( access_info ) ]
```

```
OLEDB CONNECT TO connect-string [ ( access_info ) ]
```

```
CUSTOM CONNECT TO connect-string [ ( access_info ) ]
```

```
LIB CONNECT TO connection
```

### Declare

L'istruzione **Declare** consente di creare definizioni di campo in cui è possibile definire relazioni tra i campi o le funzioni. È possibile utilizzare una serie di definizioni di campo per generare automaticamente campi derivati, che possono essere utilizzati come dimensioni. Ad esempio, è possibile creare una definizione di calendario e utilizzarla per generare dimensioni correlate, come ad esempio anno, mese, settimana e giorno da un campo della data.

```
definition_name:
```

```
Declare [Field[s]] Definition [Tagged tag_list ]
```

```
[Parameters parameter_list ]
```

```
Fields field_list
```

```
[Groups group_list ]
```

```
<definition name>:
```

```
Declare [Field][s] Definition
```

```
Using <existing_definition>
```

```
[With <parameter_assignment> ]
```

### Derive

L'istruzione **Derive** consente di generare campi derivati in base a una definizione di campo creata mediante un'istruzione **Declare**. È possibile specificare i campi dati per cui derivare i campi oppure derivarli esplicitamente o implicitamente in base ai tag di campo.

```
Derive [Field[s]] From [Field[s]] field_list Using definition
```

### 3 Istruzioni e parole chiave dello script

```
Derive [Field[s]] From Explicit [Tag[s]] (tag_list) Using definition
```

```
Derive [Field[s]] From Implicit [Tag[s]] Using definition
```

#### Direct Query

L'istruzione **DIRECT QUERY** consente di accedere alle tabelle mediante una connessione ODBC o OLE DB utilizzando la funzione Direct Discovery.

```
Direct Query [path]
```

#### Directory

L'istruzione **Directory** definisce in quale directory ricercare i file dei dati nelle istruzioni **LOAD** successive finché non viene eseguita una nuova istruzione **Directory**.

```
Directory [path]
```

#### Disconnect

L'istruzione **Disconnect** termina l'attuale connessione ODBC/OLE DB/Personalizzata. Questa istruzione è opzionale.

```
Disconnect
```

#### drop field

Durante l'esecuzione dello script, in qualsiasi momento è possibile rilasciare dal modello dati e quindi dalla memoria uno o più campi di Qlik Sense utilizzando l'istruzione **drop field**. La proprietà "distinct" di una tabella viene rimossa dopo un'istruzione **drop field**.



*Sia **drop field** che **drop fields** sono formati consentiti, senza alcuna differenza effettiva. Se nessuna tabella viene specificata, il campo verrà rilasciato da tutte le tabelle in cui ricorre.*

```
Drop field fieldname [ , fieldname2 ...] [from tablename1 [ , tablename2 ...]]
```

```
drop fields fieldname [ , fieldname2 ...] [from tablename1 [ , tablename2 ...]]
```

#### drop table

Durante l'esecuzione dello script, è possibile rilasciare dal modello dati, e quindi dalla memoria, una o più tabelle interne di Qlik Sense utilizzando l'istruzione **drop table**.



*I formati **drop table** e **drop tables** sono entrambi accettati.*

```
Drop table tablename [ , tablename2 ...]
```

```
drop tables [ tablename [ , tablename2 ...]]
```

### 3 Istruzioni e parole chiave dello script

---

#### Execute

L'istruzione **Execute** viene utilizzata per eseguire altri programmi, mentre Qlik Sense sta caricando i dati. Ad esempio, per effettuare le connessioni necessarie.

```
Execute commandline
```

#### FlushLog

L'istruzione **FlushLog** obbliga Qlik Sense a scrivere il contenuto del buffer dello script nel file di registro dello script.

```
FlushLog
```

#### Force

L'istruzione **force** impone a Qlik Sense di interpretare i nomi e i valori di campo delle istruzioni **LOAD** e **SELECT** successive in formato solo maiuscolo, solo minuscolo, sempre maiuscolo o conformemente alla visualizzazione attuale (formato misto). Questa istruzione permette di associare i valori di campo provenienti da tabelle create in base a convenzioni differenti.

```
Force ( capitalization | case upper | case lower | case mixed )
```

#### LOAD

L'istruzione **LOAD** carica i campi da un file, dai dati definiti nello script, da una tabella caricata in precedenza, da una pagina Web, dal risultato di un'istruzione **SELECT** seguente o dalla generazione automatica di dati. È anche possibile caricare dati da connessioni di analisi.

```
Load [ distinct ] *fieldlist  
  
[ ( from file [ format-spec ] |  
from_field fieldsource [format-spec]  
inline data [ format-spec ] |  
resident table-label |  
autogenerate size ) ]  
  
[ where criterion | while criterion ]  
  
[ group_by groupbyfieldlist ]  
  
[order_by orderbyfieldlist ]  
  
[extension pluginname.functionname (tabledescription) ]
```

#### Let

L'istruzione **let** è un complemento all'istruzione **set**, utilizzata per definire le variabili degli script.

L'istruzione **let**, a differenza dell'istruzione **set**, valuta l'espressione posta sul lato destro del simbolo '=' al tempo di esecuzione dello script prima dell'assegnazione alla variabile.

### 3 Istruzioni e parole chiave dello script

```
Let variablename=expression
```

#### Loosen Table

Una o più tabelle dati interne di Qlik Sense possono essere dichiarate logicamente disconnesse in modo esplicito durante l'esecuzione dello script, utilizzando un'istruzione **Loosen Table**. Quando una tabella è logicamente disconnessa, tutte le associazioni tra i valori di campo nella tabella vengono rimosse. È possibile ottenere un effetto simile caricando ogni campo della tabella logicamente disconnessa come tabelle indipendenti e scollegate. La disconnessione logica può rivelarsi utile durante il controllo per isolare temporaneamente parti differenti della struttura dei dati. Nel visualizzatore tabelle è possibile riconoscere una tabella logicamente disconnessa dalle linee punteggiate. L'utilizzo di una o più istruzioni **Loosen Table** nello script indica a Qlik Sense di ignorare ogni impostazione di tabelle logicamente disconnesse effettuata prima dell'esecuzione dello script.

```
tablename [ , tablename2 ...]
```

```
Loosen Tables tablename [ , tablename2 ...]
```

#### Map ... using

L'istruzione **map ... using** viene usata per eseguire il mapping di un certo valore di campo o una certa espressione sui valori di una tabella di mapping specifica. La tabella di mapping viene creata utilizzando l'istruzione **Mapping**.

```
Map *fieldlist Using mapname
```

#### NullAsNull

L'istruzione **NullAsNull** disattiva la conversione dei valori NULL in valori di stringa impostati in precedenza da un'istruzione **NullAsValue**.

```
NullAsNull *fieldlist
```

#### NullAsValue

L'istruzione **NullAsValue** specifica per quali campi NULL deve essere convertito in un valore.

```
NullAsValue *fieldlist
```

#### Qualify

L'istruzione **Qualify** consente di modificare la qualificazione dei nomi di campo, ad esempio il nome della tabella dei nomi di campo diventerà un prefisso.

```
Qualify *fieldlist
```

#### Rem

L'istruzione **rem** viene utilizzata per inserire osservazioni, o commenti, negli script o per disattivare temporaneamente istruzioni dello script senza rimuoverle.

```
Rem string
```

#### Rename Field

Questa funzione di script rinomina uno o più campi di Qlik Sense esistenti dopo che sono stati caricati.

```
Rename field (using mapname | oldname to newname{ , oldname to newname })
```

```
Rename Fields (using mapname | oldname to newname{ , oldname to newname })
```

### Rename Table

Questa funzione di script rinomina una o più tabelle interne di Qlik Sense esistenti dopo che sono state caricate.

```
Rename table (using mapname | oldname to newname{ , oldname to newname })
```

```
Rename Tables (using mapname | oldname to newname{ , oldname to newname })
```

### Section

L'istruzione **section** consente di definire se le istruzioni successive **LOAD** e **SELECT** devono essere considerate come dati o come una definizione dei diritti di accesso.

```
Section (access | application)
```

### Select

La selezione dei campi da una sorgente dati ODBC o da un provider OLE DB viene eseguita utilizzando le istruzioni SQL **SELECT** standard. Tuttavia, l'ambito nel quale le istruzioni **SELECT** vengono accettate dipende dal driver ODBC o dal provider OLE DB utilizzato.

```
Select [all | distinct | distinctrow | top n [percent] ] *fieldlist  
  
From tablelist  
  
[Where criterion ]  
  
[Group by fieldlist [having criterion ] ]  
  
[Order by fieldlist [asc | desc] ]  
  
[ (Inner | Left | Right | Full)Join tablename on fieldref = fieldref ]
```

### Set

L'istruzione **set** viene utilizzata per definire le variabili di script. Le variabili possono essere utilizzate per sostituire stringhe, percorsi, unità e così via.

```
Set variablename=string
```

### Sleep

L'istruzione **sleep** interrompe l'esecuzione dello script per il periodo di tempo specificato.

```
Sleep n
```

### SQL

L'istruzione **SQL** consente di inviare un comando arbitrario SQL tramite una connessione ODBC o OLE DB.

```
SQL sql_command
```

## 3 Istruzioni e parole chiave dello script

---

### SQLColumns

L'istruzione **sqlcolumns** restituisce un gruppo di campi che descrivono le colonne di una sorgente dati ODBC o OLE DB sulla quale è stata eseguita un'istruzione **connect**.

[SQLColumns](#)

### SQLTables

L'istruzione **sqltables** restituisce un gruppo di campi che descrivono le tabelle di una sorgente dati ODBC o OLE DB sulla quale è stata eseguita un'istruzione **connect**.

[SQLTables](#)

### SQLTypes

L'istruzione **sqltypes** restituisce un gruppo di campi che descrivono i tipi di una sorgente dati ODBC o OLE DB sulla quale è stata eseguita un'istruzione **connect**.

[SQLTypes](#)

### Star

La stringa utilizzata per rappresentare l'insieme di tutti i valori di un campo nel database può essere impostata tramite l'istruzione **star**. Interessa le istruzioni **LOAD** e **SELECT** successive.

[Star](#) is [ string ]

### Store

L'istruzione **Store** crea un file QVD, Parquet, CSV, o TXT.

[Store](#) [ \*fieldlist **from** ] table **into** filename [ format-spec ] ;

### Tag

Questa istruzione dello script fornisce un modo per assegnare tag a uno o più campi o tabelle. Se viene effettuato un tentativo di contrassegnare un campo o una tabella non presente nell'app, tale contrassegno verrà ignorato. In caso di conflitto nelle ricorrenze di un nome di campo o di tag, si utilizza l'ultimo valore trovato.

[Tag](#)[field|fields] fieldlist **with** tagname

[Tag](#) [field|fields] fieldlist **using** mapname

[Tag](#) table tablelist **with** tagname

### Trace

L'istruzione **trace** esegue la scrittura di una stringa nella finestra **Avanzamento dell'esecuzione dello script** e nel file di log dello script, quando viene utilizzato. Si rivela molto utile per le operazioni di debug. L'uso delle espansioni \$ delle variabili calcolate prima dell'istruzione **trace** consente di personalizzare il messaggio.

[Trace](#) string

### Unmap

L'istruzione **Unmap** disattiva il mapping del valore di campo specificato da un'istruzione **Map ... Using** precedente per i campi caricati successivamente.

```
Unmap *fieldlist
```

### Unqualify

L'istruzione **Unqualify** viene utilizzata per disattivare la qualificazione dei nomi di campo che era stata precedentemente attivata dall'istruzione **Qualify**.

```
Unqualify *fieldlist
```

### Untag

Questa istruzione dello script fornisce un modo per rimuovere tag da campi o tabelle. Se viene effettuato un tentativo di rimozione di un contrassegno da un campo o una tabella non presente nell'app, tale rimozione verrà ignorata.

```
Untag[field|fields] fieldlist with tagname
```

```
Tag [field|fields] fieldlist using mapname
```

```
Tag table tablelist with tagname
```

## Alias

L'istruzione **alias** viene utilizzata per impostare un alias in base al quale verrà rinominato un campo quando questo ricorrerà nello script che segue.

### Sintassi:

```
alias fieldname as aliasname {,fieldname as aliasname}
```

### Argomenti:

Argomenti

| Argomento | Descrizione                                |
|-----------|--------------------------------------------|
| fieldname | Il nome del campo nei dati di origine      |
| aliasname | Un nome alias che si preferisce utilizzare |

## Esempi con risultati

Per gli esempi della tabella sottostante, le modifiche del nome del campo definito verranno applicate ai riferimenti del nome campo originale in tutte le istruzioni **SELECT** e **LOAD** successive. È possibile definire un nuovo alias per un nome campo mediante una nuova istruzione **alias** in qualsiasi posizione successiva nello script.

Il campo viene anche referenziato con il nome alias nel modello dati e nel contenuto dell'app.

### 3 Istruzioni e parole chiave dello script

#### Esempi e risultati

| Esempio                                  | Risultato                                                                                                                 |
|------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| Alias ID_N as NameID;                    | Il campo <i>ID_N</i> viene rinominato in <i>NameID</i> .                                                                  |
| Alias A as Name, B as Number, C as Date; | I campi <i>A</i> , <i>B</i> e <i>C</i> vengono rinominati in <i>Name</i> , <i>Number</i> e <i>Date</i> , rispettivamente. |

Esempio di formato lungo

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

#### Script di caricamento

```
alias [A] as [AAA];
T1:
Load 'string 1' as A autogenerate 1;
T2:
Load 'string 2' as B autogenerate 1;
T3:
Load 'string 3' as A autogenerate 1;
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- AAA

Tabella  
dei risultati

|          |
|----------|
| AAA      |
| string 1 |
| string 3 |

## AutoNumber

Questa istruzione crea un valore intero univoco per ciascun valore calcolato distinto in un campo rilevato durante l'esecuzione dello script.

È possibile utilizzare la funzione [autonumber \(page 686\)](#) anche all'interno di un'istruzione **LOAD**, ma vi sono alcune limitazioni se si desidera utilizzare un caricamento ottimizzato. È possibile creare un caricamento ottimizzato caricando prima i dati da un file **QVD** e poi utilizzando l'istruzione **AutoNumber** per convertire i valori in chiavi di simboli.

#### Sintassi:

```
AutoNumber *fieldlist [Using namespace] ]
```

### Argomenti:

#### Argomenti

| Argomento  | Descrizione                                                                                                                                                                                                                                                                                                                                                                                    |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| *fieldlist | <p>Un elenco separato da virgole dei campi i cui valori devono essere sostituiti da un valore intero univoco.</p> <p>È possibile utilizzare i caratteri jolly ? e * nei nomi dei campi per includere tutti i campi con nomi corrispondenti. È anche possibile utilizzare * per includere tutti i campi. Quando si utilizzano caratteri jolly, racchiudere i nomi dei campi tra virgolette.</p> |
| namespace  | <p><b>Using namespace</b> è facoltativo. È possibile utilizzare questa opzione se si desidera creare uno spazio dei nomi, in cui valori identici in campi diversi condividono la stessa chiave.</p> <p>Se non si utilizza questa opzione, tutti i campi saranno indicizzati con una chiave separata.</p>                                                                                       |

### Limiti:

Quando nello script sono presenti più istruzioni **LOAD**, è necessario inserire l'istruzione **AutoNumber** dopo l'istruzione **LOAD** finale.

Esempio - script con AutoNumber

### Esempio di script

In questo esempio, i dati vengono prima caricati senza l'istruzione **AutoNumber**. L'istruzione **AutoNumber** viene quindi aggiunta per mostrare l'effetto.

### Dati utilizzati nell'esempio

Caricare i dati seguenti come un caricamento inline nell'editor caricamento dati per creare l'esempio di script in basso. Lasciare l'istruzione **AutoNumber** non commentata per il momento.

```
RegionSales:
LOAD *,
Region &'|'|& Year &'|'|& Month as KeyToOtherTable
INLINE
[ Region, Year, Month, Sales
North, 2014, May, 245
North, 2014, May, 347
North, 2014, June, 127
South, 2014, June, 645
South, 2013, May, 367
South, 2013, May, 221
];

Budget:
LOAD Budget,
Region &'|'|& Year &'|'|& Month as KeyToOtherTable
```

### 3 Istruzioni e parole chiave dello script

```
INLINE
[Region, Year, Month, Budget
North, 2014, May, 200
North, 2014, May, 350
North, 2014, June, 150
South, 2014, June, 500
South, 2013, May, 300
South, 2013, May, 200
];

//AutoNumber KeyToOtherTable;
```

#### Creazione di visualizzazioni

Creare due visualizzazioni di tabelle in un foglio Qlik Sense. Aggiungere **KeyToOtherTable**, **Region**, **Year**, **Month** e **Sales** come dimensioni alla prima tabella. Aggiungere **KeyToOtherTable**, **Region**, **Year**, **Month** e **Budget** come dimensioni alla seconda tabella.

#### Risultato

Tabella RegionSales

| KeyToOtherTable | Region | Year | Month | Sales |
|-----------------|--------|------|-------|-------|
| North 2014 June | North  | 2014 | June  | 127   |
| North 2014 May  | North  | 2014 | May   | 245   |
| North 2014 May  | North  | 2014 | May   | 347   |
| South 2013 May  | South  | 2013 | May   | 221   |
| South 2013 May  | South  | 2013 | May   | 367   |
| South 2014 June | South  | 2014 | June  | 645   |

Tabella budget

| KeyToOtherTable | Region | Year | Month | Budget |
|-----------------|--------|------|-------|--------|
| North 2014 June | North  | 2014 | June  | 150    |
| North 2014 May  | North  | 2014 | May   | 200    |
| North 2014 May  | North  | 2014 | May   | 350    |
| South 2013 May  | South  | 2013 | May   | 200    |
| South 2013 May  | South  | 2013 | May   | 300    |
| South 2014 June | South  | 2014 | June  | 500    |

#### Spiegazione

L'esempio mostra un campo composito **KeyToOtherTable** che collega le due tabelle. **AutoNumber** non è utilizzato. Notare la lunghezza dei valori **KeyToOtherTable**.

### Aggiungi istruzione AutoNumber

Eliminare il commento all'istruzione **AutoNumber** nello script di caricamento.

```
AutoNumber KeyToOtherTable;
```

### Risultato

Tabella RegionSales

| KeyToOtherTable | Region | Year | Month | Sales |
|-----------------|--------|------|-------|-------|
| 1               | North  | 2014 | June  | 127   |
| 1               | North  | 2014 | May   | 245   |
| 2               | North  | 2014 | May   | 347   |
| 3               | South  | 2013 | May   | 221   |
| 4               | South  | 2013 | May   | 367   |
| 4               | South  | 2014 | June  | 645   |

Tabella budget

| KeyToOtherTable | Region | Year | Month | Budget |
|-----------------|--------|------|-------|--------|
| 1               | North  | 2014 | June  | 150    |
| 1               | North  | 2014 | May   | 200    |
| 2               | North  | 2014 | May   | 350    |
| 3               | South  | 2013 | May   | 200    |
| 4               | South  | 2013 | May   | 300    |
| 4               | South  | 2014 | June  | 500    |

### Spiegazione

I valori di campo **KeyToOtherTable** sono stati sostituiti con valori interi univoci e, di conseguenza, la lunghezza dei valori dei campi è stata ridotta, conservando così la memoria. I campi chiave di entrambe le tabelle sono interessati da **AutoNumber** e le tabelle rimangono collegate. L'esempio è breve a scopo dimostrativo, ma risulterebbe significativo con una tabella contenente un elevato numero di righe.

### Binary

L'istruzione **binary** consente di caricare i dati da un'altra app Qlik Sense o documento QlikView, inclusi i dati in Section Access. Gli altri elementi dell'app non sono inclusi, ad esempio fogli, racconti, visualizzazioni, voci principali o variabili.

Nello script è consentita una sola istruzione **binary**. L'istruzione **binary** deve essere la prima istruzione dello script, ancora prima delle istruzioni SET, che sono generalmente poste all'inizio dello script.

### Sintassi:

```
binary [path] filename
```

### Argomenti:

#### Argomenti

| Argomento | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| path      | <p>Il percorso del file che deve corrispondere a un riferimento a una connessione dati della cartella. Ciò si rivela necessario se il file non si trova nella directory di lavoro di Qlik Sense.</p> <p><b>Esempio: <i>'lib://Table Files/'</i></b></p> <p>Nella modalità di creazione degli script legacy sono supportati anche i seguenti formati di percorso:</p> <ul style="list-style-type: none"><li>• assoluto</li></ul> <p><b>Esempio: <i>c:\data</i></b></p> <ul style="list-style-type: none"><li>• relativo dell'app contenente questa riga di script</li></ul> <p><b>Esempio: <i>data</i></b></p> |
| filename  | Il nome del file, inclusa l'estensione del file .qvw o .qvf.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

### Limiti:

Non è possibile utilizzare **binary** per caricare dati da un'app nella stessa distribuzione di Qlik Sense Enterprise facendo riferimento all'ID dell'app. Il caricamento può essere effettuato solo da un file .qvf.

### Esempi

| Stringa                                            | Descrizione                                                                                                                                                                                                                                                                                                                          |
|----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>Binary lib://DataFolder/customer.qvw;</code> | In questo esempio, il file deve essere inserito nella connessione dati della <b>cartella</b> . Tale posizione può essere, ad esempio, una cartella creata dall'amministratore sul server Qlik Sense. Fare clic su <b>Crea nuova connessione</b> nell'editor caricamento dati e selezionare <b>Cartella</b> in <b>Percorsi file</b> . |
| <code>Binary customer.qvf;</code>                  | In questo esempio, il file deve essere inserito nella directory di lavoro Qlik Sense.                                                                                                                                                                                                                                                |
| <code>Binary c:\qv\customer.qvw;</code>            | Questo esempio, in cui viene utilizzato un percorso di file assoluto, può essere utilizzato solo nella modalità di creazione degli script legacy.                                                                                                                                                                                    |

### Comment field

Fornisce un modo per visualizzare i commenti dei campi (metadati) da database e fogli di calcolo. I nomi di campo non presenti nell'app verranno ignorati. In caso di più ricorrenze di uno stesso nome di campo, verrà applicato l'ultimo valore utilizzato.

#### Sintassi:

```
comment [fields] *fieldlist using mapname
```

```
comment [field] fieldname with comment
```

La tabella di mapping utilizzata deve essere composta da due colonne, la prima contenente i nomi dei campi e la seconda i commenti.

#### Argomenti:

##### Argomenti

| Argomento         | Descrizione                                                                                                                                                                                                                                                                              |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>*fieldlist</i> | Un elenco separato da virgole dei campi da commentare. L'utilizzo di * per l'elenco dei campi indica tutti i campi. Nei nomi di campo sono consentiti i caratteri speciali * e ?. Se si utilizzano i caratteri speciali può essere necessario delimitare i nomi di campo tra virgolette. |
| <i>mapname</i>    | Il nome di una tabella di mapping letta in precedenza in un'istruzione mapping <b>LOAD</b> o mapping <b>SELECT</b> .                                                                                                                                                                     |
| <i>fieldname</i>  | Il nome del campo al quale si desidera aggiungere il commento.                                                                                                                                                                                                                           |
| <i>comment</i>    | Il commento da aggiungere al campo.                                                                                                                                                                                                                                                      |

#### Example 1:

```
commentmap:
```

```
mapping LOAD * inline [
```

```
a,b
```

```
Alpha,This field contains text values
```

```
Num,This field contains numeric values
```

```
];
```

```
comment fields using commentmap;
```

#### Example 2:

```
comment field Alpha with AFieldContainingCharacters;
```

```
comment field Num with '*A field containing numbers';
```

```
comment Gamma with 'Mickey Mouse field';
```

### Comment table

Fornisce un modo per visualizzare i commenti delle tabelle (metadati) da database o fogli di calcolo.

I nomi di tabella non presenti nell'app vengono ignorati. In caso di più ricorrenze di uno stesso nome di tabella, viene utilizzato l'ultimo valore trovato. La parola chiave può essere utilizzata per leggere i commenti da una sorgente dati.

#### Sintassi:

```
comment [tables] tablelist using mapname  
comment [table] tablename with comment
```

#### Argomenti:

| Argomenti        |                                                                                                                      |
|------------------|----------------------------------------------------------------------------------------------------------------------|
| Argomento        | Descrizione                                                                                                          |
| <i>tablelist</i> | (table{,table})                                                                                                      |
| <i>mapname</i>   | Il nome di una tabella di mapping letta in precedenza in un'istruzione mapping <b>LOAD</b> o mapping <b>SELECT</b> . |
| <i>tablename</i> | Il nome della tabella alla quale si desidera aggiungere il commento.                                                 |
| <i>comment</i>   | Il commento da aggiungere alla tabella.                                                                              |

#### Example 1:

```
Commentmap:  
mapping LOAD * inline [  
a,b  
Main,This is the fact table  
Currencies, Currency helper table  
];  
comment tables using Commentmap;
```

#### Example 2:

```
comment table Main with 'Main fact table';
```

### Connect

L'istruzione **CONNECT** consente di definire l'accesso di Qlik Sense a un database generico mediante l'interfaccia OLE DB/ODBC. Per ODBC, occorre innanzitutto specificare la sorgente dati utilizzando l'amministratore ODBC.

### 3 Istruzioni e parole chiave dello script



*Questa funzionalità non è disponibile in Qlik Sense SaaS.*



*Questa istruzione supporta solo le connessioni dati della cartella in modalità standard.*

#### Sintassi:

```
ODBC CONNECT TO connect-string
```

```
OLEDB CONNECT TO connect-string
```

```
CUSTOM CONNECT TO connect-string
```

```
LIB CONNECT TO connection
```

#### Argomenti:

##### Argomenti

| Argomento      | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| connect-string | <p><code>connect-string ::= datasourcename { ; conn-spec-item }</code></p> <p>La stringa di connessione è il nome della sorgente dati e un elenco opzionale di uno o più voci di specifica della connessione. Se il nome della sorgente dati contiene spazi vuoti o se vengono elencati voci di specifica della connessione, la stringa di connessione dovrà essere racchiusa tra virgolette.</p> <p><b>datasourcename</b> deve essere una sorgente dati ODBC definita o una stringa che definisce un provider OLE DB.</p> <p><code>conn-spec-item ::= DBQ=database_specifier   DriverID=driver_specifier   UID=userid   PWD=password</code></p> <p>Le possibili voci di specifica della connessione possono differire a seconda del database. Per alcuni database sono disponibili anche voci diverse da quelle indicate. Per OLE DB, alcune delle voci di specifica della connessione sono obbligatori e non facoltativi.</p> |
| connection     | <p>Il nome di una connessione dati memorizzata nell'editor caricamento dati.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

Se **ODBC** viene posizionato prima di **CONNECT**, verrà utilizzata l'interfaccia ODBC; altrimenti, verrà utilizzata OLE DB.

L'utilizzo di **LIB CONNECT TO** determina la connessione a un database mediante una connessione dati memorizzata che è stata creata nell'editor caricamento dati.

### Example 1:

```
ODBC CONNECT TO 'Sales
```

```
DBQ=C:\Program Files\Access\Samples\Sales.mdb';
```

La sorgente dati definita con questa dichiarazione viene utilizzata dalle successive istruzioni **Select (SQL)**, fino a quando non viene utilizzata una nuova istruzione **CONNECT**.

### Example 2:

```
LIB CONNECT TO 'DataConnection';
```

### Connect32

Questa istruzione viene utilizzata come l'istruzione **CONNECT**, ma impone a un sistema a 64 bit di utilizzare un provider ODBC/OLE DB a 32 bit. Non applicabile in caso di istruzione connect personalizzata.

### Connect64

Questa istruzione viene utilizzata come l'istruzione **CONNECT**, ma impone l'utilizzo di un provider a 64 bit. Non applicabile in caso di istruzione connect personalizzata.

## Declare

L'istruzione **Declare** consente di creare definizioni di campo in cui è possibile definire relazioni tra i campi o le funzioni. È possibile utilizzare una serie di definizioni di campo per generare automaticamente campi derivati, che possono essere utilizzati come dimensioni. Ad esempio, è possibile creare una definizione di calendario e utilizzarla per generare dimensioni correlate, come ad esempio anno, mese, settimana e giorno da un campo della data.

È possibile utilizzare **Declare** per impostare una nuova definizione di campo o per creare una definizione di campo basata su una definizione già esistente.

## Impostazione di una nuova definizione di campo

### Sintassi:

```
definition_name:
```

```
Declare [Field[s]] Definition [Tagged tag_list ]
```

```
[Parameters parameter_list ]
```

```
Fields field_list
```

### Argomenti:

| Argomento           | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| definition_<br>name | <p>Nome della definizione di campo, che termina con i due punti.</p> <div><i>Non utilizzare autoCalendar come definizione dei campi del calendario, in quanto questo nome è riservato per i modelli del calendario generati automaticamente.</i></div> <p><b>Esempio:</b></p> <pre>Calendar:</pre>                                                                                                            |
| tag_list            | <p>Un elenco separato da virgole di tag da applicare ai campi derivati dalla definizione di campo. L'applicazione dei tag è opzionale, tuttavia, se non si applicano i tag utilizzati per specificare l'ordinamento, ad esempio \$date, \$numeric o \$text, per impostazione predefinita il campo derivato verrà ordinato in base all'ordine di caricamento.</p> <p><b>Esempio:</b></p> <pre>'\$date' Thank you for bringing this to our attention, and apologies for the inconvenience.</pre> |
| parameter_<br>list  | <p>Un elenco separato da virgole di parametri. A un parametro definito nella forma name=value viene assegnato un valore iniziale che può essere sostituito quando viene riutilizzata la definizione di campo. Opzionale.</p> <p><b>Esempio:</b></p> <pre>first_month_of_year = 1</pre>                                                                                                                                                                                                         |
| field_list          | <p>Un elenco separato da virgole di campi da generare quando viene utilizzata la definizione di campo. Un campo è definito nella forma &lt;expression&gt; As field_name tagged tag. Utilizzare \$1 per fare riferimento al campo dati dal quale devono essere generati i campi derivati.</p> <p><b>Esempio:</b></p> <pre>Year(\$1) As Year tagged ('\$numeric')</pre>                                                                                                                          |

### Esempio:

```
Calendar:
DECLARE FIELD DEFINITION TAGGED '$date'
  Parameters
    first_month_of_year = 1
  Fields
```

```
Year($1) As Year Tagged ('$numeric'),
Month($1) as Month Tagged ('$numeric'),
Date($1) as Date Tagged ('$date'),
Week($1) as Week Tagged ('$numeric'),
Weekday($1) as Weekday Tagged ('$numeric'),
DayNumberOfYear($1, first_month_of_year) as DayNumberOfYear Tagged ('$numeric')
;
```

Il calendario è ora definito ed è possibile applicarlo ai campi data caricati, in questo caso OrderDate e ShippingDate, utilizzando una clausola **Derive**.

### Riutilizzo di una definizione di campo esistente

#### Sintassi:

```
<definition name>:
```

```
Declare [Field][s] Definition
```

```
Using <existing_definition>
```

```
[With <parameter_assignment> ]
```

#### Argomenti:

| Argomento                | Descrizione                                                                                                                                                                                                                                                                                                                                                  |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| definition_<br>name      | Nome della definizione di campo, che termina con i due punti.<br><br><b>Esempio:</b><br><br>MyCalendar:                                                                                                                                                                                                                                                      |
| existing_<br>definition  | La definizione di campo da riutilizzare durante la creazione di una nuova definizione di campo. La nuova definizione di campo funziona allo stesso modo della definizione sulla quale si basa, con l'eccezione dell'uso di parameter_assignment per modificare un valore utilizzato nelle espressioni di campo.<br><br><b>Esempio:</b><br><br>Using Calendar |
| parameter_<br>assignment | Un elenco separato da virgole di assegnazioni di parametri. Un'assegnazione di parametro viene definita nella forma name=value e sostituisce il valore del parametro impostato nella definizione di campo di base. Opzionale.<br><br><b>Esempio:</b><br><br>first_month_of_year = 4                                                                          |

### Esempio:

In questo esempio viene riutilizzata la definizione di calendario creata nell'esempio precedente. In questo caso si desidera utilizzare un anno fiscale che inizi ad aprile. Ciò si ottiene assegnando il valore 4 al parametro `first_month_of_year`, che influirà sul campo `DayNumberOfYear` definito.

Nell'esempio si suppone di utilizzare i dati campione e la definizione di campo dell'esempio precedente.

MyCalendar:

```
DECLARE FIELD DEFINITION USING Calendar WITH first_month_of_year=4;
```

```
DERIVE FIELDS FROM FIELDS OrderDate,ShippingDate USING MyCalendar;
```

Una volta ricaricato lo script di dati, i campi generati sono disponibili nell'editor dei fogli con i nomi `OrderDate.MyCalendar.*` e `ShippingDate.MyCalendar.*`.

### Derive

L'istruzione **Derive** consente di generare campi derivati in base a una definizione di campo creata mediante un'istruzione **Declare**. È possibile specificare i campi dati per cui derivare i campi oppure derivarli esplicitamente o implicitamente in base ai tag di campo.

#### Sintassi:

```
Derive [fields] From [Field[s]] field_list Using definition
```

```
Derive [Field[s]] From Explicit [Tag[s]] tag_list Using definition
```

```
Derive [Field[s]] From Implicit [Tag[s]] Using definition
```

#### Argomenti:

##### Argomenti

| Argomento  | Descrizione                                                                                                                                                                                                                               |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| definition | Nome della definizione di campo da utilizzare per la derivazione dei campi.<br><br><b>Esempio: calendar</b>                                                                                                                               |
| field_list | Un elenco separato da virgole di campi dati da cui devono essere generati i campi derivati, in base alla definizione di campo. I campi dati devono essere campi già caricati nello script.<br><br><b>Esempio: orderDate, shippingDate</b> |

## 3 Istruzioni e parole chiave dello script

| Argomento | Descrizione                                                                                                                                                                                                                                                 |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| tag_list  | Un elenco separato da virgole di tag. I campi derivati verranno generati per tutti i campi dati con uno qualsiasi dei tag inclusi nell'elenco. L'elenco dei tag deve essere racchiuso tra parentesi tonde.<br><br><b>Esempio:</b> ('\$date', '\$timestamp') |

### Esempi:

- Derivare campi per campi dati specifici.  
In questo caso è necessario specificare i campi OrderDate e ShippingDate.  
`DERIVE FIELDS FROM FIELDS OrderDate,ShippingDate USING Calendar;`
- Derivare campi per tutti i campi con un tag specifico.  
In questo caso è necessario derivare i campi in base a Calendar per tutti i campi con un tag \$date.  
`DERIVE FIELDS FROM EXPLICIT TAGS ('$date') USING Calendar;`
- Derivare campi per tutti i campi con il tag della definizione di campo.  
In questo caso è necessario derivare i campi per tutti i campi dati con lo stesso tag della definizione di campo Calendar, in questo caso \$date.  
`DERIVE FIELDS FROM IMPLICIT TAG USING Calendar;`

## Direct Query

L'istruzione **DIRECT QUERY** consente di accedere alle tabelle mediante una connessione ODBC o OLE DB utilizzando la funzione Direct Discovery.

### Sintassi:

```
DIRECT QUERY DIMENSION fieldlist [MEASURE fieldlist] [DETAIL fieldlist] FROM
tablelist
[WHERE where_clause]
```

Le parole chiave **DIMENSION**, **MEASURE** e **DETAIL** possono essere utilizzate in qualsiasi ordine.

Le clausole con parole chiave **DIMENSION** e **FROM** sono richieste in tutte le istruzioni **DIRECT QUERY**. La parola chiave **FROM** deve essere collocata dopo la parola chiave **DIMENSION**.

I campi specificati direttamente dopo la parola chiave **DIMENSION** vengono caricati in memoria e possono essere utilizzati per creare associazioni tra dati in memoria e dati Direct Discovery.



*L'istruzione **DIRECT QUERY** non può contenere le clausole **DISTINCT** o **GROUP BY**.*

La parola chiave **MEASURE** consente di definire i campi che Qlik Sense riconosce su un "metalivello". I dati effettivi di un campo di misura risiedono nel database solo durante il processo di caricamento dei dati e vengono recuperati ad hoc dalle espressioni grafiche utilizzate in una visualizzazione.

Generalmente, i campi contenenti valori discreti che verranno utilizzati come dimensioni devono essere caricati con la parola chiave **DIMENSION**, mentre i numeri che verranno utilizzati solo nelle aggregazioni devono essere selezionati con la parola chiave **MEASURE**.

### 3 Istruzioni e parole chiave dello script

I campi **DETAIL** forniscono informazioni o dettagli, ad esempio campi dei commenti, che un utente può desiderare di visualizzare in una tabella di analisi dei dettagli. I campi **DETAIL** non possono essere utilizzati nelle espressioni grafiche.

Per impostazione predefinita, l'istruzione **DIRECT QUERY** è una sorgente dati neutra per le sorgenti dati che supportano SQL. Per tale motivo, è possibile utilizzare la stessa l'istruzione **DIRECT QUERY** per database SQL diversi senza che sia necessario apportare modifiche. Direct Discovery genera query adatte al database in base alle esigenze.

La sintassi della sorgente dati nativa può essere utilizzata quando l'utente conosce il database a cui inviare le query e desidera utilizzare estensioni specifiche del database in SQL. Viene supportata la sintassi della sorgente dati nativa:

- Come espressioni di campo nelle clausole **DIMENSION** e **MEASURE**
- Come contenuto della clausola **WHERE**

Esempi:

**DIRECT QUERY**

```
DIMENSION Dim1, Dim2
MEASURE
    NATIVE ('X % Y') AS X_MOD_Y
```

FROM TableName

**DIRECT QUERY**

```
DIMENSION Dim1, Dim2
MEASURE X, Y
FROM TableName
WHERE NATIVE ('EMAIL MATCHES "\*.EDU"')
```



*I seguenti termini vengono utilizzati come parole chiave perciò non possono essere utilizzati come nomi di colonna o campo senza che vengano racchiusi tra virgolette: and, as, detach, detail, dimension, distinct, from, in, is, like, measure, native, not, or, where*

**Argomenti:**

| Argomento | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| fieldlist | Un elenco separato da virgole di specifiche dei campi, <i>fieldname {, fieldname}</i> . Una specifica di un campo può essere costituita da un nome di campo, condizione in cui lo stesso nome viene utilizzato per il nome di colonna del database e il nome di campo di Qlik Sense. Oppure una specifica di campo può essere costituita da un "alias di campo", nel cui caso a un'espressione di database o a un nome di colonna viene assegnato un nome di campo di Qlik Sense. |
| tablelist | Un elenco di nomi di tabelle o di viste del database da cui vengono caricati i dati. In genere, saranno viste contenenti un JOIN che viene eseguito sul database.                                                                                                                                                                                                                                                                                                                 |

### 3 Istruzioni e parole chiave dello script

| Argomento    | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| where_clause | <p>In questo caso, non viene definita la sintassi completa delle clausole <b>WHERE</b>, tuttavia è consentita la maggior parte delle "espressioni relazioni" di SQL, compreso l'utilizzo delle chiamate di funzione, l'operatore <b>LIKE</b> per le stringhe, <b>IS NULL</b> e <b>IS NOT NULL</b> e <b>IN</b>. <b>BETWEEN</b> non è compreso.</p> <p><b>NOT</b> è un operatore unario, contrariamente al modificatore di alcune parole chiave.</p> <p>Esempi:</p> <pre>WHERE x &gt; 100 AND "Region Code" IN ('south', 'west') WHERE Code IS NOT NULL and Code LIKE '%prospect' WHERE NOT X in (1,2,3)</pre> <p>L'ultimo esempio non può essere scritto come:</p> <pre>WHERE X NOT in (1,2,3)</pre> |

#### Esempio:

In questo esempio viene utilizzata una tabella del database denominata `TableName`, contenente i campi `Dim1`, `Dim2`, `Num1`, `Num2` e `Num3`. `Dim1` e `Dim2` verranno caricati nella serie di dati Qlik Sense.

```
DIRECT QUERY DIMENSION Dim1, Dim2 MEASURE Num1, Num2, Num3 FROM TableName ;
```

`Dim1` e `Dim2` saranno disponibili per essere utilizzati come dimensioni. `Num1`, `Num2` e `Num3` saranno disponibili per le aggregazioni. Anche `Dim1` e `Dim2` saranno disponibili per le aggregazioni. Il tipo di aggregazioni per le quali è possibile utilizzare `Dim1` e `Dim2` dipende dai relativi tipi di dati. Ad esempio, in molti casi i campi **DIMENSION** contengono dati di stringa, ad esempio nomi o numeri di account. Questi dati non possono essere sommati, tuttavia possono essere conteggiati: `count(Dim1)`.



Le istruzioni **DIRECT QUERY** vengono scritte direttamente nell'editor di script. Per semplificare la costruzione delle istruzioni **DIRECT QUERY**, è possibile generare un'istruzione **SELECT** da una connessione dati, quindi modificare lo script generato per trasformarlo in un'istruzione **DIRECT QUERY**.

Ad esempio, l'istruzione **SELECT**:

```
SQL SELECT
  SalesOrderID,
  RevisionNumber,
  OrderDate,
  SubTotal,
  TaxAmt
FROM MyDB.Sales.SalesOrderHeader;
```

può essere modificata nella seguente istruzione **DIRECT QUERY**:

```
DIRECT QUERY
DIMENSION
  SalesOrderID,
  RevisionNumber

MEASURE
  SubTotal,
  TaxAmt

DETAIL
  OrderDate

FROM MyDB.Sales.SalesOrderHeader;
```

## Elenchi dei campi Direct Discovery

Un elenco del campo è un elenco separato da virgola di specifiche di campo, *fieldname {, fieldname}*. Una specifica di un campo può essere costituita da un nome di campo, condizione in cui lo stesso nome viene utilizzato per il nome della colonna del database e per il nome di campo. Oppure una specifica di campo può essere costituita da un alias di campo, nel cui caso a un'espressione di database o a un nome di colonna viene assegnato un nome di campo di Qlik Sense.

I nomi di campo possono essere semplici oppure racchiusi tra virgolette. Un nome semplice inizia con un carattere alfabetico Unicode ed è seguito da una qualsiasi combinazione di caratteri alfanumerici o caratteri di sottolineatura. I nomi racchiusi tra virgolette iniziano con virgolette doppie e contengono qualsiasi sequenza di caratteri. Se un nome racchiuso tra virgolette contiene virgolette doppie, tali virgolette vengono rappresentate utilizzando due virgolette doppie adiacenti.

---

## 3 Istruzioni e parole chiave dello script

I nomi di campo di Qlik Sense rispettano la distinzione tra maiuscole e minuscole. La distinzione tra maiuscole e minuscole per i nomi di campo del database varia a seconda del database. Una query Direct Discovery mantiene la distinzione utilizzata in tutti gli identificatori e alias del campo. Nell'esempio seguente l'alias "MyState" viene utilizzato internamente per memorizzare i dati dalla colonna del database "STATEID".

```
DIRECT QUERY Dimension STATEID as MyState Measure AMOUNT from SALES_TABLE;
```

Il risultato sarà diverso rispetto a quello di un'istruzione **SQL Select** con un alias. Se l'alias non viene racchiuso tra virgolette in modo esplicito, il risultato presenterà la distinzione tra maiuscole e minuscole predefinita della colonna restituita dal database di destinazione. Nell'esempio seguente l'istruzione **SQL Select** in un database Oracle farà in modo che "MYSTATE," presenti solo lettere maiuscole, come l'alias interno di Qlik Sense, anche se per l'alias viene specificato il formato misto. L'istruzione **SQL Select** utilizza il nome della colonna restituito dal database, che nel caso di Oracle è tutto in formato maiuscolo.

```
SQL Select STATEID as MyState, STATENAME from STATE_TABLE;
```

Per evitare questo comportamento, utilizzare l'istruzione **LOAD** per specificare l'alias.

```
Load STATEID as MyState, STATENAME;  
SQL Select STATEID, STATEMENT from STATE_TABLE;
```

In questo esempio la colonna "STATEID" viene memorizzata internamente da Qlik Sense come "MyState".

La maggior parte delle espressioni scalari del database è consentita come specifiche di campo. Nelle specifiche di campo è possibile utilizzare anche le chiamate di funzione. Le espressioni possono contenere costanti booleane, numeriche o stringhe contenute tra virgolette singole (le virgolette singole incorporate sono rappresentate da virgolette singole adiacenti).

### Esempi:

```
DIRECT QUERY  
  
    DIMENSION  
  
        SalesOrderID, RevisionNumber  
  
    MEASURE  
  
        SubTotal AS "Sub Total"  
  
FROM Adventureworks.Sales.SalesOrderHeader;  
  
DIRECT QUERY  
  
    DIMENSION  
  
        "SalesOrderID" AS "Sales Order ID"  
  
    MEASURE  
  
        SubTotal,TaxAmt,(SubTotal-TaxAmt) AS "Net Total"
```

### 3 Istruzioni e parole chiave dello script

---

```
FROM Adventureworks.Sales.SalesOrderHeader;
```

```
DIRECT QUERY
```

```
DIMENSION
```

```
(2*Radius*3.14159) AS Circumference,
```

```
Molecules/6.02e23 AS Moles
```

```
MEASURE
```

```
Num1 AS numA
```

```
FROM TableName;
```

```
DIRECT QUERY
```

```
DIMENSION
```

```
concat(region, 'code') AS region_code
```

```
MEASURE
```

```
Num1 AS NumA
```

```
FROM TableName;
```

Direct Discovery non supporta l'utilizzo di aggregazioni nelle istruzioni **LOAD**. Se vengono utilizzate le aggregazioni, i risultati potrebbero essere imprevedibili. Si consiglia di non utilizzare un'istruzione **LOAD** come quella seguente:

```
DIRECT QUERY DIMENSION stateid, SUM(amount*7) AS MultiFirst MEASURE amount FROM sales_table;
```

La funzione **SUM** non deve essere contenuta nell'istruzione **LOAD**.

Anche Direct Discovery non supporta le funzioni di Qlik Sense nelle istruzioni **Direct Query**. Ad esempio, la specifica seguente per il campo **DIMENSION** restituisce un errore quando il campo "Mth" viene utilizzato come dimensione in una visualizzazione:

```
month(ModifiedDate) as Mth
```

## Directory

L'istruzione **Directory** definisce in quale directory ricercare i file dei dati nelle istruzioni **LOAD** successive finché non viene eseguita una nuova istruzione **Directory**.

**Sintassi:**

```
Directory[path]
```

Se l'istruzione **Directory** viene emessa senza **path** o non viene inclusa, Qlik Sense eseguirà la ricerca nella directory di lavoro di Qlik Sense.

### Argomenti:

| Argomenti   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Argomento   | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>path</b> | <p>Un testo può essere interpretato come il percorso del file data.</p> <p>Il percorso corrisponde al percorso del file, che può essere:</p> <ul style="list-style-type: none"><li>• assoluto</li></ul> <p><b>Esempio:</b> <i>c:\data\</i></p> <ul style="list-style-type: none"><li>• relativo alla directory di lavoro dell'app Qlik Sense.</li></ul> <p><b>Esempio:</b> <i>data\</i></p> <ul style="list-style-type: none"><li>• indirizzo dell'URL (HTTP o FTP), che punta a una posizione in Internet o su una Intranet</li></ul> <p><b>Esempio:</b> <i>http://www.qlik.com</i></p> |

### Esempi:

```
DIRECTORY C:\userfiles\data; // OR -> DIRECTORY data\
```

```
LOAD * FROM  
[data1.csv] // ONLY THE FILE NAME CAN BE SPECIFIED HERE (WITHOUT THE FULL PATH)  
(ansi, txt, delimiter is ',', embedded labels);
```

```
LOAD * FROM  
[data2.txt] // ONLY THE FILE NAME CAN BE SPECIFIED HERE UNTIL A NEW DIRECTORY STATEMENT IS  
MADE  
(ansi, txt, delimiter is '\t', embedded labels);
```

## Disconnect

L'istruzione **Disconnect** termina l'attuale connessione ODBC/OLE DB/Personalizzata. Questa istruzione è opzionale.

### Sintassi:

**Disconnect**

La connessione viene terminata automaticamente quando viene eseguita una nuova istruzione **connect** o quando verrà completata l'esecuzione dello script.

### Esempio:

```
Disconnect;
```

## Drop

La parola chiave dello script **Drop** può essere utilizzata per eliminare tabelle o campi dal database.

### Drop field

Durante l'esecuzione dello script, in qualsiasi momento è possibile rilasciare dal modello dati e quindi dalla memoria uno o più campi di Qlik Sense utilizzando l'istruzione **drop field**. La proprietà "distinct" di una tabella viene rimossa dopo un'istruzione **drop field**.



*Sia **drop field** che **drop fields** sono formati consentiti, senza alcuna differenza effettiva. Se nessuna tabella viene specificata, il campo verrà rilasciato da tutte le tabelle in cui ricorre.*

### Sintassi:

```
Drop field fieldname { , fieldname2 ...} [from tablename1 { , tablename2 ...}]  
Drop fields fieldname { , fieldname2 ...} [from tablename1 { , tablename2 ...}]
```

### Esempi:

```
Drop field A;  
Drop fields A,B;  
Drop field A from X;  
Drop fields A,B from X,Y;
```

## Drop table

Durante l'esecuzione dello script, è possibile rilasciare dal modello dati, e quindi dalla memoria, una o più tabelle interne di Qlik Sense utilizzando l'istruzione **drop table**.

### Sintassi:

```
drop table tablename {, tablename2 ...}
```

```
drop tables tablename {, tablename2 ...}
```



*I formati **drop table** e **drop tables** sono entrambi accettati.*

Di conseguenza i seguenti elementi verranno persi:

- La/e tabella/e attuale/i.
- Tutti i campi che non fanno parte delle tabelle rimanenti.

### 3 Istruzioni e parole chiave dello script

- I valori dei rimanenti campi, provenienti esclusivamente dalla tabella(e) scaricata.

Esempi e risultati:

| Esempio                                                                                                                                                                                                                                                                                                                                                                    | Risultato                                                                       |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| <code>drop table Orders, Salesmen, T456a;</code>                                                                                                                                                                                                                                                                                                                           | Questa riga causa la rimozione dalla memoria di tre tabelle.                    |
| <code>Tab1:</code><br><br><code>Load * Inline [</code><br><br><code>Customer, Items, UnitPrice</code><br><br><code>Bob, 5, 1.50</code><br><br><code>];</code><br><br><code>Tab2:</code><br><br><code>LOAD Customer, Sum( Items * UnitPrice ) as Sales</code><br><br><code>resident Tab1</code><br><br><code>group by Customer;</code><br><br><code>drop table Tab1;</code> | Una volta creata la tabella <i>Tab2</i> , la tabella <i>Tab1</i> viene rimossa. |

## Drop table

Durante l'esecuzione dello script, è possibile rilasciare dal modello dati, e quindi dalla memoria, una o più tabelle interne di Qlik Sense utilizzando l'istruzione **drop table**.

Sintassi:

```
drop table tablename {, tablename2 ...}
```

```
drop tables tablename {, tablename2 ...}
```



*I formati **drop table** e **drop tables** sono entrambi accettati.*

Di conseguenza i seguenti elementi verranno persi:

- La/e tabella/e attuale/i.
- Tutti i campi che non fanno parte delle tabelle rimanenti.

### 3 Istruzioni e parole chiave dello script

- I valori dei rimanenti campi, provenienti esclusivamente dalla tabella(e) scaricata.

Esempi e risultati:

| Esempio                                                                                                                                                                                                                                                                                                                                                                    | Risultato                                                                       |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| <code>drop table Orders, Salesmen, T456a;</code>                                                                                                                                                                                                                                                                                                                           | Questa riga causa la rimozione dalla memoria di tre tabelle.                    |
| <code>Tab1:</code><br><br><code>Load * Inline [</code><br><br><code>Customer, Items, UnitPrice</code><br><br><code>Bob, 5, 1.50</code><br><br><code>];</code><br><br><code>Tab2:</code><br><br><code>LOAD Customer, Sum( Items * UnitPrice ) as Sales</code><br><br><code>resident Tab1</code><br><br><code>group by Customer;</code><br><br><code>drop table Tab1;</code> | Una volta creata la tabella <i>Tab2</i> , la tabella <i>Tab1</i> viene rimossa. |

## Execute

L'istruzione **Execute** viene utilizzata per eseguire altri programmi, mentre Qlik Sense sta caricando i dati. Ad esempio, per effettuare le connessioni necessarie.



*Questa funzionalità non è disponibile in Qlik Sense SaaS.*



*Questa istruzione non è supportata in modalità standard.*

Sintassi:

```
execute commandline
```

### Argomenti:

#### Argomenti

| Argomento          | Descrizione                                                                                                                                                                          |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>commandline</i> | Un testo che può essere interpretato dal sistema operativo come riga di comando. È possibile fare riferimento a un percorso del file assoluto o a un percorso della cartella lib://. |

Se si desidera utilizzare **Execute**, devono essere soddisfatte le condizioni seguenti:

- L'esecuzione deve essere in modalità legacy (applicabile per Qlik Sense e Qlik Sense Desktop).
- `OverrideScriptSecurity` deve essere impostato su 1 in *Settings.ini* (applicabile per Qlik Sense). *Settings.ini* si trova nel percorso `C:\ProgramData\Qlik\Sense\Engine\` ed è generalmente un file vuoto.



*Se si imposta `OverrideScriptSecurity` per abilitare **Execute**, qualsiasi utente potrà eseguire i file sul server. Ad esempio, un utente può allegare un file eseguibile a un'app e quindi eseguire il file nello script di caricamento dei dati.*

### Procedere come indicato di seguito:

1. Eseguire una copia di *Settings.ini* e aprirla in un editor di testo.
2. Verificare che il file includa `[Settings 7]` nella prima riga.
3. Inserire una nuova riga e digitare `OverrideScriptSecurity=1`.
4. Inserire una riga vuota alla fine del file.
5. Salvare il file.
6. Sostituire *Settings.ini* con il file modificato.
7. Riavviare Qlik Sense Engine Service (QES).



*Se Qlik Sense viene eseguito come servizio, alcuni comandi potrebbero funzionare in modo imprevisto.*

### Esempio:

```
Execute C:\Program Files\Office12\Excel.exe;  
Execute lib://win\notepad.exe // win is a folder connection referring to c:\windows
```

## Field/Fields

Le parole chiave dello script **Field** e **Fields** vengono utilizzate nelle istruzioni **Declare**, **Derive**, **Drop**, **Comment**, **Rename** e **Tag/Untag**.

### FlushLog

L'istruzione **FlushLog** obbliga Qlik Sense a scrivere il contenuto del buffer dello script nel file di registro dello script.

#### Sintassi:

```
FlushLog
```

Il contenuto del buffer viene inserito nel file di registro. Questo comando può risultare utile ai fini di debug, in quanto si riceveranno i dati che altrimenti andrebbero persi in un'esecuzione dello script non riuscita.

#### Esempio:

```
FlushLog;
```

### Force

L'istruzione **force** impone a Qlik Sense di interpretare i nomi e i valori di campo delle istruzioni **LOAD** e **SELECT** successive in formato solo maiuscolo, solo minuscolo, sempre maiuscolo o conformemente alla visualizzazione attuale (formato misto). Questa istruzione permette di associare i valori di campo provenienti da tabelle create in base a convenzioni differenti.

L'istruzione **force** può anche modificare i nomi del campo durante un caricamento o selezionarli utilizzando le seguenti sorgenti dati:

- QVD
- CSV (file di testo)
- XLS
- QVX (file e connessioni ODBC)

L'istruzione **force** modifica i nomi del campo solo se i dati vengono caricati in modalità compatta (caricati con \*).

I nomi del campo delle seguenti sorgenti dati non vengono interessati dall'istruzione **force**:

- JSON
- Parquet
- XML
- XLSX

#### Sintassi:

```
Force ( capitalization | case upper | case lower | case mixed )
```

Se non si specifica alcun valore, viene applicato il formato misto. L'istruzione **force** rimane valida finché non si specifica una nuova istruzione **force**.

### 3 Istruzioni e parole chiave dello script

L'istruzione **force** non ha alcun effetto sulla sezione relativa al controllo degli accessi: per tutti i valori di campo caricati la distinzione tra maiuscole e minuscole non viene rispettata.

#### Esempi e risultati

| Esempio                                                                                                                                                                                                              | Risultato                                                                                                                                                            |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Questo esempio mostra come imporre l'utilizzo delle maiuscole.</p> <pre>FORCE Capitalization;<br/><br/>Capitalization:<br/><br/>LOAD * Inline [<br/><br/>ab<br/><br/>cd<br/><br/>eF<br/><br/>GH<br/><br/>];</pre> | <p>La tabella <b>Capitalization</b> contiene i seguenti valori:</p> <p>Ab</p> <p>Cd</p> <p>Ef</p> <p>Gh</p> <p>Tutti i valori sono scritti in lettere maiuscole.</p> |
| <p>Questo esempio mostra come imporre l'utilizzo dei caratteri maiuscoli.</p> <pre>FORCE Case Upper;<br/><br/>CaseUpper:<br/><br/>LOAD * Inline [<br/><br/>ab<br/><br/>cd<br/><br/>eF<br/><br/>GH<br/><br/>];</pre>  | <p>La tabella <b>CaseUpper</b> contiene i seguenti valori:</p> <p>AB</p> <p>CD</p> <p>EF</p> <p>GH</p> <p>Tutti i valori sono scritti con caratteri maiuscoli.</p>   |

### 3 Istruzioni e parole chiave dello script

---

| Esempio                                                                                                                                                                                                             | Risultato                                                                                                                                                                       |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Questo esempio mostra come imporre l'utilizzo dei caratteri minuscoli.</p> <pre>FORCE Case Lower;<br/><br/>CaseLower:<br/><br/>LOAD * Inline [<br/><br/>ab<br/><br/>cd<br/><br/>eF<br/><br/>GH<br/><br/>];</pre> | <p>La tabella <b>CaseLower</b> contiene i seguenti valori:</p> <p>ab</p> <p>cd</p> <p>ef</p> <p>gh</p> <p>Tutti i valori sono scritti con caratteri minuscoli.</p>              |
| <p>Questo esempio mostra come imporre l'utilizzo del formato misto.</p> <pre>FORCE Case Mixed;<br/><br/>CaseMixed:<br/><br/>LOAD * Inline [<br/><br/>ab<br/><br/>cd<br/><br/>eF<br/><br/>GH<br/><br/>];</pre>       | <p>La tabella <b>CaseMixed</b> contiene i seguenti valori:</p> <p>ab</p> <p>cd</p> <p>eF</p> <p>GH</p> <p>Tutti i valori sono scritti in base alla visualizzazione attuale.</p> |

---

**Vedere anche:**

## From

La parola chiave dello script **From** viene utilizzata nelle istruzioni **Load** per fare riferimento a un file e nelle istruzioni **Select** per fare riferimento a una tabella o a una vista del database.

### Load

L'istruzione **LOAD** carica i campi da un file, dai dati definiti nello script, da una tabella caricata in precedenza, da una pagina Web, dal risultato di un'istruzione **SELECT** seguente o dalla generazione automatica di dati. È anche possibile caricare dati da connessioni di analisi.

#### Sintassi:

```
LOAD [ distinct ] fieldlist
```

```
[ ( from file [ format-spec ] |
```

```
from_field fieldsource [format-spec] |
```

```
inline data [ format-spec ] |
```

```
resident table-label |
```

```
autogenerate size ) | extension pluginname.functionname([script]  
tabledescription) ]
```

```
[ where criterion | while criterion ]
```

```
[ group by groupbyfieldlist ]
```

```
[ order by orderbyfieldlist ]
```

#### Argomenti

| Argomento | Descrizione                                                                                                                                                                                                                                                                                                                                                                         |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| distinct  | <p>È possibile utilizzare <b>distinct</b> come predicato se si desidera caricare solo record univoci. Se ci sono record duplicati, verrà caricata la prima istanza.</p> <p>Se si utilizzano caricamenti precedenti, è necessario inserire <b>distinct</b> nella prima istruzione di caricamento, dal momento che <b>distinct</b> ha effetto solo sulla tabella di destinazione.</p> |

| Argomento | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| fieldlist | <p><i>fieldlist</i> ::= ( *   <i>field</i>{, *   <i>field</i> } )</p> <p>Un elenco di campi da caricare. L'utilizzo del carattere * come elenco dei campi indica tutti i campi della tabella.</p> <p><i>field</i> ::= ( <i>fieldref</i>   <i>expression</i> ) [<b>as</b> <i>aliasname</i> ]</p> <p>La definizione di campo deve sempre contenere un valore letterale, un riferimento a un campo esistente o un'espressione.</p> <p><i>fieldref</i> ::= ( <i>fieldname</i>  @<i>fieldnumber</i>  @<i>startpos:endpos</i> [ <i>I</i>   <b>U</b>   <b>R</b>   <b>B</b>   <i>T</i> ] )</p> <p><i>fieldname</i> è un testo che è identico al nome di un campo nella tabella. Tenere presente che il nome di campo deve essere racchiuso da virgolette doppie o parentesi quadre se, ad esempio, contiene spazi. Talvolta i nomi dei campi non sono disponibili in modo esplicito. Verrà quindi utilizzata una notazione differente:</p> <p>@<i>fieldnumber</i> rappresenta il numero di campo di un file tabellare delimitato. Deve essere un intero positivo preceduto da "@". La numerazione viene sempre eseguita partendo da 1 fino al numero di campi presenti.</p> <p>@<i>startpos:endpos</i> rappresenta le posizioni di inizio e di fine di un campo in un file con record a lunghezza fissa. Le posizioni devono essere entrambe numeri interi positivi. I due valori numerici devono essere preceduti dal simbolo "@" e separati da due punti. La numerazione viene sempre eseguita partendo da 1 fino al numero di posizioni presenti. Nell'ultimo campo <i>n</i> viene utilizzato come posizione finale.</p> <ul style="list-style-type: none"> <li>Se @<i>startpos:endpos</i> è seguito immediatamente dal carattere <b>I</b> o <b>U</b>, i byte letti verranno interpretati come un valore intero binario con segno (<b>I</b>) o senza segno (<b>U</b>), in base all'ordine dei byte Intel. Il numero di posizioni lette deve essere 1, 2 o 4.</li> <li>Se @<i>startpos:endpos</i> è immediatamente seguito dal carattere <b>R</b>, la lettura dei byte verrà interpretata come un numero binario reale (a virgola mobile a 32 bit o a 64 bit conforme allo standard IEEE). Il numero di posizioni lette deve essere 4 o 8.</li> <li>Se @<i>startpos:endpos</i> è immediatamente seguito dal carattere <b>B</b>, la lettura dei byte sarà interpretata come numeri BCD (Binary Coded Decimal), in base allo standard COMP-3. È possibile essere specificato un numero di byte qualsiasi.</li> </ul> <p><i>expression</i> può essere una funzione numerica o una funzione stringa basata su uno o molti altri campi della stessa tabella. Per ulteriori informazioni, vedere la sintassi delle espressioni.</p> <p><b>as</b> viene utilizzato per assegnare un nuovo nome al campo.</p> |

| Argomento | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| from      | <p><b>from</b> viene utilizzato se i dati devono essere caricati da un file mediante una connessione dati di una cartella o di un file Web</p> <p><i>file ::= [ path ] filename</i></p> <p><b>Esempio: 'lib://Table Files/'</b></p> <p>Se tale percorso viene omissso, Qlik Sense ricerca il file nella directory specificata dall'istruzione <b>Directory</b>. Se non è presente alcuna istruzione <b>Directory</b>, Qlik Sense eseguirà la ricerca nella directory di lavoro, <i>C:\Users\{user}\Documents\Qlik\Sense\Apps</i>.</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p> In un'installazione server di Qlik Sense, la directory di lavoro è specificata in Qlik Sense Repository Service e, per impostazione predefinita, è <i>C:\ProgramData\Qlik\Sense\Apps</i>.</p> </div> <p><i>filename</i> può contenere caratteri speciali nello standard DOS ( * e ? ). Tutti i file corrispondenti verranno caricati nella directory specificata.</p> <p><i>format-spec ::= ( fspec-item { , fspec-item } )</i></p> <p>La specifica del formato è costituita da un elenco di più voci di specifica del formato, racchiuse tra parentesi.</p> <p><b>Modalità di script legacy</b></p> <p>Nella modalità di creazione degli script legacy sono supportati anche i seguenti formati di percorso:</p> <ul style="list-style-type: none"> <li>• assoluto</li> </ul> <p><b>Esempio: <i>c:\data</i></b></p> <ul style="list-style-type: none"> <li>• relativo alla directory di lavoro dell'app Qlik Sense.</li> </ul> <p><b>Esempio: <i>data</i></b></p> <ul style="list-style-type: none"> <li>• indirizzo dell'URL (HTTP o FTP), che punta a una posizione in Internet o su una Intranet</li> </ul> <p><b>Esempio: <i>http://www.qlik.com</i></b></p> |

### 3 Istruzioni e parole chiave dello script

| Argomento  | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| from_field | <p><b>from_field</b> viene utilizzato nel caso in cui i dati devono essere caricati da un campo caricato in precedenza.</p> <p><i>fieldsource::=(tablename, fieldname)</i></p> <p>Il campo è il nome del valore di <i>tablename</i> e <i>fieldname</i> caricato in precedenza.</p> <p><i>format-spec ::= ( fspec-item {, fspec-item } )</i></p> <p>La specifica del formato è costituita da un elenco di più voci di specifica del formato, racchiuse tra parentesi. Per ulteriori informazioni, vedere <a href="#">Voci per la specifica del formato (page 174)</a>.</p> <div> <b>from_field</b> supporta solo le virgole come delimitatore di elenco quando si separano campi nelle tabelle.</div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| inline     | <p><b>inline</b> viene utilizzato quando i dati devono essere immessi direttamente nello script e non caricati da un file.</p> <p><i>data ::= [ text ]</i></p> <p>I dati immessi mediante una clausola <b>inline</b> devono essere racchiusi da caratteri specifici, come parentesi quadre, virgolette doppie o apici. Il testo tra parentesi o virgolette viene interpretato allo stesso modo del contenuto di un file. Pertanto, quando si desidera immettere una nuova riga in un file di testo, è necessario eseguire la stessa operazione anche nel testo di una clausola <b>inline</b>: premendo il tasto INVIO mentre si digita lo script.</p> <p>In un semplice caricamento inline, il numero di colonne è definito nella prima riga.</p> <p><i>format-spec ::= ( fspec-item {, fspec-item } )</i></p> <p>È possibile personalizzare il caricamento inline con molti degli stessi elementi di specifica di formato che sono disponibili per altre tabelle caricate. Tali elementi sono elencati tra parentesi. Per ulteriori informazioni, vedere <a href="#">Voci per la specifica del formato (page 174)</a>.</p> <p>Per ulteriori informazioni sull'utilizzo di caricamenti inline, vedere <a href="#">Utilizzo dei caricamenti inline per caricare i dati</a>.</p> |
| resident   | <p><b>resident</b> viene usato se i dati devono essere caricati da una tabella caricata in precedenza.</p> <p><i>table label</i> è un'etichetta che precede le istruzioni <b>LOAD</b> o <b>SELECT</b> che hanno creato la tabella originale. L'etichetta dovrà essere indicata con i due punti finali.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

### 3 Istruzioni e parole chiave dello script

---

| Argomento    | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| autogenerate | <p><b>autogenerate</b> viene utilizzato se i dati devono essere automaticamente generati da Qlik Sense.</p> <p><i>size ::= number</i></p> <p><i>Number</i> è un numero intero e indica il numero di record che da generare.</p> <p>L'elenco di campi non deve contenere espressioni che richiedono dati provenienti da una sorgente dati esterna o da una tabella caricata in precedenza, a meno che non si faccia riferimento a un singolo valore di campo in una tabella caricata in precedenza con la funzione <b>Peek</b>.</p> |

| Argomento | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| extension | <p>È possibile caricare dati da connessioni di analisi. È necessario utilizzare la clausola <b>extension</b> per chiamare una funzione definita nel plug-in SSE (Server-Side Extension) o per valutare uno script.</p> <p>È possibile inviare una singola tabella al plug-in SSE, che restituisce una singola tabella di dati. Se il plug-in non specifica i nomi dei campi restituiti, i campi saranno denominati Field1, Field2 e così via.</p> <pre>Extension pluginname.functionname( tabledescription );</pre> <ul style="list-style-type: none"> <li>Caricamento di dati mediante una funzione in un plug-in SSE<br/> <i>tabledescription ::= (table { ,tablefield} )</i><br/>           Se i campi della tabella non vengono specificati, verranno utilizzati in ordine di caricamento.</li> <li>Caricamento di dati mediante valutazione di uno script in un plug-in SSE<br/> <i>tabledescription ::= ( script, table { ,tablefield} )</i></li> </ul> <p><b>Gestione dei tipi di dati nella definizione dei campi della tabella</b></p> <p>I tipi di dati sono riconosciuti automaticamente nelle connessioni di analisi. Se i dati non hanno valori numerici e hanno almeno una stringa di testo non NULL, il campo è considerato di tipo testo. In tutti gli altri casi è considerato di tipo numerico.</p> <p>È possibile forzare il tipo di dati racchiudendo il nome del campo in <b>String()</b> o <b>Mixed()</b>.</p> <ul style="list-style-type: none"> <li><b>String()</b> forza il tipo del campo su testo. Se il campo è numerico, viene estratta la parte di testo del valore duale, senza eseguire alcuna conversione.</li> <li><b>Mixed()</b> forza il tipo del campo su duale.</li> </ul> <p><b>String()</b> o <b>Mixed()</b> non possono essere utilizzati esternamente alle definizioni dei campi della tabella <b>extension</b> e non è possibile utilizzare altre funzioni di Qlik Sense nella definizione di un campo della tabella.</p> <p><b>Ulteriori informazioni sulle connessioni di analisi</b></p> <p>Prima di poter utilizzare le connessioni di analisi, è necessario configurarle.</p> |
| where     | <p><b>where</b> è una clausola utilizzata per dichiarare se un record deve essere incluso o meno nella selezione. La selezione viene inclusa se <i>criterion</i> è True. <i>criterion</i> è un'espressione logica.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| while     | <p><b>while</b> è una clausola che specifica se un record deve essere letto ripetutamente. Viene letto lo stesso record finché <i>criterion</i> è True. Per risultare utile, una clausola <b>while</b> deve generalmente includere la funzione <b>IterNo( )</b>.</p> <p><i>criterion</i> è un'espressione logica.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

### 3 Istruzioni e parole chiave dello script

| Argomento | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| group by  | <p><b>group by</b> è una clausola usata per definire su quali campi devono essere aggregati (raggruppati) i dati. I campi di aggregazione devono essere inclusi in qualche modo nelle espressioni caricate. Nessun altro campo tranne i campi di aggregazione può essere usato al di fuori delle funzioni di aggregazione nelle espressioni caricate.</p> <p><i>groupbyfieldlist ::= (fieldname { ,fieldname } )</i></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| order by  | <p><b>order by</b> è una clausola utilizzata per ordinare i record di una tabella residente prima che vengano elaborati da un'istruzione <b>load</b>. La tabella residente può essere ordinata su uno o più campi, in modo crescente o decrescente. L'ordinamento viene eseguito innanzitutto in base ai valori numerici e secondariamente in base all'ordine di confronto nazionale. Questa clausola può essere utilizzata solamente quando la sorgente dati è una tabella residente. I campi di ordinamento specificano il campo in base al quale viene ordinata la tabella residente. Il campo può essere specificato da un nome o dal suo numero nella tabella residente (il primo campo è il numero 1).</p> <p><i>orderbyfieldlist ::= fieldname [ sortorder ] { , fieldname [ sortorder ] }</i></p> <p><i>sortorder</i> è <i>asc</i> per crescente o <i>desc</i> per decrescente. Se <i>sortorder</i> non è specificato, viene utilizzato <i>asc</i>.</p> <p><i>fieldname</i>, <i>path</i>, <i>filename</i> e <i>aliasname</i> sono stringhe di testo che rappresentano ciò che implica il loro rispettivo nome. Qualsiasi campo presente nella tabella sorgente può essere utilizzato come <i>fieldname</i>. Tuttavia, i campi creati mediante la clausola <i>as</i> (<i>aliasname</i>) non appartengono all'ambito e non possono essere utilizzati all'interno della stessa istruzione <b>load</b>.</p> |

Se non viene specificata alcuna sorgente dati mediante una clausola **from**, **inline**, **resident**, **from\_field**, **extension** o **autogenerate**, i dati verranno caricati dal risultato dell'istruzione **SELECT** o **LOAD** immediatamente successiva. L'istruzione successiva non dovrà avere un prefisso.

#### Esempi:

##### Caricamento di formati di file differenti

Caricare un file di dati delimitato con le opzioni predefinite:

```
LOAD * from data1.csv;
```

Caricare un file di dati delimitato da una connessione della libreria (DataFiles):

```
LOAD * from 'lib://DataFiles/data1.csv';
```

Caricare tutti i file di dati delimitati da una connessione della libreria (DataFiles):

```
LOAD * from 'lib://DataFiles/*.csv';
```

---

## 3 Istruzioni e parole chiave dello script

Caricare un file delimitato specificando la virgola come delimitatore e con etichette incorporate:

```
LOAD * from 'c:\userfiles\data1.csv' (ansi, txt, delimiter is ',', embedded labels);
```

Caricare un file delimitato specificando la tabulazione come delimitatore e con etichette incorporate:

```
LOAD * from 'c:\userfiles\data2.txt' (ansi, txt, delimiter is '\t', embedded labels);
```

Caricare un file dif con intestazioni incorporate:

```
LOAD * from file2.dif (ansi, dif, embedded labels);
```

Caricare tre campi da un file Record Fixed senza intestazioni:

```
LOAD @1:2 as ID, @3:25 as Name, @57:80 as City from data4.fix (ansi, fix, no labels, header is 0, record is 80);
```

Caricare un file QVX specificando un percorso assoluto:

```
LOAD * from C:\qdssamples\xyz.qvx (qvx);
```

### Caricamento di file Web

Caricare dall'URL predefinito impostato nella connessione dati del file Web:

```
LOAD * from [lib://MyWebFile];
```

Caricare da un URL specifico e sostituire l'URL impostato nella connessione dati del file Web:

```
LOAD * from [lib://MyWebFile] (URL is 'http://localhost:8000/foo.bar');
```

Caricare da un URL specifico impostato in una variabile mediante espansione con simbolo del dollaro:

```
SET dynamicURL = 'http://localhost/foo.bar';
```

```
LOAD * from [lib://MyWebFile] (URL is '$(dynamicURL)');
```

### Selezione di alcuni campi, ridenominazione e calcolo dei campi

Caricare solo tre campi specifici da un file delimitato:

```
LOAD FirstName, LastName, Number from data1.csv;
```

Assegnare il nome A al primo campo e il nome B al secondo campo durante il caricamento di un file senza etichette:

```
LOAD @1 as A, @2 as B from data3.txt (ansi, txt, delimiter is '\t', no labels);
```

Caricare Name come concatenazione di FirstName, uno spazio e LastName:

```
LOAD FirstName & ' ' & LastName as Name from data1.csv;
```

Caricare Quantity, Price e Value (il prodotto di Quantity e Price):

```
LOAD Quantity, Price, Quantity*Price as value from data1.csv;
```

## 3 Istruzioni e parole chiave dello script

---

### Selezione di alcuni record

Caricare solo i record univoci mentre i record duplicati verranno eliminati:

```
LOAD distinct FirstName, LastName, Number from data1.csv;
```

Caricare solo i record in cui il campo Litres presenta un valore superiore a zero:

```
LOAD * from Consumption.csv where Litres>0;
```

### Caricamento di dati non presenti nel file e di dati generati automaticamente

Caricare una tabella con dati inline, due campi denominati CatID e Category:

```
LOAD * Inline
```

```
[CatID, Category
```

```
0,Regular
```

```
1,Occasional
```

```
2,Permanent];
```

Caricare una tabella con dati inline, tre campi denominati UserID, Password e Access:

```
LOAD * Inline [UserID, Password, Access
```

```
A, ABC456, User
```

```
B, VIP789, Admin];
```

Caricare una tabella con 10.000 righe. Il campo A conterrà il numero del record di lettura (1,2,3,4,5...), mentre il campo B conterrà un numero casuale compreso tra 0 e 1:

```
LOAD RecNo( ) as A, rand( ) as B autogenerate(10000);
```



*La parentesi dopo autogenerate è consentita, ma non obbligatoria.*

### Caricamento di dati da una tabella caricata in precedenza

Come prima cosa carichiamo un file tabella delimitato assegnandogli il nome tab1:

```
tab1:
```

```
SELECT A,B,C,D from 'lib://DataFiles/data1.csv';
```

Caricare i campi dalla tabella tab1 già caricata come tab2:

```
tab2:
```

```
LOAD A,B,month(C),A*B+D as E resident tab1;
```

---

## 3 Istruzioni e parole chiave dello script

Caricare i campi dalla tabella già caricata tab1, ma solo record in cui A è maggiore di B:

tab3:

```
LOAD A,A+B+C resident tab1 where A>B;
```

Caricare i campi dalla tabella già caricata tab1 ordinata in base a A:

```
LOAD A,B*C as E resident tab1 order by A;
```

Caricare campi dalla tabella tab1 già caricata, ordinata in base al primo campo, quindi in base al secondo campo:

```
LOAD A,B*C as E resident tab1 order by 1,2;
```

Caricare i campi dalla tabella tab1 già caricata, ordinata in ordine decrescente in base a C, quindi in ordine crescente in base a B e quindi in base al primo campo in ordine decrescente:

```
LOAD A,B*C as E resident tab1 order by C desc, B asc, 1 desc;
```

**Caricamento dei dati da campi caricati in precedenza**

Caricare il campo Types da tabelle caricate in precedenza Characters come A:

```
LOAD A from_field (Characters, Types);
```

**Caricamento dei dati da una tabella successiva (anteriore al caricamento)**

Caricare A, B e i campi calcolati X e Y dalla tabella Table1, caricata nell'istruzione **SELECT** successiva:

```
LOAD A, B, if(C>0,'positive','negative') as X, weekday(D) as Y;
```

```
SELECT A,B,C,D from Table1;
```

**Raggruppamento di dati**

Caricare i campi raggruppati (aggregati) in base a ArtNo:

```
LOAD ArtNo, round(Sum(TransAmount),0.05) as ArtNoTotal from table.csv group by ArtNo;
```

Caricare i campi raggruppati (aggregati) in base a Week e ArtNo:

```
LOAD Week, ArtNo, round(Avg(TransAmount),0.05) as weekArtNoAverages from table.csv group by Week, ArtNo;
```

**Lettura ripetuta di un record**

In questo esempio è presente un file di input Grades.csv contenente i voti per ciascuno studente raccolti in un singolo campo:

### 3 Istruzioni e parole chiave dello script

---

Student, Grades

Mike, 5234

John, 3345

Pete, 1234

Paul, 3352

I voti, in una scala da 1 a 5, rappresentano le materie Math, English, Science e History. È possibile suddividere i voti in valori separati leggendo i record più volte con una clausola **while**, utilizzando la funzione **IterNo( )** come contatore. Durante ciascuna lettura, il voto viene estratto con la funzione **Mid** e memorizzato in Grade, mentre la materia viene selezionata utilizzando la funzione **pick** e memorizzata in Subject. La clausola finale **while** contiene il test per verificare che siano stati letti tutti i voti (in questo caso quattro per studente), il che significa che deve essere letto il record dello studente successivo.

MyTab:

LOAD Student,

mid(Grades, IterNo( ), 1) as Grade,

pick(IterNo( ), 'Math', 'English', 'Science', 'History') as Subject from Grades.csv

while IsNum(mid(Grades, IterNo(), 1));

Il risultato è una tabella contenente i seguenti dati:

| Student | Subject | Grade |
|---------|---------|-------|
| John    | English | 3     |
| John    | History | 5     |
| John    | Math    | 3     |
| John    | Science | 4     |
| Mike    | English | 2     |
| Mike    | History | 4     |
| Mike    | Math    | 5     |
| Mike    | Science | 3     |
| Paul    | English | 3     |
| Paul    | History | 2     |
| Paul    | Math    | 3     |
| Paul    | Science | 5     |
| Pete    | English | 2     |
| Pete    | History | 4     |
| Pete    | Math    | 1     |
| Pete    | Science | 3     |

Caricamento da connessioni di analisi

Sono utilizzati i dati campione seguenti.

Values:

Load

Rand() as A,

```
Rand() as B,  
Rand() as C  
AutoGenerate(50);
```

### Caricamento di dati mediante una funzione

In questi esempi si suppone di avere un plug-in di connessione di analisi denominato *P* contenente una funzione personalizzata *Calculate(Parameter1, Parameter2)*. La funzione restituisce la tabella *Results* contenente i campi *Field1* e *Field2*.

```
Load * Extension P.Calculate( values{A, C} );  
Carica tutti i campi restituiti quando si inviano i campi A e C alla funzione.
```

```
Load Field1 Extension P.Calculate( values{A, C} );  
Carica solo il campo Field1 quando si inviano i campi A e C alla funzione.
```

```
Load * Extension P.Calculate( values );  
Carica tutti i campi restituiti quando si inviano i campi A e B alla funzione. Dal momento che i campi non sono specificati, vengono utilizzati A e B, ossia i primi nell'ordine nella tabella.
```

```
Load * Extension P.Calculate( values {C, C});  
Carica tutti i campi restituiti quando si invia il campo C a entrambi i parametri della funzione.
```

```
Load * Extension P.Calculate( values {String(A), Mixed(B)});  
Carica tutti i campi restituiti quando si invia il campo A forzato come stringa e il campo B forzato come numerico alla funzione.
```

### Caricamento di dati mediante valutazione di uno script

```
Load A as A_echo, B as B_echo Extension R.ScriptEval( 'q;', values{A, B} );  
Carica la tabella restituita dallo script q quando si inviano i valori di A e B.
```

```
Load * Extension R.ScriptEval( '$(My_R_Script)', values{A, B} );  
Carica la tabella restituita dallo script memorizzata nella variabile My_R_Script quando si inviano i valori di A e B.
```

```
Load * Extension R.ScriptEval( '$(My_R_Script)', values{B as D, *} );  
Carica la tabella restituita dallo script memorizzato nella variabile My_R_Script quando si inviano i valori di B rinominato in D, A e C. L'uso dell'asterisco (*) consente di inviare i rimanenti campi senza riferimento.
```



*All'estensione file delle connessioni DataFiles si applica la distinzione tra maiuscole e minuscole. Ad esempio: .qvd.*

### Voci per la specifica del formato

Ogni voce di specifica del formato definisce una determinata proprietà del file tabellare:

### 3 Istruzioni e parole chiave dello script

**fspec-item ::= [ansi | oem | mac | UTF-8 | Unicode | txt | fix | dif | biff | ooxml | html | xml | kml | qvd | qvx | parquet | json | delimiter is char | no eof | embedded labels | explicit labels | no labels | table is [tablename] | header is n | header is line | header is n lines | comment is string | record is n | record is line | record is n lines | no quotes | msq | URL is string | userAgent is string]**

#### Set di caratteri

Il set di caratteri è un identificatore di file per l'istruzione **LOAD** che definisce il set di caratteri utilizzato nel file.

Gli identificatori **ansi**, **oem** e **mac** venivano utilizzati in QlikView e sono ancora validi. Tuttavia, non verranno generati quando si crea l'istruzione **LOAD** con Qlik Sense.

#### Sintassi:

```
utf8 | unicode | ansi | oem | mac | codepage is
```

#### Argomenti:

##### Argomenti

| Argomento          | Descrizione                                                                                                 |
|--------------------|-------------------------------------------------------------------------------------------------------------|
| <b>utf8</b>        | Set di caratteri UTF-8                                                                                      |
| <b>unicode</b>     | Set di caratteri Unicode                                                                                    |
| <b>ansi</b>        | Windows, codice pagina 1252                                                                                 |
| <b>oem</b>         | DOS, OS/2, AS400 e altri                                                                                    |
| <b>mac</b>         | Codice pagina 10000                                                                                         |
| <b>codepage is</b> | Con l'identificatore <b>codepage</b> è possibile utilizzare qualsiasi codice pagina Windows come <i>N</i> . |

#### Limiti:

La conversione dal set di caratteri **oem** non viene implementata per macOS. Se non si effettua alcuna selezione, in Windows verrà utilizzato il codice pagina 1252.

#### Esempio:

```
LOAD * from a.txt (utf8, txt, delimiter is ',' , embedded labels)
```

```
LOAD * from a.txt (unicode, txt, delimiter is ',' , embedded labels)
```

```
LOAD * from a.txt (codepage is 10000, txt, delimiter is ',' , no labels)
```

#### Vedere anche:

p [Load \(page 163\)](#)

#### Formato delle tabelle

Il formato delle tabelle è un identificatore di file per l'istruzione **LOAD** che definisce il tipo di file. Se non si specifica alcun valore, viene utilizzato un file *.txt*.

Tipi di formato tabella

| Tipo  | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| txt   | In un file di testo delimitato, le colonne nella tabella sono separate da un carattere di delimitazione.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| fix   | <p>In un file Record Fixed, ogni campo è costituito da un numero fisso di caratteri.</p> <p>Generalmente, molti file di lunghezza record fissa contengono record divisi da un carattere di separazione, tuttavia sono disponibili opzioni più avanzate per specificare le dimensioni dei record in byte o per includere più di una riga con <b>Record is</b>.</p> <div> <i>Se i dati contengono caratteri a più byte, le interruzioni di campo possono disallinearsi dato che il formato si basa su una lunghezza di byte fissa.</i></div> |
| dif   | In un file <i>.dif</i> (Data Interchange Format), viene utilizzato un formato speciale per la definizione della tabella.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| biff  | Qlik Sense è inoltre in grado di interpretare i dati nei file Excel standard mediante il formato <i>biff</i> (Binary Interchange File Format).                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| ooxml | <p>Excel 2007 e versioni successive utilizzano il formato ooxml <i>.xlsx</i>.</p> <p>L'identificatore <b>Table is</b> può essere utilizzato per definire il nome del foglio da caricare come tabella.</p> <p><a href="#">La tabella è (page 180)</a></p>                                                                                                                                                                                                                                                                                                                                                                    |
| html  | Se la tabella fa parte di una pagina o un file html, occorre utilizzare html.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| xml   | <p>xml (Extensible Markup Language) è un linguaggio di markup comune utilizzato per rappresentare le strutture dei dati in un formato testuale.</p> <p>L'identificatore <b>Table is</b> può essere utilizzato per definire il percorso del file XML da caricare come tabella.</p> <p><a href="#">La tabella è (page 180)</a></p>                                                                                                                                                                                                                                                                                            |
| qvd   | Il formato <i>qvd</i> è il formato file QVD proprietario, esportato da un'app Qlik Sense.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| qvx   | <i>qvx</i> è un formato file/flusso di dati per output a prestazioni elevate in Qlik Sense.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

| Tipo    | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| parquet | <p>Apache Parquet è un formato di archiviazione a colonne, molto efficiente per archiviare e interrogare grandi set di dati.</p> <p>Con i file Parquet contenenti dati annidati, è possibile specificare la tabella da caricare dal file Parquet utilizzando l'identificatore <b>Table is</b>. Ad esempio: <code>LOAD * FROM [lib://DataFiles/company.parquet] (parquet, table is [company:salesrep.salesrep]);</code>.</p> <p><a href="#">La tabella è (page 180)</a></p> |
| json    | <p>JSON (JavaScript Object Notation) è un formato dati indipendente dalla lingua.</p> <p>Se il file JSON contiene più matrici, è possibile utilizzare l'identificatore <b>Tabel is</b> per definire il percorso dell'iteratore per la sottotabella.</p> <p><a href="#">La tabella è (page 180)</a></p>                                                                                                                                                                     |

#### Delimiter is

Per i file tabellari delimitati, è possibile specificare un delimitatore arbitrario utilizzando l'identificatore **delimiter is**. Questo identificatore è pertinente solo per i file .txt delimitati.

#### Sintassi:

```
delimiter is char
```

#### Argomenti:

##### Argomenti

| Argomento | Descrizione                                             |
|-----------|---------------------------------------------------------|
| char      | Specifica un singolo carattere dai caratteri 127 ASCII. |

Inoltre, possono essere utilizzati i seguenti valori:

##### Valori opzionali

| Valore   | Descrizione                                                                                                                                                                      |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| '\t'     | che rappresenta un segno di tabulazione, con o senza virgolette.                                                                                                                 |
| '\''     | che rappresenta un carattere di barra rovesciata ( \ ).                                                                                                                          |
| 'spaces' | che rappresenta tutte le combinazioni di uno o più spazi. I caratteri non stampabili con codice ASCII inferiore a 32, ad eccezione di CR e LF, verranno interpretati come spazi. |

Se non viene specificato alcun valore, viene utilizzato **delimiter is ','**.

### Esempio:

```
LOAD * from a.txt (utf8, txt, delimiter is ',' , embedded labels);
```

---

### Vedere anche:

p [Load \(page 163\)](#)

### No eof

L'identificatore **no eof** serve ad ignorare il carattere di fine file quando si caricano i file **.txt** delimitati.

### Sintassi:

```
no eof
```

Se si utilizza l'identificatore **no eof**, i caratteri con punto di codice 26, che altrimenti denota la fine del file, vengono ignorati e possono fare parte di un valore di campo.

Questo identificatore risulta pertinente solo per i file di testo delimitati.

### Esempio:

```
LOAD * from a.txt (txt, utf8, embedded labels, delimiter is ' ', no eof);
```

---

### Vedere anche:

p [Load \(page 163\)](#)

### Labels

**Labels** è un identificatore di file per l'istruzione **LOAD** che consente di individuare i nomi dei campi all'interno di un file.

### Sintassi:

```
embedded labels|explicit labels|no labels
```

I nomi di campo possono essere collocati in posizioni differenti del file. Se il primo record contiene i nomi di campo, è consigliabile utilizzare **embedded labels**. Se non vi sono nomi di campo, è necessario utilizzare **no labels**. Talvolta, nei file *dif* viene utilizzata una sezione di intestazione separata con nomi di campo espliciti. In questi casi si consiglia di utilizzare **explicit labels**. Se non si è specificato alcun elemento, viene utilizzato **embedded labels** anche per i file *dif*.

### Example 1:

```
LOAD * from a.txt (unicode, txt, delimiter is ',' , embedded labels
```

### Example 2:

```
LOAD * from a.txt (codePage is 1252, txt, delimiter is ',' , no labels)
```

### Vedere anche:

p [Load \(page 163\)](#)

### Header is

Specifica la dimensione dell'intestazione nei file tabellari. È possibile specificare una lunghezza arbitraria per l'intestazione mediante l'identificatore **header is**. Un'intestazione è una sezione di testo non utilizzata da Qlik Sense.

### Sintassi:

```
header is n
```

```
header is line
```

```
header is n lines
```

La lunghezza dell'intestazione può essere espressa in byte (**header is n**) o in linee (**header is line** o **header is n lines**). **n** deve essere un numero intero, che rappresenti la lunghezza dell'intestazione. Se non specificato, viene utilizzato **header is 0**. L'identificatore **header is** è pertinente solo per i file tabellari.

### Esempio:

Questo è un esempio di tabella della sorgente dati contenente una riga di testo di intestazione che Qlik Sense non deve interpretare come dati.

```
*Header line  
col1,col2  
a,B  
c,D
```

Utilizzando l'identificatore **header is 1 lines**, la prima riga non verrà caricata come dati. Nell'esempio, l'identificatore **embedded labels** indica a Qlik Sense di interpretare la prima riga non esclusa come contenente etichette di campo.

```
LOAD col1, col2  
FROM 'lib://files/header.txt'  
(txt, embedded labels, delimiter is ',', msq, header is 1 lines);
```

Il risultato è una tabella con due campi Col1 e Col2.

---

### Vedere anche:

p [Load \(page 163\)](#)

### Record is

Per i file Record Fixed, la lunghezza del record deve essere specificata tramite l'identificatore **record is**.

### Sintassi:

```
Record is n
Record is line
Record is n lines
```

### Argomenti:

| Argomenti |                                                                                                                         |
|-----------|-------------------------------------------------------------------------------------------------------------------------|
| Argomento | Descrizione                                                                                                             |
| n         | Specifica la lunghezza del record in byte.                                                                              |
| line      | Specifica la lunghezza del record come riga singola.                                                                    |
| n lines   | Specifica la lunghezza del record in righe, dove n è un numero intero positivo che rappresenta la lunghezza del record. |

### Limiti:

L'identificatore **record is** è pertinente solo per i file **fix**.

---

### Vedere anche:

p [Load \(page 163\)](#)

### La tabella è

Per i file Excel, XML, Parquet o JSON, è possibile specificare la tabella da cui si desidera caricare i dati nell'identificatore di formato della tabella.

### Sintassi:

```
Table is table name
```

### Argomenti:

| Argomenti  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Argomento  | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| table name | <p>Specifica il nome della tabella. Il valore dipende dal formato della tabella:</p> <ul style="list-style-type: none"><li>• Excel: il nome del foglio.</li><li>• XML: il percorso che specifica le parti del file XML da caricare.</li><li>• Parquet: il percorso che specifica la tabella, con il formato <code>&lt;nodo&gt;.&lt;nodo&gt;.&lt;nodo&gt;</code>.</li></ul> <p>Utilizzare <b>Table is</b> quando si specifica una tabella con struttura annidata. Ad esempio, se si dispone di dati nel formato Parquet con il seguente schema:</p> <p>Schema:</p> <pre>Field(name: "Name", datatype: String),<br/>Field(name: "Age", datatype: Float),<br/>Field(name: "Phone", datatype: List(<br/>    Field(name: "Item", datatype: Struct(<br/>        Field(name: "Number", datatype: String)    ))</pre> <p>È possibile caricare Phone e i relativi campi annidati come una tabella con l'argomento <code>Table is [Schema:Phone.Item]</code>. Questa operazione genera il campo chiave <code>%Key_Phone</code> con la tabella.</p> <ul style="list-style-type: none"><li>• JSON: nei file JSON con matrici, il percorso dell'iteratore per la sottotabella. È possibile completare il caricamento di tutte le tabelle specificando <b>tutte le tabelle</b> anziché <b>Table is</b>. Ad esempio:</li></ul> <pre>LOAD * FROM [lib://DataFiles/Sales.json] (json, all tables);</pre> <p>Se i dati contengono più matrici con contenuto identico, utilizzare la parola chiave <b>DISTINCT</b> nelle istruzioni di caricamento nella sottotabella. Ad esempio:</p> <pre>LOAD DISTINCT * FROM [lib://DataFiles/Sales.json] (json, all tables);</pre> |

### Esempio: Excel

```
LOAD  
    "Item Number",  
    "Product Group",  
    "Product Line",  
    "Product Sub Group",  
    "Product Type"  
FROM [lib://AttachedFiles/Item master.xlsx]  
(ooxml, embedded labels, table is [Item master]);
```

### Esempio: XML

```
LOAD  
    city%Table,  
    %key_row_7FAC1F878EC01ECB  
FROM [lib://AttachedFiles/cities.xml]  
(XmlSimple, table is [root/row/country/city]);
```

### Esempio: Parquet

Il file `company.parquet` contiene il seguente schema:

```
company (String)
contact (String)
company:salesrep (List)
    salesrep (Group)
        salesrep (String)
company:headquarter (List)
    headquarter (Group)
        country (String)
        city (String)
        city:region (List)
        region (Group)
            region (String)
```

Le istruzioni riportate sotto consentono di caricare i contenuti del file nelle tabelle. La prima istruzione `LOAD` carica il gruppo `root`. La seconda istruzione `LOAD` carica i contenuti del gruppo `salesrep` come una tabella. La terza carica il gruppo `headquarter` come una tabella. Infine, la quarta carica il gruppo `region` come una tabella.

```
LOAD * FROM [...] (parquet);
LOAD * FROM [...] (parquet, table is [company:salesrep.salesrep]);
LOAD * FROM [...] (parquet, table is [company:headquarter.headquarter])
LOAD * FROM [...] (parquet, table is [company:headquarter.headquarter.city:region.region])
```

### Esempio: JSON

```
LOAD
customers.customerid,
customers.companyname,
customers.contactname,
customers.contacttitle,
customers.address,
customers.city,
customers.postalcode,
customers.country,
customers.phone,
customers.fax,
%Key_customers,
customers.region
FROM [lib://DataFiles/example.json]
(json, table is [*/customers]);
```

### Limiti:

L'identificatore **Table is** è rilevante solo per i file Excel, XML, Parquet o JSON.

### Quotes

**Quotes** è un identificatore di file per l'istruzione **LOAD** che stabilisce se è possibile utilizzare le virgolette e la precedenza tra virgolette e separatori. Solo per file di testo.

### Sintassi:

`no quotes`

`msq`

Se l'identificatore viene omissso, è possibile utilizzare le virgolette standard, ovvero le virgolette " " o ' ' ma solo se sono il primo e l'ultimo carattere non vuoto di un valore di campo.

### Argomenti:

#### Argomenti

| Argomento | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| no quotes | Viene utilizzato solo se le virgolette non sono accettate in un file di testo.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| msq       | <p>Viene utilizzato per specificare virgolette in stile moderno che consentono contenuti multiriga nei campi. I campi che contengono caratteri di fine riga devono essere racchiusi tra virgolette doppie.</p> <p>Un limite dell'opzione msq è che i singoli caratteri di virgoletta doppia (") che compaiono come primo o ultimo carattere nel contenuto di un campo vengono interpretati come segno iniziale o finale del contenuto multiriga. Ciò potrebbe causare risultati imprevisti nella serie di dati caricata. In questo caso, utilizzare le virgolette standard omettendo l'identificatore.</p> |

### XML

Questo identificatore di script viene utilizzato per il caricamento dei file xml. Le opzioni valide per l'identificatore **XML** sono elencate nella sintassi.



Non è possibile caricare file DTD in Qlik Sense.

### Sintassi:

`xmlsimple`

### Vedere anche:

p [Load \(page 163\)](#)

### KML

Questo identificatore di script viene utilizzato durante il caricamento dei file KML da utilizzare in una visualizzazione della mappa.

### Sintassi:

`kml`

### 3 Istruzioni e parole chiave dello script

---

Il file KML può rappresentare i dati di un'area (ad esempio paesi o regioni) rappresentati da poligoni, dati di linee (ad esempio binari o strade) o dati di punti (ad esempio città o luoghi) rappresentati da punti nella forma [long, lat].

#### URL is

Questo identificatore di script è utilizzato per impostare l'URL di una connessione dati a un file Web quando si carica un file Web.

#### Sintassi:

```
URL is string
```

#### Argomenti:

##### Argomenti

| Argomento | Descrizione                                                                                                              |
|-----------|--------------------------------------------------------------------------------------------------------------------------|
| string    | Specifica l'URL del file da caricare. Questo valore sostituirà l'URL impostato nella connessione al file Web utilizzata. |

#### Limiti:

L'identificatore **URL is** è rilevante solo per i file Web. È necessario utilizzare una connessione dati al file Web esistente.

---

#### Vedere anche:

p [Load \(page 163\)](#)

#### userAgent is

Questo identificatore di script è utilizzato per impostare l'agente utente del browser quando si carica un file Web.

#### Sintassi:

```
userAgent is string
```

#### Argomenti:

##### Argomenti

| Argomento | Descrizione                                                                                                                           |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------|
| string    | Specifica la stringa dell'agente utente del browser. Questo valore sostituirà l'agente utente del browser predefinito, "Mozilla/5.0". |

#### Limiti:

L'identificatore **userAgent is** è rilevante solo per i file Web.

Vedere anche:

p [Load \(page 163\)](#)

### Let

L'istruzione **let** è un complemento all'istruzione **set**, utilizzata per definire le variabili degli script. L'istruzione **let**, a differenza dell'istruzione **set**, valuta l'espressione posta sul lato destro del simbolo '=' al tempo di esecuzione dello script prima dell'assegnazione alla variabile.

Sintassi:

```
Let variablename=expression
```

Esempi e risultati:

| Esempio       | Risultato                                                                                                                                                                                                      |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Set x=3+4;    | \$(x) verrà valutato come ' 3+4 '                                                                                                                                                                              |
| Let y=3+4;    | \$(y) verrà valutato come ' 7 '                                                                                                                                                                                |
| z=\$(y)+1;    | \$(z) verrà valutato come ' 8 '                                                                                                                                                                                |
|               | Notare la differenza tra le istruzioni <b>Set</b> e <b>Let</b> . L'istruzione <b>Set</b> assegna la stringa '3+4' alla variabile, mentre l'istruzione <b>Let</b> valuta la stringa e assegna 7 alla variabile. |
| Let T=now( ); | \$(T) verrà restituito il valore dell'ora attuale.                                                                                                                                                             |

### Loosen Table

Una o più tabelle dati interne di Qlik Sense possono essere dichiarate logicamente disconnesse in modo esplicito durante l'esecuzione dello script, utilizzando un'istruzione **Loosen Table**. Quando una tabella è logicamente disconnessa, tutte le associazioni tra i valori di campo nella tabella vengono rimosse. È possibile ottenere un effetto simile caricando ogni campo della tabella logicamente disconnessa come tabelle indipendenti e scollegate. La disconnessione logica può rivelarsi utile durante il controllo per isolare temporaneamente parti differenti della struttura dei dati. Nel visualizzatore tabelle è possibile riconoscere una tabella logicamente disconnessa dalle linee punteggiate. L'utilizzo di una o più istruzioni **Loosen Table** nello script indica a Qlik Sense di ignorare ogni impostazione di tabelle logicamente disconnesse effettuata prima dell'esecuzione dello script.

Sintassi:

```
Loosen Tabletablename [ , tablename2 ...]
```

```
Loosen Tablestablename [ , tablename2 ...]
```

È possibile utilizzare la sintassi: **Loosen Table** o **Loosen Tables**.



*Se Qlik Sense dovesse individuare riferimenti circolari nella struttura dei dati che non possono essere interrotti da tabelle logicamente disconnesse in modo interattivo o esplicito nello script, verrà forzata l'impostazione logicamente disconnessa per una o più tabelle aggiuntive finché non verranno eliminati tutti i riferimenti circolari. In questo caso, nella finestra di dialogo relativa all'avviso di ciclo, verrà visualizzato un avviso.*

### Esempio:

Tab1:

```
SELECT * from Trans;
```

```
Loosen Table Tab1;
```

## Map

L'istruzione **map ... using** viene usata per eseguire il mapping di un certo valore di campo o una certa espressione sui valori di una tabella di mapping specifica. La tabella di mapping viene creata utilizzando l'istruzione **Mapping**.

### Sintassi:

```
Map fieldlist Using mapname
```

Il mapping automatico viene eseguito per i campi caricati dopo l'istruzione **Map ... Using** fino alla fine dello script o finché non viene rilevata un'istruzione **Unmap**.

Il mapping viene eseguito al termine della catena di eventi che conducono alla memorizzazione del campo nella tabella interna in Qlik Sense. Questo significa che le operazioni di mapping non vengono eseguite ogni volta che si incontra un nome di campo come parte di un'espressione, ma più propriamente quando il valore viene salvato nel nome di campo nella tabella interna. Se è richiesto il mapping a livello di espressione, occorre utilizzare la funzione **Applymap()**.

### Argomenti:

#### Argomenti

| Argomento        | Descrizione                                                                                                                                                                                                                                                                                                                                |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>fieldlist</i> | Un elenco separato da virgole dei campi di cui occorre eseguire il mapping da questo punto nello script. L'utilizzo di * per l'elenco dei campi indica tutti i campi. Nei nomi di campo sono consentiti i caratteri speciali * e ?. Se si utilizzano i caratteri speciali può essere necessario delimitare i nomi di campo tra virgolette. |
| <i>mapname</i>   | Il nome di una tabella di mapping letta in precedenza in un'istruzione <b>mapping load</b> o <b>mapping select</b> .                                                                                                                                                                                                                       |

## 3 Istruzioni e parole chiave dello script

Esempi e risultati:

| Esempio                 | Risultato                                                      |
|-------------------------|----------------------------------------------------------------|
| Map Country Using Cmap; | Esegue il mapping del campo Country utilizzando la mappa Cmap. |
| Map A, B, C Using X;    | Esegue il mapping dei campi A, B e C utilizzando la mappa X.   |
| Map * Using GenMap;     | Esegue il mapping di tutti i campi utilizzando GenMap.         |

### NullAsNull

L'istruzione **NullAsNull** disattiva la conversione dei valori NULL in valori di stringa impostati in precedenza da un'istruzione **NullAsValue**.

Sintassi:

```
NullAsNull *fieldlist
```

L'istruzione **NullAsValue** funge da interruttore e può essere attivata o disattivata diverse volte all'interno di uno script utilizzando un'istruzione **NullAsValue** o un'istruzione **NullAsNull**.

Argomenti:

Argomenti

| Argomento  | Descrizione                                                                                                                                                                                                                                                                                                                                   |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| *fieldlist | Un elenco separato da virgole dei campi per il quale è necessario attivare l'istruzione <b>NullAsNull</b> . L'utilizzo di * per l'elenco dei campi indica tutti i campi. Nei nomi di campo sono consentiti i caratteri speciali * e ?. Se si utilizzano i caratteri speciali può essere necessario delimitare i nomi di campo tra virgolette. |

Esempio:

```
NullAsNull A,B;  
LOAD A,B from x.csv;
```

### NullAsValue

L'istruzione **NullAsValue** specifica per quali campi NULL deve essere convertito in un valore.

Sintassi:

```
NullAsValue *fieldlist
```

Per impostazione predefinita, Qlik Sense considera i valori NULL come entità mancanti o non definite. Tuttavia, alcuni database considerano i valori NULL valori speciali piuttosto che semplici valori mancanti. È possibile sospendere il divieto di collegamento reciproco dei valori NULL con altri valori NULL mediante l'istruzione **NullAsValue**.

## 3 Istruzioni e parole chiave dello script

L'istruzione **NullAsValue** funge da interruttore e viene applicata alle istruzioni di caricamento successive. Questa istruzione può essere disattivata di nuovo utilizzando l'istruzione **NullAsNull**.

**Argomenti:**

Argomenti

| Argomento  | Descrizione                                                                                                                                                                                                                                                                                                                                    |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| *fieldlist | Un elenco separato da virgole dei campi per il quale è necessario attivare l'istruzione <b>NullAsValue</b> . L'utilizzo di * per l'elenco dei campi indica tutti i campi. Nei nomi di campo sono consentiti i caratteri speciali * e ?. Se si utilizzano i caratteri speciali può essere necessario delimitare i nomi di campo tra virgolette. |

**Esempio:**

```
NullAsValue A,B;  
Set NullValue = 'NULL';  
LOAD A,B from x.csv;
```

### Qualify

L'istruzione **Qualify** consente di modificare la qualificazione dei nomi di campo, ad esempio il nome della tabella dei nomi di campo diventerà un prefisso.

**Sintassi:**

```
Qualify *fieldlist
```

L'unione automatica dei campi con lo stesso nome in tabelle differenti può essere sospesa con l'utilizzo dell'istruzione **qualify**, che qualifica il nome di campo con il proprio nome di tabella. Se qualificato, il nome campo verrà ridenominato quando verrà trovato in una tabella. Il nuovo nome sarà nel formato *tablename.fieldname*. *Tablename* è equivalente all'etichetta della tabella attuale oppure, se non esiste alcuna etichetta, al nome visualizzato dopo **from** nelle istruzioni **LOAD** e **SELECT**.

La qualificazione viene effettuata per tutti i campi caricati dopo l'istruzione **qualify**

e, per impostazione predefinita, è sempre disattivata all'inizio dell'esecuzione dello script. La qualificazione di un nome di campo può essere attivata in qualsiasi momento utilizzando l'istruzione **qualify**. La qualificazione può essere disattivata in qualsiasi momento utilizzando l'istruzione **Unqualify**.



*L'istruzione **qualify** non deve essere utilizzata insieme al ricaricamento parziale.*

### Argomenti:

#### Argomenti

| Argomento  | Descrizione                                                                                                                                                                                                                                                                                                                    |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| *fieldlist | Un elenco separato da virgola dei campi per i quali è necessario attivare la qualificazione. L'utilizzo di * per l'elenco dei campi indica tutti i campi. Nei nomi di campo sono consentiti i caratteri speciali * e ?. Se si utilizzano i caratteri speciali può essere necessario delimitare i nomi di campo tra virgolette. |

### Example 1:

```
Qualify B;
```

```
LOAD A,B from x.csv;
```

```
LOAD A,B from y.csv;
```

Le due tabelle **x.csv** e **y.csv** vengono associate mediante **A**. Tre campi daranno come risultato: A, x.B, y.B.

### Example 2:

In un database poco familiare si rivela spesso utile iniziare associando le varie tabelle secondo un unico campo o un numero minore di campi, come illustrato nel seguente esempio:

```
qualify *;
```

```
unqualify TransID;
```

```
SQL SELECT * from tab1;
```

```
SQL SELECT * from tab2;
```

```
SQL SELECT * from tab3;
```

Solo il campo **TransID** viene utilizzato per le associazioni tra le tre tabelle *tab1*, *tab2* e *tab3*.

## Rem

L'istruzione **rem** viene utilizzata per inserire osservazioni, o commenti, negli script o per disattivare temporaneamente istruzioni dello script senza rimuoverle.

### Sintassi:

```
Rem string
```

Tutti i contenuti compresi tra **rem** e il punto e virgola ; successivo vengono interpretati come un commento.

Per inserire commenti negli script, sono disponibili altri due metodi:

## 3 Istruzioni e parole chiave dello script

1. È possibile creare un commento in un punto qualsiasi dello script, eccetto tra due virgolette, inserendo la sezione interessata tra `/*` e `*/`.
2. Digitando `//` all'interno dello script, tutto il testo che segue a destra nella stessa riga diventa un commento. (Tenere presente l'eccezione di `//`: che può far parte di un indirizzo Internet).

### Argomenti:

#### Argomenti

| Argomento | Descrizione          |
|-----------|----------------------|
| string    | Un testo arbitrario. |

### Esempio:

```
Rem ** This is a comment **;  
/* This is also a comment */  
// This is a comment as well
```

## Rename

La parola chiave dello script **Rename** consente di rinominare tabelle o campi già caricati.

### Rename field

Questa funzione di script rinomina uno o più campi di Qlik Sense esistenti dopo che sono stati caricati.



*Si sconsiglia di denominare una variabile con lo stesso nome utilizzato per un campo o una funzione in Qlik Sense.*

È possibile utilizzare la sintassi: **rename field** o **rename fields**.

### Sintassi:

```
Rename Field (using mapname | oldname to newname{ , oldname to newname })
```

```
Rename Fields (using mapname | oldname to newname{ , oldname to newname })
```

### Argomenti:

| Argomento | Descrizione                                                                                                              |
|-----------|--------------------------------------------------------------------------------------------------------------------------|
| mapname   | Il nome di una tabella di mapping caricata in precedenza, contenente una o più coppie di nomi di campo obsoleti e nuovi. |
| oldname   | Il nome di campo obsoleto.                                                                                               |
| newname   | Il nome di campo nuovo.                                                                                                  |

### Limiti:

Non è possibile rinominare due campi in modo che abbiano lo stesso nome.

### Example 1:

```
Rename Field XAZ0007 to Sales;
```

### Example 2:

FieldMap:

```
Mapping SQL SELECT oldnames, newnames from datadictionary;
```

```
Rename Fields using FieldMap;
```

## Rename table

Questa funzione di script rinomina una o più tabelle interne di Qlik Sense esistenti dopo che sono state caricate.

È possibile utilizzare la sintassi: **rename table** o **rename tables**.

### Sintassi:

```
Rename Table (using mapname | oldname to newname{ , oldname to newname })  
Rename Tables (using mapname | oldname to newname{ , oldname to newname })
```

### Argomenti:

| Argomenti |                                                                                                                            |
|-----------|----------------------------------------------------------------------------------------------------------------------------|
| Argomento | Descrizione                                                                                                                |
| mapname   | Il nome di una tabella di mapping caricata in precedenza, contenente una o più coppie di nomi di tabella obsoleti e nuovi. |
| oldname   | Il nome di tabella obsoleto.                                                                                               |
| newname   | Il nome di tabella nuovo.                                                                                                  |

### Limiti:

Non è possibile ridenominare nello stesso modo due tabelle con nomi diversi. Lo script genererà un errore se si tenta di rinominare una tabella con lo stesso nome di una tabella esistente.

### Example 1:

Tab1:

```
SELECT * from Trans;
```

```
Rename Table Tab1 to Xyz;
```

### Example 2:

TabMap:  
Mapping LOAD oldnames, newnames from tabnames.csv;  
Rename Tables using TabMap;

## Search

L'istruzione **Search** viene utilizzata per includere o escludere campi nella ricerca intelligente.

### Sintassi:

```
Search Include *fieldlist  
Search Exclude *fieldlist
```

È possibile utilizzare diverse istruzioni Search per affinare la selezione dei campi da includere. Le istruzioni vengono valutate dall'alto verso il basso.

### Argomenti:

#### Argomenti

| Argomento  | Descrizione                                                                                                                                                                                                                                                                                                                                   |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| *fieldlist | Un elenco separato da virgole dei campi da includere o escludere dalle ricerche nella ricerca intelligente. L'utilizzo di * per l'elenco dei campi indica tutti i campi. Nei nomi di campo sono consentiti i caratteri speciali * e ?. Se si utilizzano i caratteri speciali può essere necessario delimitare i nomi di campo tra virgolette. |

### Esempio:

#### Esempi di ricerche

| Istruzione                | Descrizione                                                                                         |
|---------------------------|-----------------------------------------------------------------------------------------------------|
| Search Include *;         | Consente di includere tutti i campi nelle ricerche della ricerca intelligente.                      |
| Search Exclude [*ID];     | Consente di escludere tutti i campi che terminano con ID dalle ricerche della ricerca intelligente. |
| Search Exclude '*ID';     | Consente di escludere tutti i campi che terminano con ID dalle ricerche della ricerca intelligente. |
| Search Include ProductID; | Consente di includere il campo ProductID nelle ricerche della ricerca intelligente.                 |

Il risultato combinato di queste tre istruzioni, in questa sequenza, è che dalle ricerche della ricerca intelligente vengono esclusi tutti i campi che terminano con ID tranne ProductID.

### Section

L'istruzione **section** consente di definire se le istruzioni successive **LOAD** e **SELECT** devono essere considerate come dati o come una definizione dei diritti di accesso.

**Sintassi:**

```
Section (access | application)
```

Se non viene specificato alcun valore, viene utilizzato **section application**. La definizione **section** è valida finché non si specifica una nuova istruzione **section**.

**Esempio:**

```
Section access;
```

```
Section application;
```

### Select

La selezione dei campi da una sorgente dati ODBC o da un provider OLE DB viene eseguita utilizzando le istruzioni SQL **SELECT** standard. Tuttavia, l'ambito nel quale le istruzioni **SELECT** vengono accettate dipende dal driver ODBC o dal provider OLE DB utilizzato. L'utilizzo dell'istruzione **SELECT** richiede una connessione dati aperta all'origine.

**Sintassi:**

```
Select [all | distinct | distinctrow | top n [percent] ] fieldlist  
  
From tablelist  
  
[where criterion ]  
  
[group by fieldlist [having criterion ] ]  
  
[order by fieldlist [asc | desc] ]  
  
[ (Inner | Left | Right | Full) join tablename on fieldref = fieldref ]
```

Inoltre, talvolta è possibile concatenare diverse istruzioni **SELECT** in un'unica istruzione tramite l'utilizzo dell'operatore **union**:

```
selectstatement Union selectstatement
```

L'istruzione **SELECT** viene interpretata dal driver ODBC o dal provider OLE DB e, pertanto, possono verificarsi deviazioni dalla sintassi generale SQL a seconda delle caratteristiche dei driver ODBC o del provider OLE DB, ad esempio:

### 3 Istruzioni e parole chiave dello script

- Talvolta **as** non è consentito, vale a dire *aliasname* deve seguire immediatamente *fieldname*.
- Talvolta **as** è obbligatorio se si utilizza un *aliasname*.
- **distinct**, **as**, **where**, **group by**, **order by** o **union** a volte non sono supportati.
- Il driver ODBC talvolta non accetta tutte le diverse virgolette elencate in precedenza.



La descrizione qui fornita dell'istruzione SQL **SELECT** non è completa. Ad esempio, le istruzioni **SELECT** possono essere nidificate, più unioni possono essere inserite in un'unica istruzione **SELECT**, a volte il numero di funzioni consentito nelle espressioni può essere molto alto e così via.

#### Argomenti:

##### Argomenti

| Argomento   | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| distinct    | <b>distinct</b> è un predicato che viene utilizzato se le combinazioni duplicate dei valori nei campi selezionati devono essere caricate una sola volta.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| distinctrow | <b>distinctrow</b> è un predicato che viene utilizzato se i record duplicati presenti nella tabella sorgente devono essere caricati una sola volta.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| fieldlist   | <b>fieldlist</b> ::= ( *   field ) { , field }<br>Un elenco dei campi da selezionare. L'utilizzo del simbolo * come elenco dei campi indica tutti i campi della tabella.<br><b>fieldlist</b> ::= field { , field }<br>Un elenco di uno o più campi separati da virgole.<br><b>field</b> ::= ( fieldref   expression ) [ as aliasname ]<br>L'espressione può essere, ad esempio, una funzione numerica oppure una funzione di stringa basata su uno o più campi diversi. Tra gli operatori e le funzioni generalmente accettati si annoverano: +, -, *, /, & (concatenazione di stringhe), sum(fieldname), count(fieldname), avg(fieldname)(average), month(fieldname) ecc. Per ulteriori informazioni, consultare la documentazione del driver ODBC.<br><b>fieldref</b> ::= [ tablename. ] fieldname<br><b>tablename</b> e <b>fieldname</b> sono stringhe di testo identiche a ciò che implicano. Se contengono spazi, ad esempio, devono essere incluse fra doppie virgolette diritte. La clausola <b>as</b> viene utilizzata per assegnare un nuovo nome al campo. |
| from        | <b>tablelist</b> ::= table { , table }<br>Elenco di tabelle da cui vengono selezionati i campi.<br><b>table</b> ::= tablename [ [ as ] aliasname ]<br><b>tablename</b> può essere inserito o meno tra virgolette.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

### 3 Istruzioni e parole chiave dello script

| Argomento | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| where     | <b>where</b> è una clausola utilizzata per dichiarare se un record deve essere incluso o meno nella selezione.<br><b>criterion</b> è un'espressione logica che a volte può risultare molto complessa. Alcuni degli operatori accettati sono: funzioni e operatori numerici, =, <> o #(diverso da), >, >=, <, <=, <b>and</b> , <b>or</b> , <b>not</b> , <b>exists</b> , <b>some</b> , <b>all</b> , <b>in</b> e anche le nuove istruzioni <b>SELECT</b> . Consultare la documentazione del driver ODBC o del provider OLE DB per ulteriori informazioni. |
| group by  | <b>group by</b> è una clausola utilizzata per aggregare (raggruppare) più record in uno solo. All'interno di un gruppo, per un determinato campo, tutti i record devono avere lo stesso valore o il campo può essere utilizzato solo all'interno di un'espressione, ad esempio una somma o una media. L'espressione basata su uno o più campi viene definita nell'espressione del simbolo del campo.                                                                                                                                                   |
| having    | <b>having</b> è una clausola utilizzata per qualificare i gruppi analogamente al modo in cui la clausola <b>where</b> viene utilizzata per qualificare i record.                                                                                                                                                                                                                                                                                                                                                                                       |
| order by  | <b>order by</b> è una clausola utilizzata per dichiarare la sequenza di ordinamento della tabella risultante dall'istruzione <b>SELECT</b> .                                                                                                                                                                                                                                                                                                                                                                                                           |
| join      | <b>join</b> è un qualificatore che dichiara se diverse tabelle devono essere unite in una sola. I nomi dei campi e delle tabelle devono essere delimitati da virgolette se contengono spazi vuoti o lettere dell'alfabeto nazionale. Quando lo script viene generato automaticamente da Qlik Sense, le virgolette utilizzate sono quelle preferite dal driver ODBC o dal provider OLE DB specificato nella definizione della sorgente dati nell'istruzione <b>Connect</b> .                                                                            |

#### Example 1:

```
SELECT * FROM `Categories`;
```

#### Example 2:

```
SELECT `Category ID`, `Category Name` FROM `Categories`;
```

#### Example 3:

```
SELECT `Order ID`, `Product ID`,  
`Unit Price` * Quantity * (1-Discount) as NetSales  
FROM `Order Details`;
```

#### Example 4:

```
SELECT `Order Details`.`Order ID`,  
Sum(`Order Details`.`Unit Price` * `Order Details`.Quantity) as `Result`  
FROM `Order Details`, Orders  
where Orders.`Order ID` = `Order Details`.`Order ID`  
group by `Order Details`.`Order ID`;
```

### Set

L'istruzione **set** viene utilizzata per definire le variabili di script. Le variabili possono essere utilizzate per sostituire stringhe, percorsi, unità e così via.

#### Sintassi:

```
Set variablename=string
```

#### Example 1:

```
Set FileToUse=Data1.csv;
```

#### Example 2:

```
Set Constant="My string";
```

#### Example 3:

```
Set BudgetYear=2012;
```

### Sleep

L'istruzione **sleep** interrompe l'esecuzione dello script per il periodo di tempo specificato.

#### Sintassi:

```
Sleep n
```

#### Argomenti:

| Argomento | Descrizione                                                                                                                                                      |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| n         | Indicato in millisecondi, in cui <i>n</i> rappresenta un numero intero positivo che non supera 3600000 (ad esempio, 1 ora). Il valore può essere un'espressione. |

#### Example 1:

```
Sleep 10000;
```

#### Example 2:

```
Sleep t*1000;
```

### SQL

L'istruzione **SQL** consente di inviare un comando arbitrario SQL tramite una connessione ODBC o OLE DB.

#### Sintassi:

```
SQL sql_command
```

### 3 Istruzioni e parole chiave dello script

L'invio di istruzioni SQL che aggiornano il database restituisce un errore se Qlik Sense ha aperto la connessione ODBC in modalità di sola lettura.

La sintassi:

```
SQL SELECT * from tab1;
```

È consentita e, per ragioni di uniformità, rappresenta la sintassi preferita per l'istruzione **SELECT**. Il prefisso SQL rimarrà comunque opzionale per le istruzioni **SELECT**.

**Argomenti:**

| Argomento          | Descrizione            |
|--------------------|------------------------|
| <i>sql_command</i> | Un comando SQL valido. |

**Example 1:**

```
SQL leave;
```

**Example 2:**

```
SQL Execute <storedProc>;
```

## SQLColumns

L'istruzione **sqlcolumns** restituisce un gruppo di campi che descrivono le colonne di una sorgente dati ODBC o OLE DB sulla quale è stata eseguita un'istruzione **connect**.

**Sintassi:**

```
SQLcolumns
```

Questi campi possono essere combinati con i campi generati dai comandi **sqltables** e **sqltypes** per ottenere una visione d'insieme di un determinato database. I dodici campi standard sono:

TABLE\_QUALIFIER

TABLE\_OWNER

TABLE\_NAME

COLUMN\_NAME

DATA\_TYPE

TYPE\_NAME

PRECISION

LENGTH

SCALE

RADIX

NULLABLE

REMARKS

Per una descrizione dettagliata di questi campi, consultare un manuale di riferimento di ODBC.

### Esempio:

```
Connect to 'MS Access 7.0 Database; DBQ=C:\Course3\DataSrc\QWT.mbd';  
SQLcolumns;
```



*Alcuni driver ODBC potrebbero non supportare questo comando. Alcuni driver ODBC potrebbero produrre campi aggiuntivi.*

## SQLTables

L'istruzione **sqltables** restituisce un gruppo di campi che descrivono le tabelle di una sorgente dati ODBC o OLE DB sulla quale è stata eseguita un'istruzione **connect**.

### Sintassi:

**SQLTables**

Questi campi possono essere combinati con i campi generati dai comandi **sqlcolumns** e **sqltypes** per ottenere una visione d'insieme di un determinato database. I cinque campi standard sono:

TABLE\_QUALIFIER

TABLE\_OWNER

TABLE\_NAME

TABLE\_TYPE

REMARKS

Per una descrizione dettagliata di questi campi, consultare un manuale di riferimento di ODBC.

### Esempio:

```
Connect to 'MS Access 7.0 Database; DBQ=C:\Course3\DataSrc\QWT.mbd';  
SQLTables;
```



*Alcuni driver ODBC potrebbero non supportare questo comando. Alcuni driver ODBC potrebbero produrre campi aggiuntivi.*

## SQLTypes

L'istruzione **sqltypes** restituisce un gruppo di campi che descrivono i tipi di una sorgente dati ODBC o OLE DB sulla quale è stata eseguita un'istruzione **connect**.

### Sintassi:

#### SQLTypes

Questi campi possono essere combinati con i campi generati dai comandi **sqlcolumns** e **sqltables** per ottenere una visione d'insieme di un determinato database. I quindici campi standard sono:

TYPE\_NAME  
DATA\_TYPE  
PRECISION  
LITERAL\_PREFIX  
LITERAL\_SUFFIX  
CREATE\_PARAMS  
NULLABLE  
CASE\_SENSITIVE  
SEARCHABLE  
UNSIGNED\_ATTRIBUTE  
MONEY  
AUTO\_INCREMENT  
LOCAL\_TYPE\_NAME  
MINIMUM\_SCALE  
MAXIMUM\_SCALE

Per una descrizione dettagliata di questi campi, consultare un manuale di riferimento di ODBC.

### Esempio:

```
Connect to 'MS Access 7.0 Database; DBQ=C:\Course3\DataSrc\QWT.mbd';  
SQLTypes;
```



*Alcuni driver ODBC potrebbero non supportare questo comando. Alcuni driver ODBC potrebbero produrre campi aggiuntivi.*

### Star

La stringa utilizzata per rappresentare l'insieme di tutti i valori di un campo nel database può essere impostata tramite l'istruzione **star**. Interessa le istruzioni **LOAD** e **SELECT** successive.

#### Sintassi:

```
Star is[ string ]
```

#### Argomenti:

| Argomenti |                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Argomento | Descrizione                                                                                                                                                                                                                                                                                                                                                                       |
| string    | <p>Un testo arbitrario. Tenere presente che se la stringa contiene spazi deve essere racchiusa tra virgolette.</p> <p>Se non si specifica alcun valore, viene utilizzato <b>star is;</b>, ossia non è disponibile alcun simbolo star finché non viene specificato in maniera esplicita. Questa definizione è valida finché non si specifica una nuova istruzione <b>star</b>.</p> |

Si sconsiglia l'uso dell'istruzione **Star is** nella parte dei dati dello script (sotto **Section Application**) se si utilizza **Section Access**. Il carattere asterisco è tuttavia pienamente supportato per i campi protetti nella parte **Section Access** dello script. In questo caso non è necessario utilizzare l'istruzione **Star is** esplicita in quanto è sempre implicita in **Section Access**.

### Limiti

- Non è possibile utilizzare il carattere asterisco con i campi chiave, ovvero i campi che collegano le tabelle.
- Non è possibile utilizzare il carattere asterisco con qualsiasi campo interessato dall'istruzione **Unqualify**, in quanto ciò può avere ripercussioni sui campi che collegano le tabelle.
- Non è possibile utilizzare il carattere asterisco con tabelle non logiche, ad esempio tabelle info-load o tabelle mapping-load.
- Quando viene utilizzato in un campo di riduzione (un campo collegato ai dati) in **Section Access**, il carattere asterisco rappresenta i valori elencati in questo campo in **Section Access**. Non rappresenta altri valori che possono esistere nei dati, ma non sono elencati in **Section Access**.
- Non è possibile utilizzare il carattere asterisco con campi interessati da qualsiasi forma di riduzione dei dati al di fuori dell'area **Section Access**.

### Esempio

L'esempio seguente è un estratto di uno script di caricamento dei dati contenente la sezione relativa al controllo degli accessi.

```
Star is *;
```

Section Access;

LOAD \* INLINE [

ACCESS, USERID, OMIT

ADMIN, ADMIN,

USER, USER1, SALES

USER, USER2, WAREHOUSE

USER, USER3, EMPLOYEES

USER, USER4, SALES

USER, USER4, WAREHOUSE

USER, USER5, \*

];

Section Application;

LOAD \* INLINE [

SALES, WAREHOUSE, EMPLOYEES, ORDERS

1, 2, 3, 4

];

Viene applicato quanto segue:

- Il segno *Star* corrisponde a \*.
- L'utente *ADMIN* visualizza tutti i campi. Non viene omissso nulla.
- L'utente *USER1* non può visualizzare il campo *SALES*.
- L'utente *USER2* non può visualizzare il campo *WAREHOUSE*.
- L'utente *USER3* non può visualizzare il campo *EMPLOYEES*.
- L'utente *USER4* è stato aggiunto due volte alla soluzione per omettere (OMIT) due campi relativi a questo utente, *SALES* e *WAREHOUSE*.
- *USER5* presenta un "\*" aggiunto che significa che tutti i campi elencati in OMIT sono indisponibili, ovvero, l'utente *USER5* non può vedere i campi *SALES*, *WAREHOUSE* e *EMPLOYEES* ma questo utente può visualizzare il campo *ORDERS*.

## Store

L'istruzione **Store** crea un file QVD, Parquet, CSV, o TXT.

### Sintassi:

```
Store [ fieldlist from] table into filename [ format-spec ];
```

L'istruzione creerà un file con nome esplicito QVD, Parquet o un file di testo.

L'istruzione può esportare campi solo da una tabella dati, a meno che non si effettui l'archiviazione in Parquet. Se occorre esportare i campi da più tabelle in un file QVD, CSV o TXT, prima è necessario eseguire un'operazione join esplicita nello script per la creare la tabella dati da esportare. È possibile archiviare più tabelle in un unico Parquet annidando i dati nei file Parquet.

I valori di testo sono esportati nel file CSV in UTF-8 formato BOM. È possibile specificare un delimitatore, vedere **LOAD**. L'istruzione **store** associata a un file CSV non supporta l'esportazione BIFF.



*In alcuni casi in cui i dati non sono formattati correttamente, i campi vengono racchiusi da virgolette doppie, per assicurare che i dati siano interpretati correttamente. Questo può succedere, per esempio, quando il campo contiene caratteri come virgolette, virgole, spazi o interruzioni di riga.*

### Argomenti:

#### Argomenti del comando Store

| Argomento                                     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>fieldlist::= ( *   field ) { , field }</i> | Un elenco dei campi da selezionare. L'utilizzo del carattere * per l'elenco dei campi indica tutti i campi.<br><br><i>field::= fieldname [as aliasname]</i><br><br><i>fieldname</i> è un testo che è identico al nome di campo in <i>table</i> . (Tenere presente che il nome di campo deve essere racchiuso da virgolette doppie diritte o parentesi quadre se, ad esempio, contiene spazi o altri caratteri non standard).<br><br><i>aliasname</i> è un nome alternativo per il campo da utilizzare nel file QVD o CSV risultante. |
| <i>table</i>                                  | Un'etichetta dello script che rappresenta una tabella già caricata da utilizzare come sorgente dei dati.                                                                                                                                                                                                                                                                                                                                                                                                                             |

### 3 Istruzioni e parole chiave dello script

---

| Argomento       | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>filename</i> | <p>Il nome del file di destinazione che include un percorso valido a una connessione dati della cartella esistente.</p> <p><b>Esempio: <i>'lib://Table Files/target.qvd'</i></b></p> <p>Nella modalità di creazione degli script legacy sono supportati anche i seguenti formati di percorso:</p> <ul style="list-style-type: none"><li>• assoluto</li></ul> <p><b>Esempio: <i>c:\data\sales.qvd</i></b></p> <ul style="list-style-type: none"><li>• relativo alla directory di lavoro dell'app Qlik Sense.</li></ul> <p><b>Esempio: <i>data\sales.qvd</i></b></p> <p>Se il percorso viene omissso, Qlik Sense memorizza il file nella directory specificata dall'istruzione <b>Directory</b>. Se non è presente alcuna istruzione <b>Directory</b>, Qlik Sense memorizza il file nella directory di lavoro, <i>C:\Users\{user}\Documents\Qlik\Sense\Apps</i>.</p> |

### 3 Istruzioni e parole chiave dello script

| Argomento                                                                 | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|---------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>format-spec ::= ( ( txt   qvd   parquet), la compressione è codec)</i> | <p>È possibile impostare la specifica del formato per entrambi i formati file. Se si omette la specifica del formato, viene utilizzato <b>qvd</b>.</p> <ul style="list-style-type: none"><li>• <b>txt</b> per i file CSV e TXT.</li><li>• <b>qvd</b> per i file QVD.</li><li>• <b>parquet</b> per i file Parquet.</li></ul> <p>Se si utilizza <b>parquet</b>, è possibile anche impostare quale codec di compressione utilizzare con <b>compression is</b>. Se non si specifica il codec di compressione con <b>compression is</b>, viene utilizzato snappy. Sono disponibili le seguenti impostazioni di compressione:</p> <ul style="list-style-type: none"><li>• uncompressed</li><li>• snappy</li><li>• gzip</li><li>• lz4</li><li>• brotli</li><li>• zstd</li><li>• lz4_hadoop</li></ul> <p>Esempio:</p> <pre>store mytable into [lib://AttachedFiles/myfile.parquet] (parquet, compression is lz4);</pre> |

#### Esempi:

```
store mytable into xyz.qvd (qvd);
```

```
store * from mytable into 'lib://FolderConnection/myfile.qvd';
```

```
store Name, RegNo from mytable into xyz.qvd;
```

```
store Name as a, RegNo as b from mytable into 'lib://FolderConnection/myfile.qvd';
```

```
store mytable into myfile.txt (txt);
```

```
store mytable into [lib://FolderConnection/myfile.csv] (txt);
```

```
store mytable into myfile.parquet (parquet);
```

```
store * from mytable into 'lib://FolderConnection/myfile.qvd';
```

### Archiviazione nei file parquet

Parquet è un formato file fortemente tipizzato, in cui ogni campo contenente un singolo tipo di dati specifico (come ad esempio INT32, doppio, timestamp, o testo). Qlik Sense archivia i dati interni come doppi poco tipizzati, in cui i dati da sorgenti differenti possono essere mescolati negli stessi campi. Poiché nei file Parquet è possibile archiviare solo una parte dei doppi in ogni campo, è importante sapere cosa contiene ciascuno di essi. Per impostazione predefinita, Qlik Sense utilizza i campi per determinare come devono essere archiviati. Quando si archiviano dati nei file Parquet in un formato specifico, è necessario specificare qual è il tipo di dati dei campi al caricamento. Se si tenta di archiviare i dati in campi non compatibili in un file Parquet, come ad esempio dei numeri in un campo di testo o del testo in un campo timestamp, verranno generati valori null.

Quando si caricano dati che si desidera archiviare in Parquet, è possibile modificare il comportamento predefinito. È possibile formattarlo per modificare il tipo di dati o aggiungere tag per applicare forzatamente i tipi di colonna in Parquet.

### Formattazione dei dati per l'archiviazione in Parquet

È possibile utilizzare le funzioni di formattazione di Qlik Sense per classificare i dati. Ad esempio, **Text()**, **Num()**, **Interval()** o **Timestamp()** possono applicare forzatamente i formati dati per l'archiviazione in Parquet. Qlik Sense può archiviare i dati in quasi 20 tipi di dati differenti, a seconda degli attributi del campo e dei tag di campo automatici. Per ulteriori informazioni, vedere [Funzioni di interpretazione \(page 1433\)](#).

#### Esempio: Formattazione dei dati con Num() e Text()

Nel seguente esempio viene descritta la preparazione dati per l'archiviazione in Parquet. Il valore **Num()** è applicato ai campi numerici, mentre **Text()** è applicato ai campo di testo e misti. Nel caso dei campi misti, **Text()** impedisce che vengano trattati come campi numerici in Parquet e che i valori di testo vengano modificati in null.

Data:

```
LOAD * INLINE [  
  num, text, mixed  
  123.321, abc, 123  
  456.654, def, xyz  
  789.987, ghi, 321  
];
```

Format:

```
NoConcatenate  
LOAD num, text, Text(mixed) as mixed RESIDENT Data;  
STORE Format INTO [lib://AttachedFiles/Tmp.parquet] (parquet);
```

### Tagging dei dati per l'archiviazione in Parquet

È possibile applicare tag ai dati con i tag \$parquet per forzare l'applicazione di tipi di colonne specifici durante l'archiviazione dei dati in Parquet. È possibile forzare ogni tipo di dati aggiungendo i corrispondenti tag di controllo. Per esempio, per archiviare un campo come INT32 in Parquet, aggiungere un tag utilizzando \$parquet-int32 nello script di caricamento. A seconda del tipo di dati, verranno archiviate la stringa o la rappresentazione numerica dei dati doppi.

### 3 Istruzioni e parole chiave dello script

I seguenti tag di controllo Parquet possono essere utilizzati per taggare i campi per l'archiviazione nei file Parquet.

| Tag di controllo Parquet |         |                      |             |                  |
|--------------------------|---------|----------------------|-------------|------------------|
| Tag di controllo         | Duale   | Tipo fisico          | Tipo logico | Tipo convertito  |
| \$parquet-boolean        | Numero  | BOOLEAN              | NONE        | NONE             |
| \$parquet-int32          | Numero  | INT32                | NONE        | NONE             |
| \$parquet-int64          | Numero  | INT64                | NONE        | NONE             |
| \$parquet-float          | Numero  | FLOAT                | NONE        | NONE             |
| \$parquet-double         | Numero  | DOUBLE               | NONE        | NONE             |
| \$parquet-bytearray      | Stringa | BYTE_ARRAY           | NONE        | UTF8             |
| \$parquet-bytearrayfix   | Numero  | FIXED_LEN_BYTE_ARRAY | NONE        | DECIMAL          |
| \$parquet-decimal        | Numero  | INT64                | DECIMAL     | DECIMAL          |
| \$parquet-date           | Numero  | INT32                | DATE        | DATE             |
| \$parquet-time           | Numero  | INT64                | TIME        | TIME_MICROS      |
| \$parquet-timestamp      | Numero  | INT64                | TIMESTAMP   | TIMESTAMP_MICROS |
| \$parquet-string         | Stringa | BYTE_ARRAY           | STRING      | UTF8             |
| \$parquet-enum           | Stringa | BYTE_ARRAY           | ENUM        | ENUM             |
| \$parquet-interval       | Numero  | FIXED_LEN_BYTE_ARRAY | INTERVAL    | INTERVAL         |
| \$parquet-json           | Stringa | BYTE_ARRAY           | JSON        | JSON             |
| \$parquet-bson           | Stringa | BYTE_ARRAY           | BSON        | BSON             |
| \$parquet-uuid           | Stringa | FIXED_LEN_BYTE_ARRAY | UUID        | NONE             |

#### Esempio: Tagging dei dati per l'archiviazione in Parquet

In questo esempio, vengono utilizzati due tag per definire i dati per Parquet. Il campo *num* è taggato come \$parquet-int32 per definirlo come campo numerico che verrà impostato come INT32 in Parquet.

```
Data:
LOAD * INLINE [
num, text,
123.321, abc
456.654, def
789.987, ghi
];
TAG num WITH '$parquet-int32';
STORE Format INTO [lib://AttachedFiles/Tmp.parquet] (parquet);
```

### Archiviazione dei dati annidati in file Parquet

È possibile archiviare più tabelle in un file Parquet annidandole in dati strutturati. **Store** supporta nodi strutturati e nodi elenco in uno schema stella. È possibile anche archiviare singole tabelle in modalità annidata utilizzando l'identificatore **Il delimitatore è**.

Quando si archiviano le tabelle, specificare quelle che si desidera includere separate da virgole. Ad esempio: `STORE Table1, Table2, Table3 INTO [lib://<file location>/<file name>.parquet] (parquet);`. È possibile controllare i campi da archiviare utilizzando un elenco campo nell'istruzione **Store**. Ad esempio `STORE Field1, Field2, FROM Table1, Table2 INTO [lib://<file location>/<file name>.parquet] (parquet);`. Tutti i campi nell'elenco campo devono essere inclusi in una o più delle tabelle elencate. La prima tabella nell'istruzione **Store** verrà utilizzata come tabella dei fatti nello schema stella.

I nomi del campo sono utilizzati per controllare come i gruppi verranno creati e annidati. Per impostazione predefinita, i nomi del campo sono divisi in nodi con un periodo (.). Il delimitatore può essere modificato impostando la variabile di sistema *FieldNameDelimiter* o utilizzando l'identificatore **Il delimitatore è**. L'identificatore sovrascriverà la variabile di sistema..

I nomi dei campi sono divisi dal delimitatore e le parti sono utilizzate per creare lo schema con gruppi annidati. Per esempio, `STORE Field1, Field1.Field2, Field1.Field3, Field1.Field4 FROM Table1 INTO [nested.parquet] (parquet, delimiter is '.');` creerà due gruppi (*Gruppo1* e *Gruppo2*) con *Campi1*, *Campi2* e *Campi3*, *Campi4*.



*I gruppi e i campi potrebbero non avere lo stesso nome in un nodo nello schema. Per esempio, l'istruzione `STORE Address, Address.Street INTO [nested.parquet] (parquet, delimiter is ' ');` non riuscirà poiché il campo *Indirizzo* è ambiguo ed è sia un campo dati sia un gruppo.*

Quando si archiviano dati annidati in Parquet, le chiavi tra le tabelle sono trasformate in nodi di collegamento nello schema. Le tabelle vengono trasformate in nodi strutturati nello schema. È possibile sovrascrivere la trasformazione predefinita utilizzando i nomi di campo.

### Esempio: Archiviazione dei dati annidati in un file Parquet

```
company:
LOAD * INLINE [
company, contact
A&G, Amanda Honda
Cabro, Cary Frank
Fenwick, Dennis Fisher
Camros, Molly McKenzie
];
```

```
salesrep:
LOAD * INLINE [
company, salesrep
A&G, Bob Park
Cabro, Cezar Sandu
Fenwick, Ken Roberts
```

```
Camros, Max Smith  
];
```

```
headquarter:  
LOAD * INLINE [  
company, country, city  
A&G, USA, Los Angeles  
Cabro, USA, Albuquerque  
Fenwick, USA, Baltimore  
Camros, USA, Omaha  
];
```

```
region:  
LOAD * INLINE [  
region, city  
West, Los Angeles  
Southwest, Albuquerque  
East, Baltimore  
Central, Omaha  
];
```

```
STORE company, salesrep, headquarter, region INTO [lib://AttachedFiles/company.parquet]  
(parquet)  
DROP TABLES company, salesrep, headquarter, region;  
Il file Parquet risultante ha il seguente schema:
```

```
company (String)  
contact (String)  
company:salesrep (List)  
    salesrep (Group)  
        salesrep (String)  
company:headquarter (List)  
    headquarter (Group)  
        country (String)  
        city (String)  
        city:region (List)  
            region (Group)  
                region (String)
```

### Limitazioni

L'archiviazione dei dati annidati in Parquet presenta le seguenti limitazioni:

- L'archiviazione non supporta i nodi mappa.
- L'archiviazione non include i campi chiave generati dal caricamento dei file parquet annidati.
- Non è possibile archiviare i dati da più tabelle insieme che non siano collegate con campi chiave.
- Il file annidato denormalizza il modello dati. I valori senza riferimento non verranno salvati e i valori con riferimenti multipli verranno copiati.

### Limitazioni

- Le istruzioni **STORE** dello script di caricamento non sono supportate da Connettori a provider di archiviazione Web in Qlik Sense Enterprise on Windows e Qlik Sense Desktop.

### Table/Tables

Le parole chiave dello script **Table** e **Tables** vengono utilizzate sia nelle istruzioni **Drop**, **Comment** e **Rename** che come identificatori di formato nelle istruzioni **Load**.

### Tag

Questa istruzione dello script fornisce un modo per assegnare tag a uno o più campi o tabelle. Se viene effettuato un tentativo di contrassegnare un campo o una tabella non presente nell'app, tale contrassegno verrà ignorato. In caso di conflitto nelle ricorrenze di un nome di campo o di tag, si utilizza l'ultimo valore trovato.

Sintassi:

```
Tag [field|fields] fieldlist with tagname
```

```
Tag [field|fields] fieldlist using mapname
```

```
Tag table tablelist with tagname
```

#### Argomenti

| Argomento | Descrizione                                                                                                             |
|-----------|-------------------------------------------------------------------------------------------------------------------------|
| fieldlist | Uno o più campi che devono essere taggati, in un elenco separato da virgole.                                            |
| mapname   | Il nome di una tabella di mapping caricata in precedenza in un'istruzione <b>mapping Load</b> o <b>mapping Select</b> . |
| tablelist | Un elenco separato da virgole delle tabelle che devono essere taggate.                                                  |
| tagname   | Il nome del tag da applicare al campo.                                                                                  |

#### Example 1:

tagmap:

```
mapping LOAD * inline [
```

```
a,b
```

```
Alpha,MyTag
```

```
Num,MyTag
```

```
];
```

```
tag fields using tagmap;
```

#### Example 2:

```
tag field Alpha with 'MyTag2';
```

### Trace

L'istruzione **trace** esegue la scrittura di una stringa nella finestra **Avanzamento dell'esecuzione dello script** e nel file di log dello script, quando viene utilizzato. Si rivela molto utile per le operazioni di debug. L'uso delle espansioni \$ delle variabili calcolate prima dell'istruzione **trace** consente di personalizzare il messaggio.

#### Sintassi:

```
Trace string
```

#### Example 1:

L'istruzione seguente può essere utilizzata subito dopo l'istruzione LOAD che carica la tabella 'Principale'.

```
Trace Main table loaded;
```

Ciò visualizzerà il testo 'Tabella principale caricata' nella finestra di dialogo esecuzione script e nel file di log.

#### Example 2:

Le istruzioni seguenti possono essere utilizzate subito dopo l'istruzione LOAD che carica la tabella 'Principale'.

```
Let MyMessage = NoOfRows('Main') & ' rows in Main table';
```

```
Trace $(MyMessage);
```

Ciò visualizzerà un testo che mostra il numero di righe nella finestra di dialogo di esecuzione script e nel file di log, ad esempio '265.391 righe nella tabella principale'.

### Unmap

L'istruzione **Unmap** disattiva il mapping del valore di campo specificato da un'istruzione **Map ... Using** precedente per i campi caricati successivamente.

#### Sintassi:

```
Unmap *fieldlist
```

#### Argomenti:

##### Argomenti

| Argomento  | Descrizione                                                                                                                                                                                                                                                                                                                                                 |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| *fieldlist | Un elenco separato da virgole dei campi per i quali non deve più essere eseguito il mapping da questo punto nello script. L'utilizzo di * per l'elenco dei campi indica tutti i campi. Nei nomi di campo sono consentiti i caratteri speciali * e ?. Se si utilizzano i caratteri speciali può essere necessario delimitare i nomi di campo tra virgolette. |

Esempi e risultati:

| Esempio        | Risultato                                 |
|----------------|-------------------------------------------|
| Unmap Country; | Disabilita il mapping del campo Country.  |
| Unmap A, B, C; | Disabilita il mapping dei campi A, B e C. |
| Unmap * ;      | Disabilita il mapping di tutti i campi.   |

### Unqualify

L'istruzione **Unqualify** viene utilizzata per disattivare la qualificazione dei nomi di campo che era stata precedentemente attivata dall'istruzione **Qualify**.

Sintassi:

```
Unqualify *fieldlist
```

Argomenti:

| Argomenti  |                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Argomento  | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                           |
| *fieldlist | Un elenco separato da virgola dei campi per i quali è necessario attivare la qualificazione. L'utilizzo di * per l'elenco dei campi indica tutti i campi. Nei nomi di campo sono consentiti i caratteri speciali * e ?. Se si utilizzano i caratteri speciali può essere necessario delimitare i nomi di campo tra virgolette.<br><br>Consultare la documentazione relativa all'istruzione <b>Qualify</b> per ulteriori informazioni. |

#### Example 1:

In un database poco familiare si rivela spesso utile iniziare associando le varie tabelle secondo un unico campo o un numero minore di campi, come illustrato nel seguente esempio:

```
qualify *;  
unqualify TransID;  
SQL SELECT * from tab1;  
SQL SELECT * from tab2;  
SQL SELECT * from tab3;
```

Inizialmente, la qualificazione è attivata per tutti i campi.

Dopodiché, la qualificazione è disattivata per **TransID**.

Solo il campo **TransID** viene utilizzato per le associazioni tra le tre tabelle *tab1*, *tab2* e *tab3*. Tutti gli altri campi verranno qualificati con lo stesso nome tabella.

### Untag

Questa istruzione dello script fornisce un modo per rimuovere tag da campi o tabelle. Se viene effettuato un tentativo di rimozione di un contrassegno da un campo o una tabella non presente nell'app, tale rimozione verrà ignorata.

### Sintassi:

```
Untag [field|fields] fieldlist with tagname
```

```
Untag [field|fields] fieldlist using mapname
```

```
Untag table tablelist with tagname
```

### Argomenti:

#### Argomenti

| Argomento | Descrizione                                                                                                             |
|-----------|-------------------------------------------------------------------------------------------------------------------------|
| fieldlist | Uno o più campi dai quali rimuovere i tag, in un elenco separato da virgole.                                            |
| mapname   | Il nome di una tabella di mapping caricata in precedenza in un'istruzione mapping <b>LOAD</b> o mapping <b>SELECT</b> . |
| tablelist | Un elenco separato da virgole delle tabelle da cui rimuovere i tag.                                                     |
| tagname   | Il nome del tag che deve essere rimosso dal campo.                                                                      |

### Example 1:

tagmap:

```
mapping LOAD * inline [
```

```
a,b
```

```
Alpha,MyTag
```

```
Num,MyTag
```

```
];
```

```
Untag fields using tagmap;
```

### Example 2:

```
Untag field Alpha with MyTag2;
```

## 3.4 Directory di lavoro

Se in un'istruzione dello script si fa riferimento a un file di cui viene omissso il percorso, Qlik Sense eseguirà la ricerca del file nell'ordine seguente:

1. La directory specificata da un'istruzione **Directory** (supportata solo nella modalità di creazione degli script legacy).

2. Se non è presente alcuna istruzione **Directory**, Qlik Sense eseguirà la ricerca nella directory di lavoro.

### Directory di lavoro Qlik Sense Desktop

In Qlik Sense Desktop, la directory di lavoro è *C:\Users\{user}\Documents\Qlik\Sense\Apps*.

### Directory di lavoro Qlik Sense

In un'installazione server di Qlik Sense, la directory di lavoro è specificata in Qlik Sense Repository Service e, per impostazione predefinita, è *C:\ProgramData\Qlik\Sense\Apps*. Per ulteriori informazioni, vedere la Guida di Console di gestione Qlik.

## 4 Utilizzo delle variabili nell'editor caricamento dati

Una variabile in Qlik Sense può essere definita come un raccoglitore che memorizza un valore statico o un calcolo, ad esempio un valore numerico o alfanumerico. Quando si utilizza la variabile nell'app, qualsiasi modifica apportata alla variabile viene applicata ovunque venga utilizzata. È possibile definire le variabili utilizzando la relativa panoramica o nello script tramite l'Editor di caricamento dati. Per impostare il valore di una variabile si utilizzano le istruzioni **Let** o **Set** nello script di caricamento dei dati.



*È inoltre possibile lavorare con le variabili di Qlik Sense dalla relativa panoramica mentre si modifica un foglio.*

### 4.1 Panoramica

Se il primo carattere del valore di una variabile è un segno di uguale '=', Qlik Sense tenta di valutare il valore come se si trattasse di una formula (espressione di Qlik Sense), quindi visualizza o restituisce il risultato anziché il testo effettivo della formula.

Quando vengono utilizzate, la variabile viene sostituita dal suo valore. Le variabili possono essere utilizzate nello script per l'espansione con simbolo del dollaro e in diverse istruzioni di controllo. Ciò si rivela molto utile se la stessa stringa viene ripetuta molte volte nello script, ad esempio un percorso.

Alcune variabili di sistema speciali vengono impostate da Qlik Sense all'inizio dell'esecuzione dello script, indipendentemente dai loro valori precedenti.

### 4.2 Definizione di una variabile

Le variabili forniscono la possibilità di memorizzare valori statici o il risultato di un calcolo. Quando si definisce una variabile, utilizzare la sintassi seguente:

```
set variablename = string
```

o

```
let variable = expression
```

L'istruzione **Set** viene utilizzata per la valutazione della stringa. Assegna il testo a destra del segno di uguale alla variabile. L'istruzione **Let** valuta un'espressione a destra del segno di uguale al momento di eseguire lo script e assegna il risultato dell'espressione alla variabile.

Le variabili sono soggette alla distinzione tra maiuscole e minuscole.



*Si sconsiglia di denominare una variabile con lo stesso nome utilizzato per un campo o una funzione in Qlik Sense.*

### Esempi:

```
set x = 3 + 4; // la variabile otterrà la stringa '3 + 4' come valore.
```

```
let x = 3 + 4; // restituisce 7 come valore.
```

```
set x = Today(); // restituisce 'Today()' come valore.
```

```
let x = Today(); // restituisce la data di oggi come valore, per esempio, '27/9/2021'.
```

## Denominazione delle variabili

Come best practice, si consiglia di utilizzare una convenzione di denominazione standardizzata per le variabili create dall'utente in un'app. Ad esempio, si può fare in modo che tutti i nomi delle variabili inizino con *v*. Ad esempio: *vUserText*. In questo modo, si garantisce che le variabili vengano riconosciute rapidamente come tali e differenziate da misure, campi e funzioni.

## 4.3 Eliminazione di una variabile

Rimuovendo una variabile dallo script e ricaricando i dati, la variabile resta nell'app. Se si desidera rimuovere completamente la variabile dall'app, è necessario eliminarla anche dalla finestra di dialogo delle variabili.

## 4.4 Caricamento di un valore della variabile come valore di campo

Se si desidera caricare un valore della variabile come valore di campo in un'istruzione **LOAD** e il risultato dell'espansione del simbolo del dollaro restituisce del testo invece che dei numeri o un'espressione, la variabile espansa deve essere racchiusa tra virgolette singole.

### Esempio:

In questo esempio, la variabile di sistema contenente l'elenco degli errori di script viene caricata in una tabella. Si può notare che l'espansione di `ScriptErrorCount` nella clausola **If** non richiede virgolette, mentre l'espansione di `ScriptErrorList` richiede le virgolette.

```
IF $(ScriptErrorCount) >= 1 THEN
    LOAD '$(ScriptErrorList)' AS Error AutoGenerate 1;
END IF
```

## 4.5 Calcolo della variabile

Esistono diversi modi per utilizzare le variabili con i valori calcolati in Qlik Sense e il risultato dipende dal metodo di definizione e dal metodo di richiamo in un'espressione.

## 4 Utilizzo delle variabili nell'editor caricamento dati

In questo esempio, sono stati caricati dei dati in linea:

```
LOAD * INLINE [  
    Dim, Sales  
    A, 150  
    A, 200  
    B, 240  
    B, 230  
    C, 410  
    C, 330  
];
```

Si definiscano due variabili:

```
Let vSales = 'Sum(Sales)' ;  
Let vSales2 = '=Sum(Sales)' ;
```

Nella seconda variabile un segno di uguale verrà aggiunto prima dell'espressione. Ciò attiverà il calcolo della variabile prima che ne venga eseguita l'espansione e prima che l'espressione venga valutata.

Se si utilizza la variabile vSales senza modifiche, ad esempio in una misura, il risultato sarà la stringa Sum (Sales), vale a dire non verrà eseguito alcun calcolo.

Se si aggiunge un'espansione con simbolo del dollaro e si richiama \$(vSales) nell'espressione, la variabile viene estesa e viene visualizzata la somma di Sales.

Infine, se si richiama \$(vSales2), la variabile verrà calcolata prima di essere espansa. Ciò significa che il risultato visualizzato è la somma totale di Sales. In questo grafico, la differenza tra l'uso di =(vSales) e =(vSales2) come espressioni di misura restituisce i risultati:

| Risultati |            |             |
|-----------|------------|-------------|
| Dim       | \$(vSales) | \$(vSales2) |
| A         | 350        | 1560        |
| B         | 470        | 1560        |
| C         | 740        | 1560        |

Come si può vedere, \$(vSales) risulta nella somma parziale di un valore di dimensione, mentre \$(vSales2) risulta nella somma totale.

Sono disponibili le seguenti variabili di script:

- [Variabili di errore \(page 290\)](#)
- [Variabili di interpretazione numerica \(page 225\)](#)
- [Variabili di sistema \(page 216\)](#)
- [Variabili di gestione del valore \(page 223\)](#)

### 4.6 Variabili di sistema

Le variabili di sistema, talvolta definite dal sistema stesso, forniscono informazioni sul sistema e sull'app Qlik Sense.

### Prospetto delle variabili di sistema

Alcune funzioni vengono ulteriormente descritte dopo la panoramica. Per tali funzioni, è inoltre possibile fare clic sul nome della funzione nella sintassi per accedere immediatamente ai dettagli per tale funzione specifica.

#### CreateSearchIndexOnReload

Questa variabile definisce se durante il ricaricamento dati devono essere creati i file dell'indice di ricerca.

CreateSearchIndexOnReload

#### Floppy

Restituisce la lettera relativa alla prima unità floppy rilevata, in genere *a:*. Questa è una variabile definita dal sistema.

Floppy



*Questa variabile non è supportata in modalità standard.*

#### CD

Restituisce la lettera relativa alla prima unità CD-ROM rilevata. Se non viene rilevata alcuna unità CD-ROM, viene restituito *c:*. Questa è una variabile definita dal sistema.

CD



*Questa variabile non è supportata in modalità standard.*

#### HidePrefix

Tutti i nomi di campo che iniziano con questa stringa di testo verranno nascosti nella stessa maniera dei campi di sistema. Questa è una variabile definita dall'utente.

HidePrefix

#### HideSuffix

Tutti i nomi di campo che finiscono con questa stringa di testo verranno nascosti nella stessa maniera dei campi di sistema. Questa è una variabile definita dall'utente.

HideSuffix

#### Include

La **Include/Must\_Include** variabile specifica un file che contiene testo che deve essere incluso nello script e valutato come codice di script. Non è utilizzato per aggiungere dati. È possibile memorizzare parti del codice di script in un file di testo separato e riutilizzarlo in diverse app. Questa è una variabile definita dall'utente.

```
$ (Include=filename)
```

## 4 Utilizzo delle variabili nell'editor caricamento dati

```
$(Must_Include=filename)
```

### OpenUrlTimeout

Questa variabile definisce l'attesa in secondi che Qlik Sense deve rispettare quando recupera i dati da sorgenti URL (ad esempio HTML pagine). Se viene omessa, l'attesa è di circa 20 minuti.

OpenUrlTimeout

### QvPath

Restituisce la stringa costituita dal percorso del file eseguibile di Qlik Sense. Questa è una variabile definita dal sistema.

QvPath



*Questa variabile non è supportata in modalità standard.*

### QvRoot

Restituisce la directory principale del file eseguibile di Qlik Sense. Questa è una variabile definita dal sistema.

QvRoot



*Questa variabile non è supportata in modalità standard.*

### QvWorkPath

Restituisce la stringa costituita dal percorso dell'app Qlik Sense attuale. Questa è una variabile definita dal sistema.

QvWorkPath



*Questa variabile non è supportata in modalità standard.*

### QvWorkRoot

Restituisce la directory principale dell'app Qlik Sense attuale. Questa è una variabile definita dal sistema.

QvWorkRoot



*Questa variabile non è supportata in modalità standard.*

### StripComments

Se questa variabile è impostata su 0, la rimozione dei commenti `/*..*/` e `//` nello script viene bloccata. Se questa variabile non è definita, la rimozione dei commenti verrà sempre eseguita.

StripComments

### Verbatim

In genere, gli spazi vuoti (ASCII 32) iniziali e finali vengono rimossi automaticamente in tutti i valori di campo prima di essere caricati nel database di Qlik Sense. L'impostazione di questa variabile su 1 sospende la rimozione dei caratteri vuoti. La tabulazione (ASCII 9) e lo spazio unificatore (ANSI 160) non vengono mai rimossi.

#### Verbatim

### WinPath

Restituisce la stringa costituita dal percorso di Windows. Questa è una variabile definita dal sistema.

#### WinPath



*Questa variabile non è supportata in modalità standard.*

### WinRoot

Restituisce la directory principale di Windows. Questa è una variabile definita dal sistema.

#### WinRoot



*Questa variabile non è supportata in modalità standard.*

### CollationLocale

Specifica le impostazioni locali da utilizzare per l'ordinamento e le corrispondenze della ricerca. Il valore corrisponde al nome della lingua di un'impostazione locale, ad esempio 'en-US'. Questa è una variabile definita dal sistema.

#### CollationLocale

## CreateSearchIndexOnReload

Questa variabile definisce se durante il ricaricamento dati devono essere creati i file dell'indice di ricerca.

### Sintassi:

#### CreateSearchIndexOnReload

È possibile definire se i file dell'indice di ricerca devono essere creati durante il ricaricamento dei dati oppure dopo la prima richiesta di ricerca fatta da un utente. Il vantaggio della creazione dei file dell'indice di ricerca durante il ricaricamento dei dati è rappresentato dall'eliminazione dei tempi di attesa per il primo utente che esegue una ricerca. D'altra parte, però, la creazione dell'indice di ricerca aumenterà i tempi di ricaricamento dei dati.

Se questa variabile è omessa, i file dell'indice di ricerca non verranno creati durante il ricaricamento dei dati.



*Per le app della sessione, i file dell'indice di ricerca non verranno creati durante il ricaricamento dei dati, indipendentemente dall'impostazione di questa variabile.*



*Le nuove app includono `set CreateSearchIndexOnReload=1` nelle istruzioni predefinite SET. Quando si creano script nel centro attività Analisi, le espressioni SET non includono `CreateSearchIndexOnReload` poiché le risorse di script non forniscono dati persistenti, rendendo superfluo un valore di indice.*

### Example 1: Creazione dei campi dell'indice di ricerca durante il ricaricamento dei dati

```
set CreateSearchIndexOnReload=1;
```

### Example 2: Creazione dei campi dell'indice di ricerca dopo la prima richiesta di ricerca

```
set CreateSearchIndexOnReload=0;
```

## HidePrefix

Tutti i nomi di campo che iniziano con questa stringa di testo verranno nascosti nella stessa maniera dei campi di sistema. Questa è una variabile definita dall'utente.

### Sintassi:

**HidePrefix**

### Esempio:

```
set HidePrefix='_ ' ;
```

Se si utilizza questa istruzione, i nomi di campo che iniziano con un carattere di sottolineatura non vengono visualizzati negli elenchi dei nomi di campo quando i campi del sistema sono nascosti.

## HideSuffix

Tutti i nomi di campo che finiscono con questa stringa di testo verranno nascosti nella stessa maniera dei campi di sistema. Questa è una variabile definita dall'utente.

### Sintassi:

**HideSuffix**

### Esempio:

```
set HideSuffix='% ' ;
```

Se si utilizza questa istruzione, i nomi di campo che terminano con un segno percentuale non vengono visualizzati nell'elenco dei nomi di campo quando i campi del sistema sono nascosti.

### Include

La **Include/Must\_Include** variabile specifica un file che contiene testo che deve essere incluso nello script e valutato come codice di script. Non è utilizzato per aggiungere dati. È possibile memorizzare parti del codice di script in un file di testo separato e riutilizzarlo in diverse app. Questa è una variabile definita dall'utente.



*In Qlik Sense e Qlik Sense Desktop, questa variabile supporta solo i riferimenti alle connessioni dati della cartella. Non sono supportati i riferimenti a file nei provider di archiviazione basati sul cloud.*

#### Sintassi:

```
$(Include=filename)
```

```
$(Must_Include=filename)
```

Esistono due versioni della variabile:

- **Include** non genera un errore quando non viene individuato il file e non visualizza alcun messaggio.
- **Must\_Include** genera un errore quando non viene individuato il file.

Se non si specifica un percorso, il nome del file sarà relativo rispetto alla directory di lavoro dell'app Qlik Sense. È inoltre possibile specificare un percorso del file assoluto o un percorso della connessione della cartella lib://. Non inserire un carattere di spazio prima o dopo il segno di uguale.



*La costruzione **set Include =filename** non è applicabile.*

#### Esempi:

```
$(Include=abc.txt);
```

```
$(Must_Include=lib://DataFiles/abc.txt);
```

### Limitazioni

Compatibilità incrociata limitata tra i file codificati UTF-8 sotto Windows rispetto a Linux.

In via opzionale è possibile usare UTF-8 con il BOM (Byte Order Mark). Il BOM può interferire con l'uso di UTF-8 nei software che non si aspettano byte non-ASCII all'inizio di un file, ma che potrebbe altrimenti gestire il flusso di testo.

- I sistemi Windows usano il BOM in UTF-8 per identificare che un file è codificato in UTF-8, nonostante non ci sia ambiguità nella memorizzazione dei byte.
- Unix/Linux usa UTF-8 per Unicode, ma non usano il BOM perché ciò interferisce con la sintassi dei file di comando.

## 4 Utilizzo delle variabili nell'editor caricamento dati

Sono presenti alcune implicazioni per Qlik Sense.

- In Windows qualsiasi file che inizia con un BOM UTF-8 è considerato un file di script UTF-8. Altrimenti si presume l'uso della codifica ANSI.
- In Linux, la pagina di codice a 8 bit predefinita del sistema è UTF-8. È per questo che UTF-8 funziona anche se non contiene il BOM.

Di conseguenza, la portabilità non può essere garantita. Non è sempre possibile creare un file su Windows che possa essere interpretato da Linux e viceversa. Non esiste alcuna compatibilità incrociata tra i due sistemi per quanto riguarda i file codificati UTF-8, a causa della diversa gestione del BOM.

### OpenUrlTimeout

Questa variabile definisce l'attesa in secondi che Qlik Sense deve rispettare quando recupera i dati da sorgenti URL (ad esempio HTML pagine). Se viene omessa, l'attesa è di circa 20 minuti.

**Sintassi:**

```
OpenUrlTimeout
```

**Esempio:**

```
set OpenUrlTimeout=10;
```

### StripComments

Se questa variabile è impostata su 0, la rimozione dei commenti `/*..*/` e `//` nello script viene bloccata. Se questa variabile non è definita, la rimozione dei commenti verrà sempre eseguita.

**Sintassi:**

```
StripComments
```

Alcuni driver del database utilizzano `/*..*/` come suggerimento di ottimizzazione nelle istruzioni **SELECT**. In questo caso, non rimuovere i commenti prima di inviare l'istruzione **SELECT** al driver del database.



*È consigliabile reimpostare questa variabile su 1 subito dopo l'istruzione o le istruzioni per le quali è stata necessaria.*

**Esempio:**

```
set StripComments=0;  
SQL SELECT * /* <optimization directive> */ FROM Table ;  
set StripComments=1;
```

### Verbatim

In genere, gli spazi vuoti (ASCII 32) iniziali e finali vengono rimossi automaticamente in tutti i valori di campo prima di essere caricati nel database di Qlik Sense. L'impostazione di questa variabile su 1 sospende la rimozione dei caratteri vuoti. La tabulazione (ASCII 9) e lo spazio unificatore (ANSI 160) non vengono mai rimossi.

#### Sintassi:

```
Verbatim
```

#### Esempio:

```
set Verbatim = 1;
```

## 4.7 Variabili di gestione del valore

In questa sezione vengono descritte le variabili utilizzate per gestire il valore NULL e gli altri valori.

### Prospetto delle variabili di gestione del valore

Ciascuna funzione viene descritta ulteriormente dopo la panoramica. È inoltre possibile fare clic sul nome della funzione nella sintassi per accedere immediatamente ai dettagli per tale funzione specifica.

#### NullDisplay

Il simbolo definito sostituisce tutti i valori NULL da ODBC e i connettori al livello di dati più basso. Questa è una variabile definita dall'utente.

```
NullDisplay
```

#### NullInterpret

Quando il simbolo definito compare in un file di testo, un file CSV o in un'istruzione inline, viene interpretato come NULL. Questa è una variabile definita dall'utente.

```
NullInterpret
```

#### NullValue

Se viene utilizzata l'istruzione **NullAsValue**, il simbolo definito sostituirà tutti i valori NULL nei campi **NullAsValue** specificati con la stringa specificata.

```
NullValue
```

#### OtherSymbol

Fa in modo che un simbolo venga trattato come 'tutti gli altri valori' prima di un'istruzione **LOAD/SELECT**. Questa è una variabile definita dall'utente.

```
OtherSymbol
```

### NullDisplay

Il simbolo definito sostituisce tutti i valori NULL da ODBC e i connettori al livello di dati più basso. Questa è una variabile definita dall'utente.

#### Sintassi:

```
NullDisplay
```

#### Esempio:

```
set NullDisplay='<NULL>';
```

### NullInterpret

Quando il simbolo definito compare in un file di testo, un file CSV o in un'istruzione inline, viene interpretato come NULL. Questa è una variabile definita dall'utente.

#### Sintassi:

```
NullInterpret
```

#### Esempi:

```
set NullInterpret=' ';
```

```
set NullInterpret =;
```

```
set NullInterpret ='';
```

### NullValue

Se viene utilizzata l'istruzione **NullAsValue**, il simbolo definito sostituirà tutti i valori NULL nei campi **NullAsValue** specificati con la stringa specificata.

#### Sintassi:

```
NullValue
```

#### Esempio:

```
NullAsValue Field1, Field2;  
set NullValue='<NULL>';
```

### OtherSymbol

Fa in modo che un simbolo venga trattato come 'tutti gli altri valori' prima di un'istruzione **LOAD/SELECT**. Questa è una variabile definita dall'utente.

#### Sintassi:

```
OtherSymbol
```

### Esempio:

```
set OthersSymbol='+';  
LOAD * inline  
[X, Y  
a, a  
b, b];  
LOAD * inline  
[X, Z  
a, a  
+, c];
```

Il valore di campo Y='b' sarà ora collegato al valore Z='c' mediante l'altro simbolo.

## 4.8 Variabili di interpretazione numerica

Le variabili di interpretazione del numero sono definite dal sistema. Le variabili sono incluse all'inizio dello script di caricamento e applicano le impostazioni di formattazione dei numeri al momento dell'esecuzione dello script. Possono essere eliminate, modificate o duplicate.

Le variabili di interpretazione numerica sono generate automaticamente in base alle impostazioni regionali attuali del sistema operativo al momento della creazione di una nuova app. In Qlik Sense Desktop, questo dipende dalle impostazioni del sistema operativo del computer. In Qlik Sense riflettono le impostazioni del sistema operativo del server dove è installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Formattazione valuta

#### MoneyDecimalSep

Il separatore dei decimali specificato sostituisce il simbolo decimale per la valuta impostato nelle impostazioni locali.

[MoneyDecimalSep](#)

#### MoneyFormat

Il simbolo specificato sostituisce il simbolo della valuta impostato nelle impostazioni locali.

[MoneyFormat](#)

#### MoneyThousandSep

Il separatore delle migliaia specificato sostituisce il simbolo di raggruppamento delle cifre per la valuta configurata nelle impostazioni locali.

[MoneyThousandSep](#)

### Formattazione numero

#### DecimalSep

Il separatore dei decimali specificato sostituisce il simbolo decimale impostato nelle impostazioni locali.

[DecimalSep](#)

#### ThousandSep

Il separatore delle migliaia specificato sostituisce il simbolo di raggruppamento delle cifre del sistema operativo (impostazioni locali).

[ThousandSep](#)

#### NumericalAbbreviation

L'abbreviazione numerica definisce l'abbreviazione da utilizzare per i prefissi in scala dei numeri, ad esempio M per mega o un milione ( $10^6$ ) e  $\mu$  per micro ( $10^{-6}$ ).

[NumericalAbbreviation](#)

### Formattazione dell'ora

#### DateFormat

Questa variabile d'ambiente definisce il formato di data utilizzato come predefinito nell'app. Il formato viene utilizzato sia per interpretare che per formattare le date. Se la variabile non è definita, il formato della data delle impostazioni regionali del sistema operativo sarà recuperato quando lo script viene eseguito.

[DateFormat](#)

#### TimeFormat

Il formato specificato sostituisce il formato dell'ora del sistema operativo (impostazioni locali).

[TimeFormat](#)

#### TimestampFormat

Il formato specificato sostituisce il formato della data e dell'ora del sistema operativo (impostazioni locali).

[TimestampFormat](#)

#### MonthNames

Il formato specificato sostituisce la convenzione usata per i nomi dei mesi nelle impostazioni locali.

[MonthNames](#)

#### LongMonthNames

Il formato specificato sostituisce la convenzione usata per i nomi lunghi dei mesi nelle impostazioni locali.

[LongMonthNames](#)

#### DayNames

Il formato specificato sostituisce la convenzione usata per i nomi dei giorni della settimana configurata nelle impostazioni locali.

## 4 Utilizzo delle variabili nell'editor caricamento dati

### DayNames

#### LongDayNames

Il formato specificato sostituisce la convenzione usata per i nomi lunghi dei giorni della settimana nelle impostazioni locali.

### LongDayNames

#### FirstWeekDay

Numero intero che definisce il giorno da utilizzare come primo giorno della settimana.

### FirstWeekDay

#### BrokenWeeks

Questa impostazione definisce se le settimane sono interrotte o meno.

### BrokenWeeks

#### ReferenceDay

L'impostazione definisce il giorno del mese di gennaio da impostare come giorno di riferimento per definire la settimana 1.

### ReferenceDay

#### FirstMonthOfYear

L'impostazione definisce il mese da utilizzare come primo mese dell'anno, un'opzione utile per definire gli anni finanziari che fanno uso di un offset per i mesi, iniziando ad esempio il 1 aprile.



*Attualmente questa impostazione non viene utilizzata, tuttavia è stata riservata per un uso futuro.*

Le impostazioni valide vanno dal 1 (gennaio) al 12 (dicembre). L'impostazione predefinita è 1.

#### Sintassi:

### FirstMonthOfYear

#### Esempio:

```
Set FirstMonthOfYear=4; //Sets the year to start in April
```

## BrokenWeeks

Questa impostazione definisce se le settimane sono interrotte o meno.

#### Sintassi:

### BrokenWeeks

In Qlik Sense, le impostazioni regionali vengono recuperate alla creazione dell'app e le impostazioni corrispondenti vengono memorizzate nello script come variabili d'ambiente.

---

## 4 Utilizzo delle variabili nell'editor caricamento dati

---

Uno sviluppatore di app nordamericano ottiene spesso `set BrokenWeeks=1`; nello script, corrispondente a settimane interrotte. Uno sviluppatore di app europeo ottiene spesso `set BrokenWeeks=0`; nello script, corrispondente a settimane intere.

Settimane ininterrotte significa che:

- In alcuni anni, la settimana 1 inizia a dicembre, mentre in altri anni l'ultima settimana dell'anno precedente continua a gennaio.
- Secondo la norma ISO 8601, la settimana 1 ha sempre almeno 4 giorni in gennaio. In Qlik Sense, questo può essere configurato utilizzando la variabile `ReferenceDay`.

Settimane interrotte significa che:

- L'ultima settimana dell'anno non prosegue mai a gennaio.
- La settimana 1 inizia il primo gennaio e, nella maggior parte dei casi, non è una settimana completa.

È possibile utilizzare i seguenti valori:

- 0 (= utilizzo di settimane intere)
- 1 (= utilizzo di settimane suddivise)

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempi:

Se si desiderano le impostazioni ISO per le settimane e i numeri di settimana, assicurarsi di inserire nello script quanto segue:

```
set FirstWeekDay=0;
set BrokenWeeks=0;    //(use unbroken weeks)
set ReferenceDay=4;
```

Se si desiderano le impostazioni USA, assicurarsi che nello script sia presente quanto segue:

```
set FirstWeekDay=6;
set BrokenWeeks=1;    //(use broken weeks)
set ReferenceDay=1;
```

### DateFormat

Questa variabile di ambiente definisce il formato della data utilizzato come predefinito nell'app e per data che restituisce funzioni come `date()` e `date#()`. Il formato viene utilizzato per interpretare e formattare le date. Se la variabile non è definita, il formato della data delle impostazioni regionali sarà recuperato quando viene eseguito lo script.

#### Sintassi:

##### DateFormat

##### Esempi di funzioni DateFormat

| Esempio                                                              | Risultato                                                                                                                           |
|----------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| <pre>Set DateFormat='M/D/YY'; //<br/>(US format)</pre>               | Quando viene utilizzata in questo modo, la funzione <code>DateFormat</code> definisce la data con il formato USA, mese/giorno/anno. |
| <pre>Set DateFormat='DD/MM/YY';<br/>//(UK date format)</pre>         | Quando viene utilizzata in questo modo, la funzione <code>DateFormat</code> definisce la data con il formato GB, giorno/mese/anno.  |
| <pre>Set<br/>DateFormat='YYYY/MM/DD'; //<br/>(ISO date format)</pre> | Quando viene utilizzata in questo modo, la funzione <code>DateFormat</code> definisce la data con il formato ISO, anno/mese/giorno. |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - Variabili di sistema predefinite

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

---

## 4 Utilizzo delle variabili nell'editor caricamento dati

---

- Un set di dati di date.
- La funzione `DateFormat`, che utilizzerà il formato data USA.

Nel seguente esempio, un set di dati è caricato in una tabella denominata 'Transactions'. Include un campo `date`. Viene utilizzata la definizione USA `DateFormat`. Questo modello verrà utilizzato per il testo implicito per la conversione data quando vengono caricate date di testo.

### Script di caricamento

```
Set DateFormat='MM/DD/YYYY';
```

```
Transactions:
LOAD
date,
month(date) as month,
id,
amount
INLINE
[
date,id,amount
01/01/2022,1,1000
02/01/2022,2,2123
03/01/2022,3,4124
04/01/2022,4,2431
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `date`
- `month`

Creare questa misura:

```
=sum(amount)
```

Tabella dei risultati

| date       | mese | =sum(amount) |
|------------|------|--------------|
| 01/01/2022 | Jan  | 1000         |
| 02/01/2022 | Feb  | 2123         |
| 03/01/2022 | Mar  | 4124         |
| 04/01/2022 | Apr  | 2431         |

La definizione `DateFormat MM/GG/AAAA` è usata per la conversione implicita del testo in date, per cui il campo `date` è correttamente interpretato come una data. Lo stesso formato è utilizzato per visualizzare la data, come mostrato nella tabella dei risultati.

### Esempio 2 - Modifica della variabile di sistema

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Lo stesso set di dati dall'esempio precedente.
- La funzione `DateFormat`, che utilizzerà il formato 'GG/MM/AAAA'.

#### Script di caricamento

```
SET DateFormat='DD/MM/YYYY';
Transactions:
LOAD
date,
month(date) as month,
id,
amount
INLINE
[
date,id,amount
01/01/2022,1,1000
02/01/2022,2,2123
03/01/2022,3,4124
04/01/2022,4,2431
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `date`
- `month`

Creare questa misura:

```
=sum(amount)
```

Tabella dei risultati

| date       | mese | =sum(amount) |
|------------|------|--------------|
| 01/01/2022 | Jan  | 1000         |
| 02/01/2022 | Jan  | 2123         |
| 03/01/2022 | Jan  | 4124         |
| 04/01/2022 | Jan  | 2431         |

---

## 4 Utilizzo delle variabili nell'editor caricamento dati

---

Poiché la definizione di `dateFormat` è stata impostata su 'GG/MM/AAAA', si può notare che le due cifre dopo il primo simbolo "/" sono state interpretate come il mese, con il risultato che tutti i record sono del mese di gennaio.

### Esempio 3 - Interpretazione delle date

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati con date in formato numerico.
- La variabile `dateFormat`, che utilizzerà il formato 'GG/MM/AAAA'.
- La variabile `date()`.

#### Script di caricamento

```
SET dateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
date(numerical_date),
```

```
month(date(numerical_date)) as month,
```

```
id,
```

```
amount
```

```
Inline
```

```
[
```

```
numerical_date,id,amount
```

```
43254,1,1000
```

```
43255,2,2123
```

```
43256,3,4124
```

```
43258,4,2431
```

```
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- month

Creare questa misura:

```
=sum(amount)
```

## 4 Utilizzo delle variabili nell'editor caricamento dati

---

Tabella dei risultati

| date       | mese | =sum(amount) |
|------------|------|--------------|
| 06/03/2022 | Jun  | 1000         |
| 06/04/2022 | Jun  | 2123         |
| 06/05/2022 | Jun  | 4124         |
| 06/07/2022 | Jun  | 2431         |

Nello script di caricamento, utilizzare la funzione `date()` per convertire la data numerica in un formato data. Dato che non si fornisce un formato specificato come argomento secondario nella funzione, viene utilizzato `DateFormat`. Ciò fa sì che il campo data utilizzi il formato 'MM/GG/AAAA'.

### Esempio 4 - Formattazione di una data straniera

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati di date.
- La variabile `DateFormat`, che utilizza il formato 'GG/MM/AAAA' ma non presenta commenti con barre terminali.

#### Script di caricamento

```
// SET DateFormat='DD/MM/YYYY';
```

```
Transactions:
Load
date,
month(date) as month,
id,
amount
Inline
[
date,id,amount
22-05-2022,1,1000
23-05-2022,2,2123
24-05-2022,3,4124
25-05-2022,4,2431
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

## 4 Utilizzo delle variabili nell'editor caricamento dati

- date
- month

Creare questa misura:

```
=sum(amount)
```

Tabella dei risultati

| date       | mese | =sum(amount) |
|------------|------|--------------|
| 22-05-2022 | -    | 1000         |
| 23-05-2022 | -    | 2123         |
| 24-05-2022 | -    | 4124         |
| 25-05-2022 | -    | 2431         |

Nello script di caricamento iniziale, il `DateFormat` in uso è il formato predefinito 'MM/GG/AAAA'. Poiché il campo `date` nel set di dati delle transazioni non è in questo formato, il campo non viene interpretato come una data. Ciò è mostrato nella tabella dei risultati in cui i valori del campo `month` sono NULL.

È possibile verificare i tipi di dati interpretati nel sistema di visualizzazione del modello dati verificando le proprietà "Tag" del campo `date`.

*Anteprima della tabella Transactions. Notare i "Tag" per il campo date che indicano che i dati di input testuali non sono stati convertiti implicitamente in una data/data e ora.*

| date                    |               | Transactions |       |    |        |
|-------------------------|---------------|--------------|-------|----|--------|
| Density                 | 100%          | date         | month | id | amount |
| Subset ratio            | 100%          | 22-05-2022   | -     | 1  | 1000   |
| Has duplicates          | false         | 23-05-2022   | -     | 2  | 2123   |
| Total distinct values   | 4             | 24-05-2022   | -     | 3  | 4124   |
| Present distinct values | 4             | 25-05-2022   | -     | 4  | 2431   |
| Non-null values         | 4             |              |       |    |        |
| Tags                    | Sascii \$text |              |       |    |        |

Questo problema può essere risolto abilitando la variabile di sistema `DateFormat` :

```
// SET DateFormat='DD/MM/YYYY';
```

Rimuovere le doppie barre terminali e ricaricare i dati.

## 4 Utilizzo delle variabili nell'editor caricamento dati

Anteprima della tabella *Transactions*. Notare i "Tag" per il campo *date* che indicano che i dati di input testuali sono stati convertiti implicitamente in una data/data e ora.

| date                    |                                        | Transactions |       |    |        |
|-------------------------|----------------------------------------|--------------|-------|----|--------|
| Density                 | 100%                                   | date         | month | id | amount |
| Subset ratio            | 100%                                   | 22-05-2022   | May   | 1  | 1000   |
| Has duplicates          | false                                  | 23-05-2022   | May   | 2  | 2123   |
| Total distinct values   | 4                                      | 24-05-2022   | May   | 3  | 4124   |
| Present distinct values | 4                                      | 25-05-2022   | May   | 4  | 2431   |
| Non-null values         | 4                                      |              |       |    |        |
| Tags                    | \$numeric \$integer \$timestamp \$date |              |       |    |        |

### DayNames

Il formato specificato sostituisce la convenzione usata per i nomi dei giorni della settimana configurata nelle impostazioni locali.

Sintassi:

#### DayNames

Quando si modifica la variabile, è necessario inserire un punto e virgola ; per separare i singoli valori.

Esempi della funzione DayName

#### Esempio di funzione

Set

DayNames='Mon;Tue;Wed;Thu;Fri;Sat;Sun';

Set DayNames='M;Tu;W;Th;F;Sa;Su';

#### Definizione del risultato

La funzione DayNames consente di definire i nomi dei giorni nella loro forma abbreviata.

La funzione DayNames consente di definire i nomi dei giorni utilizzando le lettere iniziali.

La funzione DayNames viene spesso utilizzata in combinazione con le seguenti funzioni:

Funzioni correlate

#### Funzione

#### Interazione

[weekday \(page 1184\)](#)

Funzione script per restituire DayNames come valori di campo.

[Date \(page 1378\)](#)

Funzione script per restituire DayNames come valori di campo.

[LongDayNames \(page 246\)](#)

Valori nel formato lungo di DayNames.

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione SET DateFormat nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema,

## 4 Utilizzo delle variabili nell'editor caricamento dati

---

a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - Variabili di sistema predefinite

Script di caricamento e risultati

#### Panoramica

In questo esempio, le date nel set di dati sono impostate nel formato MM/GG/AAAA.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati con date, che verrà caricato in una tabella denominata `Transactions`.
- Un campo `date`.
- La definizione predefinita `DayNames`.

#### Script di caricamento

```
SET DayNames='Mon;Tue;Wed;Thu;Fri;Sat;Sun';
```

```
Transactions:
LOAD
date,
weekDay(date) as dayname,
id,
amount
INLINE
[
date,id,amount
01/01/2022,1,1000
02/01/2022,2,2123
03/01/2022,3,4124
04/01/2022,4,2431
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

---

## 4 Utilizzo delle variabili nell'editor caricamento dati

---

- date
- dayname

Creare questa misura:

```
sum(amount)
```

Tabella dei risultati

| date       | dayname | sum(amount) |
|------------|---------|-------------|
| 01/01/2022 | Sat     | 1000        |
| 02/01/2022 | Tue     | 2123        |
| 03/01/2022 | Tue     | 4124        |
| 04/01/2022 | Fri     | 2431        |

Nello script di caricamento, la funzione `weekDay` viene utilizzata con il campo `date` come argomento fornito. Nella tabella dei risultati, l'output di questa funzione `weekDay` visualizza i giorni della settimana nel formato della definizione `DayNames`.

### Esempio 2 - Modifica della variabile di sistema

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor di caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda. Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, all'inizio dello script, la definizione `DayNames` viene modificata per utilizzare i giorni abbreviati della settimana in afrikaans.

#### Script di caricamento

```
SET DayNames='Ma;Di;Wo;Do;Vr;Sa;So';
```

```
Transactions:
```

```
Load
```

```
date,
```

```
weekDay(date) as dayname,
```

```
id,
```

```
amount
```

```
InLine
```

```
[
```

```
date,id,amount
```

```
01/01/2022,1,1000
```

```
02/01/2022,2,2123
```

```
03/01/2022,3,4124
```

```
04/01/2022,4,2431
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- dayname

Creare questa misura:

```
sum(amount)
```

| Tabella dei risultati |         |             |
|-----------------------|---------|-------------|
| date                  | dayname | sum(amount) |
| 01/01/2022            | Sa      | 1000        |
| 02/01/2022            | Di      | 2123        |
| 03/01/2022            | Di      | 4124        |
| 04/01/2022            | Vr      | 2431        |

Nella tabella dei risultati, l'output di questa funzione `weekDay` visualizza i giorni della settimana nel formato della definizione `DayNames`.

È importante ricordare che se la lingua per `DayNames` viene modificata come in questo esempio, `LongDayNames` conterrà comunque i giorni della settimana in inglese. È necessario modificare anche questo valore se entrambe le variabili vengono utilizzate nell'applicazione.

### Esempio 3 - Funzione data

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati con `date`, che verrà caricato in una tabella denominata `Transactions`.
- Un campo `date`.
- La definizione predefinita `DayNames`.

#### Script di caricamento

```
SET DayNames='Mon;Tue;Wed;Thu;Fri;Sat;Sun';
```

```
Transactions:
```

```
Load  
date,  
Date(date,'www') as dayname,  
id,
```

## 4 Utilizzo delle variabili nell'editor caricamento dati

---

```
amount
inline
[
date,id,amount
01/01/2022,1,1000
02/01/2022,2,2123
03/01/2022,3,4124
04/01/2022,4,2431
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- dayname

Creare questa misura:

```
sum(amount)
```

Tabella dei risultati

| date       | dayname | sum(amount) |
|------------|---------|-------------|
| 01/01/2022 | Sat     | 1000        |
| 02/01/2022 | Tue     | 2123        |
| 03/01/2022 | Tue     | 4124        |
| 04/01/2022 | Fri     | 2431        |

Viene utilizzata la definizione predefinita `DayNames`. Nello script di caricamento, la funzione `date` viene utilizzata con il campo `date` come primo argomento. Il secondo argomento è `www`. Questa formattazione converte il risultato nei valori archiviati nella definizione `DayNames`. Quest'ultimo viene visualizzato nell'output della tabella dei risultati.

### DecimalSep

Il separatore dei decimali specificato sostituisce il simbolo decimale impostato nelle impostazioni locali.

Qlik Sense interpreta automaticamente il testo come numeri ogni volta che rileva uno schema numerico riconoscibile. Le variabili di sistema `ThousandSep` e `DecimalSep` determinano la composizione dei modelli applicati durante l'analisi del testo come numeri. Le variabili `ThousandSep` e `DecimalSep` impostano il modello di formattazione numerico predefinito durante la visualizzazione di contenuti numerici nei grafici e nelle tabelle front-end. Quindi, ha un impatto diretto sulle opzioni di **Formattazione numero** per qualsiasi espressione front-end.

L'adozione di un separatore delle migliaia di virgola ',' e un separatore decimale di '.', sono esempi di modelli che verrebbero convertiti implicitamente in valori numerici equivalenti:

```
0,000.00
```

0000.00

0,000

Questi sono esempi di modelli che rimarrebbero invariati come testo; cioè, non verrebbero convertiti in un valore numerico:

0.000,00

0,00

### Sintassi:

#### DecimalSep

#### Esempi di funzioni

| Esempio             | Risultato                             |
|---------------------|---------------------------------------|
| Set DecimalSep='.'; | Imposta '.' come separatore decimale. |
| Set DecimalSep=','; | Imposta ',' come separatore decimale. |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio - Effetto dell'impostazione delle variabili di separazione dei numeri su dati di input diversi

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati di somme e date con le somme impostate in modelli di formato differenti.
- Una tabella denominata `Transactions`.
- La variabile di sistema `DecimalSep` che è impostata su '.'.

## 4 Utilizzo delle variabili nell'editor caricamento dati

---

- La variabile di sistema ThousandSep che è impostata su ','.
- La variabile delimiter che è impostata come carattere '|' per separare i diversi campi in una riga.

### Script di caricamento

```
Set ThousandSep=',';
Set DecimalSep='.';

Transactions:
Load date,
id,
amount as amount
Inline
[
date|id|amount
01/01/2022|1|1.000-45
01/02/2022|2|23.344
01/03/2022|3|4124,35
01/04/2022|4|2431.36
01/05/2022|5|4,787
01/06/2022|6|2431.84
01/07/2022|7|4132.5246
01/08/2022|8|3554.284
01/09/2022|9|3.756,178
01/10/2022|10|3,454.356
] (delimiter is '|');
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: amount.

Creare questa misura:

=sum(amount)

| Tabella dei risultati |              |            |
|-----------------------|--------------|------------|
| Importo               | =Sum(amount) |            |
| Totali                |              | 20814.7086 |
| 1.000-45              |              |            |
| 3.756,178             |              |            |
| 4124,35               |              |            |
|                       | 23.344       | 23.344     |
|                       | 2431.36      | 2431.36    |
|                       | 2431.84      | 2431.84    |
|                       | 3,454.356    | 3454.356   |

## 4 Utilizzo delle variabili nell'editor caricamento dati

| Importo | =Sum(amount) |           |
|---------|--------------|-----------|
|         | 3554.284     | 3554.284  |
|         | 4132.5246    | 4132.5246 |
|         | 4,787        | 4787      |

Qualsiasi valore non interpretato come numero viene trattato come testo ed è allineato a sinistra per impostazione predefinita. Tutti i valori convertiti correttamente vengono allineati a destra, mantenendo il formato di input originale.

La colonna dell'espressione mostra l'equivalente numerico, che per impostazione predefinita è formattato solo con un separatore decimale '.'. È possibile modificare la formattazione tramite l'impostazione del menu a discesa **Formattazione numero** nella configurazione dell'espressione.

### FirstWeekDay

Numero intero che definisce il giorno da utilizzare come primo giorno della settimana.

#### Sintassi:

##### FirstWeekDay

Il lunedì è il primo giorno della settimana secondo la norma ISO 8601, lo standard internazionale per la rappresentazione di date e orari. Il lunedì è anche utilizzato come primo giorno della settimana in diversi Paesi, ad esempio nel Regno Unito, in Francia, in Germania e in Svezia.

Ma in altri Paesi, come gli Stati Uniti e il Canada, la domenica è considerata l'inizio della settimana.

In Qlik Sense, le impostazioni locali vengono recuperate alla creazione dell'app e le impostazioni corrispondenti vengono memorizzate nello script come variabili d'ambiente.

Uno sviluppatore di app nordamericano ottiene spesso `set FirstWeekDay=6;` nello script, corrispondente alla domenica. Uno sviluppatore di app europeo ottiene spesso `set FirstWeekDay=0;` nello script, corrispondente al lunedì.

Valori impostabili per  
FirstWeekDay

| Valore | Giorno    |
|--------|-----------|
| 0      | Lunedì    |
| 1      | Martedì   |
| 2      | Mercoledì |
| 3      | Giovedì   |
| 4      | Venerdì   |
| 5      | Sabato    |
| 6      | Domenica  |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempi:

Se si desiderano le impostazioni ISO per le settimane e i numeri di settimana, assicurarsi di inserire nello script quanto segue:

```
Set FirstWeekDay=0; // Monday as first week day
Set BrokenWeeks=0;
Set ReferenceDay=4;
```

Se si desiderano le impostazioni USA, assicurarsi che nello script sia presente quanto segue:

```
Set FirstWeekDay=6; // Sunday as first week day
Set BrokenWeeks=1;
Set ReferenceDay=1;
```

### Esempio 1: utilizzo del valore predefinito (script)

Script di caricamento e risultati

#### Panoramica

Aprire Editor caricamento dati, quindi aggiungere lo script di caricamento in basso in una nuova scheda.

In questo esempio, lo script di caricamento utilizzare il valore della variabile di sistema Qlik Sense predefinita, `FirstWeekDay=6`. Questi dati contengono i dati per i primi 14 giorni nel 2020.

#### Script di caricamento

```
// Example 1: Load Script using the default value of FirstWeekDay=6, i.e. Sunday
```

```
SET FirstWeekDay = 6;
```

```
Sales:
```

```
LOAD
    date,
    sales,
    week(date) as week,
```

## 4 Utilizzo delle variabili nell'editor caricamento dati

---

```
weekday(date) as weekday  
Inline [  
date,sales  
01/01/2021,6000  
01/02/2021,3000  
01/03/2021,6000  
01/04/2021,8000  
01/05/2021,5000  
01/06/2020,7000  
01/07/2020,3000  
01/08/2020,5000  
01/09/2020,9000  
01/10/2020,5000  
01/11/2020,7000  
01/12/2020,7000  
01/13/2020,7000  
01/14/2020,7000  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- week
- weekday

Tabella dei risultati

| Data       | settimana | weekday |
|------------|-----------|---------|
| 01/01/2021 | 1         | Mer     |
| 01/02/2021 | 1         | Gio     |
| 01/03/2021 | 1         | Fri     |
| 01/04/2021 | 1         | Sat     |
| 01/05/2021 | 2         | Sun     |
| 01/06/2020 | 2         | Mon     |
| 01/07/2020 | 2         | Tue     |
| 01/08/2020 | 2         | Wed     |
| 01/09/2020 | 2         | Thu     |
| 01/10/2020 | 2         | Fri     |
| 01/11/2020 | 2         | Sat     |
| 01/12/2020 | 3         | Dom     |
| 01/13/2020 | 3         | Lun     |
| 01/14/2020 | 3         | Mar     |

---

## 4 Utilizzo delle variabili nell'editor caricamento dati

---

Dato che vengono utilizzate le impostazioni predefinite, la variabile di sistema `FirstWeekDay` è impostata su 6. Nella tabella dei risultati, ciascuna nuova settimana può essere visualizzata iniziando dalla domenica (il 5 e il 12 di gennaio).

### Esempio 2 - Cambiamento della variabile `FirstWeekDay` (script)

Script di caricamento e risultati

#### Panoramica

Aprire Editor caricamento dati, quindi aggiungere lo script di caricamento in basso in una nuova scheda.

In questo esempio, i dati contengono i primi 14 giorni nel 2020. All'inizio dello script, abbiamo impostato la variabile `FirstWeekDay` a 3.

#### Script di caricamento

```
// Example 2: Load Script setting the value of FirstWeekDay=3, i.e. Thursday
```

```
SET FirstWeekDay = 3;
```

```
Sales:
```

```
LOAD
```

```
    date,  
    sales,  
    week(date) as week,  
    weekday(date) as weekday
```

```
Inline [
```

```
date,sales
```

```
01/01/2021,6000
```

```
01/02/2021,3000
```

```
01/03/2021,6000
```

```
01/04/2021,8000
```

```
01/05/2021,5000
```

```
01/06/2020,7000
```

```
01/07/2020,3000
```

```
01/08/2020,5000
```

```
01/09/2020,9000
```

```
01/10/2020,5000
```

```
01/11/2020,7000
```

```
01/12/2020,7000
```

```
01/13/2020,7000
```

```
01/14/2020,7000
```

```
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- week
- weekday

Tabella dei risultati

| Data       | settimana | weekday |
|------------|-----------|---------|
| 01/01/2021 | 52        | Mer     |
| 01/02/2021 | 1         | Gio     |
| 01/03/2021 | 1         | Fri     |
| 01/04/2021 | 1         | Sat     |
| 01/05/2021 | 1         | Sun     |
| 01/06/2020 | 1         | Mon     |
| 01/07/2020 | 1         | Tue     |
| 01/08/2020 | 1         | Wed     |
| 01/09/2020 | 2         | Thu     |
| 01/10/2020 | 2         | Fri     |
| 01/11/2020 | 2         | Sat     |
| 01/12/2020 | 2         | Dom     |
| 01/13/2020 | 2         | Lun     |
| 01/14/2020 | 2         | Mar     |

Dato che la variabile di sistema `Firstweekday` è impostata su 3, il primo giorno di ogni settimana sarà un giovedì. Nella tabella dei risultati, ciascuna nuova settimana può essere vista iniziare il giovedì (il 2 e il 9 di gennaio).

### LongDayNames

Il formato specificato sostituisce la convenzione usata per i nomi lunghi dei giorni della settimana nelle impostazioni locali.

#### Sintassi:

##### **LongDayNames**

Il seguente esempio della funzione `LongDayNames` definisce i nomi dei giorni per intero:

```
Set LongDayNames='Monday;Tuesday;Wednesday;Thursday;Friday;Saturday;Sunday';
```

Quando si modifica la variabile, è necessario inserire un punto e virgola ; per separare i singoli valori.

La funzione `LongDayNames` può essere utilizzata in combinazione con la funzione [Date \(page 1378\)](#) che restituisce `DayNames` come valori di campo.

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema,

---

## 4 Utilizzo delle variabili nell'editor caricamento dati

---

a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - Variabili di sistema predefinite

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati con date, che verrà caricato in una tabella denominata `Transactions`.
- Un campo `date`.
- La definizione predefinita `LongDayNames`.

#### Script di caricamento

```
SET LongDayNames='Monday;Tuesday;Wednesday;Thursday;Friday;Saturday;Sunday';
```

```
Transactions:
LOAD
date,
Date(date,'www') as dayname,
id,
amount
INLINE
[
date,id,amount
01/01/2022,1,1000
02/01/2022,2,2123
03/01/2022,3,4124
04/01/2022,4,2431
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `date`
- `dayname`

Creare questa misura:

## 4 Utilizzo delle variabili nell'editor caricamento dati

---

=sum(amount)

Tabella dei risultati

| date       | dayname  | =sum(amount) |
|------------|----------|--------------|
| 01/01/2022 | Saturday | 1000         |
| 02/01/2022 | Tuesday  | 2123         |
| 03/01/2022 | Tuesday  | 4124         |
| 04/01/2022 | Friday   | 2431         |

Nello script di caricamento, per creare un campo denominato dayname, viene utilizzata la funzione date viene utilizzata con il campo date come primo argomento. Il secondo argomento nella funzione è la formattazione `www`.

L'utilizzo di questa formattazione converte i valori del primo argomento nel nome del giorno completo corrispondente impostato nella variabile `LongDayNames`. Questo viene mostrato nella tabella dei risultati, nei valori di campo per il campo creato dayname.

### Esempio 2 - Modifica della variabile di sistema

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor di caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio. Tuttavia, all'inizio dello script, la definizione `LongDayNames` viene modificata per utilizzare i giorni della settimana in spagnolo.

#### Script di caricamento

```
SET LongDayNames='Lunes;Martes;Miércoles;Jueves;Viernes;Sábado;Domingo';
```

```
Transactions:
```

```
LOAD
date,
Date(date,'www') as dayname,
id,
amount
INLINE
[
date,id,amount
01/01/2022,1,1000
02/01/2022,2,2123
03/01/2022,3,4124
04/01/2022,4,2431
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

## 4 Utilizzo delle variabili nell'editor caricamento dati

---

- date
- dayname

Creare questa misura:

=sum(amount)

Tabella dei risultati

| date       | dayname | =sum(amount) |
|------------|---------|--------------|
| 01/01/2022 | Sábado  | 1000         |
| 02/01/2022 | Martes  | 2123         |
| 03/01/2022 | Martes  | 4124         |
| 04/01/2022 | Viernes | 2431         |

Nello script di caricamento, la variabile LongDayNames viene modificata per elencare i giorni della settimana in spagnolo.

Quindi, è necessario creare un campo denominato dayname, che è la funzione date utilizzata con il campo date come primo argomento.

Il secondo argomento nella funzione è la formattazione www. L'utilizzo della formattazione Qlik Sense converte i valori del primo argomento nel nome del giorno completo corrispondente impostato nella variabile LongDayNames.

Nella tabella dei risultati, i valori di campo per il campo creato dayname visualizza i giorni della settimana scritti per intero in spagnolo.

### LongMonthNames

Il formato specificato sostituisce la convenzione usata per i nomi lunghi dei mesi nelle impostazioni locali.

**Sintassi:**

**LongMonthNames**

Quando si modifica la variabile, è necessario utilizzare ; per separare i singoli valori.

Il seguente esempio della funzione LongMonthNames definisce i nomi completi dei mesi:

Set

LongMonthNames='January;February;March;April;May;June;July;August;September;October;November;December';

La funzione LongMonthNames viene spesso utilizzata in combinazione con le seguenti funzioni:

#### Funzioni correlate

| Funzione                                | Interazione                                                   |
|-----------------------------------------|---------------------------------------------------------------|
| <a href="#">Date (page 1378)</a>        | Funzione script per restituire DayNames come valori di campo. |
| <a href="#">LongDayNames (page 246)</a> | Valori nel formato lungo di DayNames.                         |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - Variabili di sistema predefinite

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati caricato in una tabella denominata `Transactions`.
- Un campo `date`.
- La definizione predefinita `LongMonthNames`.

#### Script di caricamento

```
SET  
LongMonthNames='January;February;March;April;May;June;July;August;September;October;November;December';
```

```
Transactions:  
Load  
date,  
Date(date,'MMMM') as monthname,  
id,  
amount  
Inline  
[  
date,id,amount  
01/01/2022,1,1000.45  
01/02/2022,2,2123.34  
01/03/2022,3,4124.35  
01/04/2022,4,2431.36  
01/05/2022,5,4787.78  
01/06/2022,6,2431.84
```

## 4 Utilizzo delle variabili nell'editor caricamento dati

---

```
01/07/2022,7,2854.83
01/08/2022,8,3554.28
01/09/2022,9,3756.17
01/10/2022,10,3454.35
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni.

- date
- monthname

Creare questa misura:

```
=sum(amount)
```

Tabella dei risultati

| date       | monthname | sum(amount) |
|------------|-----------|-------------|
| 01/01/2022 | Gennaio   | 1000.45     |
| 01/02/2022 | Gennaio   | 2123.34     |
| 01/03/2022 | Gennaio   | 4124.35     |
| 01/04/2022 | Gennaio   | 2431.36     |
| 01/05/2022 | Gennaio   | 4787.78     |
| 01/06/2022 | Gennaio   | 2431.84     |
| 01/07/2022 | Gennaio   | 2854.83     |
| 01/08/2022 | Gennaio   | 3554.28     |
| 01/09/2022 | Gennaio   | 3756.17     |
| 01/10/2022 | Gennaio   | 3454.35     |

Viene utilizzata la definizione predefinita `LongMonthNames`. Nello script di caricamento, per creare un campo denominato `month`, viene utilizzata la funzione `Date` viene utilizzata con il campo `date` come primo argomento. Il secondo argomento nella funzione è la formattazione `MMMM`.

L'utilizzo della formattazione Qlik Sense converte i valori del primo argomento nel nome completo del mese corrispondente impostato nella variabile `LongMonthNames`. Questo viene mostrato nella tabella dei risultati, nei valori di campo per il campo creato `month`.

### Esempio 2 - Modifica della variabile di sistema

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

## 4 Utilizzo delle variabili nell'editor caricamento dati

---

Lo script di caricamento contiene:

- Un set di dati caricato in una tabella denominata `Transactions`.
- Un campo `date`.
- La variabile `LongMonthNames` modificata per utilizzare i giorni abbreviati della settimana in spagnolo.

### Script di caricamento

```
SET  
LongMonthNames='Enero;Febrero;Marzo;Abril;Mayo;Junio;Julio;Agosto;Septiembre;OctubreNoviembre;  
Diciembre';
```

```
Transactions:  
LOAD  
date,  
Date(date,'MMMM') as monthname,  
id,  
amount  
INLINE  
[  
date,id,amount  
01/01/2022,1,1000  
02/01/2022,2,2123  
03/01/2022,3,4124  
04/01/2022,4,2431  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere `sum(amount)` come misura e i seguenti campi come dimensioni:

- `date`
- `monthname`

Creare questa misura:

```
=sum(amount)
```

| Tabella dei risultati |                  |                    |
|-----------------------|------------------|--------------------|
| <b>date</b>           | <b>monthname</b> | <b>sum(amount)</b> |
| 01/01/2022            | Enero            | 1000.45            |
| 01/02/2022            | Enero            | 2123.34            |
| 01/03/2022            | Enero            | 4124.35            |
| 01/04/2022            | Enero            | 2431.36            |
| 01/05/2022            | Enero            | 4787.78            |
| 01/06/2022            | Enero            | 2431.84            |

## 4 Utilizzo delle variabili nell'editor caricamento dati

| date       | monthname | sum(amount) |
|------------|-----------|-------------|
| 01/07/2022 | Enero     | 2854.83     |
| 01/08/2022 | Enero     | 3554.28     |
| 01/09/2022 | Enero     | 3756.17     |
| 01/10/2022 | Enero     | 3454.35     |

Nello script di caricamento, la variabile `LongMonthNames` viene modificata per elencare i mesi dell'anno in spagnolo. Quindi, per creare un campo denominato `monthname`, la funzione `Date` viene utilizzata con il campo `date` come primo argomento. Il secondo argomento nella funzione è la formattazione `MMMM`.

L'utilizzo della formattazione Qlik Sense converte i valori del primo argomento nel nome completo del mese corrispondente impostato nella variabile `LongMonthNames`. Nella tabella dei risultati, i valori di campo per il campo creato `monthname` visualizzano il nome del mese in spagnolo.

### MoneyDecimalSep

Il separatore dei decimali specificato sostituisce il simbolo decimale per la valuta impostato nelle impostazioni locali.



*Per impostazione predefinita, in Qlik Sense i numeri e il testo vengono visualizzati in modo diverso nei grafici a tabella. I numeri sono allineati a destra e il testo è allineato a sinistra. In questo modo è facile individuare i problemi di conversione da testo a numero. Tutte le tabelle di questa pagina che mostrano i risultati Qlik Sense utilizzeranno questa formattazione.*

#### Sintassi:

##### MoneyDecimalSep

Le applicazioni Qlik Sense interpretano i campi di testo conformi a questa formattazione come valori monetari. Il campo di testo deve contenere il simbolo di valuta definito nella variabile di sistema `MoneyFormat`. Il valore `MoneyDecimalSep` è particolarmente utile quando si gestiscono sorgenti dati ricevute da più impostazioni regionali diverse.

Il seguente esempio mostra un possibile utilizzo della variabile di sistema `MoneyDecimalSep`:

```
Set MoneyDecimalSep='.';
```

Questa funzione viene spesso utilizzata insieme alle funzioni seguenti:

#### Funzioni correlate

| Funzione                      | Interazione                                                                                                                                                                                                                                                           |
|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>MoneyFormat</code>      | Nei casi di interpretazione di campi di testo, il simbolo <code>MoneyFormat</code> verrà utilizzato come parte dell'interpretazione. Per la Formattazione dei numeri, la formattazione <code>MoneyFormat</code> verrà utilizzata da Qlik Sense negli Oggetti grafico. |
| <code>MoneyThousandSep</code> | Nei casi di interpretazione di campi di testo, è necessario rispettare anche la funzione <code>MoneyThousandSep</code> .                                                                                                                                              |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - Notazione MoneyDecimalSep punto (.)

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati caricato in una tabella denominata `Transactions`.
- Dati forniti che hanno il loro campo monetario in formato testo con un punto '.' usato come separatore decimale. Ogni record è inoltre preceduto dal simbolo "\$", tranne l'ultimo record, che è preceduto dal simbolo "£".

Tenere presente che la variabile di sistema `MoneyFormat` definisce il dollaro '\$' come valuta predefinita.

#### Script di caricamento

```
SET MoneyThousandSep=',';
SET MoneyDecimalSep='.';
SET MoneyFormat='$###0.00;-###0.00';
```

`Transactions:`

```
Load
date,
id,
amount
Inline
[
date,id,amount
01/01/2022,1,'$14.41'
01/02/2022,2,'$2,814.32'
01/03/2022,3,'$249.36'
01/04/2022,4,'$24.37'
```

## 4 Utilizzo delle variabili nell'editor caricamento dati

```
01/05/2022,5,'$7.54'  
01/06/2022,6,'$243.63'  
01/07/2022,7,'$545.36'  
01/08/2022,8,'$3.55'  
01/09/2022,9,'$3.436'  
01/10/2022,10,'£345.66'  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:amount.

Aggiungere le seguenti misure:

- isNum(amount)
- sum(amount)

Esaminare i risultati qui sotto, dimostrando la corretta interpretazione di tutti i valori del dollaro "\$".

Tabella dei risultati

| importo    | =isNum(amount) | =Sum(amount) |
|------------|----------------|--------------|
| Totali     | 0              | \$3905.98    |
| £345.66    | 0              | \$0.00       |
| \$3.436    | -1             | \$3.44       |
| \$3.55     | -1             | \$3.55       |
| \$7.54     | -1             | \$7.54       |
| \$14.41    | -1             | \$14.41      |
| \$24.37    | -1             | \$24.37      |
| 243.63     | -1             | \$243.63     |
| \$249.36   | -1             | \$249.36     |
| \$545.36   | -1             | \$545.36     |
| \$2,814.32 | -1             | \$2814.32    |

La tabella dei risultati sopra mostra come il campo amount sia stato interpretato correttamente per tutti i valori con prefisso dollaro (\$), mentre il prefisso sterlina (£) amount non è stato convertito in un valore monetario.

### Esempio 2 - Notazione MoneyDecimalSep virgola (,)

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati caricato in una tabella denominata `Transactions`.
- Dati forniti che hanno il loro campo monetario in formato testo con una virgola '.' usato come separatore decimale. Ogni record è inoltre preceduto dal simbolo '\$', ad eccezione dell'ultimo record, che utilizza erroneamente il separatore decimale a punti ','.

Tenere presente che la variabile di sistema `MoneyFormat` definisce il dollaro '\$' come valuta predefinita.

#### Script di caricamento

```
SET MoneyThousandSep='.';
SET MoneyDecimalSep=',';
SET MoneyFormat='$###0.00;-###0.00';
```

```
Transactions:
Load
date,
id,
amount
Inline
[
date,id,amount
01/01/2022,1,'$14,41'
01/02/2022,2,'$2.814,32'
01/03/2022,3,'$249,36'
01/04/2022,4,'$24,37'
01/05/2022,5,'$7,54'
01/06/2022,6,'$243,63'
01/07/2022,7,'$545,36'
01/08/2022,8,'$3,55'
01/09/2022,9,'$3,436'
01/10/2022,10,'$345.66'
];
```

#### Risultati

Testo del paragrafo per i Risultati.

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: `amount`.

Aggiungere le seguenti misure:

## 4 Utilizzo delle variabili nell'editor caricamento dati

- `isNum(amount)`
- `sum(amount)`

Esaminare i risultati qui sotto, dimostrando la corretta interpretazione di tutti i valori, ad eccezione dell'importo in cui il separatore decimale utilizza la notazione del punto '. In questo caso si sarebbe dovuta usare una virgola.

Tabella dei risultati

| importo    | =isNum(amount) | =Sum(amount) |
|------------|----------------|--------------|
| Totali     | 0              | \$3905.98    |
| \$345.66   | 0              | \$0.00       |
| \$3,436    | -1             | \$3.44       |
| \$3,55     | -1             | \$3.55       |
| \$7,54     | -1             | \$7.54       |
| \$14,41    | -1             | \$14.41      |
| \$24,37    | -1             | \$24.37      |
| \$243,63   | -1             | \$243.63     |
| \$249,36   | -1             | \$249.36     |
| \$545,36   | -1             | \$545.36     |
| \$2.814,32 | -1             | \$2814.32    |

### MoneyFormat

Questa variabile di sistema definisce il modello di formattazione utilizzato da Qlik per la traduzione automatica di testo in numero quando il numero è preceduto da un simbolo monetario. Definisce anche il modo in cui le misure le cui proprietà di formattazione dei numeri sono impostate su "denaro" saranno visualizzate negli oggetti del grafico.

Il simbolo definito come parte del modello di formattazione nella variabile di sistema `MoneyFormat` sostituisce il simbolo di valuta impostato dalle impostazioni regionali.



*Per impostazione predefinita, in Qlik Sense i numeri e il testo vengono visualizzati in modo diverso nei grafici a tabella. I numeri sono allineati a destra e il testo è allineato a sinistra. In questo modo è facile individuare i problemi di conversione da testo a numero. Tutte le tabelle di questa pagina che mostrano i risultati Qlik Sense utilizzeranno questa formattazione.*

#### Sintassi:

##### `MoneyFormat`

```
Set MoneyFormat='$ #,##0.00; ($ #,##0.00)';
```

## 4 Utilizzo delle variabili nell'editor caricamento dati

Questa formattazione viene visualizzata negli oggetti del grafico quando la proprietà di un campo numerico `Number Formatting` è impostata su `Money`. Inoltre, quando i campi di testo numerici sono interpretati da Qlik Sense, se il simbolo di valuta del campo di testo corrisponde a quello del simbolo definito nella variabile `MoneyFormat`, Qlik Sense interpreterà il campo come un valore monetario.

Questa funzione viene spesso utilizzata insieme alle funzioni seguenti:

### Funzioni correlate

| Funzione                                                                     | Interazione                                                                                                                  |
|------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| <a href="#"><u>MoneyDecimalSep</u></a><br><a href="#"><u>(page 253)</u></a>  | Per la formattazione dei numeri, verrà utilizzato <code>MoneyDecimalSep</code> nella formattazione dei campi degli oggetti.  |
| <a href="#"><u>MoneyThousandSep</u></a><br><a href="#"><u>(page 261)</u></a> | Per la formattazione dei numeri, verrà utilizzato <code>MoneyThousandSep</code> nella formattazione dei campi degli oggetti. |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - MoneyFormat

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor di caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento contiene un set di dati che viene caricato in una tabella denominata `Transactions`. Viene utilizzata la definizione di variabile `MoneyFormat` predefinita.

#### Script di caricamento

```
SET MoneyThousandSep=' ';  
SET MoneyDecimalSep='.';  
SET MoneyFormat='$###0.00;-$$$0.00';
```

`Transactions:`

## 4 Utilizzo delle variabili nell'editor caricamento dati

```
Load
date,
id,
amount
Inline
[
date,id,amount
01/01/2022,1,$10000000441
01/02/2022,2,$21237492432
01/03/2022,3,$249475336
01/04/2022,4,$24313369837
01/05/2022,5,$7873578754
01/06/2022,6,$24313884663
01/07/2022,7,$545883436
01/08/2022,8,$35545828255
01/09/2022,9,$37565817436
01/10/2022,10,$3454343566
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- amount

Aggiungere questa misura:

=Sum(amount)

In **Formattazione numeri**, selezionare **Denaro** per configurare sum(amount) come valore monetario.

Tabella dei risultati

| date       | Importo       | =Sum(amount)      |
|------------|---------------|-------------------|
| Totali     |               | \$165099674156.00 |
| 01/01/2022 | \$10000000441 | \$10000000441.00  |
| 01/02/2022 | \$21237492432 | \$21237492432.00  |
| 01/03/2022 | \$249475336   | \$249475336.00    |
| 01/04/2022 | \$24313369837 | \$24313369837.00  |
| 01/05/2022 | \$7873578754  | \$7873578754.00   |
| 01/06/2022 | \$24313884663 | \$24313884663.00  |
| 01/07/2022 | \$545883436   | \$545883436.00    |
| 01/08/2022 | \$35545828255 | \$35545828255.00  |
| 01/09/2022 | \$37565817436 | \$37565817436.00  |
| 01/10/2022 | \$3454343566  | \$3454343566.00   |

---

## 4 Utilizzo delle variabili nell'editor caricamento dati

---

Viene utilizzata la definizione predefinita `MoneyFormat`. L'aspetto sarà il seguente: `####0.00; -####0.00`.

Nella tabella dei risultati, il formato del campo `amount` mostra il simbolo della valuta e la virgola e i decimali sono stati inclusi.

### Esempio 2 - MoneyFormat con separatore delle migliaia e formati di input misti

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati in formato misto, che viene caricato in una tabella denominata `Transactions` con separatori di migliaia e separatori decimali intercalati.
- Una modifica della definizione `MoneyFormat` è stata apportata per includere una virgola come separatore delle migliaia.
- Una delle righe di dati è stata erroneamente delimitata con migliaia di virgole di separazione nei punti sbagliati. Si noti che questo importo viene lasciato come testo e non è interpretabile come numero.

#### Script di caricamento

```
SET MoneyThousandSep=',';
SET MoneyDecimalSep='.';
SET MoneyFormat = '$#,##0.00;-#,##0.00';
```

```
Transactions:
Load
date,
id,
amount
Inline
[
date,id,amount
01/01/2022,1,'$10,000,000,441.45'
01/02/2022,2,'$212,3749,24,32.23'
01/03/2022,3,$249475336.45
01/04/2022,4,$24,313,369,837
01/05/2022,5,$7873578754
01/06/2022,6,$24313884663
01/07/2022,7,$545883436
01/08/2022,8,$35545828255
01/09/2022,9,$37565817436
01/10/2022,10,$3454343566
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

## 4 Utilizzo delle variabili nell'editor caricamento dati

- date
- amount

Aggiungere questa misura:

=Sum(amount)

In **Formattazione numeri**, selezionare **Denaro** per configurare sum(amount) come valore monetario.

Tabella dei risultati

| date       | Importo             | =Sum(amount)         |
|------------|---------------------|----------------------|
| Totali     |                     | \$119,548,811,911.90 |
| 01/01/2022 | \$10,000,000,441.45 | \$10,000,000,441.45  |
| 01/02/2022 | \$212,3749,24,32.23 | \$0.00               |
| 01/03/2022 | \$249475336.45      | \$249,475,336.45     |
| 01/04/2022 | \$24                | \$24.00              |
| 01/05/2022 | \$7873578754        | \$7,873,578,754.00   |
| 01/06/2022 | \$24313884663       | \$24,313,884,663.00  |
| 01/07/2022 | \$545883436         | \$545,883,436.00     |
| 01/08/2022 | \$35545828255       | \$35,545,828,255.00  |
| 01/09/2022 | \$37565817436       | \$37,565,817,436.00  |
| 01/10/2022 | \$3454343566        | \$3,454,343,566.00   |

All'inizio dello script, la variabile di sistema MoneyFormat viene modificata per includere una virgola come separatore delle migliaia. Nella tabella Qlik Sense, si può notare che la formattazione include questo separatore. Inoltre, la riga con il separatore errato non è stata interpretata correttamente e rimane come testo. Per questo motivo non contribuisce alla somma dell'importo.

### MoneyThousandSep

Il separatore delle migliaia specificato sostituisce il simbolo di raggruppamento delle cifre per la valuta configurata nelle impostazioni locali.



*Per impostazione predefinita, in Qlik Sense i numeri e il testo vengono visualizzati in modo diverso nei grafici a tabella. I numeri sono allineati a destra e il testo è allineato a sinistra. In questo modo è facile individuare i problemi di conversione da testo a numero. Tutte le tabelle di questa pagina che mostrano i risultati Qlik Sense utilizzeranno questa formattazione.*

**Sintassi:**

**MoneyThousandSep**

## 4 Utilizzo delle variabili nell'editor caricamento dati

Le applicazioni Qlik Sense interpretano i campi di testo conformi a questa formattazione come valori monetari. Il campo di testo deve contenere il simbolo di valuta definito nella variabile di sistema `MoneyFormat`. Il valore `MoneyThousandSep` è particolarmente utile quando si gestiscono sorgenti dati ricevute da più impostazioni regionali diverse.

Il seguente esempio mostra un possibile utilizzo della variabile di sistema `MoneyThousandSep`:

```
Set MoneyDecimalSep=',';
```

Questa funzione viene spesso utilizzata insieme alle funzioni seguenti:

### Funzioni correlate

| Funzione                     | Interazione                                                                                                                                                                                                                                                               |
|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>MoneyFormat</code>     | Nei casi di interpretazione di campi di testo, il simbolo <code>MoneyFormat</code> verrà utilizzato come parte dell'interpretazione. Per la Formattazione dei numeri, la formattazione <code>MoneyFormat</code> verrà utilizzata da Qlik Sense negli oggetti del grafico. |
| <code>MoneyDecimalSep</code> | Nei casi di interpretazione di campi di testo, è necessario rispettare anche la funzione <code>MoneyDecimalSep</code> .                                                                                                                                                   |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - Notazione MoneyThousandSep virgola (,)

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati caricato in una tabella denominata `Transactions`.
- Dati forniti che hanno il loro campo monetario in formato testo con una virgola usato come separatore delle migliaia. Ogni record è inoltre preceduto dal simbolo "\$".

## 4 Utilizzo delle variabili nell'editor caricamento dati

Tenere presente che la variabile di sistema `MoneyFormat` definisce il dollaro '\$' come valuta predefinita.

### Script di caricamento

```
SET MoneyThousandSep=',';
SET MoneyDecimalSep='.';
SET MoneyFormat='$###0.00;-$$$0.00';
```

Transactions:

```
Load
date,
id,
amount
Inline
[
date,id,amount
01/01/2022,1,'$10,000,000,441'
01/02/2022,2,'$21,237,492,432'
01/03/2022,3,'$249,475,336'
01/04/2022,4,'$24,313,369,837'
01/05/2022,5,'$7,873,578,754'
01/06/2022,6,'$24,313,884,663'
01/07/2022,7,'$545,883,436'
01/08/2022,8,'$35,545,828,255'
01/09/2022,9,'$37,565,817,436'
01/10/2022,10,'$3.454.343.566'
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: `amount`.

Aggiungere le seguenti misure:

- `isNum(amount)`
- `sum(amount)`

Esaminare i risultati di seguito. La tabella mostra la corretta interpretazione di tutti i valori utilizzando la virgola ',' come separatore delle migliaia.

Il campo `amount` è stato interpretato correttamente per tutti i valori, ad eccezione di un valore che utilizzava un punto '.' come separatore delle migliaia.

Tabella dei risultati

| importo         | =isNum(amount) | =Sum(amount)      |
|-----------------|----------------|-------------------|
| Totali          | 0              | \$161645330590.00 |
| \$3.454.343.566 | 0              | \$0.00            |
| \$249,475,336   | -1             | \$249475336.00    |

## 4 Utilizzo delle variabili nell'editor caricamento dati

| importo          | =isNum(amount) | =Sum(amount)     |
|------------------|----------------|------------------|
| \$545,883,436    | -1             | \$545883436.00   |
| \$7,873,578,754  | -1             | \$7873578754.00  |
| \$10,000,000,441 | -1             | \$10000000441.00 |
| \$21,237,492,432 | -1             | \$21237492432.00 |
| \$24,313,369,837 | -1             | \$24313369837.00 |
| \$24,33,884,663  | -1             | \$24313884663.00 |
| \$35,545,828,255 | -1             | \$35545828255.00 |
| \$37,565,817,436 | -1             | \$37565817436.00 |

### Esempio 2 - Notazione MoneyThousandSep punto (.)

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati caricato in una tabella denominata `Transactions`.
- Dati forniti che hanno il loro campo monetario in formato testo con un punto '.' usato come separatore delle migliaia. Ogni record è inoltre preceduto dal simbolo "\$".

Tenere presente che la variabile di sistema `MoneyFormat` definisce il dollaro '\$' come valuta predefinita.

#### Script di caricamento

```
SET MoneyThousandSep='.';
SET MoneyDecimalSep='.';
SET MoneyFormat='$###0.00;-$$$0.00';
```

`Transactions:`

```
Load
date,
id,
amount
Inline
[
date,id,amount
01/01/2022,1,'$10.000.000.441'
01/02/2022,2,'$21.237.492.432'
01/03/2022,3,'$249.475.336'
01/04/2022,4,'$24.313.369.837'
01/05/2022,5,'$7.873.578.754'
01/06/2022,6,'$24.313.884.663'
```

## 4 Utilizzo delle variabili nell'editor caricamento dati

```
01/07/2022,7,'$545.883.436'  
01/08/2022,8,'$35.545.828.255'  
01/09/2022,9,'$37.565.817.436'  
01/10/2022,10,'$3,454,343,566'  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:amount.

Aggiungere le seguenti misure:

- isNum(amount)
- sum(amount)

Esaminare i risultati qui sotto, dimostrando la corretta interpretazione di tutti i valori utilizzando la notazione del punto '.' come separatore delle migliaia.

Il campo amount è stato interpretato correttamente per tutti i valori, ad eccezione di un valore che utilizzava una virgola ',' come separatore delle migliaia.

Tabella dei risultati

| importo          | =isNum(amount) | =Sum(amount)      |
|------------------|----------------|-------------------|
| Totali           | 0              | \$161645330590.00 |
| \$3,545,343,566  | 0              | \$0.00            |
| \$249.475.336    | -1             | \$249475336.00    |
| \$545.883.436    | -1             | 545883436.00      |
| \$7.873.578.754  | -1             | \$7873578754.00   |
| \$10.000.000.441 | -1             | \$10000000441.00  |
| \$21.237.492.432 | -1             | \$21237492432.00  |
| \$24.313.884.663 | -1             | \$24313884663.00  |
| \$24.313.884.663 | -1             | \$24313884663.00  |
| \$35.545.828.255 | -1             | \$35545828255.00  |
| \$37.565.817.436 | -1             | \$37565817436.00  |

### MonthNames

Il formato specificato sostituisce la convenzione usata per i nomi dei mesi nelle impostazioni locali.

#### Sintassi:

##### MonthNames

Quando si modifica la variabile, è necessario utilizzare ; per separare i singoli valori.

## 4 Utilizzo delle variabili nell'editor caricamento dati

---

### Esempi di funzioni

#### Esempio

Set MonthNames='Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec';

#### Risultati

L'utilizzo della funzione MonthNames consente di definire i nomi dei mesi in inglese e nella loro forma abbreviata.

Set

MonthNames='Enero;Feb;Marzo;Abr;Mayo;Jun;Jul;Agosto;Set;Oct;Nov;Dic';

L'utilizzo della funzione MonthNames consente di definire i nomi dei mesi in spagnolo e nella loro forma abbreviata.

La funzione MonthNames può essere utilizzata in combinazione con le seguenti funzioni:

### Funzioni correlate

#### Funzione

#### Interazione

[month \(page 1024\)](#)

Funzione di script per restituire i valori definiti in MonthNames come valori di campo

[Date \(page 1378\)](#)

Funzione script per restituire i valori definiti in MonthNames come valori di campo in base all'argomento di formattazione fornito

[LongMonthNames \(page 249\)](#)

Valori nel formato lungo di MonthNames

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione SET DateFormat nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - Variabili di sistema predefinite

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati caricato in una tabella denominata `Transactions`.
- Un campo `date`.
- La definizione predefinita `MonthNames`.

#### Script di caricamento

```
SET MonthNames='Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec';
```

```
Transactions:
```

```
LOAD
```

```
date,
```

```
Month(date) as monthname,
```

```
id,
```

```
amount
```

```
INLINE
```

```
[
```

```
date,id,amount
```

```
01/01/2022,1,1000.45
```

```
01/02/2022,2,2123.34
```

```
01/03/2022,3,4124.35
```

```
01/04/2022,4,2431.36
```

```
01/05/2022,5,4787.78
```

```
01/06/2022,6,2431.84
```

```
01/07/2022,7,2854.83
```

```
01/08/2022,8,3554.28
```

```
01/09/2022,9,3756.17
```

```
01/10/2022,10,3454.35
```

```
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `date`
- `monthname`

Creare questa misura:

```
=sum(amount)
```

## 4 Utilizzo delle variabili nell'editor caricamento dati

---

Tabella dei risultati

| date       | monthname | sum(amount) |
|------------|-----------|-------------|
| 01/01/2022 | Jan       | 1000.45     |
| 01/02/2022 | Jan       | 2123.34     |
| 01/03/2022 | Jan       | 4124.35     |
| 01/04/2022 | Jan       | 2431.36     |
| 01/05/2022 | Jan       | 4787.78     |
| 01/06/2022 | Jan       | 2431.84     |
| 01/07/2022 | Jan       | 2854.83     |
| 01/08/2022 | Jan       | 3554.28     |
| 01/09/2022 | Jan       | 3756.17     |
| 01/10/2022 | Jan       | 3454.35     |

Viene utilizzata la definizione predefinita `MonthNames`. Nello script di caricamento, la funzione `Month` viene utilizzata con il campo `date` come argomento fornito.

Nella tabella dei risultati, l'output della funzione `Month` visualizza i giorni della settimana nel formato della definizione `MonthNames`.

### Esempio 2 - Modifica della variabile di sistema

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati caricato in una tabella denominata `Transactions`.
- Un campo `date`.
- La variabile `MonthNames` modificata per utilizzare i mesi abbreviati in spagnolo.

#### Script di caricamento

```
Set
MonthNames=' Enero;Feb;Marzo;Abr;Mayo;Jun;Jul;Agosto;Set;Oct;Nov;Dic';

Transactions:
LOAD
date,
month(date) as month,
id,
amount
INLINE
[
```

## 4 Utilizzo delle variabili nell'editor caricamento dati

---

```
date, id, amount
01/01/2022, 1, 1000
02/01/2022, 2, 2123
03/01/2022, 3, 4124
04/01/2022, 4, 2431
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- monthname

Creare questa misura:

```
=sum(amount)
```

Tabella dei risultati

| date       | monthname | sum(amount) |
|------------|-----------|-------------|
| 01/01/2022 | Enero     | 1000.45     |
| 01/02/2022 | Enero     | 2123.34     |
| 01/03/2022 | Enero     | 4124.35     |
| 01/04/2022 | Enero     | 2431.36     |
| 01/05/2022 | Enero     | 4787.78     |
| 01/06/2022 | Enero     | 2431.84     |
| 01/07/2022 | Enero     | 2854.83     |
| 01/08/2022 | Enero     | 3554.28     |
| 01/09/2022 | Enero     | 3756.17     |
| 01/10/2022 | Enero     | 3454.35     |

Nello script di caricamento, per prima cosa la variabile `MonthNames` viene modificata per elencare i mesi dell'anno abbreviati in spagnolo. La funzione `Month` viene utilizzata con il campo `date` come argomento fornito.

Nella tabella dei risultati, l'output della funzione `Month` visualizza i giorni della settimana nel formato della definizione `MonthNames`.

È importante ricordare che se la lingua per la variabile `MonthNames` viene modificata come in questo esempio, la variabile `LongMonthNames` conterrà comunque i mesi dell'anno in inglese. Se entrambe le variabili vengono utilizzate nell'applicazione, è necessario modificare la variabile `LongMonthNames`.

### Esempio 3 - Funzione data

Script di caricamento e risultati

## 4 Utilizzo delle variabili nell'editor caricamento dati

---

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati caricato in una tabella denominata `Transactions`.
- Un campo `date`.
- La definizione predefinita `MonthNames`.

### Script di caricamento

```
SET MonthNames='Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec';
```

```
Transactions:
LOAD
date,
Month(date, 'MMM') as monthname,
id,
amount
INLINE
[
date,id,amount
01/01/2022,1,1000.45
01/02/2022,2,2123.34
01/03/2022,3,4124.35
01/04/2022,4,2431.36
01/05/2022,5,4787.78
01/06/2022,6,2431.84
01/07/2022,7,2854.83
01/08/2022,8,3554.28
01/09/2022,9,3756.17
01/10/2022,10,3454.35
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `date`
- `monthname`

Creare questa misura:

```
=sum(amount)
```

Tabella dei risultati

| <b>date</b> | <b>monthname</b> | <b>sum(amount)</b> |
|-------------|------------------|--------------------|
| 01/01/2022  | Jan              | 1000.45            |

## 4 Utilizzo delle variabili nell'editor caricamento dati

| date       | monthname | sum(amount) |
|------------|-----------|-------------|
| 01/02/2022 | Jan       | 2123.34     |
| 01/03/2022 | Jan       | 4124.35     |
| 01/04/2022 | Jan       | 2431.36     |
| 01/05/2022 | Jan       | 4787.78     |
| 01/06/2022 | Jan       | 2431.84     |
| 01/07/2022 | Jan       | 2854.83     |
| 01/08/2022 | Jan       | 3554.28     |
| 01/09/2022 | Jan       | 3756.17     |
| 01/10/2022 | Jan       | 3454.35     |

Viene utilizzata la definizione predefinita `MonthNames`. Nello script di caricamento, la funzione `date` viene utilizzata con il campo `date` come primo argomento. Il secondo argomento è `MMM`.

L'utilizzo della formattazione Qlik Sense converte i valori del primo argomento nel nome del mese corrispondente impostato nella variabile `MonthNames`. Questo viene mostrato nella tabella dei risultati, nei valori di campo per il campo creato `month`.

### NumericalAbbreviation

L'abbreviazione numerica definisce l'abbreviazione da utilizzare per i prefissi in scala dei numeri, ad esempio M per mega o un milione ( $10^6$ ) e  $\mu$  per micro ( $10^{-6}$ ).

#### Sintassi:

##### **NumericalAbbreviation**

La variabile `NumericalAbbreviation` deve essere impostata su una stringa contenente un elenco di coppie che definiscono le abbreviazioni, delimitate da punti e virgola. Ogni coppia di definizione abbreviazioni deve contenere la scala (l'esponente in base decimale) e l'abbreviazione separata da due punti, ad esempio `6:M` per i milioni.

L'impostazione predefinita è `'3:k;6:M;9:G;12:T;15:P;18:E;21:Z;24:Y;-3:m;-6:μ;-9:n;-12:p;-15:f;-18:a;-21:z;-24:y'`.

#### Esempi:

Questa impostazione cambia il prefisso delle migliaia in `t` e il prefisso dei miliardi in `B`. Ciò è utile in applicazioni finanziarie statunitensi, dove si utilizzano abbreviazioni come `t$`, `M$` e `B$`.

```
Set NumericalAbbreviation='3:t;6:M;9:B;12:T;15:P;18:E;21:Z;24:Y;-3:m;-6:μ;-9:n;-12:p;-15:f;-18:a;-21:z;-24:y';
```

### ReferenceDay

L'impostazione definisce quale giorno di gennaio impostare come giorno di riferimento per definire la settimana 1. In altre parole, questa impostazione stabilisce il numero di giorni della settimana 1 che devono essere inclusi a gennaio.

#### Sintassi:

##### ReferenceDay

ReferenceDay imposta il numero di giorni da includere nella prima settimana dell'anno. È possibile impostare ReferenceDay su qualsiasi valore compreso tra 1 e 7. Qualsiasi valore al di fuori dell'intervallo 1-7 viene interpretato come il punto centrale della settimana (4), che equivale a ReferenceDay impostato su 4.

Se non si seleziona un valore per l'impostazione ReferenceDay, il valore predefinito mostrerà ReferenceDay=0, che verrà interpretato come il punto centrale della settimana (4), come mostrato nella tabella dei valori ReferenceDay di seguito.

La funzione ReferenceDay viene spesso utilizzata in combinazione con le seguenti funzioni:

#### Funzioni correlate

| Variabile                                  | Interazione                                                                                                                                                                                                                                                                                                                                     |
|--------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">BrokenWeeks</a><br>(page 227)  | Se nell'app Qlik Sense vengono utilizzate settimane ininterrotte, verrà applicata l'impostazione della variabile ReferenceDay. Tuttavia, se vengono utilizzate settimane interrotte, la settimana 1 inizierà il 1 gennaio e terminerà contemporaneamente all'impostazione della variabile FirstWeekDay e ignorerà il contrassegno ReferenceDay. |
| <a href="#">FirstWeekDay</a><br>(page 242) | Numero intero che definisce il giorno da utilizzare come primo giorno della settimana.                                                                                                                                                                                                                                                          |

Qlik Sense consente di impostare i seguenti valori per ReferenceDay:

#### Valori ReferenceDay

| Valore          | Giorno di riferimento |
|-----------------|-----------------------|
| 0 (predefinito) | January 4             |
| 1               | January 1             |
| 2               | Gennaio 2             |
| 3               | January 3             |
| 4               | January 4             |
| 5               | January 5             |
| 6               | January 6             |
| 7               | January 7             |

Nell'esempio seguente il valore ReferenceDay = 3 definisce il 3 gennaio come giorno di riferimento:

## 4 Utilizzo delle variabili nell'editor caricamento dati

---

```
SET ReferenceDay=3; //(set January 3 as the reference day)
```

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempi:

Se si desiderano le impostazioni ISO per le settimane e i numeri di settimana, assicurarsi di inserire nello script quanto segue:

```
Set FirstWeekDay=0;  
Set BrokenWeeks=0;  
Set ReferenceDay=4;    // Jan 4th is always in week 1
```

Se si desiderano le impostazioni USA, assicurarsi che nello script sia presente quanto segue:

```
Set FirstWeekDay=6;  
Set BrokenWeeks=1;  
Set ReferenceDay=1;    // Jan 1st is always in week 1
```

### Esempio 1 - Script di caricamento con il valore predefinito; ReferenceDay=0

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- La variabile di sistema `ReferenceDay` è impostata su 0.
- La variabile `BrokenWeeks` impostata su 0 che fa in modo che l'app utilizzi settimane ininterrotte.
- Un set di dati delle date dalla fine del 2019 all'inizio del 2020.

#### Script di caricamento

```
SET BrokenWeeks = 0;  
SET ReferenceDay = 0;
```

Sales:

## 4 Utilizzo delle variabili nell'editor caricamento dati

---

```
LOAD
date,
sales,
week(date) as week,
weekday(date) as weekday
Inline [
date,sales
12/27/2019,5000
12/28/2019,6000
12/29/2019,7000
12/30/2019,4000
12/31/2019,3000
01/01/2020,6000
01/02/2020,3000
01/03/2020,6000
01/04/2020,8000
01/05/2020,5000
01/06/2020,7000
01/07/2020,3000
01/08/2020,5000
01/09/2020,9000
01/10/2020,5000
01/11/2020,7000
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- week
- weekday

Tabella dei risultati

| date       | week | weekday |
|------------|------|---------|
| 12/27/2019 | 52   | Fri     |
| 12/28/2019 | 52   | Sat     |
| 12/29/2019 | 1    | Sun     |
| 12/30/2019 | 1    | Mon     |
| 12/31/2019 | 1    | Tue     |
| 01/01/2020 | 1    | Wed     |
| 01/02/2020 | 1    | Thu     |
| 01/03/2020 | 1    | Fri     |
| 01/04/2020 | 1    | Sat     |
| 01/05/2020 | 2    | Sun     |

---

## 4 Utilizzo delle variabili nell'editor caricamento dati

---

| date       | week | weekday |
|------------|------|---------|
| 01/06/2020 | 2    | Mon     |
| 01/07/2020 | 2    | Tue     |
| 01/08/2020 | 2    | Wed     |
| 01/09/2020 | 2    | Thu     |
| 01/10/2020 | 2    | Fri     |
| 01/11/2020 | 2    | Sat     |

La settimana 52 si conclude sabato 28 dicembre. Poiché `ReferenceDay` richiede che il 4 gennaio sia incluso nella settimana 1, pertanto la settimana 1 inizia il 29 dicembre e si conclude sabato 4 gennaio.

### Esempio: Variabile `ReferenceDay` impostata su 5

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- La variabile di sistema `ReferenceDay` è impostata su 5.
- La variabile `BrokenWeeks` impostata su 0 che fa in modo che l'app utilizzi settimane ininterrotte.
- Un set di dati delle date dalla fine del 2019 all'inizio del 2020.

#### Script di caricamento

```
SET BrokenWeeks = 0;
SET ReferenceDay = 5;

Sales:
LOAD
date,
sales,
week(date) as week,
weekday(date) as weekday
Inline [
date,sales
12/27/2019,5000
12/28/2019,6000
12/29/2019,7000
12/30/2019,4000
12/31/2019,3000
01/01/2020,6000
01/02/2020,3000
01/03/2020,6000
01/04/2020,8000
01/05/2020,5000
01/06/2020,7000
```

## 4 Utilizzo delle variabili nell'editor caricamento dati

---

```
01/07/2020,3000
01/08/2020,5000
01/09/2020,9000
01/10/2020,5000
01/11/2020,7000
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- week
- weekday

Tabella dei risultati

| date       | week | weekday |
|------------|------|---------|
| 12/27/2019 | 52   | Fri     |
| 12/28/2019 | 52   | Sat     |
| 12/29/2019 | 53   | Sun     |
| 12/30/2019 | 53   | Mon     |
| 12/31/2019 | 53   | Tue     |
| 01/01/2020 | 53   | Wed     |
| 01/02/2020 | 53   | Thu     |
| 01/03/2020 | 53   | Fri     |
| 01/04/2020 | 53   | Sat     |
| 01/05/2020 | 1    | Sun     |
| 01/06/2020 | 1    | Mon     |
| 01/07/2020 | 1    | Tue     |
| 01/08/2020 | 1    | Wed     |
| 01/09/2020 | 1    | Thu     |
| 01/10/2020 | 1    | Fri     |
| 01/11/2020 | 1    | Sat     |

La settimana 52 si conclude sabato 28 dicembre. La variabile `brokenweeks` fa in modo che l'app utilizzi settimane ininterrotte. Il valore del giorno di riferimento di 5 richiede che il 5 gennaio sia incluso nella settimana 1.

Tuttavia, questo giorno è otto giorni dopo la conclusione della settimana 52 dell'anno precedente. Pertanto, la settimana 53 inizia il 29 dicembre e si conclude il 4 gennaio. La settimana 1 inizia domenica 5 gennaio.

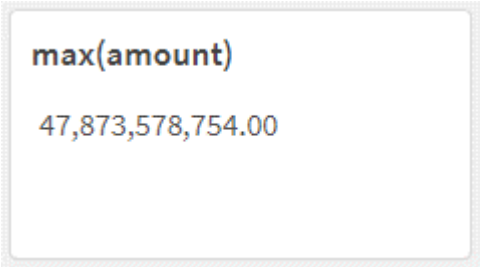
### ThousandSep

Il separatore delle migliaia specificato sostituisce il simbolo di raggruppamento delle cifre del sistema operativo (impostazioni locali).

Sintassi:

**ThousandSep**

Oggetto Qlik Sense che utilizza la variabile ThousandSep (con separatore delle migliaia)



**max(amount)**  
47,873,578,754.00

Le app Qlik Sense interpretano i campi di testo conformi a questa formattazione come numeri. Questa formattazione viene visualizzata negli oggetti del grafico quando la proprietà di **Formattazione numero** per un campo numerico è impostata su **Numero**.

ThousandSep è utile quando si gestiscono sorgenti dati ricevute da più impostazioni regionali.



*Se la variabile ThousandSep viene modificata dopo che gli oggetti sono già stati creati e formattati nell'applicazione, l'utente dovrà riformattare ogni campo pertinente deselegionando e poi riselezionando la proprietà di **formattazione numerica Numero**.*

I seguenti esempi mostrano i possibili utilizzi della variabile di sistema ThousandSep:

```
Set ThousandSep=','; //(for example, seven billion will be displayed as: 7,000,000,000)
```

```
Set ThousandSep=' '; //(for example, seven billion will be displayed as: 7 000 000 000)
```

I seguenti argomenti possono aiutarti a lavorare con questa funzione:

#### Argomenti correlati

| Argomento                             | Descrizione                                                                                                                                                                                                                                             |
|---------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">DecimalSep (page 239)</a> | Nei casi di interpretazione di campi di testo, devono essere rispettate anche le impostazioni del separatore decimale fornite da questa funzione. Per la formattazione dei numeri, <b>DecimalSep</b> verrà utilizzato da Qlik Sense laddove necessario. |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema,

---

## 4 Utilizzo delle variabili nell'editor caricamento dati

---

a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - Variabili di sistema predefinite

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati caricato in una tabella denominata `Transactions`.
- Utilizzo della definizione di variabile `Thousandsep` predefinita.

#### Script di caricamento

```
Transactions:
Load
date,
id,
amount
Inline
[
date,id,amount
01/01/2022,1,10000000441
01/02/2022,2,21237492432
01/03/2022,3,41249475336
01/04/2022,4,24313369837
01/05/2022,5,47873578754
01/06/2022,6,24313884663
01/07/2022,7,28545883436
01/08/2022,8,35545828255
01/09/2022,9,37565817436
01/10/2022,10,3454343566
];
```

### Risultati

Procedere come indicato di seguito:

1. Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:date.
2. Aggiungere la misura seguente:  
=sum(amount)
3. nel pannello delle **proprietà**, alla voce Dati, selezionare la misura.
4. In **Formattazione numeri**, selezionare **Numero**.

*Regolazione della formattazione dei numeri per una misura del grafico*

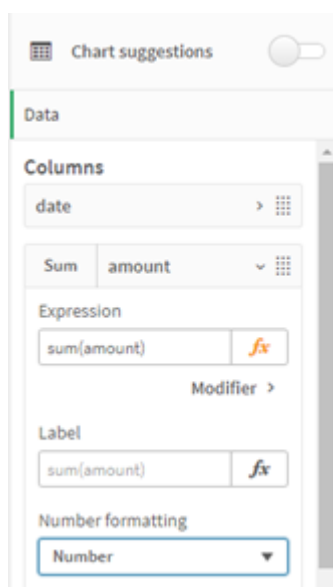


Tabella dei risultati

| date       | sum(amount)       |
|------------|-------------------|
| 01/01/2022 | 10,000,000,441.00 |
| 01/02/2022 | 21,237,492,432.00 |
| 01/03/2022 | 41,249,475,336.00 |
| 01/04/2022 | 24,313,369,837.00 |
| 01/05/2022 | 47,873,578,754.00 |
| 01/06/2022 | 24,313,884,663.00 |
| 01/07/2022 | 28,545,883,436.00 |
| 01/08/2022 | 35,545,828,255.00 |
| 01/09/2022 | 37,565,817,436.00 |
| 01/10/2022 | 3,454,343,566.00  |

---

## 4 Utilizzo delle variabili nell'editor caricamento dati

---

In questo esempio, viene utilizzata la definizione predefinita `ThousandSep`, impostata sul formato virgola (',' ). Nella tabella dei risultati, il formato del campo `importo` visualizza una virgola tra i raggruppamenti delle migliaia.

### Esempio 2 - Modifica della variabile di sistema

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Lo stesso set di dati del primo esempio, caricato in una tabella denominata `Transactions`.
- Modifica della definizione `ThousandSep`, all'inizio dello script, per visualizzare un carattere '\*' come separatore delle migliaia. Si tratta di un esempio estremo, utilizzato esclusivamente per dimostrare la funzionalità della variabile.

La modifica utilizzata in questo esempio è estrema e non comunemente utilizzata, ma viene mostrata per dimostrare la funzionalità della variabile.

#### Script di caricamento

```
SET ThousandSep='*';
```

```
Transactions:
```

```
Load
```

```
date,
```

```
id,
```

```
amount
```

```
Inline
```

```
[
```

```
date,id,amount
```

```
01/01/2022,1,10000000441
```

```
01/02/2022,2,21237492432
```

```
01/03/2022,3,41249475336
```

```
01/04/2022,4,24313369837
```

```
01/05/2022,5,47873578754
```

```
01/06/2022,6,24313884663
```

```
01/07/2022,7,28545883436
```

```
01/08/2022,8,35545828255
```

```
01/09/2022,9,37565817436
```

```
01/10/2022,10,3454343566
```

```
];
```

### Risultati

Procedere come indicato di seguito:

1. Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:date.
2. Aggiungere la misura seguente:  
`=sum(amount)`
3. Nel pannello delle proprietà, alla voce **Dati**, selezionare la misura.
4. In **Formattazione numeri**, selezionare **Personalizzata**.

Tabella dei risultati

| date       | sum(amount)       |
|------------|-------------------|
| 01/01/2022 | 10*000*000*441.00 |
| 01/02/2022 | 21*237*492*432.00 |
| 01/03/2022 | 41*249*475*336.00 |
| 01/04/2022 | 24*313*369*837.00 |
| 01/05/2022 | 47*873*578*754.00 |
| 01/06/2022 | 24*313*884*663.00 |
| 01/07/2022 | 28*545*883*436.00 |
| 01/08/2022 | 35*545*828*255.00 |
| 01/09/2022 | 37.565.817.436,00 |
| 01/10/2022 | 3*454*343*566.00  |

All'inizio dello script, la variabile di sistema `ThousandSep` viene modificata con un `""`. Nella tabella dei risultati, il formato del campo importo può essere visualizzato come un `""` tra il raggruppamento delle migliaia.

### Esempio 3 - Interpretazione del testo

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati caricato in una tabella denominata `Transactions`.
- Dati che hanno il loro campo numerico in formato testo, con una virgola come separatore delle

migliaia.

- Utilizzo della variabile di sistema predefinita ThousandSep.

### Script di caricamento

```
Transactions:
Load
date,
id,
amount
Inline
[
date,id,amount
01/01/2022,1,'10,000,000,441'
01/02/2022,2,'21,492,432'
01/03/2022,3,'4,249,475,336'
01/04/2022,4,'24,313,369,837'
01/05/2022,5,'4,873,578,754'
01/06/2022,6,'313,884,663'
01/07/2022,7,'2,545,883,436'
01/08/2022,8,'545,828,255'
01/09/2022,9,'37,565,817,436'
01/10/2022,10,'3,454,343,566'
];
```

### Risultati

Procedere come indicato di seguito:

1. Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:date.
2. Aggiungere la misura seguente:  
=sum(amount)
3. Nel pannello delle proprietà, alla voce **Dati**, selezionare la misura.
4. In **Formattazione numeri**, selezionare **Numero**.
5. Aggiungere la seguente misura per valutare se il campo importo è un valore numerico o meno:  
=isnum(amount)

Tabella dei risultati

| date       | sum(amount)       | =isnum(amount) |
|------------|-------------------|----------------|
| 01/01/2022 | 10,000,000,441.00 | -1             |
| 01/02/2022 | 21,492,432.00     | -1             |
| 01/03/2022 | 4,249,475,336.00  | -1             |
| 01/04/2022 | 24.313.369.837,00 | -1             |
| 01/05/2022 | 4,873,578,754.00  | -1             |

## 4 Utilizzo delle variabili nell'editor caricamento dati

| date       | sum(amount)       | =isnum(amount) |
|------------|-------------------|----------------|
| 01/06/2022 | 313,884,663.00    | -1             |
| 01/07/2022 | 2,545,883,436.00  | -1             |
| 01/08/2022 | 545,828,255.00    | -1             |
| 01/09/2022 | 37.565.817.436,00 | -1             |
| 01/10/2022 | 3.454.343.566,00  | -1             |

Una volta caricati i dati, si può notare che il campo importo è stato interpretato da Qlik Sense come un valore numerico, grazie alla conformità dei dati alla variabile `ThousandSep`. Ciò è dimostrato dalla funzione `isnum()`, che valuta ogni voce a -1, o a `TRUE`.



*In Qlik Sense, il valore booleano vero è rappresentato da -1 e il valore falso è rappresentato da 0.*

### TimeFormat

Il formato specificato sostituisce il formato dell'ora del sistema operativo (impostazioni locali).

Sintassi:

**TimeFormat**

Esempio:

```
Set TimeFormat='hh:mm:ss';
```

### TimestampFormat

Il formato specificato sostituisce il formato della data e dell'ora del sistema operativo (impostazioni locali).

Sintassi:

**TimestampFormat**

Esempio:

Gli esempi seguenti utilizzano `1983-12-14T13:15:30Z` come data e ora per mostrare i risultati di diverse istruzioni **SET TimestampFormat**. Il formato data utilizzato è `YYYYMMDD` e il formato ora è `h:mm:ss TT`. Il formato data è specificato nell'istruzione **SET DateFormat** e il formato dell'ora è specificato nell'istruzione **SET TimeFormat** all'inizio dello script di caricamento dei dati.

Risultati

| Esempio                                      | Risultato |
|----------------------------------------------|-----------|
| <code>SET TimestampFormat='YYYYMMDD';</code> | 19831214  |

## 4 Utilizzo delle variabili nell'editor caricamento dati

| Esempio                                             | Risultato             |
|-----------------------------------------------------|-----------------------|
| SET TimestampFormat='M/D/YY hh:mm:ss[.fff]';        | 12/14/83 13:15:30     |
| SET TimestampFormat='DD/MM/YYYY hh:mm:ss[.fff]';    | 14/12/1983 13:15:30   |
| SET TimestampFormat='DD/MM/YYYY hh:mm:ss[.fff] TT'; | 14/12/1983 1:15:30 PM |
| SET TimestampFormat='YYYY-MM-DD hh:mm:ss[.fff] TT'; | 1983-12-14 01:15:30   |

### Esempi: script di caricamento

#### Esempio: script di caricamento

Nel primo script di caricamento viene utilizzata l'istruzione `SET TimestampFormat='DD/MM/YYYY h:mm:ss[.fff] TT'`. Nel secondo script di caricamento, il formato di data e ora viene modificato con l'istruzione `SET TimestampFormat='MM/DD/YYYY hh:mm:ss[.fff]'`. I diversi risultati mostrano come funziona l'istruzione **SET TimeFormat** con diversi formati di dati temporali.

La tabella seguente mostra il set di dati utilizzato negli script di caricamento che seguono. La seconda colonna della tabella mostra il formato di ogni indicatore di data e ora nel set di dati. I primi cinque indicatori di data e ora seguono le regole ISO 8601, a differenza del sesto.

#### Set di dati

*Tabella che mostra i dati temporali utilizzati e il formato di ogni indicatore di data e ora nel set di dati.*

| transaction_timestamp    | time data format        |
|--------------------------|-------------------------|
| 2018-08-30               | YYYY-MM-DD              |
| 20180830T193614.857      | YYYYMMDDhhmmss.sss      |
| 20180830T193614.857+0200 | YYYYMMDDhhmmss.sss±hhmm |
| 2018-09-16T12:30-02:00   | YYYY-MM-DDhh:mm±hh:mm   |
| 2018-09-16T13:15:30Z     | YYYY-MM-DDhh:mmZ        |
| 9/30/18 19:36:14         | M/D/YY hh:mm:ss         |

Nell'**editor caricamento dati** creare una nuova sezione, aggiungere lo script di esempio ed eseguirlo. Aggiungere quindi a un foglio nell'app almeno i campi elencati nella colonna dei risultati per visualizzare il risultato.

#### Script di caricamento

```
SET FirstWeekDay=0;
SET BrokenWeeks=1;
SET ReferenceDay=0;
SET DayNames='Mon;Tue;Wed;Thu;Fri;Sat;Sun';
SET LongDayNames='Monday;Tuesday;Wednesday;Thursday;Friday;Saturday;Sunday';
SET DateFormat='YYYYMMDD';
SET TimestampFormat='DD/MM/YYYY h:mm:ss[.fff] TT';
```

## 4 Utilizzo delle variabili nell'editor caricamento dati

---

Transactions:

Load

\*,

Timestamp(transaction\_timestamp, 'YYYY-MM-DD hh:mm:ss[.fff]') as LogTimeStamp

;

Load \* Inline [

transaction\_id, transaction\_timestamp, transaction\_amount, transaction\_quantity, discount,  
customer\_id, size, color\_code

3750, 2018-08-30, 12423.56, 23, 0, 2038593, L, Red

3751, 20180830T193614.857, 5356.31, 6, 0.1, 203521, m, orange

3752, 20180830T193614.857+0200, 15.75, 1, 0.22, 5646471, S, blue

3753, 2018-09-16T12:30-02:00, 1251, 7, 0, 3036491, l, Black

3754, 2018-09-16T13:15:30Z, 21484.21, 1356, 75, 049681, xs, Red

3755, 9/30/18 19:36:14, -59.18, 2, 0.3333333333333333, 2038593, M, Blue

];

### Risultati

*Tabella Qlik Sense che mostra i risultati della variabile di interpretazione  
TimestampFormat utilizzata nello script di caricamento. L'ultimo indicatore di  
data e ora nel set di dati non restituisce una data corretta.*

| transaction_id | transaction_timestamp    | LogTimeStamp        |
|----------------|--------------------------|---------------------|
| 3750           | 2018-08-30               | 2018-08-30 00:00:00 |
| 3751           | 20180830T193614.857      | 2018-08-30 19:36:14 |
| 3752           | 20180830T193614.857+0200 | 2018-08-30 17:36:14 |
| 3753           | 2018-09-16T12:30-02:00   | 2018-09-16 14:30:00 |
| 3754           | 2018-09-16T13:15:30Z     | 2018-09-16 13:15:30 |
| 3755           | 9/30/18 19:36:14         | -                   |

Lo script di caricamento successivo utilizza lo stesso set di dati, ma usando l'istruzione *SET  
TimestampFormat='MM/DD/YYYY hh:mm:ss[.fff]'* per consentire il riconoscimento del formato non  
conforme a ISO-8601 del sesto indicatore di data e ora.

Nell'**editor caricamento dati** sostituire lo script di esempio precedente con quello sottostante ed eseguirlo.  
Aggiungere quindi a un foglio nell'app almeno i campi elencati nella colonna dei risultati per visualizzare il  
risultato.

### Script di caricamento

SET FirstWeekDay=0;

SET BrokenWeeks=1;

SET ReferenceDay=0;

SET DayNames='Mon;Tue;Wed;Thu;Fri;Sat;Sun';

SET LongDayNames='Monday;Tuesday;Wednesday;Thursday;Friday;Saturday;Sunday';

SET DateFormat='YYYYMMDD';

SET TimestampFormat='MM/DD/YYYY hh:mm:ss[.fff]';

## 4 Utilizzo delle variabili nell'editor caricamento dati

Transactions:

Load

\*,

Timestamp(transaction\_timestamp, 'YYYY-MM-DD hh:mm:ss[.fff]') as LogTimeStamp

;

Load \* Inline [

transaction\_id, transaction\_timestamp, transaction\_amount, transaction\_quantity, discount,  
customer\_id, size, color\_code

3750, 2018-08-30, 12423.56, 23, 0, 2038593, L, Red

3751, 20180830T193614.857, 5356.31, 6, 0.1, 203521, m, orange

3752, 20180830T193614.857+0200, 15.75, 1, 0.22, 5646471, S, blue

3753, 2018-09-16T12:30-02:00, 1251, 7, 0, 3036491, l, black

3754, 2018-09-16T13:15:30Z, 21484.21, 1356, 75, 049681, xs, Red

3755, 9/30/18 19:36:14, -59.18, 2, 0.3333333333333333, 2038593, M, Blue

];

### Risultati

*Tabella Qlik Sense che mostra i risultati della variabile di interpretazione  
TimestampFormat utilizzata nello script di caricamento.*

| transaction_id | transaction_timestamp    | LogTimeStamp        |
|----------------|--------------------------|---------------------|
| 3750           | 2018-08-30               | 2018-08-30 00:00:00 |
| 3751           | 20180830T193614.857      | 2018-08-30 19:36:14 |
| 3752           | 20180830T193614.857+0200 | 2018-08-30 17:36:14 |
| 3753           | 2018-09-16T12:30-02:00   | 2018-09-16 14:30:00 |
| 3754           | 2018-09-16T13:15:30Z     | 2018-09-16 13:15:30 |
| 3755           | 9/30/18 19:36:14         | 2018-09-16 19:36:14 |

## 4.9 Variabili di Direct Discovery

### Variabili di sistema di Direct Discovery

#### DirectCacheSeconds

È possibile impostare un limite di memorizzazione nella cache per i risultati della query Direct Discovery per le visualizzazioni. Una volta raggiunto tale limite, Qlik Sense svuota la cache quando vengono eseguite nuove query Direct Discovery. Qlik Sense esegue una query sui dati sorgente per le selezioni e crea nuovamente la cache per il limite di tempo indicato. Il risultato per ciascuna combinazione di selezioni viene memorizzato nella cache in modo indipendente. Ciò significa che la cache viene aggiornata in modo indipendente per ciascuna selezione, quindi una selezione aggiorna la cache solo per i campi selezionati e una seconda selezione la aggiorna per i campi rilevanti. Se la seconda selezione include i campi aggiornati nella prima selezione, questi non vengono aggiornati nuovamente nella cache se il limite di memorizzazione nella cache non è stato raggiunto.

La cache Direct Discovery non è applicabile alle visualizzazioni di **tabella**. Le selezioni di tabella interrogano la sorgente dei dati tutte le volte.

## 4 Utilizzo delle variabili nell'editor caricamento dati

---

Il valore limite deve essere impostato in secondi. Il limite predefinito della cache è 1800 secondi (30 minuti).

Il valore utilizzato per **DirectCacheSeconds** è il valore impostato al momento dell'esecuzione dell'istruzione **DIRECT QUERY**. Tale valore non può essere modificato in fase di esecuzione.

### Esempio:

```
SET DirectCacheSeconds=1800;
```

### DirectConnectionMax

È possibile effettuare chiamate asincrone e parallele al database utilizzando la funzione di pooling delle connessioni. Di seguito è riportata la sintassi dello script di caricamento per impostare la funzionalità di pooling:

```
SET DirectConnectionMax=10;
```

L'impostazione numerica specifica il numero massimo di connessioni di database che il codice Direct Discovery deve utilizzare durante l'aggiornamento di un foglio. L'impostazione predefinita è 1.



*Questa variabile deve essere utilizzata con attenzione. Se è impostata su un valore superiore a 1 causa problemi di connessione a Microsoft SQL Server.*

### DirectUnicodeStrings

Direct Discovery è in grado di supportare la selezione dei dati Unicode estesi mediante l'utilizzo del formato standard SQL per i valori letterali delle stringhe di caratteri estesi (N'<stringa estesa>') come richiesto da alcuni database (ad esempio SQL Server). È possibile attivare l'utilizzo della sintassi per Direct Discovery con la variabile di script **DirectUnicodeStrings**.

Se si imposta questa variabile su "true", sarà possibile attivare l'utilizzo del marcatore di caratteri wide ANSI "N" standard davanti ai valori letterali di stringa. Non tutti i database supportano questo standard. L'impostazione predefinita è 'false'.

### DirectDistinctSupport

Quando il valore di un campo **DIMENSIONE** viene selezionato in un oggetto di Qlik Sense, viene generata una query per il database di origine. Quando la query richiede il raggruppamento, Direct Discovery utilizza la parola chiave **DISTINCT** per selezionare solo valori univoci. Alcuni database, tuttavia, richiedono la parola chiave **GROUP BY**. Impostare **DirectDistinctSupport** su 'false' per generare **GROUP BY** anziché **DISTINCT** nelle query per i valori univoci.

```
SET DirectDistinctSupport='false';
```

Se **DirectDistinctSupport** è impostato su true, viene utilizzato **DISTINCT**. Se non è impostato su alcun valore, il comportamento predefinito prevede l'utilizzo di **DISTINCT**.

### DirectEnableSubquery

Negli scenari con tabelle multiple ad alta cardinalità, è possibile generare una query secondaria nella query SQL invece di generare una lunga clausola IN. È possibile attivarla impostando **DirectEnableSubquery** su 'true'. il valore predefinito è 'false'.



Quando viene attivato **DirectEnableSubquery**, non è possibile caricare tabelle che non si trovino nella modalità **Direct Discovery**.

```
SET DirectEnableSubquery='true';
```

### Variabili di unione di query Teradata

L'unione di query Teradata è una funzione che consente alle applicazioni aziendali di collaborare con il database Teradata sottostante, al fine di fornire una migliore contabilità, prioritizzazione e gestione del carico di lavoro. L'utilizzo di un'unione di query permette di aggiungere a una query metadati come le credenziali utente.

Sono disponibili due variabili, entrambe sono stringhe che vengono valutate e inviate al database.

#### SQLSessionPrefix

Questa stringa viene inviata al momento della creazione di una connessione al database.

```
SET SQLSessionPrefix = 'SET QUERY_BAND = ' & Chr(39) & 'who=' & OSuser() & ';' & Chr(39) & '
FOR SESSION;';
```

Se, ad esempio, **OSuser()** restituisce **WAIsbt**, questa verrà valutata per **SET QUERY\_BAND = 'who=WA\sbt;'** **FOR SESSION;**, che viene inviata al database quando viene creata la connessione.

#### SQLQueryPrefix

Questa stringa viene inviata per ciascuna singola query.

```
SET SQLSessionPrefix = 'SET QUERY_BAND = ' & Chr(39) & 'who=' & OSuser() & ';' & Chr(39) & '
FOR TRANSACTION;';
```

### Direct DiscoveryVariabili di carattere

#### DirectFieldColumnDelimiter

È possibile impostare il carattere utilizzato come delimitatore di campo nelle istruzioni **Direct Query** per i database che richiedono un carattere diverso dalla virgola come delimitatore di campo. Il carattere specificato deve essere racchiuso tra virgolette semplici nell'istruzione **SET**.

```
SET DirectFieldColumnDelimiter= '|'
```

#### DirectStringQuoteChar

È possibile specificare un carattere da utilizzare per racchiudere tra virgolette le stringhe in una query generata. Il valore predefinito è costituito dalle virgolette singole. Il carattere specificato deve essere racchiuso tra virgolette semplici nell'istruzione **SET**.

```
SET DirectStringQuoteChar= ''';
```

#### DirectIdentifierQuoteStyle

È possibile specificare l'utilizzo di virgolette non ANSI per racchiudere gli identificatori nelle query generate. In tal caso, le uniche virgolette non ANSI disponibili sono GoogleBQ. Il valore predefinito è ANSI. È possibile utilizzare lettere maiuscole, minuscole e miste (ANSI, ansi, Ansi).

```
SET DirectIdentifierQuoteStyle="GoogleBQ";
```

---

## 4 Utilizzo delle variabili nell'editor caricamento dati

---

Ad esempio, le virgolette ANSI vengono utilizzate nella seguente istruzione **SELECT**:

```
SELECT [Quarter] FROM [qvTest].[sales] GROUP BY [Quarter]
```

Quando l'opzione **DirectIdentifierQuoteStyle** è impostata su "GoogleBQ", l'istruzione **SELECT** utilizza le virgolette come indicato di seguito:

```
SELECT [Quarter] FROM [qvTest.sales] GROUP BY [Quarter]
```

### **DirectIdentifierQuoteChar**

È possibile specificare un carattere per controllare le virgolette che racchiudono gli identificatori in una query generata. L'impostazione può includere uno o due caratteri: ad esempio, virgolette doppie o parentesi quadre. L'impostazione predefinita è costituita dalle virgolette doppie.

```
SET DirectIdentifierQuoteChar='[]';
```

```
SET DirectIdentifierQuoteChar='``';
```

```
SET DirectIdentifierQuoteChar=' ';
```

```
SET DirectIdentifierQuoteChar='\"';
```

### **DirectTableBoxListThreshold**

Quando si utilizzano i campi Direct Discovery in una visualizzazione di **tabella**, viene impostata una soglia per limitare il numero di righe visualizzate. La soglia predefinita è 1000 record. L'impostazione predefinita della soglia può essere modificata impostando la variabile **DirectTableBoxListThreshold** nello script di caricamento. Ad esempio:

```
SET DirectTableBoxListThreshold=5000;
```

L'impostazione della soglia è valida solo per le visualizzazioni di **tabella** che contengono campi Direct Discovery. Le visualizzazioni di **tabella** che contengono solo campi in memoria non sono limitate dall'impostazione **DirectTableBoxListThreshold**.

Nella visualizzazione di **tabella** non viene visualizzato alcun campo finché la selezione non include un numero di record inferiore al limite di soglia.

## Variabili di interpretazione numerica Direct Discovery

### **DirectMoneyDecimalSep**

Il separatore decimale definito sostituisce il simbolo decimale per la valuta nell'istruzione SQL generata per caricare i dati con Direct Discovery. Questo carattere deve corrispondere a quello utilizzato in **DirectMoneyFormat**.

Il valore predefinito è ' . '.

### **Esempio:**

```
Set DirectMoneyDecimalSep='.';
```

### **DirectMoneyFormat**

Il simbolo definito sostituisce il formato della valuta nell'istruzione SQL generata per caricare i dati con Direct Discovery. Il simbolo della valuta per il separatore delle migliaia non deve essere incluso.

Il valore predefinito è '#.0000'

### Esempio:

```
Set DirectMoneyFormat='#.0000';
```

### DirectTimeFormat

Il formato dell'ora definito sostituisce quello dell'istruzione SQL generata per caricare i dati con Direct Discovery.

### Esempio:

```
Set DirectTimeFormat='hh:mm:ss';
```

### DirectDateFormat

Il formato della data definito sostituisce quello dell'istruzione SQL generata per caricare i dati con Direct Discovery.

### Esempio:

```
Set DirectDateFormat='MM/DD/YYYY';
```

### DirectTimeStampFormat

Il formato definito sostituisce il formato di data e ora dell'istruzione SQL generata per caricare i dati con Direct Discovery.

### Esempio:

```
Set DirectTimestampFormat='M/D/YY hh:mm:ss[.fff]';
```

## 4.10 Variabili di errore

I valori di tutte le variabili di errore vengono mantenuti anche dopo l'esecuzione dello script. La prima variabile, `ErrorMode`, è l'input dell'utente, mentre le ultime tre sono l'output di Qlik Sense con informazioni sugli errori rilevati nello script.

### Prospetto delle variabili di errore

Ciascuna variabile è descritta ulteriormente dopo la panoramica. È inoltre possibile fare clic sul nome della variabile nella sintassi per accedere immediatamente ai dettagli per tale variabile specifica.

Per ulteriori informazioni sulla variabile, fare riferimento alla Guida in linea di Qlik Sense.

#### ErrorMode

Questa variabile di errore determina quale azione Qlik Sense deve intraprendere in caso di errore durante l'esecuzione dello script.

#### [ErrorMode](#)

#### ScriptError

Questa variabile di errore restituisce il codice di errore nell'istruzione dell'ultimo script eseguito.

### ScriptError

#### **ScriptErrorCount**

Questa variabile di errore restituisce il numero totale di istruzioni che hanno provocato errori durante l'esecuzione dello script attuale. Questa variabile viene sempre reimpostata su 0 all'inizio dell'esecuzione dello script.

### ScriptErrorCount

#### **ScriptErrorList**

Questa variabile di errore conterrà un elenco concatenato di tutti gli errori di script che si sono verificati durante l'ultima esecuzione dello script. Ogni errore è separato da un avanzamento riga.

### ScriptErrorList

## ErrorMode

Questa variabile di errore determina quale azione Qlik Sense deve intraprendere in caso di errore durante l'esecuzione dello script.

Sintassi:

**ErrorMode**

Argomenti:

Argomenti

| Argomento           | Descrizione                                                                                                                                             |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ErrorMode=1</b>  | Impostazione predefinita. L'esecuzione dello script viene interrotta e viene richiesto un intervento da parte dell'utente (modalità non batch).         |
| <b>ErrorMode =0</b> | Qlik Sense ignora l'errore e prosegue nell'esecuzione dello script passando all'istruzione successiva.                                                  |
| <b>ErrorMode =2</b> | Qlik Sense restituirà immediatamente un messaggio di errore "Esecuzione dello script non riuscita...", senza richiedere prima alcuna azione all'utente. |

Esempio:

```
set ErrorMode=0;
```

## ScriptError

Questa variabile di errore restituisce il codice di errore nell'istruzione dell'ultimo script eseguito.

Sintassi:

**ScriptError**

---

## 4 Utilizzo delle variabili nell'editor caricamento dati

---

Questa variabile verrà reimpostata su 0 dopo ogni esecuzione corretta delle istruzioni dello script. Se si verifica un errore, verrà impostato su un codice di errore di Qlik Sense interno. Tali codici sono valori doppi con un componente numerico e uno testuale. Esistono i seguenti codici di errore:

Codici di errore script

| Codice di errore | Descrizione                                   |
|------------------|-----------------------------------------------|
| 0                | Nessun errore. Il testo valore duale è vuoto. |
| 1                | Errore generico.                              |
| 2                | Errore di sintassi.                           |
| 3                | Errore generico ODBC.                         |
| 4                | Errore generico OLE DB.                       |
| 5                | Errore database personalizzato generico.      |
| 6                | Errore generico XML.                          |
| 7                | Errore generico HTML.                         |
| 8                | File non trovato.                             |
| 9                | Database non trovato.                         |
| 10               | Tabella non trovata.                          |
| 11               | Campo non trovato.                            |
| 12               | Formato file errato.                          |
| 16               | Errore semantica.                             |

### Esempio:

```
set ErrorMode=0;

LOAD * from abc.qvf;

if ScriptError=8 then

exit script;

//no file;

end if
```

### ScriptErrorCount

Questa variabile di errore restituisce il numero totale di istruzioni che hanno provocato errori durante l'esecuzione dello script attuale. Questa variabile viene sempre reimpostata su 0 all'inizio dell'esecuzione dello script.

**Sintassi:**

```
ScriptErrorCount
```

### ScriptErrorList

Questa variabile di errore conterrà un elenco concatenato di tutti gli errori di script che si sono verificati durante l'ultima esecuzione dello script. Ogni errore è separato da un avanzamento riga.

**Sintassi:**

```
ScriptErrorList
```

## 5 Espressioni nello script

Le espressioni possono essere utilizzate sia nelle istruzioni **LOAD** che nelle istruzioni **SELECT**. La sintassi e le funzioni descritte in questa sezione riguardano l'istruzione **LOAD** e non l'istruzione **SELECT** poiché quest'ultima viene interpretata dal driver ODBC e non da Qlik Sense. In ogni caso, la maggior parte dei driver ODBC è spesso in grado di interpretare molte delle funzioni descritte di seguito.

Le espressioni sono costituite da funzioni, campi e operatori, combinati in una sintassi.

Tutte le espressioni in uno script di Qlik Sense restituiscono un numero e/o una stringa, a seconda di quale dei due risultati è appropriato. Le funzioni logiche e gli operatori restituiscono 0 per False e -1 per True. Le conversioni da numero a stringa e viceversa sono implicite. Gli operatori logici e le funzioni interpretano 0 come False e tutto il resto come True.

La sintassi generale di un'espressione è:

Sintassi generale

| Espressione              | Campi                           | Operatore |
|--------------------------|---------------------------------|-----------|
| expression ::= (constant | constant                        |           |
| expression ::= (constant | fieldref                        |           |
| expression ::= (constant | operator1 expression            |           |
| expression ::= (constant | expression operator2 expression |           |
| expression ::= (constant | function                        |           |
| expression ::= (constant | ( expression )                  | )         |

dove:

- **constant** è una stringa (un testo, una data o un'ora) racchiusa tra virgolette singole diritte o un numero. Le costanti sono scritte senza separatore delle migliaia e con un punto decimale come separatore decimale.
- **fieldref** è un nome di campo della tabella caricata.
- **operator1** è un operatore unario (che agisce su un'unica espressione, quella a destra).
- **operator2** è un operatore binario (che agisce su due espressioni, una per ogni lato).
- **function ::= functionname( parameters)**
- **parameters ::= expression { , expression }**

Il numero e i tipi dei parametri non sono arbitrari. Dipendono dal tipo di funzione utilizzata.

Le espressioni e le funzioni possono essere nidificate liberamente. Finché l'espressione restituisce un valore interpretabile, in Qlik Sense non verrà visualizzato alcun messaggio di errore.

## 6 Espressioni del grafico

Un'espressione del grafico (visualizzazione) è una combinazione di funzioni, campi e operatori matematici (+ \* / =) e altre misure. Le espressioni vengono utilizzate per elaborare i dati nell'app al fine di produrre un risultato visibile all'interno di una visualizzazione. Il loro utilizzo non è limitato alle misure. È possibile creare visualizzazioni più dinamiche e avanzate con espressioni per titoli, sottotitoli, note a piè di pagina e persino dimensioni.

Questo significa, ad esempio, che al posto del titolo di una visualizzazione costituito da testo statico, è possibile utilizzare un'espressione il cui risultato cambia in base alle selezioni effettuate.



*Per un riferimento dettagliato sulle funzioni degli script e dei grafici, vedere Sintassi dello script e funzioni grafiche.*

### 6.1 Definizione dell'ambito di aggregazione

Vi sono in genere due fattori che concorrono a determinare quali record vengono utilizzati per definire il valore dell'aggregazione in un'espressione. Quando si utilizzano le visualizzazioni, tali fattori sono:

- Valore dimensionale (dell'aggregazione in un'espressione di grafico)
- Selezioni

Insieme, questi fattori definiscono l'ambito dell'aggregazione. È possibile incontrare situazioni in cui si desidera che il calcolo ignori la selezione, la dimensione o entrambe. Nelle funzioni del grafico è possibile raggiungere questo obiettivo utilizzando il qualificatore TOTAL, l'analisi di gruppo o una combinazione dei due. In sintesi:

- Per ignorare o regolare il raggruppamento dimensionale: utilizzare il qualificatore TOTAL
- Per ignorare o regolare la selezione: utilizzare l'analisi di gruppo

## Aggregazione: metodo e descrizione

| Metodo                                  | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Qualificatore TOTAL                     | <p>Utilizzando il qualificatore TOTAL all'interno della funzione di aggregazione viene ignorato il valore dimensionale.</p> <p>Il qualificatore totale può essere utilizzato all'interno di un'aggregazione per ridefinire l'ambito dell'aggregazione, ignorando i valori dimensionali. L'aggregazione verrà eseguita per tutti i valori di campo possibili.</p> <p>Il qualificatore <b>TOTAL</b> può essere seguito da un elenco di uno o più nomi di campo tra parentesi acute.</p> <p>Ad esempio:</p> <pre>Sum(TOTAL &lt;Quarter&gt; {&lt;Year={2013}, Quarter={'Q2'}}&gt; Amount)</pre> <p>Questi nomi di campo devono essere un sottogruppo delle dimensioni del grafico. In questo caso, il calcolo verrà effettuato ignorando tutte le dimensioni del grafico eccetto quelle elencate, ad esempio un valore verrà restituito per ogni combinazione di valori di campo nei campi delle dimensioni elencati. Anche i campi che non sono attualmente una dimensione in un grafico possono essere inclusi nell'elenco. Questo può essere utile in caso di dimensioni di gruppo, dove i campi di dimensione non sono fissi. Se si elencano tutte le dimensioni nel gruppo, la funzione fornirà risultati di aggregazione coerenti quando il livello di drill-down cambia.</p> <p>Per un esempio su come aggiungere eccezioni tra parentesi angolari, vedere <a href="#">Esempio: Qualificatore TOTAL con eccezione e analisi di gruppo (page 299)</a>.</p> |
| Analisi set                             | Se si utilizza l'analisi di gruppo all'interno dell'aggregazione, la selezione verrà ignorata. L'aggregazione verrà eseguita su tutti i valori suddivisi nelle dimensioni.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Qualificatore TOTAL e analisi di gruppo | Se si utilizza il qualificatore <b>TOTAL</b> e l'analisi di gruppo all'interno dell'aggregazione, la selezione viene ignorata e tutte le dimensioni vengono escluse, a eccezione di quelle elencate tra parentesi angolari dopo il qualificatore <b>TOTAL</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Qualificatore ALL                       | <p>Se si utilizza il qualificatore <b>ALL</b> all'interno dell'aggregazione, la selezione e le dimensioni verranno ignorate. È possibile ottenere gli stessi risultati con l'istruzione dell'analisi di gruppo {1} e il qualificatore <b>TOTAL</b>:</p> <pre>=sum(All Sales)</pre> <pre>=sum({1} Total Sales)</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

## Esempio: Qualificatore TOTAL

Nell'esempio seguente viene mostrato come TOTAL può essere utilizzato per calcolare una quota relativa. Presupponendo che sia stato selezionato Q2, se si utilizza TOTAL viene calcolata la somma di tutti i valori ignorando le dimensioni.

Esempio: qualificatore totale

| Year | Quarter | Sum(Amount) | Sum(TOTAL Amount) | Sum(Amount)/Sum(TOTAL Amount) |
|------|---------|-------------|-------------------|-------------------------------|
|      |         | 3000        | 3000              | 100%                          |
| 2012 | Q2      | 1700        | 3000              | 56,7%                         |
| 2013 | Q2      | 1300        | 3000              | 43,3%                         |



Per visualizzare i numeri sotto forma di percentuale, nel pannello delle proprietà relativo alla misura che si desidera visualizzare come valore percentuale, in **Formattazione numero**, selezionare **Numero** e da **Formattazione**, scegliere **Semplice** e uno dei formati %.

## Esempio: analisi di gruppo

Nell'esempio seguente viene mostrato come l'analisi di gruppo può essere utilizzata per eseguire un confronto tra serie di dati prima che venga eseguita qualsiasi selezione. Presupponendo che sia stato selezionato Q2, se si utilizza l'analisi di gruppo con la definizione di gruppo {1}, viene calcolata la somma di tutti i valori ignorando eventuali selezioni, ma operando una suddivisione in base alle dimensioni.

Esempio: analisi di gruppo

| Year | Quarter | Sum(Amount) | Sum({1} Amount) | Sum(Amount)/Sum({1} Amount) |
|------|---------|-------------|-----------------|-----------------------------|
|      |         | 3000        | 10800           | 27,8%                       |
| 2012 | Q1      | 0           | 1100            | 0%                          |
| 2012 | Q3      | 0           | 1400            | 0%                          |
| 2012 | Q4      | 0           | 1800            | 0%                          |
| 2012 | Q2      | 1700        | 1700            | 100%                        |
| 2013 | Q1      | 0           | 1000            | 0%                          |
| 2013 | Q3      | 0           | 1100            | 0%                          |
| 2013 | Q4      | 0           | 1400            | 0%                          |
| 2013 | Q2      | 1300        | 1300            | 100%                        |

## Esempio: qualificatore TOTAL e analisi di gruppo (identificatore {1})

Nell'esempio seguente viene mostrato come l'analisi di gruppo e il qualificatore TOTAL possono essere combinati per eseguire un confronto tra i set di dati prima che venga eseguita qualsiasi selezione e per tutte le dimensioni. Presupponendo che sia stato selezionato Q2, se si utilizza l'analisi di gruppo con la definizione di gruppo e il qualificatore TOTAL calcola la somma di tutti i valori ignorando eventuali selezioni e le dimensioni.

Esempio: qualificatore TOTAL e analisi di gruppo

| Year | Quarter | Sum<br>(Amount) | Sum({1} TOTAL<br>Amount) | Sum(Amount)/Sum({1} TOTAL<br>Amount) |
|------|---------|-----------------|--------------------------|--------------------------------------|
|      |         | 3000            | 10800                    | 27,8%                                |
| 2012 | Q2      | 1700            | 10800                    | 15,7%                                |
| 2013 | Q2      | 1300            | 10800                    | 12%                                  |

## Esempio: Qualificatore TOTAL e analisi di gruppo (confronto dei dati rispetto a un valore di riferimento)

L'esempio seguente mostra come è possibile utilizzare l'analisi di gruppo e il qualificatore TOTAL per creare delle colonne di riferimento nella tabella, per semplificare il confronto delle metriche a livello di riga. Per rendere i dati più facili da capire e da utilizzare, è necessario assegnare un'etichetta alle colonne di misura.

I dati del secondo trimestre del 2013 sono utilizzati come riferimento per il confronto con tutti gli altri valori nella tabella.

### Procedere come indicato di seguito:

1. Aggiungere una tabella al foglio.
2. Aggiungere i seguenti campi come dimensioni:
  - Year
  - Quarter
3. Aggiungere la misura seguente:  
Sum(Amount)
4. Nelle proprietà della misura appena aggiunta, assegnare la seguente etichetta alla misura: *Total Sales*
5. Aggiungere la misura seguente:  
Sum(TOTAL {<Year={2013}, Quarter={'Q2'}>} Amount)
6. Nelle proprietà della misura appena aggiunta, assegnare la seguente etichetta alla misura:  
*Reference Quarter (2013-Q2) Total Sales*
7. Aggiungere la misura seguente:  
round((sum(Amount) / sum(total {<Year={2013}, Quarter={'Q2'}>} Amount))\*100,'0.01') & '%'
8. Nelle proprietà della misura appena aggiunta, assegnare la seguente etichetta alla misura:  
*Rounded Quotient - Total Sales Compared to Reference Quarter (2013-Q2)*

Senza alcuna selezione applicata, la tabella dovrebbe avere il seguente aspetto:

Esempio: Qualificatore TOTAL e analisi di riferimento (colonna di riferimento)

| Year | Quarter | Total Sales | Reference Quarter (2013-Q2) Total Sales | Rounded Quotient - Total Sales Compared to Reference Quarter (2013-Q2) |
|------|---------|-------------|-----------------------------------------|------------------------------------------------------------------------|
| 2012 | Q1      | 1100        | 1300                                    | 84.62%                                                                 |
| 2012 | Q2      | 1700        | 1300                                    | 130.77%                                                                |
| 2012 | Q3      | 1400        | 1300                                    | 107.69%                                                                |
| 2012 | Q4      | 1800        | 1300                                    | 138.46%                                                                |
| 2013 | Q1      | 1000        | 1300                                    | 76.92%                                                                 |
| 2013 | Q2      | 1300        | 1300                                    | 100.00%                                                                |
| 2013 | Q3      | 1100        | 1300                                    | 84.62%                                                                 |
| 2013 | Q4      | 1400        | 1300                                    | 107.69%                                                                |

## Esempio: Qualificatore TOTAL con eccezione e analisi di gruppo

L'esempio seguente mostra come utilizzare il qualificatore TOTAL, ma con un'eccezione al qualificatore indicata tra parentesi angolari. Le espressioni di esempio utilizzano anche l'analisi di gruppo.

**Procedere come indicato di seguito:**

1. Aggiungere una tabella al foglio.
2. Aggiungere i seguenti campi come dimensioni:
  - *Year*
  - *Quarter*
3. Aggiungere la misura seguente:  
`Sum(Amount)`
4. Nelle proprietà della misura appena aggiunta, assegnare la seguente etichetta alla misura: *Total Sales*
5. Aggiungere la misura seguente:  
`Sum(TOTAL <Quarter> {<Year={2013}, Quarter={'Q2'}>} Amount)`
6. Nelle proprietà della misura appena aggiunta, assegnare la seguente etichetta alla misura:  
*Reference Quarter (2013-Q2) Total Sales*

Esempi: Qualificatore TOTAL con eccezione e analisi di gruppo

| Year | Quarter | Total Sales | Reference Quarter (2013-Q2) Total Sales |
|------|---------|-------------|-----------------------------------------|
| 2012 | Q1      | 1100        | 0                                       |
| 2012 | Q2      | 1700        | 1300                                    |
| 2012 | Q3      | 1400        | 0                                       |
| 2012 | Q4      | 1800        | 0                                       |

| Year | Quarter | Total Sales | Reference Quarter (2013-Q2) Total Sales |
|------|---------|-------------|-----------------------------------------|
| 2013 | Q1      | 1000        | 0                                       |
| 2013 | Q2      | 1300        | 1300                                    |
| 2013 | Q3      | 1100        | 0                                       |
| 2013 | Q4      | 1400        | 0                                       |

L'espressione della colonna di riferimento viene interpretata come segue:

- L'espressione di gruppo {<Year={2013}, quarter={'Q2'}>} limita i dati di input ai record con il valore 2013 per Year e Q2 per quarter.
- Il qualificatore TOTAL assicura che tutte le dimensioni della tabella vengano ignorate. Tuttavia, con l'identificatore <quarter>, viene fatta un'eccezione per la dimensione quarter.

## Dati utilizzati negli esempi di questa pagina:

Dati utilizzati negli esempi:

```
AggregationScope:
LOAD * inline [
Year Quarter Amount
2012 Q1 1100
2012 Q2 1700
2012 Q3 1400
2012 Q4 1800
2013 Q1 1000
2013 Q2 1300
2013 Q3 1100
2013 Q4 1400] (delimiter is ' ');
```

## 6.2 Analisi di gruppo

Quando si effettua una selezione in un'app, si definisce un sottogruppo di record nei dati. Le funzioni di aggregazione, come `Sum()`, `Max()`, `Min()`, `Avg()` e `Count()`, vengono calcolate in base a questo sottogruppo.

In altre parole, la selezione dell'utente definisce l'ambito dell'aggregazione; definisce il set di record su cui vengono effettuati i calcoli.

L'analisi di gruppo offre un modo per definire un ambito diverso dal set di record definito dalla selezione corrente. Tale nuovo ambito può anche essere considerato come una selezione alternativa.

Ciò può essere utile se si desidera confrontare la selezione corrente con un particolare valore, ad esempio il valore per lo scorso anno o la quota di mercato globale.

## Espressioni set

Le espressioni set possono essere utilizzate all'interno e all'esterno delle funzioni di aggregazione, racchiudendole tra parentesi graffe.

### Esempio: Espressione set interna

```
Sum( {$<Year={2021}>} Sales )
```

### Esempio: Espressione set esterna

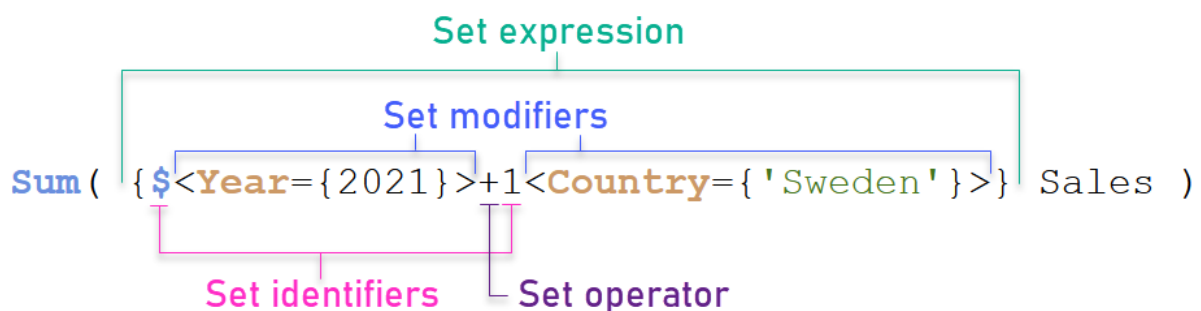
```
{<Year={2021}>} Sum(Sales) / Count(distinct Customer)
```

Un'espressione set è composta da una combinazione dei seguenti elementi:

- **Identificatori.** Un identificatore set rappresenta una selezione, definita altrove. Rappresenta anche un set specifico di record nei dati. Può trattarsi della selezione corrente, di una selezione da un segnalibro o di una selezione da uno stato alternato. Un'espressione di gruppo semplice è composta da un unico identificatore, come ad esempio il simbolo del dollaro, {\$}, che rappresenta tutti i record della selezione attuale.  
Esempi: \$, 1, BookMark1, State2
- **Operatori.** Un operatore set può essere utilizzato per creare unioni, differenze o intersezioni tra diversi identificatori set. In questo modo, è possibile creare un sottogruppo o un sovragruppo delle selezioni definite dagli identificatori set.  
Esempi: +, -, \*, /
- **Modificatori.** Un modificatore set può essere aggiunto all'identificatore set per modificarne la selezione. Un modificatore può anche essere utilizzato autonomamente, in modo che modifichi successivamente l'identificatore predefinito. Un modificatore deve essere racchiuso tra parentesi angolari <...>.  
Esempi: <Year={2020}><Supplier={ACME}>
- **Flag di conservazione del set vuoto.** Un elemento avanzato che può essere utilizzato in qualsiasi espressione set esterna, ad eccezione dell'ultimo elemento di una catena di espressioni set esterne. Questo flag annulla la cancellazione implicita del set per tutti i set definiti in queste espressioni set esterne, semplificando la configurazione delle espressioni avanzate.  
Sintassi: &

Gli elementi sono combinati in modo da formare espressioni set.

*Elementi in un'espressione set*



L'espressione set di cui sopra, ad esempio, viene costruita dall'aggregazione `sum(Sales)`.

Il primo operando restituisce le vendite per l'anno 2021 per la selezione corrente, indicato dall'identificatore `set $` e dal modificatore contenente la selezione dell'anno 2021. Il secondo operando restituisce `sa1es` per Sweden, e ignora la selezione corrente, indicata dall'identificatore `set 1`.

Infine, l'espressione restituisce un set composto dai record appartenenti a uno qualsiasi dei due operandi `set`, come indicato dall'operatore `set +`.

### Esempi

Esempi che combinano gli elementi espressione `set` sopra disponibili nei seguenti argomenti:

### Set naturali

In genere, un'espressione `set` rappresenta sia un set di record nel modello dati, sia una selezione che definisce tale sottogruppo di dati. In questo caso, il set viene definito *set naturale*.

Gli identificatori `set`, con o senza modificatori `set`, rappresentano sempre *set naturali*.

Tuttavia, un'espressione `set` che utilizza operatori di gruppo rappresenta anche un sottogruppo dei record, ma in genere non può comunque essere descritta usando una selezione di valori di campo. Tale espressione rappresenta un *set non naturale*.

Ad esempio, il set fornito da `{1-$}` non può essere sempre definito da una selezione. Non rappresenta dunque un *set naturale*. Ciò può essere mostrato caricando i seguenti dati, aggiungendoli a una tabella ed effettuando quindi selezioni usando le caselle di filtro.

```
Load * Inline
[Dim1, Dim2, Number
A, X, 1
A, Y, 1
B, X, 1
B, Y, 1];
```

Effettuando selezioni per `Dim1` e `Dim2`, si ottiene la visualizzazione mostrata nella tabella seguente.

*Tabella con set naturali e non naturali*

| Dim1 | Dim2 | Sum({\$} Number) | Sum({1-\$} Number) |
|------|------|------------------|--------------------|
| A    | X    | 1                | 0                  |
| A    | Y    | 0                | 1                  |
| B    | X    | 0                | 1                  |
| B    | Y    | 0                | 1                  |

L'espressione `set` nella prima misura utilizza un *set naturale*; corrisponde alla selezione effettuata `{$}`.

La seconda misura è diversa. Utilizza `{1-$}`. Non è possibile effettuare una selezione corrispondente a questo set, pertanto rappresenta un *set non naturale*.

Tale distinzione ha una serie di conseguenze:

- I modificatori set possono essere applicati solo agli identificatori set. Non possono essere applicati a un'espressione set arbitraria. Ad esempio, non è possibile utilizzare un'espressione set come:  
`{ (BM01 * BM02) <Field={x,y}> }`  
 Qui, le normali parentesi (tonde) implicano che l'intersezione tra BM01 e BM02 debba essere valutata prima che sia possibile applicare il modificatore set. Il motivo è l'assenza di un set di elementi modificabili.
- Non è possibile utilizzare set non naturali all'interno delle funzioni di elementi `P()` e `E()`. Tali funzioni restituiscono un set di elementi, ma non è possibile dedurre il set di elementi da un set non naturali.
- Una misura che utilizza un set non naturale non sempre può essere attribuita al giusto valore dimensionale se il modello dati presenta molte tabelle. Ad esempio, nel grafico seguente, alcuni numeri esclusi sulle vendite sono attribuiti al valore Country corretto, mentre altri hanno NULL come Country.

*Grafico con set non naturale*

| ProductCategory    | Country | Sum([\$] Sales) | Sum([1-\$] Sales) |
|--------------------|---------|-----------------|-------------------|
| Baby Clothes       |         | 127791.28       | 0                 |
| Children's Clothes |         | 0               | 81681.54          |
| Men's Clothes      |         | 0               | 140987.45         |
| Men's Footwear     |         | 0               | 232747.44         |
| Sportswear         |         | 0               | 270272.76         |
| Swimwear           |         | 0               | 29548.6           |
| Women's Clothes    |         | 0               | 649348.5          |
| Women's Footwear   |         | 0               | 140654.44         |
| -                  |         | 0               | 131935.86         |
| Belgium            |         | 0               | 1005.02           |
| Germany            |         | 0               | 773.3             |
| Portugal           |         | 0               | 1279.74           |

Il fatto che l'assegnazione venga effettuata correttamente o meno dipende dal modello dati. In questo caso, il numero non può essere assegnato se appartiene a un paese escluso dalla selezione.

| Identificatore | Descrizione                                                                                                                                         |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| 1              | Rappresenta la serie completa di tutti i record nell'applicazione, indipendentemente dalle selezioni eseguite.                                      |
| \$             | Rappresenta i record della selezione attuale. L'espressione set <code>{ \$ }</code> equivale pertanto alla non dichiarazione di un'espressione set. |
| \$1            | Rappresenta la selezione precedente. <code>\$2</code> rappresenta la selezione precedente, con uno scarto di uno, e così via.                       |

| Identificatore | Descrizione                                                                                                                   |
|----------------|-------------------------------------------------------------------------------------------------------------------------------|
| \$_1           | Rappresenta la selezione successiva (in avanti). \$_2 rappresenta la selezione successiva, con uno scarto di uno, e così via. |
| BM01           | È possibile utilizzare qualsiasi ID o nome del segnalibro.                                                                    |
| MyAltState     | È possibile fare riferimento alle selezioni effettuate in uno stato alternativo utilizzando il nome del relativo stato.       |

| Esempio              | Risultato                                                                              |
|----------------------|----------------------------------------------------------------------------------------|
| sum ({1} Sales)      | Restituisce le vendite totali per l'app, ignorando le selezioni, ma non la dimensione. |
| sum ({ \$ } Sales)   | Restituisce le vendite per la selezione attuale, vale a dire lo stesso di sum(Sales).  |
| sum ({ \$ 1 } Sales) | Restituisce le vendite per la selezione precedente.                                    |
| sum ({ BM01 } Sales) | Restituisce le vendite per il segnalibro denominato <i>BM01</i> .                      |

| Esempio                                        | Risultato                                                                                                                   |
|------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| sum({ \$ <OrderDate = DeliveryDate> } Sales)   | Restituisce le vendite per la selezione attuale in cui OrderDate = DeliveryDate.                                            |
| sum({ 1 <Region = {US}> } Sales)               | Restituisce le vendite per la regione US ignorando la selezione attuale.                                                    |
| sum({ \$ <Region = > } Sales)                  | Restituisce le vendite per la selezione, da cui è stata però rimossa la selezione in 'Region'.                              |
| sum({ <Region = > } Sales)                     | Restituisce lo stesso risultato dell'esempio precedente. Quando il gruppo da modificare viene omissso, viene utilizzato \$. |
| sum({ \$ <Year={2000}, Region="{U*}"> } Sales) | Restituisce le vendite per la selezione corrente, ma con nuove selezioni sia in <i>Year</i> che in <i>Region</i> .          |

## Identificatori set

Un identificatore set rappresenta un set di record nei dati; tutti i dati o un sottogruppo di essi. Rappresenta il set di record definiti da una selezione. Può trattarsi della selezione corrente, di tutti i dati (nessuna selezione), di una selezione da un segnalibro, o di una selezione da uno stato alternato.

Nell'esempio `sum( { $ <Year = {2009}> } sales )`, l'identificatore è il simbolo del dollaro: \$. Ciò rappresenta la selezione corrente. Rappresenta anche tutti i possibili record. Questo set può quindi essere alterato dalla parte con il modificatore dell'espressione set: la selezione 2009 in *year* viene aggiunta.

L'identificatore set \$ equivale a non indicare un identificatore set. Ad esempio, con l'esempio precedente, l'espressione `sum( { $ <Year = {2009}> } sales )` è equivalente a `sum( { <Year = {2009}> } sales )`.

In un'espressione set più complessa, è possibile utilizzare insieme due identificatori con un operatore per formare un'unione, oppure un'intersezione dei due set di record.

La tabella seguente mostra alcuni identificatori comuni.

Esempi con identificatori comuni

| Identificatore                        | Descrizione                                                                                                                                                             |
|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1                                     | Rappresenta la serie completa di tutti i record nell'applicazione, indipendentemente dalle selezioni eseguite.                                                          |
| \$ (oppure nessun identificatore set) | Rappresenta i record della selezione corrente nello stato predefinito. L'espressione set {\$} equivale pertanto in genere alla non dichiarazione di un'espressione set. |
| \$1                                   | Rappresenta la selezione precedente nello stato predefinito. \$2 rappresenta la selezione precedente con uno scarto di uno, e così via.                                 |
| \$_1                                  | Rappresenta la prossima selezione (in avanti). \$_2 rappresenta la selezione successiva, con uno scarto di uno, e così via.                                             |
| BM01                                  | È possibile utilizzare qualsiasi ID o nome del segnalibro.                                                                                                              |
| AltState                              | È possibile fare riferimento a uno stato alternato mediante il proprio nome stato.                                                                                      |
| AltState::BM01                        | Un segnalibro contiene le selezioni di tutti gli stati, ed è possibile creare un riferimento a un segnalibro specifico qualificandone il nome.                          |

La tabella seguente mostra esempi con identificatori diversi.

Esempi con identificatori diversi

| Esempio              | Risultato                                                                              |
|----------------------|----------------------------------------------------------------------------------------|
| Sum ({1} Sales)      | Restituisce le vendite totali per l'app, ignorando le selezioni, ma non la dimensione. |
| Sum ({ \$ } Sales)   | Restituisce le vendite per la selezione attuale, vale a dire lo stesso di Sum(Sales).  |
| Sum ({ \$1 } Sales)  | Restituisce le vendite per la selezione precedente.                                    |
| Sum ({ BM01 } Sales) | Restituisce le vendite per il segnalibro denominato BM01.                              |

## Operatori set

Gli operatori set sono utilizzati per includere, escludere o intersecare serie di dati. Tutti gli operatori utilizzano i gruppi come operandi e restituiscono un gruppo come risultato.

È possibile utilizzare gli operatori set in due situazioni diverse:

- Per eseguire un'operazione set sugli identificatori set, rappresentando i set di record nei dati.
- Per eseguire un'operazione sui set di elementi, sui valori di campo o all'interno di un modificatore set.

La tabella seguente mostra gli operatori utilizzabili nelle espressioni set.

### Operatori

| Operatore | Descrizione                                                                                                                                                                                                                                                 |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| +         | Unione. Questa operazione binaria restituisce un gruppo costituito dai record o dagli elementi che appartengono a uno qualsiasi dei due operandi set.                                                                                                       |
| -         | Esclusione. Questa operazione binaria restituisce un gruppo costituito dai record o dagli elementi che appartengono solo al primo dei due operandi set e non al secondo. Inoltre, se utilizzata come operatore unario, restituisce il gruppo complementare. |
| *         | Intersezione. Questa operazione binaria restituisce un gruppo costituito dai record o dagli elementi che appartengono a entrambi gli operandi set.                                                                                                          |
| /         | Differenza simmetrica (XOR). Questa operazione binaria restituisce un gruppo costituito dai record o dagli elementi che appartengono a uno dei due operandi set, ma non a entrambi.                                                                         |

La tabella seguente mostra esempi con operatori.

### Esempi con operatori

| Esempio                                                                    | Risultato                                                                                                                                                 |
|----------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>Sum ({1-\$} Sales)</code>                                            | Restituisce le vendite per tutto ciò che è escluso dalla selezione corrente.                                                                              |
| <code>Sum ({*\$BM01} Sales)</code>                                         | Restituisce le vendite per l'intersezione tra la selezione e il preferito #160;BM01.                                                                      |
| <code>Sum ({-(\$+BM01)} Sales)</code>                                      | Restituisce le vendite escluse dalla selezione e dal segnalibro BM01.                                                                                     |
| <code>Sum ({&lt;Year={2009}&gt;+1&lt;Country={'Sweden'}&gt;} Sales)</code> | Restituisce le vendite per l'anno 2009 associate alle selezioni correnti e aggiunge il set di dati completo associato al paese Sweden per tutti gli anni. |
| <code>Sum ({&lt;Country={'S*'}+{'*land'}&gt;} Sales)</code>                | Restituisce le vendite per i paesi che iniziano con s o terminano con land.                                                                               |

## Modificatori set

Le espressioni set vengono utilizzate per definire l'ambito di un calcolo. La parte centrale dell'espressione set è il modificatore set che specifica una selezione. Ciò è utilizzabile per modificare la selezione utente, oppure la selezione nell'identificatore set, e il risultato definisce un nuovo ambito per il calcolo.

Il modificatore del set è composto da uno o più nomi di campo, ciascuno seguito da una selezione che dovrebbe essere effettuata sul campo. Il modificatore è racchiuso da parentesi angolari: < >

Ad esempio:

- `Sum ( {<Year = {2015}>} Sales )`
- `Count ( {1<Country = {Germany}>} distinct OrderID )`
- `Sum ( {<Year = {2015}, Country = {Germany}>} Sales )`

### Set di elementi

Un set di elementi può essere definito usando quanto segue:

- Un elenco di valori
- Una ricerca
- Un riferimento a un altro campo
- Una funzione set

Se la definizione del set elementi viene omessa, il modificatore set cancellerà qualsiasi selezione in questo campo. Ad esempio:

```
Sum( {$<Year = >} Sales )
```

### Esempi: espressioni del grafico per i modificatori set basati su set di elementi

Esempi: espressioni del grafico

#### Script di caricamento

Caricare i dati seguenti come un caricamento inline nell'editor caricamento dati per creare gli esempi di espressione del grafico in basso.

```
MyTable:
Load * Inline [
Country, Year, Sales
Argentina, 2014, 66295.03
Argentina, 2015, 140037.89
Austria, 2014, 54166.09
Austria, 2015, 182739.87
Belgium, 2014, 182766.87
Belgium, 2015, 178042.33
Brazil, 2014, 174492.67
Brazil, 2015, 2104.22
Canada, 2014, 101801.33
Canada, 2015, 40288.25
Denmark, 2014, 45273.25
Denmark, 2015, 106938.41
Finland, 2014, 107565.55
Finland, 2015, 30583.44
France, 2014, 115644.26
France, 2015, 30696.98
Germany, 2014, 8775.18
Germany, 2015, 77185.68
];
```

### Espressioni del grafico

Generare una tabella in un foglio Qlik Sense con le seguenti espressioni del grafico.

Tabella - modificatori set basati su set di elementi

| Paese     | Somma<br>(Vendite) | Somma<br>({1<Paese=<br>{Belgium}>}<br>Vendite) | Somma<br>({1<Paese=<br>{"*A*"}>}<br>Vendite) | Somma<br>({1<Paese=<br>{"A*"}>}<br>Vendite) | Somma<br>({1<Anno=<br>{\$(=Max<br>(Anno))}>}<br>Vendite) |
|-----------|--------------------|------------------------------------------------|----------------------------------------------|---------------------------------------------|----------------------------------------------------------|
| Totali    | 1645397.3          | 360809.2                                       | 1284588.1                                    | 443238.88                                   | 788617.07                                                |
| Argentina | 206332.92          | 0                                              | 206.332,92                                   | 206.332,92                                  | 140037.89                                                |
| Austria   | 236905.96          | 0                                              | 236905.96                                    | 236905.96                                   | 182739.87                                                |
| Belgio    | 360809.2           | 360809.2                                       | 0                                            | 0                                           | 178042.33                                                |
| Brasile   | 176596.89          | 0                                              | 176596.89                                    | 0                                           | 2104.22                                                  |
| Canada    | 142089.58          | 0                                              | 142089,58                                    | 0                                           | 40288.25                                                 |
| Danimarca | 152211.66          | 0                                              | 152211,66                                    | 0                                           | 106938.41                                                |
| Finlandia | 138148.99          | 0                                              | 138148,99                                    | 0                                           | 30583.44                                                 |
| Francia   | 146341.24          | 0                                              | 146.341,24                                   | 0                                           | 30696.98                                                 |
| Germania  | 85960.86           | 0                                              | 85.960,86                                    | 0                                           | 77185.68                                                 |

### Spiegazione

- Dimensioni:
  - Country
- Misure:
  - Sum(Sales)  
Sum sales senza espressione set.
  - Sum({1<Country={Belgium}>}Sales)  
Selezionare Belgium, quindi sommare il corrispondente sales.
  - Sum({1<Country={"\*A\*"}>}Sales)  
Selezionare tutti i paesi che hanno un A, quindi sommare il corrispondente sales.
  - Sum({1<Country={"A\*"}>}Sales)  
Selezionare tutti i paesi che iniziano con un A, quindi sommare il corrispondente sales.
  - Sum({1<Year={\$(=Max(Year))}>}Sales)  
Calcolare il valore Max(Year), che corrisponde a 2015, quindi sommare il valore corrispondente sales.

Modificatori set basati su set di elementi

My new sheet

| Country       | Sum (Sales)      | Sum( {1<Country = {Belgium}>} Sales ) | Sum( {1<Country = {"*A*"}>} Sales ) | Sum( {1<Country = {"A*"}>} Sales ) | Sum( {1<Year = {\$(=Max(Year))}>} Sales ) |
|---------------|------------------|---------------------------------------|-------------------------------------|------------------------------------|-------------------------------------------|
| <b>Totals</b> | <b>1645397.3</b> | <b>360809.2</b>                       | <b>1284588.1</b>                    | <b>443238.88</b>                   | <b>788617.07</b>                          |
| Argentina     | 206332.92        | 0                                     | 206332.92                           | 206332.92                          | 140037.89                                 |
| Austria       | 236905.96        | 0                                     | 236905.96                           | 236905.96                          | 182739.87                                 |
| Belgium       | 360809.2         | 360809.2                              | 0                                   | 0                                  | 178042.33                                 |
| Brazil        | 176596.89        | 0                                     | 176596.89                           | 0                                  | 2104.22                                   |
| Canada        | 142089.58        | 0                                     | 142089.58                           | 0                                  | 40288.25                                  |
| Denmark       | 152211.66        | 0                                     | 152211.66                           | 0                                  | 106938.41                                 |
| Finland       | 138148.99        | 0                                     | 138148.99                           | 0                                  | 30583.44                                  |
| France        | 146341.24        | 0                                     | 146341.24                           | 0                                  | 30696.98                                  |
| Germany       | 85960.86         | 0                                     | 85960.86                            | 0                                  | 77185.68                                  |

## Valori elencati

L'esempio più comune di un set di elementi è quello basato su un elenco di valori di campo racchiusi tra parentesi graffe. Ad esempio:

- {<Country = {Canada, Germany, Singapore}>}
- {<Year = {2015, 2016}>}

Le parentesi graffe interne definiscono il set di elementi. I singoli valori sono separati da virgole.

## Distinzione per virgolette e maiuscole/minuscole

Se i valori contengono caratteri vuoti o speciali, i valori devono essere racchiusi tra virgolette. Le virgolette singole indicano una corrispondenza letterale, con distinzione tra maiuscole e minuscole, con un valore di campo singolo. Le virgolette doppie implicano una corrispondenza senza distinzione tra maiuscole e minuscole in uno o più valori di campo. Ad esempio:

- <Country = {'New Zealand'}>  
Corrisponde solo a New Zealand.
- <Country = {"New Zealand"}>  
Corrisponde a New Zealand, NEW ZEALAND e new zealand.

Le date devono essere racchiuse tra virgolette e utilizzare il formato data del campo in questione. Ad esempio:

- <ISO\_Date = {'2021-12-31'}>
- <US\_Date = {'12/31/2021'}>
- <UK\_Date = {'31/12/2021'}>

Le virgolette doppie possono essere sostituite dalle parentesi quadre o da accenti gravi.

### Ricerche

I set di elementi possono anche essere creati attraverso le ricerche. Ad esempio:

- `<Country = {"C*"}>`
- `<Ingredient = {"*garlic*"}>`
- `<Year = {">2015"}>`
- `<Date = {">12/31/2015"}>`

Nelle ricerche di testo è possibile utilizzare i caratteri jolly: un asterisco (\*) rappresenta un numero qualsiasi di caratteri, mentre un punto interrogativo (?) rappresenta un carattere singolo. Per definire le ricerche numeriche possono essere utilizzati gli operatori relazionali.

Utilizzare sempre le virgolette doppie per le ricerche. Le ricerche non distinguono tra maiuscole e minuscole.

### Espansioni con dollaro

Le espansioni con dollaro sono necessarie se si desidera utilizzare un calcolo all'interno del proprio set di elementi. Ad esempio, se si desidera ricercare solo l'ultimo anno possibile, è possibile usare:

```
<Year = {$(=Max(Year))}>
```

### Valori selezionati in altri campi

I modificatori possono essere basati sui valori selezionati di un altro campo. Ad esempio:

```
<OrderDate = DeliveryDate>
```

Questo modificatore prenderà i valori selezionati da `DeliveryDate` e li applicherà come una selezione a `OrderDate`. Se sono presenti molti valori diversi, più di duecento, evitare di eseguire questa operazione, in quanto prevede l'uso di una quantità elevata di CPU.

### Funzioni dei set di elementi

Il set di elementi può anche basarsi sulle funzioni `set P()` (possibili valori) e `E()` (valori esclusi).

Ad esempio, se si desidera selezionare i paesi in cui è stato venduto il prodotto `cap`, è possibile utilizzare:

```
<Country = P({1<Product={Cap}>} Country)>
```

In modo simile, se si desidera selezionare i paesi in cui il prodotto `cap` non è stato venduto, è possibile utilizzare:

```
<Country = E({1<Product={Cap}>} Country)>
```

### Modificatori set con ricerche

È possibile creare set di elementi attraverso le ricerche con i modificatori `set`.

Ad esempio:

- `<Country = {"C*"}>`
- `<Year = {">2015"}>`
- `<Ingredient = {"*garlic*"}>`

Le ricerche devono sempre essere racchiuse tra virgolette doppie, parentesi quadre o accenti gravi. È possibile utilizzare un elenco con un mix di stringhe letterali (virgolette singole) e ricerche (virgolette doppie). Ad esempio:

```
<Product = {'Nut', "*Bolt", washer}>
```

### Ricerche testuali

Nelle ricerche testuali possono essere utilizzati i caratteri jolly e altri simboli:

- Un asterisco (\*) rappresenterà un numero qualsiasi di caratteri.
- Un punto interrogativo (?) rappresenterà un carattere singolo.
- Un accento circonflesso (^) contrassegnerà l'inizio di una parola.

Ad esempio:

- `<Country = {"C*", "*land"}>`  
Mostra le corrispondenze di tutti i paesi che iniziano con un c o terminano con land.
- `<Country = {"*^z*"}>`  
Ciò corrisponderà a tutti i paesi che hanno una parola che inizia con z, come New Zealand.

### Ricerche numeriche

È possibile effettuare ricerche numeriche usando questi operatori relazionali: `>`, `>=`, `<`, `<=`

Una ricerca numerica inizia sempre con uno di questi operatori. Ad esempio:

- `<Year = {">2015"}>`  
Corrisponde al 2016 e a tutti gli anni successivi.
- `<Date = {">=1/1/2015<1/1/2016"}>`  
Corrisponde a tutte le date durante il 2015. Notare la sintassi per la descrizione di un intervallo temporale tra due date. Il formato della data deve corrispondere al formato data del campo in questione.

### Ricerche tramite espressioni

È possibile utilizzare le ricerche tramite espressioni per effettuare ricerche più avanzate. Un'aggregazione viene quindi valutata per ciascun valore di campo nel campo di ricerca. Verranno selezionati tutti i valori per i quali l'espressione di ricerca restituisce valori veri.

Una ricerca tramite espressione inizia sempre con il segno uguale: `=`

Ad esempio:

```
<Customer = {"=Sum(Sales)>1000"}>
```

Ciò restituirà tutti i clienti con un valore delle vendite superiore a 1000. `Sum(Sales)` è calcolato sulla selezione corrente. Ciò significa che se si dispone di una selezione in un altro campo, come il campo `Product`, si otterranno i clienti che hanno soddisfatto la condizione di vendita solo per i prodotti selezionati.

Se si desidera che la condizione sia indipendente dalla selezione, è necessario utilizzare l'analisi set all'interno della stringa di ricerca. Ad esempio:

```
<Customer = {"=Sum({1} Sales)>1000"}>
```

Le espressioni dopo il segno uguale verranno interpretate come un valore booleano. Ciò significa che in caso di valutazione di qualcosa di diverso, qualsiasi numero diverso da zero verrà interpretato come vero, mentre lo zero e le stringhe vengono interpretate come false.

### Virgolette

Utilizzare le virgolette quando le stringhe di ricerca contengono caratteri vuoti o speciali. Le virgolette singole implicano una corrispondenza letterale, con distinzione tra maiuscole e minuscole, con un valore di campo singolo. Le virgolette doppie implicano una ricerca senza distinzione tra maiuscole e minuscole che corrisponde potenzialmente a più valori di campo.

Ad esempio:

- `<Country = {'New Zealand'}>`  
Corrispondenza solo con New Zealand.
- `<Country = {"New Zealand"}>`  
Corrispondenza con New Zealand, NEW ZEALAND e new zealand

Le virgolette doppie possono essere sostituite dalle parentesi quadre o da accenti gravi.



*Nelle precedenti versioni di Qlik Sense non vi era distinzione tra virgolette singole e virgolette doppie e tutte le stringhe racchiuse tra virgolette venivano trattate come ricerche. Per questione di compatibilità, le app create con le versioni precedenti di Qlik Sense continueranno a funzionare come prima. Le app create con Qlik Sense November 2017 o versioni successive rispetteranno la differenza tra i due tipi di virgolette.*

### Esempi: le espressioni del grafico per i modificatori set con ricerche

Esempi: espressioni del grafico

#### Script di caricamento

Caricare i dati seguenti come un caricamento inline nell'editor caricamento dati per creare gli esempi di espressione del grafico in basso.

```
MyTable:
Load
Year(Date) as Year,
Date#(Date,'YYYY-MM-DD') as ISO_Date,
Date(Date#(Date,'YYYY-MM-DD'),'M/D/YYYY') as US_Date,
Country, Product, Amount
Inline
[Date, Country, Product, Amount
2018-02-20, Canada, washer, 6
2018-07-08, Germany, Anchor bolt, 10
```

2018-07-14, Germany, Anchor bolt, 3  
 2018-08-31, France, Nut, 2  
 2018-09-02, Czech Republic, Bolt, 1  
 2019-02-11, Czech Republic, Bolt, 3  
 2019-07-31, Czech Republic, Washer, 6  
 2020-03-13, France, Anchor bolt, 1  
 2020-07-12, Canada, Anchor bolt, 8  
 2020-09-16, France, Washer, 1];

### Esempio 1: espressioni del grafico con ricerche testuali

Generare una tabella in un foglio Qlik Sense con le seguenti espressioni del grafico.

Tabella - Modificatori set con ricerche testuali

| Paese              | Sum<br>(Amount) | Sum({<Country=<br>{"C*"}>} Amount) | Sum({<Country=<br>{"^R*"}>} Amount) | Sum({<Product=<br>{"*bolt*"}>} Amount) |
|--------------------|-----------------|------------------------------------|-------------------------------------|----------------------------------------|
| <b>Totali</b>      | <b>41</b>       | <b>24</b>                          | <b>10</b>                           | <b>26</b>                              |
| Canada             | 14              | 14                                 | 0                                   | 8                                      |
| Repubblica<br>Ceca | 10              | 10                                 | 10                                  | 4                                      |
| Francia            | 4               | 0                                  | 0                                   | 1                                      |
| Germania           | 13              | 0                                  | 0                                   | 13                                     |

### Spiegazione

- Dimensioni:
  - Country
- Misure:
  - Sum(Amount)  
Sum Amount senza espressione set.
  - Sum({<Country={"C\*"}>} Amount)  
Sum Amount per tutti i paesi che iniziano con C, come Canada e Czech Republic.
  - Sum({<Country={"^R\*"}>} Amount)  
Sum Amount per tutti i paesi che presentano una parola che inizia con R, come Czech Republic.
  - Sum({<Product={"\*bolt\*"}>} Amount)  
Sum Amount per tutti i prodotti che contengono la stringa bolt, come bolt e Anchor bolt.

Modificatori set con ricerche testuali

My new sheet

| Country        | Q | Sum<br>(Amount) | Sum({<Country={""C*"}>}<br>Amount) | Sum({<Country={""^R*"}>}<br>Amount) | Sum({<Product={""*bolt*"}>}<br>Amount) |
|----------------|---|-----------------|------------------------------------|-------------------------------------|----------------------------------------|
| <b>Totals</b>  |   | <b>41</b>       | <b>24</b>                          | <b>10</b>                           | <b>26</b>                              |
| Canada         |   | 14              | 14                                 | 0                                   | 8                                      |
| Czech Republic |   | 10              | 10                                 | 10                                  | 4                                      |
| France         |   | 4               | 0                                  | 0                                   | 1                                      |
| Germany        |   | 13              | 0                                  | 0                                   | 13                                     |

### Esempio 2: espressioni del grafico con ricerche numeriche

Generare una tabella in un foglio Qlik Sense con le seguenti espressioni del grafico.

Tabella - Modificatori set con ricerche numeriche

| Paese              | Sum<br>(Amount) | Sum({<Year={"">2019"}>}<br>Amount) | Sum({<ISO_<br>Date={"">=2019-07-<br>01"}>}<br>Amount) | Sum({<US_Date={"">=4/1/2018<=12/31/2018"}>}<br>Amount) |
|--------------------|-----------------|------------------------------------|-------------------------------------------------------|--------------------------------------------------------|
| <b>Totali</b>      | <b>41</b>       | <b>10</b>                          | <b>16</b>                                             | <b>16</b>                                              |
| Canada             | 14              | 8                                  | 8                                                     | 0                                                      |
| Repubblica<br>Ceca | 10              | 0                                  | 6                                                     | 1                                                      |
| Francia            | 4               | 2                                  | 2                                                     | 2                                                      |
| Germania           | 13              | 0                                  | 0                                                     | 13                                                     |

### Spiegazione

- Dimensioni:
  - Country
- Misure:
  - Sum(Amount)  
Sum Amount senza espressione set.
  - Sum({<Year={"">2019"}>}Amount)  
Sum Amount per tutti gli anni dopo 2019.
  - Sum({<ISO\_Date={"">=2019-07-01"}>}Amount)  
Sum Amount per tutte le date al o dopo il 2019-07-01. Il formato della data nella ricerca deve corrispondere al formato del campo.
  - Sum({<US\_Date={"">=4/1/2018<=12/31/2018"}>}Amount)

Sum Amount per tutte le date dal 4/1/2018 al 12/31/2018, incluse le date di inizio e fine. Il formato delle date nella ricerca deve corrispondere al formato del campo.

*Modificatori set con ricerche numeriche*

My new sheet

| Country        | Sum<br>(Amount) | Sum({<Year={">2019"}>}<br>Amount) | Sum({<ISO_Date={">=2019-07-01"}>}<br>Amount) | Sum({<US_Date={">=4/1/2018<=12/31/2018"}>}<br>Amount) |
|----------------|-----------------|-----------------------------------|----------------------------------------------|-------------------------------------------------------|
| Totals         | 41              | 10                                | 16                                           | 16                                                    |
| Canada         | 14              | 8                                 | 8                                            | 0                                                     |
| Czech Republic | 10              | 0                                 | 6                                            | 1                                                     |
| France         | 4               | 2                                 | 2                                            | 2                                                     |
| Germany        | 13              | 0                                 | 0                                            | 13                                                    |

### Esempio 3: espressioni del grafico con ricerche tramite espressione

Generare una tabella in un foglio Qlik Sense con le seguenti espressioni del grafico.

Table - Set modifiers with expression searches

| Country           | Sum<br>(Amount) | Sum({<Country={<br>"=Sum<br>(Amount)>10">}>}<br>Amount) | Sum({<Country={<br>"=Count(distinct<br>Product)=1">}>}<br>Amount) | Sum({<Product={<br>"=Count<br>(Amount)>3">}>}<br>Amount) |
|-------------------|-----------------|---------------------------------------------------------|-------------------------------------------------------------------|----------------------------------------------------------|
| Totals            | 41              | 27                                                      | 13                                                                | 22                                                       |
| Canada            | 14              | 14                                                      | 0                                                                 | 8                                                        |
| Czech<br>Republic | 10              | 0                                                       | 0                                                                 | 0                                                        |
| France            | 4               | 0                                                       | 0                                                                 | 1                                                        |
| Germany           | 13              | 13                                                      | 13                                                                | 13                                                       |

### Spiegazione

- Dimensioni:
  - Country
- Misure:
  - Sum(Amount)  
Sum Amount senza espressione set.
  - Sum({<Country={"=Sum(Amount)>10">}>}Amount)  
Sum Amount per tutti i paesi che presentano una somma aggregata di Amount superiore a 10.
  - Sum({<Country={"=Count(distinct Product)=1">}>}Amount)  
Sum Amount per tutti i paesi associati con esattamente un prodotto distinto.

- `Sum({<Product={"=Count(Amount)>3"}>}Amount)`  
Sum Amount per tutti i paesi che presentano più di tre transazioni nei dati.

Modificatori set con ricerche tramite espressione

My new sheet

| Country        | Sum<br>(Amount) | Sum({<Country=<br>{"=Sum(Amount)>10"}>}<br>Amount) | Sum({<Country={"=Count(distinct<br>Product)=1"}>} Amount) | Sum({<Product=<br>{"=Count(Amount)>3"}>}<br>Amount) |
|----------------|-----------------|----------------------------------------------------|-----------------------------------------------------------|-----------------------------------------------------|
| Totals         | 41              | 27                                                 | 13                                                        | 22                                                  |
| Canada         | 14              | 14                                                 | 0                                                         | 8                                                   |
| Czech Republic | 10              | 0                                                  | 0                                                         | 0                                                   |
| France         | 4               | 0                                                  | 0                                                         | 1                                                   |
| Germany        | 13              | 13                                                 | 13                                                        | 13                                                  |

| Esempi                                                                                                            | Risultati                                                                                                                                                                              |
|-------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>sum( {\$-1&lt;Product =<br/>{"*Internal*", "*Domestic*"}&gt;}<br/>Sales )</code>                            | Restituisce le vendite per la selezione corrente, escluse le transazioni relative ai prodotti con la stringa 'Internal' o 'Domestic' nel nome del prodotto.                            |
| <code>sum( {\$&lt;Customer = {"=Sum<br/>( {1&lt;Year = {2007}&gt;} Sales ) &gt;<br/>1000000"}&gt;} Sales )</code> | Restituisce le vendite per la selezione corrente, ma con nuove selezione nel campo 'Customer': solo i clienti che nel 2007 hanno totalizzato un numero di vendite superiore a 1000000. |

### Modificatori di gruppo con espansioni con simbolo del dollaro

Le espansioni con simbolo del dollaro sono costrutti calcolati prima che l'espressione sia analizzata e valutata. Il risultato viene quindi inserito nell'espressione al posto di `$(...)`. Il calcolo dell'espressione viene quindi effettuato usando il risultato dell'espansione con simbolo del dollaro.

L'editor delle espressioni mostra un'anteprima dell'espansione con simbolo del dollaro, in modo che sia possibile verificare per cosa valuta la propria espressione con simbolo del dollaro.

*Anteprima dell'espansione con simbolo del dollaro nell'editor delle espressioni*



Utilizzare le espansioni con simbolo del dollaro quando si desidera utilizzare un calcolo all'interno del proprio set di elementi.

Ad esempio, se si desidera ricercare solo l'ultimo anno possibile, è possibile utilizzare la seguente costruzione:

```
<Year = {$(=Max(Year))}>
```

Max(Year) viene calcolato per primo e il risultato verrà inserito nell'espressione al posto di \$(...).

Il risultato dopo l'espansione con dollaro sarà un'espressione simile alla seguente:

```
<Year = {2021}>
```

L'espressione all'interno dell'espansione con simbolo del dollaro viene calcolata in base alla selezione corrente. Ciò significa che se si effettua una selezione in un altro campo, il risultato dell'espressione ne viene influenzato.

Se si desidera che il calcolo sia indipendente dalla selezione, utilizzare l'analisi set all'interno dell'espansione con il simbolo del dollaro. Ad esempio:

```
<Year = {$(=Max({1} Year))}>
```

### Stringhe

Quando si desidera che l'espansione con simbolo del dollaro risulti in una stringa, si applicano le normali regole associate alle citazioni. Ad esempio:

```
<Country = {'$(=FirstSortedValue(Country,Date))'}>
```

Il risultato dopo l'espansione con dollaro sarà un'espressione simile alla seguente:

```
<Country = {'New Zealand'}>
```

Si otterrà un errore di sintassi se non si utilizzano le virgolette.

### Numeri

Quando si desidera che l'espansione con simbolo del dollaro risulti in un numero, assicurarsi che l'espansione abbia la stessa formattazione del campo. Questo significa che a volte è necessario disporre l'espressione in una funzione di formattazione.

Ad esempio:

```
<Amount = {$ (=Num(Max(Amount), '###0.00'))}>
```

Il risultato dopo l'espansione con dollaro sarà un'espressione simile alla seguente:

```
<Amount = {12362.00}>
```

Usare un hash per forzare l'espansione a usare sempre il punto decimale e nessun separatore delle migliaia. Ad esempio:

```
<Amount = {$ (#=Max(Amount))}>
```

### Date

Quando si desidera che l'espansione con simbolo del dollaro risulti in una data, assicurarsi che l'espansione abbia la formattazione corretta. Questo significa che a volte è necessario disporre l'espressione in una funzione di formattazione.

Ad esempio:

```
<Date = {'$ (=Date(Max(Date)))'}>
```

Il risultato dopo l'espansione con dollaro sarà un'espressione simile alla seguente:

```
<Date = {'12/31/2015'}>
```

Proprio come con le stringhe, è necessario utilizzare le virgolette corrette.

Un caso d'uso comune è quello di volere che il proprio calcolo sia limitato all'ultimo mese (o anno). Quindi è possibile utilizzare una ricerca numerica in combinazione con la funzione `AddMonths()`.

Ad esempio:

```
<Date = {">= $(=AddMonths(Today(), -1))"}>
```

Il risultato dopo l'espansione con dollaro sarà un'espressione simile alla seguente:

```
<Date = {">=9/31/2021"}>
```

Questa operazione sceglierà tutti gli eventi che si sono verificati nell'ultimo mese.

Esempio: espressioni del grafico per i modificatori set con espansioni con simbolo del dollaro

Esempio: espressioni del grafico

### Script di caricamento

Caricare i dati seguenti come un caricamento inline nell'editor caricamento dati per creare gli esempi di espressione del grafico in basso.

```
Let vToday = Today();
MyTable:
Load
Year(Date) as Year,
Date#(Date,'YYYY-MM-DD') as ISO_Date,
Date(Date#(Date,'YYYY-MM-DD'),'M/D/YYYY') as US_Date,
Country, Product, Amount
Inline
[Date, Country, Product, Amount
2018-02-20, Canada, Washer, 6
2018-07-08, Germany, Anchor bolt, 10
2018-07-14, Germany, Anchor bolt, 3
2018-08-31, France, Nut, 2
2018-09-02, Czech Republic, Bolt, 1
2019-02-11, Czech Republic, Bolt, 3
2019-07-31, Czech Republic, Washer, 6
2020-03-13, France, Anchor bolt, 1
2020-07-12, Canada, Anchor bolt, 8
2021-10-15, France, Washer, 1];
```

### Espressioni del grafico con espansioni con simbolo del dollaro

Generare una tabella in un foglio Qlik Sense con le seguenti espressioni del grafico.

Tabella - Modificatori set con espansioni con simbolo del dollaro

| Paese              | Sum<br>(Amount) | Sum({<US_<br>Date=<br>{'\$(vToday)'}>}<br>Amount) | Sum({<ISO_Date=<br>{'\$(=Date(Min(ISO_<br>Date),'YYYY-MM-<br>DD'))'}>} Amount) | Sum({<US_Date=<br>{'>=\$(=AddYears<br>(Max(US_Date),-<br>1))'}>} Amount) |
|--------------------|-----------------|---------------------------------------------------|--------------------------------------------------------------------------------|--------------------------------------------------------------------------|
| <b>Totali</b>      | <b>41</b>       | <b>1</b>                                          | <b>6</b>                                                                       | <b>1</b>                                                                 |
| Canada             | 14              | 0                                                 | 6                                                                              | 0                                                                        |
| Repubblica<br>Ceca | 10              | 0                                                 | 0                                                                              | 0                                                                        |
| Francia            | 4               | 1                                                 | 0                                                                              | 1                                                                        |
| Germania           | 13              | 0                                                 | 0                                                                              | 0                                                                        |

## Spiegazione

- Dimensioni:
  - Country
- Misure:
  - Sum(Amount)  
Sum Amount senza espressione set.
  - Sum({<US\_Date={ '\$(vToday)' }>}Amount)  
Sum Amount per tutti i record in cui us\_date è lo stesso della variabile vToday.
  - Sum({<ISO\_Date={"\$(=Date(Min(ISO\_Date), 'YYYY-MM-DD'))"}>}Amount)  
Sum Amount per tutti i record in cui ISO\_date è lo stesso del primo (più piccolo) ISO\_date possibile. La funzione Date() è necessaria per assicurare che il formato della data corrisponda a quello del campo.
  - Sum({<US\_Date={">=\$(=AddYears(Max(US\_Date), -1))"}>}Amount)  
Sum Amount per tutti i record che presentano un us\_date dopo o nella data un anno prima del più recente (più grande) us\_date possibile. La funzione AddYears() restituirà una data nel formato specificato dalla variabile DateFormat, e questo deve corrispondere al formato del campo US\_Date.

Modificatori di gruppo con espansioni con simbolo del dollaro

My new sheet

| Country        | Sum (Amount) | Sum({<US_Date={"\$(vToday)"}>} Amount) | Sum({<ISO_Date={"\$(=Date(Min(ISO_Date), 'YYYY-MM-DD'))"}>} Amount) | Sum({<US_Date={">=\$(=AddYears(Max(US_Date), -1))"}>} Amount) |
|----------------|--------------|----------------------------------------|---------------------------------------------------------------------|---------------------------------------------------------------|
| Totals         | 41           | 1                                      | 6                                                                   | 1                                                             |
| Canada         | 14           | 0                                      | 6                                                                   | 0                                                             |
| Czech Republic | 10           | 0                                      | 0                                                                   | 0                                                             |
| France         | 4            | 1                                      | 0                                                                   | 1                                                             |
| Germany        | 13           | 0                                      | 0                                                                   | 0                                                             |

| Esempi                                      | Risultati                                                                                                                                                                                                            |
|---------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| sum( {<Year = {\$(#vLastYear)}>} Sales )    | Restituisce le vendite per l'anno precedente in relazione alla selezione attuale. In questo esempio, una variabile vLastYear contenente l'anno pertinente viene utilizzata in un'espansione del simbolo del dollaro. |
| sum( {<Year = {\$(#Only(Year)-1)}>} Sales ) | Restituisce le vendite per l'anno precedente in relazione alla selezione attuale. In questo esempio, viene utilizzata un'espansione del simbolo del dollaro per calcolare l'anno precedente.                         |

## Modificatori di gruppo con operatori di gruppo

Gli operatori set sono utilizzati per includere, escludere o intersecare vari set di elementi. Combinano i vari metodi per definire set di elementi.

Gli operatori corrispondono a quelli utilizzati per gli identificatori set.

### Operatori

| Operatore | Descrizione                                                                                                                                                                                                                                                 |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| +         | Unione. Questa operazione binaria restituisce un gruppo costituito dai record o dagli elementi che appartengono a uno qualsiasi dei due operandi set.                                                                                                       |
| -         | Esclusione. Questa operazione binaria restituisce un gruppo costituito dai record o dagli elementi che appartengono solo al primo dei due operandi set e non al secondo. Inoltre, se utilizzata come operatore unario, restituisce il gruppo complementare. |
| *         | Intersezione. Questa operazione binaria restituisce un gruppo costituito dai record o dagli elementi che appartengono a entrambi gli operandi set.                                                                                                          |
| /         | Differenza simmetrica (XOR). Questa operazione binaria restituisce un gruppo costituito dai record o dagli elementi che appartengono a uno dei due operandi set, ma non a entrambi.                                                                         |

Ad esempio, i due modificatori seguenti definiscono lo stesso set di valori di campo:

- `<Year = {1997, "20*"}>`
- `<Year = {1997} + {"20*"}>`

Entrambe le espressioni selezionano 1997 e gli anni che iniziano con 20. In altre parole, si tratta dell'unione delle due condizioni.

Gli operatori set consentono anche definizioni più complesse. Ad esempio:

`<Year = {1997, "20*"} - {2000}>`

Questa espressione selezionerà gli stessi anni di quanto riportato sopra, ma in aggiunta esclude l'anno 2000.

.

**Esempi: espressioni del grafico per i modificatori set con operatori set**

**Esempi: espressioni del grafico**

### Script di caricamento

Caricare i dati seguenti come un caricamento inline nell'editor caricamento dati per creare gli esempi di espressione del grafico in basso.

```
MyTable:
Load
Year(Date) as Year,
Date#(Date, 'YYYY-MM-DD') as ISO_Date,
Date(Date#(Date, 'YYYY-MM-DD'), 'M/D/YYYY') as US_Date,
Country, Product, Amount
Inline
[Date, Country, Product, Amount
```

2018-02-20, Canada, Washer, 6  
 2018-07-08, Germany, Anchor bolt, 10  
 2018-07-14, Germany, Anchor bolt, 3  
 2018-08-31, France, Nut, 2  
 2018-09-02, Czech Republic, Bolt, 1  
 2019-02-11, Czech Republic, Bolt, 3  
 2019-07-31, Czech Republic, Washer, 6  
 2020-03-13, France, Anchor bolt, 1  
 2020-07-12, Canada, Anchor bolt, 8  
 2020-09-16, France, Washer, 1];

### Espressioni del grafico

Generare una tabella in un foglio Qlik Sense con le seguenti espressioni del grafico.

Tabella - Modificatori set con operatori set

| Paese              | Sum<br>(Amount) | Sum({<Year=<br>{">2018"}-<br>{2020}>}<br>Amount) | Sum<br>({{<Country=-<br>{Germany}>}<br>Amount) | Sum({<Country=<br>{Germany}+P({<Product=<br>{Nut}>}Country)>} Amount) |
|--------------------|-----------------|--------------------------------------------------|------------------------------------------------|-----------------------------------------------------------------------|
| <b>Totali</b>      | <b>41</b>       | <b>9</b>                                         | <b>28</b>                                      | <b>17</b>                                                             |
| Canada             | 14              | 0                                                | 14                                             | 0                                                                     |
| Repubblica<br>Ceca | 10              | 9                                                | 10                                             | 0                                                                     |
| Francia            | 4               | 0                                                | 4                                              | 4                                                                     |
| Germania           | 13              | 0                                                | 0                                              | 13                                                                    |

### Spiegazione

- Dimensioni:
  - Country
- Misure:
  - Sum(Amount)  
Sum Amount senza espressione set.
  - Sum({<Year={">2018"}-{2020}>}Amount)  
Sum Amount per tutti gli anni dopo 2018, tranne 2020.
  - Sum({<Country=-{Germany}>}Amount)  
Sum Amount per tutti i paesi tranne la Germany. Notare l'operatore di esclusione unario.
  - Sum({<Country={Germany}+P({<Product={Nut}>}Country)>}Amount)  
Sum Amount per Germany e tutti i paesi associati al prodotto Nut.

## Modificatori di gruppo con operatori di gruppo

My new sheet

| Country        | Sum<br>(Amount) | Sum({<Year={ ">2018" }-[2020]>}<br>Amount) | Sum({<Country= - {Germany}>}<br>Amount) | Sum({<Country={Germany}>+P({<Product={<br>Nut}>} Country)>} Amount) |
|----------------|-----------------|--------------------------------------------|-----------------------------------------|---------------------------------------------------------------------|
| Totals         | 41              | 9                                          | 28                                      | 17                                                                  |
| Canada         | 14              | 0                                          | 14                                      | 0                                                                   |
| Czech Republic | 10              | 9                                          | 10                                      | 0                                                                   |
| France         | 4               | 0                                          | 4                                       | 4                                                                   |
| Germany        | 13              | 0                                          | 0                                       | 13                                                                  |

| Esempi                                                                          | Risultati                                                                                                                                                                                                                                                                                          |
|---------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| sum( {<Product =<br>Product +<br>{OurProduct1} -<br>{OurProduct2} >}<br>Sales ) | Restituisce le vendite per la selezione corrente, ma con il prodotto "OurProduct1" aggiunto all'elenco dei prodotti selezionati e "OurProduct2" rimosso dall'elenco dei prodotti selezionati.                                                                                                      |
| sum( {<Year =<br>Year +<br>{ "20*", 1997 } -<br>{2000} >} Sales )               | Restituisce le vendite per la selezione corrente ma con selezioni aggiuntive nel campo "Year": 1997 e tutti gli anni che iniziano per "20", ad eccezione dell'anno 2000.<br><br>Tenere presente che se 2000 è incluso nella selezione attuale, continuerà a essere incluso anche dopo la modifica. |
| sum( {<Year =<br>(Year +<br>{ "20*", 1997 }) -<br>{2000} >} Sales )             | Restituisce quasi lo stesso risultato dell'espressione precedente, ma l'anno 2000 verrà escluso, anche se inizialmente viene incluso nella selezione corrente. L'esempio dimostra l'importanza dello specifico utilizzo delle parentesi per definire un ordine di precedenza.                      |
| sum( {<Year = { "*" }<br>- {2000}, Product =<br>{ "*"bearing*" } >}<br>Sales )  | Restituisce le vendite per la selezione corrente, ma con una nuova selezione in "Year": tutti gli anni eccetto il 2000 e solo per i prodotti contenenti la stringa "bearing".                                                                                                                      |

## Modificatori set con operatori set impliciti

Il modo standard di scrivere le selezioni in un modificatore set consiste nell'utilizzare un segno uguale. Ad esempio:

```
Year = { ">2015" }
```

L'espressione a destra del segno uguale nel modificatore set è chiamata set di elementi. Definisce un set di valori di campo distinti, in altre parole una selezione.

Questa notazione definisce una nuova selezione ignorando la selezione attuale nel campo. Pertanto, se l'identificatore set contiene una selezione in questo campo, la vecchia selezione verrà sostituita da quella nel set di elementi.

Quando si desidera basare la propria selezione sulla selezione corrente nel campo, è necessario usare un'espressione diversa

Per esempio, se si desidera rispettare la vecchia selezione e aggiungere il requisito che l'anno sia successivo al 2015, è possibile scrivere quanto segue:

```
Year = Year * {">2015"}
```

L'asterisco è un operatore set che definisce un'intersezione, quindi si otterrà l'intersezione tra la selezione corrente in Year e il requisito aggiuntivo che l'anno sia successivo al 2015. Un modo alternativo per scriverlo è il seguente:

```
Year *= {">2015"}
```

Ovvero, l'operatore di assegnazione (\*=) definisce implicitamente un'intersezione.

Allo stesso modo, le unioni, le esclusioni e le differenze simmetriche implicite possono essere definite utilizzando quanto segue: +=, -=, /=

### Esempi: espressioni del grafico per i modificatori set con operatori set impliciti

Esempi: espressioni del grafico

#### Script di caricamento

Caricare i dati seguenti come un caricamento inline nell'editor caricamento dati per creare gli esempi di espressione del grafico in basso.

```
MyTable:
Load
Year(Date) as Year,
Date#(Date,'YYYY-MM-DD') as ISO_Date,
Date(Date#(Date,'YYYY-MM-DD'),'M/D/YYYY') as US_Date,
Country, Product, Amount
Inline
[Date, Country, Product, Amount
2018-02-20, Canada, Washer, 6
2018-07-08, Germany, Anchor bolt, 10
2018-07-14, Germany, Anchor bolt, 3
2018-08-31, France, Nut, 2
2018-09-02, Czech Republic, Bolt, 1
2019-02-11, Czech Republic, Bolt, 3
2019-07-31, Czech Republic, Washer, 6
2020-03-13, France, Anchor bolt, 1
2020-07-12, Canada, Anchor bolt, 8
2020-09-16, France, Washer, 1];
```

### Espressioni del grafico con operatori set impliciti

Generare una tabella in un foglio Qlik Sense con le seguenti espressioni del grafico.

Selezionare Canada e Czech Republic da un elenco di paesi.

Tabella - Espressioni del grafico con operatori set impliciti

| Paese              | Sum<br>(Amount) | Sum({<Country*=<br>{Canada}>} Amount) | Sum({<Country-=<br>{Canada}>} Amount) | Sum({<Country+=<br>{France}>} Amount) |
|--------------------|-----------------|---------------------------------------|---------------------------------------|---------------------------------------|
| <b>Totali</b>      | <b>24</b>       | <b>14</b>                             | <b>10</b>                             | <b>28</b>                             |
| Canada             | 14              | 14                                    | 0                                     | 14                                    |
| Repubblica<br>Ceca | 10              | 0                                     | 10                                    | 10                                    |
| Francia            | 0               | 0                                     | 0                                     | 4                                     |

### Spiegazione

- Dimensioni:
  - Country
- Misure:
  - Sum(Amount)  
Sum Amount per la selezione corrente. Notare che solo Canada e Czech Republic presentano valori diversi da zero.
  - Sum({<Country\*={Canada}>}Amount)  
Sum Amount per la selezione corrente, intersecato con il requisito secondo cui il country deve essere Canada. Se Canada non fa parte della selezione dell'utente, l'espressione set restituisce un set vuoto e la colonna avrà 0 su tutte le righe.
  - Sum({<Country-={Canada}>}Amount)  
Sum Amount per la selezione corrente, ma prima Canada viene escluso dalla selezione country. Se Canada non fa parte della selezione utente, l'espressione set non cambierà alcun numero.
  - Sum({<Country+={France}>}Amount)  
Sum Amount per la selezione corrente, ma prima si aggiunge France alla selezione country. Se France fa già parte della selezione utente, l'espressione set non cambierà alcun numero.

Modificatori set con operatori set impliciti.

| Country<br>2 of 4 |                |              |                                   |                                   |                                   |
|-------------------|----------------|--------------|-----------------------------------|-----------------------------------|-----------------------------------|
| My new sheet      |                |              |                                   |                                   |                                   |
| Country           | Country        | Sum (Amount) | Sum({<Country*={Canada}>} Amount) | Sum({<Country-={Canada}>} Amount) | Sum({<Country+={France}>} Amount) |
| Canada ✓          | Totals         | 24           | 14                                | 10                                | 28                                |
| Czech Republic ✓  | Canada         | 14           | 14                                | 0                                 | 14                                |
| France            | Czech Republic | 10           | 0                                 | 10                                | 10                                |
| Germany           | France         | 0            | 0                                 | 0                                 | 4                                 |

| Esempi                                                    | Risultati                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| sum( {\$<Product += {OurProduct1, OurProduct2} >} Sales ) | Restituisce le vendite per la selezione attuale, ma utilizzando un'unione implicita per aggiungere i prodotti 'OurProduct1' e 'OurProduct2' all'elenco dei prodotti selezionati.                                                                                                                                                                                                              |
| sum( {\$<Year += {"20*", 1997} - {2000} >} Sales )        | Restituisce le vendite per la selezione attuale ma utilizzando un'unione implicita per aggiungere un numero di anni alla selezione: 1997 e tutti gli anni che iniziano per "20", ad eccezione dell'anno 2000.<br><br>Tenere presente che se 2000 è incluso nella selezione attuale, continuerà a essere incluso anche dopo la modifica. Lo stesso di <Year=Year + ({ "20*", 1997} - {2000})>. |
| sum( {\$<Product *= {OurProduct1} >} Sales )              | Restituisce le vendite per la selezione attuale, ma solo per l'intersezione dei prodotti attualmente selezionati e del prodotto OurProduct1.                                                                                                                                                                                                                                                  |

## Modificatori set che utilizzano le funzioni set

A volte è necessario definire un set di valori di campo utilizzando una definizione di gruppo nidificata. Ad esempio, è possibile voler selezionare tutti i clienti che hanno portato un prodotto specifico, senza selezionare il prodotto.

In tali casi, usare le funzioni set di elementi  $P()$  e  $E()$ . Questi restituiscono set di elementi dei valori possibili e dei valori esclusi di un campo, rispettivamente. All'interno delle parentesi, è possibile specificare il campo in questione e un'espressione set che definisce l'ambito. Ad esempio:

```
P({1<Year = {2021}>} Customer)
```

Ciò restituirà il set di clienti che hanno avuto delle transazioni nel 2021. È quindi possibile utilizzarlo in un modificatore set. Ad esempio:

```
Sum({<Customer = P({1<Year = {2021}>} Customer)>} Amount)
```

Questa espressione set selezionerà tali clienti, ma non limiterà la restrizione della selezione al 2021.

Queste funzioni non possono essere utilizzate in altre espressioni.

In aggiunta, solo i set naturali possono essere utilizzati all'interno delle funzioni dei set di elementi. ossia un set di record che può essere definito mediante una semplice selezione.

Ad esempio, il set restituito da {1-\$} non può essere sempre definito mediante una selezione e non è dunque un set naturale. L'utilizzo di queste funzioni su set non naturali restituirà risultati inattesi.

### Esempi: espressioni del grafico per i modificatori set che utilizzano funzioni set

Esempi: espressioni del grafico

#### Script di caricamento

Caricare i dati seguenti come un caricamento inline nell'editor caricamento dati per creare gli esempi di espressione del grafico in basso.

```
MyTable:
Load
Year(Date) as Year,
Date#(Date,'YYYY-MM-DD') as ISO_Date,
Date(Date#(Date,'YYYY-MM-DD'),'M/D/YYYY') as US_Date,
Country, Product, Amount
Inline
[Date, Country, Product, Amount
2018-02-20, Canada, Washer, 6
2018-07-08, Germany, Anchor bolt, 10
2018-07-14, Germany, Anchor bolt, 3
2018-08-31, France, Nut, 2
2018-09-02, Czech Republic, Bolt, 1
2019-02-11, Czech Republic, Bolt, 3
2019-07-31, Czech Republic, Washer, 6
2020-03-13, France, Anchor bolt, 1
2020-07-12, Canada, Anchor bolt, 8
2020-09-16, France, Washer, 1];
```

#### Espressioni del grafico

Generare una tabella in un foglio Qlik Sense con le seguenti espressioni del grafico.

Tabella - Modificatori set che utilizzano le funzioni set

| Paese         | Sum<br>(Amount) | Sum({<Country=P<br>{<Year=<br>{2019}>}Country>}<br>Amount) | Sum({<Product=P<br>{<Year=<br>{2019}>}Product>}<br>Amount) | Sum({<Country=E<br>{<Product=<br>{Washer}>}Country>}<br>Amount) |
|---------------|-----------------|------------------------------------------------------------|------------------------------------------------------------|-----------------------------------------------------------------|
| <b>Totali</b> | <b>41</b>       | <b>10</b>                                                  | <b>17</b>                                                  | <b>13</b>                                                       |
| Canada        | 14              | 0                                                          | 6                                                          | 0                                                               |

| Paese              | Sum<br>(Amount) | Sum({<Country=P<br>{<Year=<br>{2019}>}Country>}<br>Amount) | Sum({<Product=P<br>{<Year=<br>{2019}>}Product>}<br>Amount) | Sum({<Country=E<br>{<Product=<br>{Washer}>}Country>}<br>Amount) |
|--------------------|-----------------|------------------------------------------------------------|------------------------------------------------------------|-----------------------------------------------------------------|
| <b>Totali</b>      | <b>41</b>       | <b>10</b>                                                  | <b>17</b>                                                  | <b>13</b>                                                       |
| Repubblica<br>Ceca | 10              | 10                                                         | 10                                                         | 0                                                               |
| Francia            | 4               | 0                                                          | 1                                                          | 0                                                               |
| Germania           | 13              | 0                                                          | 0                                                          | 13                                                              |

### Spiegazione

- Dimensioni:
  - Country
- Misure:
  - Sum(Amount)  
Sum Amount senza espressione set.
  - Sum({<Country=P({<Year={2019}>} Country)>} Amount)  
Sum Amount per i paesi associati all'anno 2019. Tuttavia, non limiterà il calcolo a 2019.
  - Sum({<Product=P({<Year={2019}>} Product)>} Amount)  
Sum Amount per i prodotti associati all'anno 2019. Tuttavia, non limiterà il calcolo a 2019.
  - Sum({<Country=E({<Product={washer}>} Country)>} Amount)  
Sum Amount per i paesi non associati al prodotto washer.

*Modifieri set che utilizzano le funzioni set*

My new sheet

| Country        | Q | Sum<br>(Amount) | Sum({<Country=P({<Year=<br>{2019}>} Country)>} Amount) | Sum({<Product=P({<Year=<br>{2019}>} Product)>} Amount) | Sum({<Country=E({<Product=<br>{Washer}>} Country)>} Amount) |
|----------------|---|-----------------|--------------------------------------------------------|--------------------------------------------------------|-------------------------------------------------------------|
| Totals         |   | 41              | 10                                                     | 17                                                     | 13                                                          |
| Canada         |   | 14              | 0                                                      | 6                                                      | 0                                                           |
| Czech Republic |   | 10              | 10                                                     | 10                                                     | 0                                                           |
| France         |   | 4               | 0                                                      | 1                                                      | 0                                                           |
| Germany        |   | 13              | 0                                                      | 0                                                      | 13                                                          |

| Esempi                                                                          | Risultati                                                                                                                                                                                                                                                                                                                                                                                     |
|---------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| sum(<br>{\$<Customer =<br>P({1<Product=<br>'Shoe'})>}<br>Customer)>}<br>Sales ) | Restituisce le vendite per la selezione corrente, ma solo i clienti che hanno acquistato il prodotto 'Shoe'. La funzione di elemento P( ) qui restituisce un elenco di possibili clienti; quelli che sono interessati dalla selezione 'Shoe' nel campo Product.                                                                                                                               |
| sum(<br>{\$<Customer =<br>P({1<Product=<br>'Shoe'})>})>}<br>Sales )             | Restituisce lo stesso risultato riportato in precedenza. Se il campo nella funzione di elemento viene omissa, la funzione restituirà i possibili valori del campo specificato nell'assegnazione esterna.                                                                                                                                                                                      |
| sum(<br>{\$<Customer =<br>P({1<Product=<br>'Shoe'})>}<br>Supplier)>}<br>Sales ) | Restituisce le vendite per la selezione corrente, ma solo per i clienti che hanno fornito il prodotto 'Shoe', ovvero, laddove il cliente è anche un fornitore. La funzione di elemento P( ) qui restituisce un elenco di possibili fornitori; quelli che sono interessati dalla selezione 'Shoe' nel campo Product. L'elenco di fornitori viene utilizzato come selezione nel campo Customer. |
| sum(<br>{\$<Customer =<br>E({1<Product=<br>'Shoe'})>})>}<br>Sales )             | Restituisce le vendite per la selezione corrente, ma solo i clienti che non hanno mai acquistato il prodotto 'Shoe'. La funzione di elemento E( ) qui restituisce l'elenco di clienti esclusi; quelli che sono esclusi dalla selezione 'Shoe' nel campo Product.                                                                                                                              |

## Flag di conservazione del set vuoto

È possibile utilizzare il simbolo & (il flag di conservazione del set vuoto) all'inizio di alcune espressioni set esterne per annullare la [cancellazione implicita del set di selezione](#) durante l'analisi utente. Il flag di conservazione del set vuoto consente di modificare il modo in cui le aggregazioni vengono calcolate e rappresentate quando gli utenti effettuano le selezioni, senza richiedere la riscrittura di espressioni set lunghe e avanzate.

Ad esempio, nell'espressione set che segue, viene utilizzato il flag di conservazione del set vuoto:

```
&{$<Country*={'Sweden', 'Denmark'}>} {<ProductCategory={'shirts'}>} Sum( Sales )
```

## Sfondo

### Condizioni per l'uso

Il flag di conservazione del set vuoto può essere utilizzato nelle espressioni set che hanno uno dei seguenti componenti:

- Almeno un'espressione set interna e una esterna
- Due o più espressioni set esterne

Utilizzare il flag di conservazione del set vuoto quando si vuole evitare [Cancellazione implicita del set di selezione \(page 330\)](#).

### Cancellazione implicita del set di selezione

La cancellazione implicita del set di selezione è un comportamento interno che a volte si desidera evitare per l'analisi dell'utente finale, quando si utilizzano catene complesse di espressioni set. La cancellazione implicita del set di selezione può avvenire per le espressioni set che sono strutturate secondo [Condizioni per l'uso \(page 329\)](#).

Il comportamento indesiderato è il seguente:

1. Una o più espressioni set esterne nella catena – in particolare, la prima (o unica) e tutte le espressioni set esterne successive, tranne l'ultima – definiscono i set di dati. Ad esempio:  
`Country*={ 'Sweden', 'Denmark' }>>`
2. Si verifica una condizione che genera un set vuoto per uno o più di questi componenti dell'espressione set esterna. Nella maggior parte dei casi, viene effettuata una selezione che presenta conflitti nell'app.
3. Per i set vuoti, i set definiti da queste espressioni vengono cancellati quando l'espressione viene valutata. Ciò significa che i dati utilizzati nelle aggregazioni non sono più ridotti ai set definiti, ma vengono utilizzati tutti i dati.  
D'altra parte, i set definiti in qualsiasi espressione set esterna che producono set non vuoti, così come l'ultima espressione set esterna nella catena, vengono rispettati durante la valutazione.

Questo comportamento determina valori di aggregazione che possono essere inaspettati e possono portare ad analisi incoerenti. Per evitare la cancellazione implicita del set di selezione, è possibile utilizzare il flag di conservazione del set vuoto. Vedere [La soluzione: utilizzo del flag di conservazione del set vuoto \(page 330\)](#).

### La soluzione: utilizzo del flag di conservazione del set vuoto

Utilizzare il simbolo & (il flag di conservazione del set vuoto) per annullare la cancellazione implicita del set di selezione. Questo flag assicura che i set definiti in tutte le espressioni set esterne in una catena, ad eccezione dell'ultima, non vengano cancellate, ma vengano rispettate durante la valutazione dell'intera espressione.

Il flag di conservazione del set vuoto fornisce:

- Flessibilità e consente di risparmiare tempo quando si dispone di espressioni set lunghe avanzate che si desidera mantenere intatte, anziché riscriverle.
- La capacità di modificare la logica di aggregazione per adattarla a casi d'uso avanzati.

### Sintassi

In un'espressione set esterna, il flag di conservazione del set vuoto è posizionato come un singolo carattere & all'inizio dell'espressione, cioè prima dei modificatori set, degli identificatori e degli operatori.

**Esempi:**

Esempi: flag di conservazione del set vuoto

| Espressione                                                                        | Espressione senza flag di conservazione del set vuoto                   |
|------------------------------------------------------------------------------------|-------------------------------------------------------------------------|
| {&<Country*={'Sweden', 'Denmark'}>}<br>{<ProductCategory={'shirts'}>} Sum( Sales ) | {<Country={'Sweden'}>}<br>{<ProductCategory={'shirts'}>} Sum<br>(Sales) |
| {& <Year=, Product={XI345}>} {<Region={Europe}>}<br>Sum (Sales)                    | {<Year=, Product={XI345}>} {<Region=<br>{Europe}>} Sum (Sales)          |

**Dettagli ed esempi**

Per informazioni più dettagliate, vedere [Catene di espressioni set esterne e interne multiple \(page 333\)](#).

**Espressioni set interne ed esterne**

Le espressioni set possono essere utilizzate all'interno e all'esterno delle funzioni di aggregazione, racchiudendole tra parentesi graffe.

Quando si utilizza un'espressione set all'interno di una funzione di aggregazione, l'aspetto può essere il seguente:

**Esempio: Espressione set interna**

```
Sum( {<Year={2021}>} Sales )
```

Utilizzare un'espressione set al di fuori della funzione di aggregazione se si hanno espressioni con più aggregazioni e si vuole evitare di scrivere la stessa espressione set in ogni funzione di aggregazione.

Se si utilizza un'espressione di set esterna, questa deve essere posizionata all'inizio dell'ambito.

**Esempio: Espressione set esterna**

```
{<Year={2021}>} Sum(Sales) / Count(distinct Customer)
```

Se si utilizza un'espressione set al di fuori della funzione di aggregazione, è possibile applicarla anche alle misure principali esistenti.

**Esempio: Espressione set esterna applicata alla misura principale**

```
{<Year={2021}>} [Master Measure]
```

Un'espressione set utilizzata al di fuori delle funzioni di aggregazione influisce sull'intera espressione, a meno che non sia racchiusa tra parentesi; in tal caso le parentesi definiscono l'ambito. Nell'esempio di scoping lessicale che segue, l'espressione set viene applicata solo all'aggregazione all'interno delle parentesi.

**Esempio: Scoping lessicale**

```
( {<Year={2021}>} Sum(Amount) / Count(distinct Customer) ) - Avg(CustomerSales)
```

### Regole

#### Ambito lessicale

L'espressione set influisce sull'intera espressione, a meno che non sia racchiusa tra parentesi. In tal caso, le parentesi definiscono l'ambito lessicale.

#### Posizione

L'espressione set deve essere collocata all'inizio dell'ambito lessicale.

#### Contesto

Il contesto è la selezione rilevante per l'espressione. Tradizionalmente, il contesto è sempre stato lo stato predefinito della selezione corrente. Ma se un oggetto è impostato su uno stato alternativo, il contesto è lo stato alternativo della selezione corrente.

È possibile definire un contesto anche sotto forma di espressione set esterna.

#### Eredità

Le espressioni set interne hanno la precedenza su quelle esterne. Se l'espressione set interna contiene un identificatore set, sostituisce il contesto definito nell'espressione set esterna. Altrimenti, le espressioni set interne ed esterne vengono entrambe valutate.

#### Example 1: Espressione set interna con identificatore set

```
{<Year={2023}>} Sum(Sales) / Count({1} distinct OrderNumber)
```

Nell'espressione riportata sopra, l'espressione set interna `Count({1} distinct OrderNumber)` consiste nell'identificatore set `{1}`. Quando questa espressione set interna viene valutata, l'ambito definito nell'espressione set esterna `{<Year={2023}>}` non viene applicato.

#### Example 2: Espressioni set interne con identificatori set

```
{<Year={2023}>} Sum ({<Status='Confirmed'>} Sales_Stream1) + Sum ({<UpdatedStatus='Confirmed'>} Sales_Stream2)
```

Nell'espressione riportata sopra, le espressioni set interne `Sum ({<Status='Confirmed'>} Sales_Stream1)` e `Sum ({<UpdatedStatus='Confirmed'>} Sales_Stream2)` non contengono identificatori set. Pertanto, l'espressione set esterna `{<Year={2023}>}` ed entrambe le espressioni set interne vengono tutte applicate quando si valuta il risultato.

#### Assegnazione set elementi

L'assegnazione set elementi determina il modo in cui le due selezioni vengono unite. Se viene utilizzato un normale segno di uguale, la selezione nell'espressione set interna ha la precedenza. Altrimenti, verrà utilizzato l'operatore set implicito.

- `{<Field={value}>}` - questa selezione interna sostituisce qualsiasi selezione esterna in "Field".
- `{<Field+=value>}` - questa selezione interna viene unita alla selezione esterna in "Field", utilizzando l'operatore di unione.

- {<Field\*={value}>} - questa selezione interna viene unita alla selezione esterna in "Field", utilizzando l'operatore di intersezione.

### Ereditarietà in più fasi

L'ereditarietà può avvenire in più fasi. Esempi:

- Selezione corrente → Sum(Amount)  
La funzione di aggregazione utilizzerà il contesto, che in questo caso è la selezione corrente.
- Selezione corrente → {<Set1>} Sum(Amount)  
set1 erediterà dalla selezione corrente e il risultato sarà il contesto per la funzione di aggregazione.
- Selezione corrente → {<Set1>} ({<Set2>} Sum(Amount))  
set2 erediterà da set1, che a sua volta eredita dalla selezione corrente e il risultato sarà il contesto per la funzione di aggregazione.

### Ereditarietà per le catene di espressioni set interne ed esterne

Vi sono ulteriori considerazioni per l'ereditarietà quando si utilizzano espressioni set multiple in catene complesse, in particolare quando un'espressione set esterna è seguita da altre espressioni set esterne e/o da un'espressione set interna. Per i dettagli, vedere [Catene di espressioni set esterne e interne multiple \(page 333\)](#).

### La funzione Aggr()

La funzione Aggr() crea un'aggregazione nidificata con due aggregazioni indipendenti. Nell'esempio in basso, viene calcolato un count() per ciascun valore di Dim, e l'array risultante viene aggregato utilizzando la funzione sum().

### Esempio:

```
Sum(Aggr(Count(X),Dim))
```

count() è un'aggregazione interna mentre sum() è un'aggregazione esterna.

- L'aggregazione interna non eredita alcun contesto dall'aggregazione esterna.
- L'aggregazione interna eredita il contesto dalla funzione Aggr(), che può contenere un'espressione set.
- Sia la funzione Aggr() sia la funzione di aggregazione esterna ereditano il contesto da un'espressione set esterna.

### Catene di espressioni set esterne e interne multiple

Notare che è possibile disporre di più di un'espressione set esterna. In questo esempio vi sono due espressioni set esterne e una interna:

```
{<Year={2021}>} {<Region={"Europe"}>} Sum( {$<Product={"XI345"}>} Sales )
```

Le espressioni set verranno applicate da sinistra a destra. Ciò significa che se due espressioni set utilizzano dei set in conflitto, l'espressione più a destra avrà la precedenza.

Sebbene non vi sia alcun limite al numero di espressioni set esterne concatenate in questo modo, il numero di espressioni set interne è limitato a uno. La presenza di più espressioni set interne l'una accanto all'altra non genererà alcun errore, ma solo l'espressione più a destra verrà utilizzata al momento della valutazione.

### Eredità delle espressioni set

Quando un'espressione di aggregazione contiene più di un'espressione set come parte di una catena, le espressioni set verranno valutate una per una, da sinistra a destra. Il risultato di ogni espressione set è un gruppo di set, potenzialmente uno per ogni dimensione. I set vengono passati all'espressione set successiva, che può apportare ulteriori modifiche prima di essere passata alla successiva.

Finché ogni espressione set opera su dimensioni diverse, non vi saranno conflitti tra loro e tutti i set saranno conservati fino alla fine della catena (e applicati durante l'aggregazione). Questa regola ha un'eccezione, che a volte si applica quando una dimensione in un'espressione set genera un set vuoto. Per maggiori informazioni, vedere:

- [Set di selezione vuoti \(page 334\)](#)
- [Cancellazione implicita del set di selezione \(page 334\)](#)

### Set di selezione vuoti

Quando viene valutata una dimensione in un'espressione set, è possibile che il set non contenga dati. Questo è noto come un set vuoto. I set vuoti possono verificarsi in molti scenari, come ad esempio:

- Definizione errata dell'espressione. Per esempio, quando si desidera cercare il valore 2025 con un'espressione set {<OrderYear={ ' 2025 ' }>}, l'espressione viene scritta come {<OrderYear={ '0025' }>}.
- Durante l'analisi, viene effettuata una selezione in una dimensione diversa da quella a cui fa riferimento l'espressione e questa selezione non produce dati disponibili per la dimensione dell'espressione set. Ad esempio, un utente effettua una selezione per Region='Europe' e un'espressione set applica il filtro per {<ProductCategory={ 'Shirts' }>} Sum (Sales). Se non vengono vendute camicie in Europa, l'espressione set contiene un set vuoto.
- Il set vuoto è definito esplicitamente nell'espressione. Questo non è comune o utile, ma è tecnicamente possibile. Ad esempio: {<Year={}>}

### Cancellazione implicita del set di selezione

Se il risultato dopo l'applicazione di un'espressione set è un set vuoto per una delle dimensioni (vedere [Set di selezione vuoti \(page 334\)](#)), e c'è un'altra espressione set che viene applicata dopo questa, il set di selezione per quella dimensione verrà cancellato (ossia verrà reimpostato al set completo) appena prima che venga applicata l'espressione set successiva. Un set di selezione cancellato significa che tutti i valori sono inclusi. Notare, tuttavia, che la cancellazione avviene solo se c'è un'altra espressione set esterna che viene applicata dopo l'espressione set in cui viene generato il set vuoto.

Ecco un esempio di quando si verifica una selezione cancellata:

```
{<Year={}>} {<Region={"Europe"}>} Sum( Sales )
```

year={} creerà il set vuoto per la dimensione year. Ciò significa che il set di selezione per year viene cancellato quando il set viene passato all'espressione set successiva, che contiene Region={Europe}. Notare che la cancellazione della selezione avviene solo per una singola dimensione. I set in altre dimensioni, compresi i set all'interno della stessa espressione set, verranno passati in ogni caso senza modifiche.

Ecco un altro esempio di quando si verifica una selezione cancellata:

```
{<Year={}, Product={"XI345"}>} {<Region={"Europe"}>} Sum(Sales)
```

Nell'esempio precedente, si verifica quanto segue:

- Vengono applicati i set di selezione `Product={"XI345"}` e `Region={"Europe"}`.
- Il valore `Year={}` non viene applicato. Questo perché quel set viene cancellato quando viene applicata l'ultima espressione set, `{<Region={"Europe"}>}`.

Lo spostamento di un componente dell'espressione set nell'ultima espressione set esterna impedirà la cancellazione implicita. Prendiamo questo esempio:

```
{<Product={"XI345"}>} {<Year={}, Region={"Europe"}>} Sum(Sales)
```

In questo esempio, l'espressione `Year` è stata spostata nell'ultima espressione set della catena. In questo modo, si eviterà la cancellazione implicita, mentre il set vuoto per la dimensione `Year` verrà effettivamente utilizzato per l'aggregazione `Sum`.

In ogni caso, può verificarsi un comportamento indesiderato quando i set di selezione vengono cancellati. Per migliorare l'esperienza di analisi e facilitare la scrittura di catene di espressioni set, utilizzare il flag di conservazione del set vuoto. Per ulteriori informazioni, vedere [Flag di conservazione del set vuoto \(espressioni del set esterno\) \(page 335\)](#).

### Flag di conservazione del set vuoto (espressioni del set esterno)

Utilizzare il simbolo `&` (il flag di conservazione del set vuoto) per semplificare il modello di valutazione per le espressioni dei set esterni. Ad esempio:

```
{& <Year={}, Product={"XI345"}>} {<Region={"Europe"}>} Sum( Sales )
```

In questo esempio, `Year={}` definisce un set di selezione vuoto per `Year`. Senza il simbolo `&`, l'espressione `{<Region={"Europe"}>}` causerebbe la cancellazione del set di selezione vuoto `Year`, con conseguente utilizzo di tutti i dati nel calcolo dell'aggregazione. Il simbolo `&` obbliga l'aggregazione a rispettare il set di selezione vuoto `Year`, ottenendo un risultato più comprensibile.

Per tutti i dettagli sul flag di conservazione del set vuoto, vedere [Flag di conservazione del set vuoto \(page 329\)](#).

### Espressioni set esterne ed espansioni con il simbolo del dollaro

Quando una (o più) espansioni con il simbolo del dollaro sono utilizzate in un'espressione più ampia, ogni espansione con simbolo del dollaro verrà valutata isolatamente prima che il risultato sia inserito nel resto dell'espressione. Un'implicazione di questo metodo di valutazione è che le espressioni set al di fuori di un'espressione espansa con il segno del dollaro non influiscono sul risultato. Questo crea un effetto in cui c'è una doppia valutazione delle informazioni all'interno di un'espressione.

### Esempio: metodo di valutazione delle espressioni che contengono un'espressione set esterna ed espansioni con simbolo del dollaro

Supponiamo di avere questo esempio di espressione:

```
{<Year={2024}>} $(=Sum(Sales))
```

All'interno dell'espressione precedente, c'è un'espressione set per selezionare i valori 2024 per Year: {<Year={2024}>}. Quando l'intera espressione di esempio viene valutata, tale espressione set non viene applicata.

Invece, l'espressione con il segno del dollaro \$(=sum(Sales)) viene valutata per prima.

Per questo esempio, supponiamo che sum(Sales) valuti un risultato di 1500311. Quando si valuta l'espressione {<Year={2024}>} \$(=sum(Sales)), questo numero viene inserito nell'espressione circostante. Questo ordine di valutazione dà come risultato la seguente configurazione parzialmente valutata (si noti che questa non è un'espressione utile di Qlik Sense e viene mostrata solo per dimostrare l'ordine di valutazione):

```
{<Year={2024}>} 1500311
```

Successivamente, l'espressione {<Year={2024}>} 1500311 viene valutata nel suo complesso. L'espressione set {<Year={2024}>} non ha alcun effetto perché l'espressione alla sua destra è semplicemente un numero, non un tipo di espressione aggregata.

Pertanto, se si utilizza il valore sum(Sales) fornito nell'esempio riportato sopra, il risultato valutato per l'intera espressione {<Year={2024}>} \$(=sum(Sales)) sarebbe 1500311.

### Creazione di un'espressione dalle parti

L'utilizzo delle espansioni con il simbolo del dollaro in un'espressione comporta un effetto di doppia valutazione. È possibile approfittare di questo effetto di doppia valutazione creando un'espressione utilizzando delle parti.

#### Esempio

Supponiamo di avere questo esempio di espressione:

```
{<Year={2024}>} $=('sum(' & 'Sales' & ''))
```

Nel primo ciclo di valutazione, viene valutata l'espansione con il simbolo del dollaro \$=('sum(' & 'Sales' & '')). In questa fase, le tre stringhe vengono combinate in un'unica stringa 'sum(Sales)'. Poiché si tratta di una stringa e non di un'espressione, l'espressione all'interno non viene momentaneamente valutata.

In questa fase della valutazione, l'espressione è impostata per essere interpretata da Qlik Sense come segue:

```
{<Year={2024}>} sum(Sales)
```

In questo caso, l'espressione dell'analisi set esterna sarà applicata all'aggregazione sum, poiché ora è stata prodotta un'espressione normale, anziché un semplice numero (come nel [primo esempio](#)). Notare che i caratteri di citazione hanno consumato il primo ciclo di valutazione, in modo che l'espressione è stata conservata (e concatenata utilizzando &) per il secondo ciclo di valutazione.

### Ricerca delle variabili

Quando si utilizza un'espansione con il simbolo del dollaro per eseguire una ricerca delle variabili, il contenuto della variabile viene inserito nell'espressione circostante prima che il contenuto venga valutato.

### Esempio

Supponiamo di avere questo esempio di espressione:

```
{<Year={2024}>} $(Var1)
```

In questo esempio, il contenuto della variabile *Var1* verrà inserito nell'espressione circostante prima che il contenuto venga valutato.

### Esito A: definizione di una variabile senza il segno uguale

Se il valore *Var1* è definito come *Sum(Sales)*, dopo l'espansione si presenterebbe così (si noti che la configurazione sottostante è un'espressione parzialmente valutata):

```
{<Year={2024}>} Sum(Sales)
```

La finestra di dialogo *Variabili* mostra la definizione e il valore della variabile *Var1*, quando il segno uguale non è posto all'inizio della definizione della variabile.

Variables ✕

Create new Delete

|                          | Name ▼ | Definition | Value      | Tags ▼ | Include in bookmarks |     |
|--------------------------|--------|------------|------------|--------|----------------------|-----|
| <input type="checkbox"/> | Var1   | Sum(Sales) | Sum(Sales) |        | No                   | ... |

Showing: 1-1 of 1. |< < > >|

Quindi, per questo esempio, verrà applicata l'espressione set esterna `{<Year={2024}>}`.

### Esito B: definizione di una variabile con il segno uguale

D'altra parte, se la definizione in *Var1* inizia con il segno uguale, cioè `=Sum(Sales)`, l'espressione set esterna non viene applicata. Ciò avviene perché il segno uguale fa sì che l'espressione della variabile venga calcolata in anticipo e in modo isolato. Questo porta a un risultato simile al [primo esempio](#), in cui c'è un'espressione set esterna da valutare insieme a un numero, che non restituisce un risultato utile.

La finestra di dialogo Variabili mostra la definizione e il valore della variabile Var1, quando il segno uguale è posto all'inizio della definizione della variabile.

Variables ×

Create new Delete

|                          | Name ▼ | Definition  | Value   | Tags ▼ | Include in bookmarks |     |
|--------------------------|--------|-------------|---------|--------|----------------------|-----|
| <input type="checkbox"/> | Var1   | =Sum(Sales) | 1500311 |        | No                   | ... |

Showing: 1-1 of 1. |< < > >|

### Tutorial - Creazione di un'espressione set

È possibile creare espressioni set in Qlik Sense per supportare l'analisi dati. In questo contesto, si fa spesso riferimento all'analisi come analisi set. L'analisi set offre un modo per definire un ambito diverso dal set di record definiti dalla selezione corrente in un'app.

#### Cosa imparerai

Questo tutorial fornisce i dati e le espressioni del grafico per creare espressioni set che utilizzano modificatori, identificatori e operatori set.

#### Chi dovrebbe completare questo tutorial

Questo tutorial è destinato agli sviluppatori di app abituati a lavorare con l'editor di script e con le espressioni del grafico.

#### Cosa bisogna fare prima di iniziare

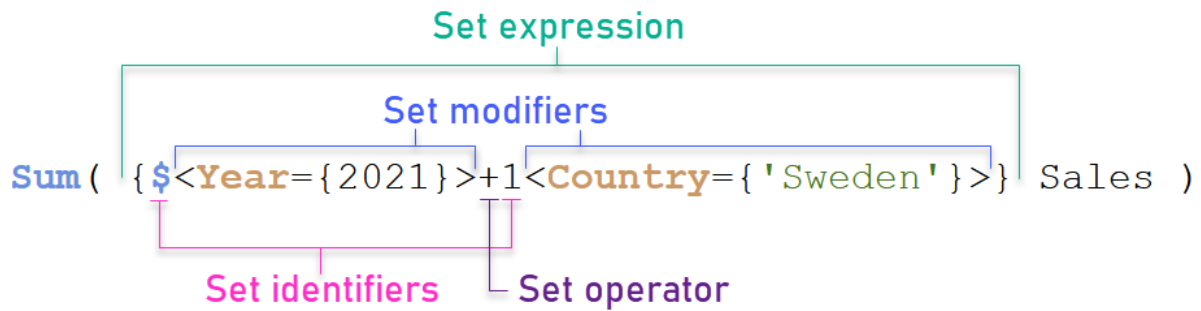
Un'allocatione per l'accesso a Qlik Sense Enterprise Professional, che consente di caricare dati e creare app.

- [Analisi parte 1: introduzione per i nuovi utenti](#)
- [Analisi di gruppo parte 2](#)

#### Elementi in un'espressione set

Le espressioni set sono racchiuse in una funzione di aggregazione, come `sum()`, `Max()`, `Min()`, `Avg()` o `count()`. Le espressioni set sono formate da blocchi costitutivi noti come elementi. Tali elementi sono modificatori, identificatori e operatori set.

## Elementi in un'espressione set



L'espressione set di cui sopra, ad esempio, viene costruita dall'aggregazione `sum(Sales)`. L'espressione set è racchiusa da parentesi graffe esterne: `{ }`

Il primo operando nell'espressione è: `$<Year={2021}>`

Questo operando restituisce le vendite per l'anno 2021 per la selezione corrente. Il modificatore, `<Year={2021}>`, contiene la selezione dell'anno 2021. L'identificatore set `$` indica che l'espressione set è basata sulla selezione corrente.

Il secondo operando nell'espressione è: `1<Country={ 'Sweden' }>`

Questo operando restituisce Sales per Sweden. Il modificatore, `<Country={ 'Sweden' }>`, contiene la selezione del paese Sweden. L'identificatore del set `1` indica che le selezioni effettuate nell'app verranno ignorate.

Infine, l'operatore set `+` indica che l'espressione restituisce un set composto dai record appartenenti a uno qualsiasi dei due operandi set.

## Creazione di un tutorial di espressioni set

Completare le procedure seguenti per creare le espressioni set mostrate in questo tutorial.

### Creare una nuova app e caricare i dati

Procedere come indicato di seguito:

1. Consente di creare una nuova app.
2. Fare clic su **Editor di script**. In alternativa, aprire il menu di navigazione dell'app e selezionare **Editor caricamento dati**.
3. Creare una nuova sezione nell'**Editor caricamento dati**.
4. Copiare i dati seguenti e incollarli nella nuova sezione: [Dati tutorial espressioni set \(page 346\)](#)
5. Fare clic su **Carica dati**. I dati vengono caricati come caricamento inline.

### Creare espressioni set con i modificatori

Il modificatore del set è composto da uno o più nomi di campo, ciascuno seguito da una selezione che dovrebbe essere effettuata sul campo. Il modificatore è racchiuso tra parentesi angolari. Ad esempio, in questa espressione set:

```
sum ( {<Year = {2015}>} Sales )
```

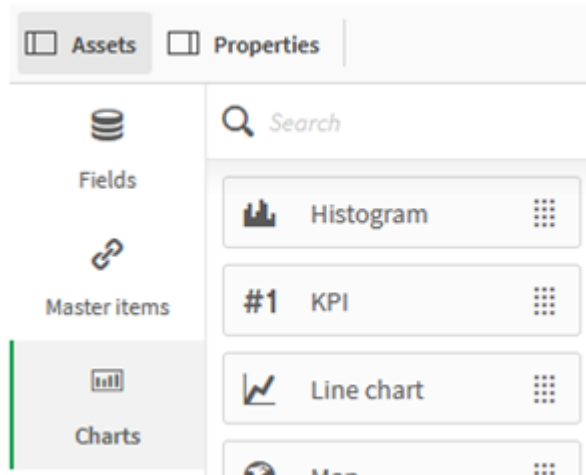
Il modificatore è:

<Year = {2015}>

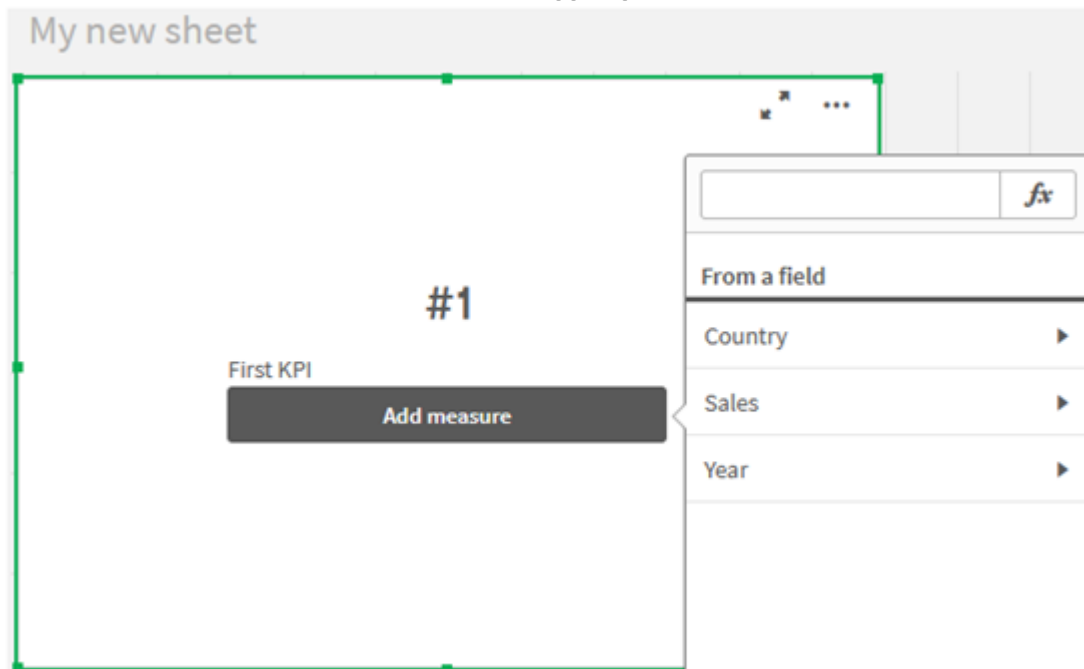
Questo modificatore specifica che verranno selezionati dati relativi all'anno 2015. Le parentesi graffe in cui il modificatore è racchiuso indicano un'espressione set.

**Procedere come indicato di seguito:**

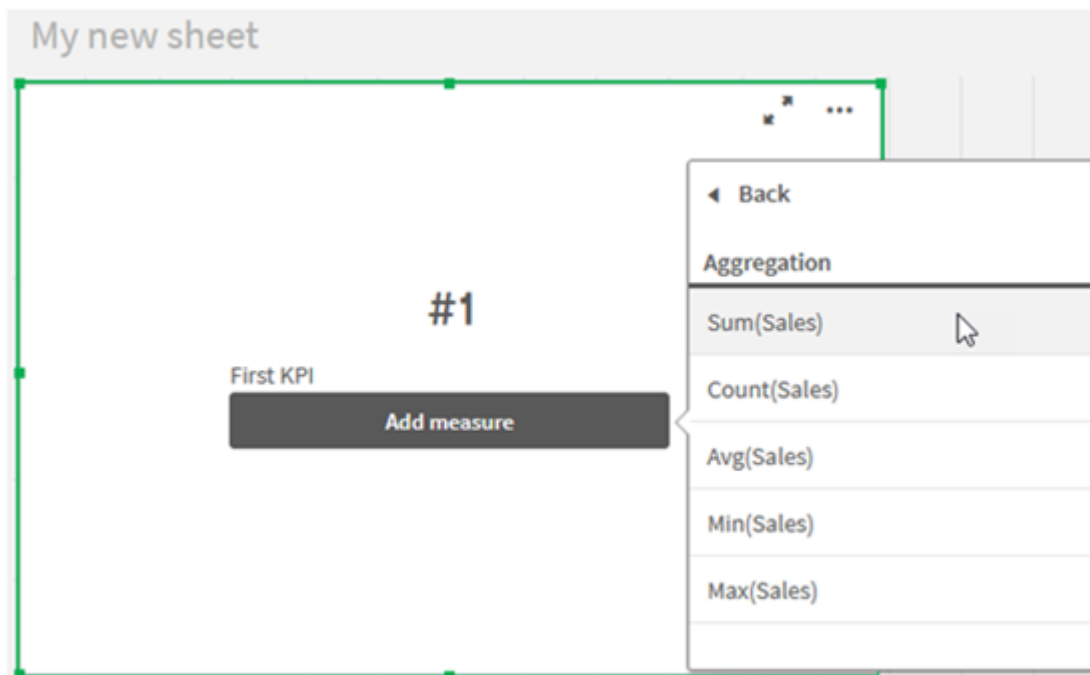
1. In un foglio, aprire il pannello **Asset** dalla barra di navigazione, quindi fare clic su **Grafici**.



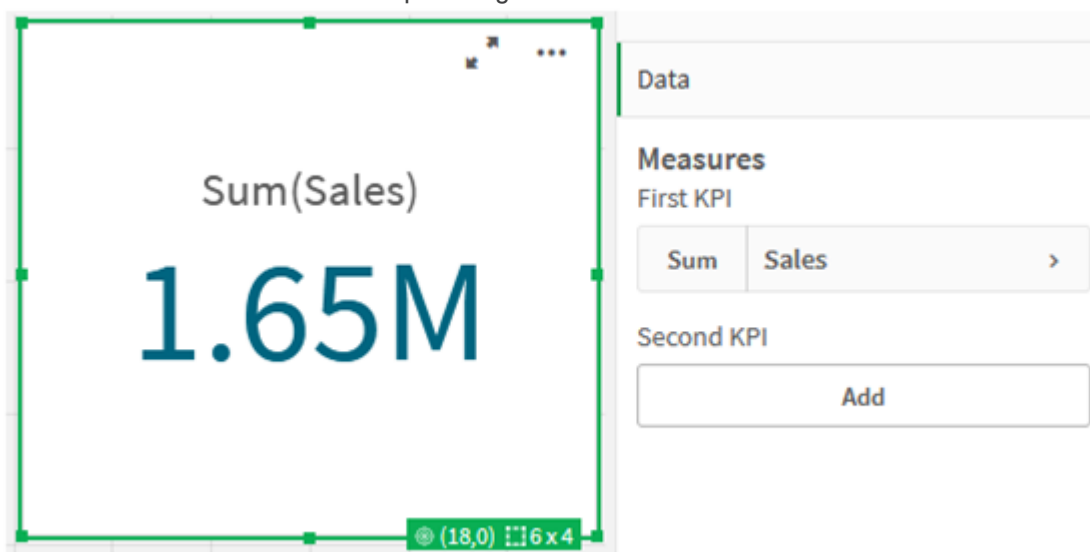
2. Trascinare un **KPI** sul foglio, quindi fare clic su **Aggiungi misura**.



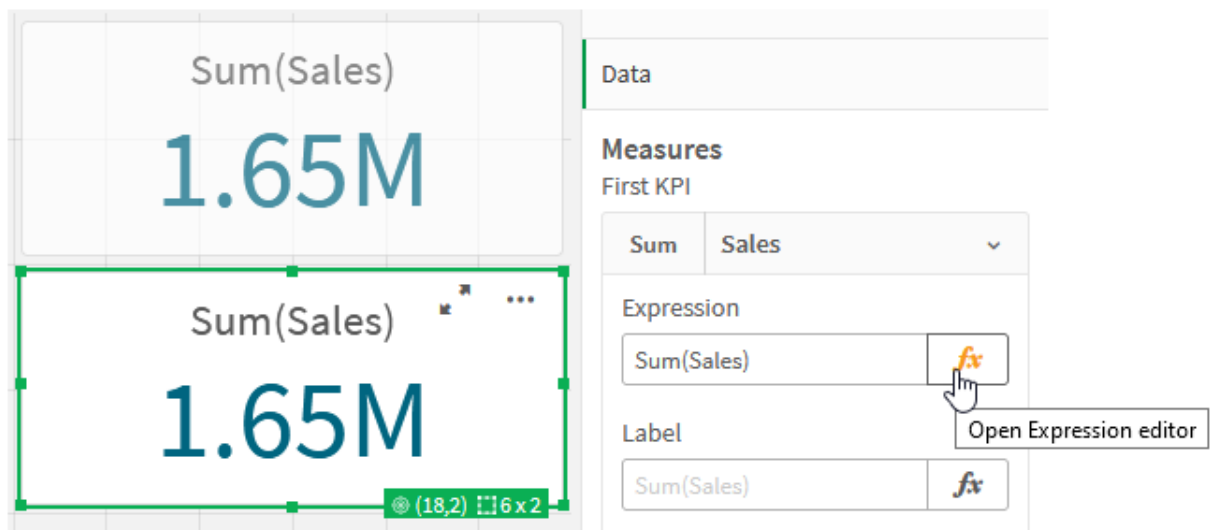
3. Fare clic su `sa1es`, quindi selezionare `sum(sa1es)` per l'aggregazione.



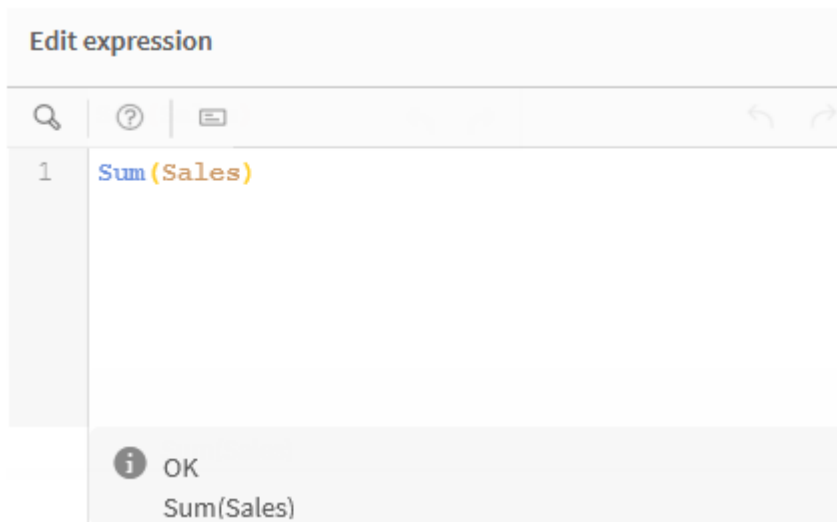
KPI mostra la somma delle vendite per tutti gli anni.



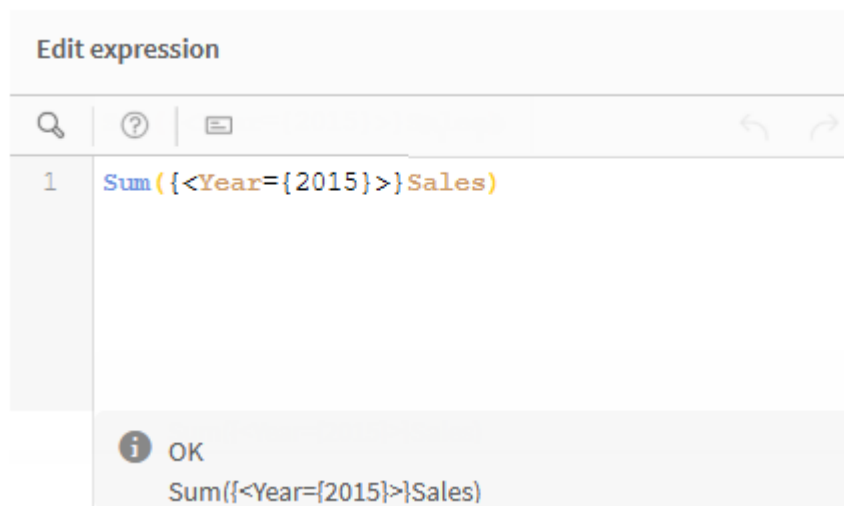
4. Copiare e incollare KPI per creare un nuovo KPI.
5. Fare clic sul nuovo KPI, fare clic su **Vendite** sotto **Misure** e quindi fare clic su **Apri editor delle espressioni**.



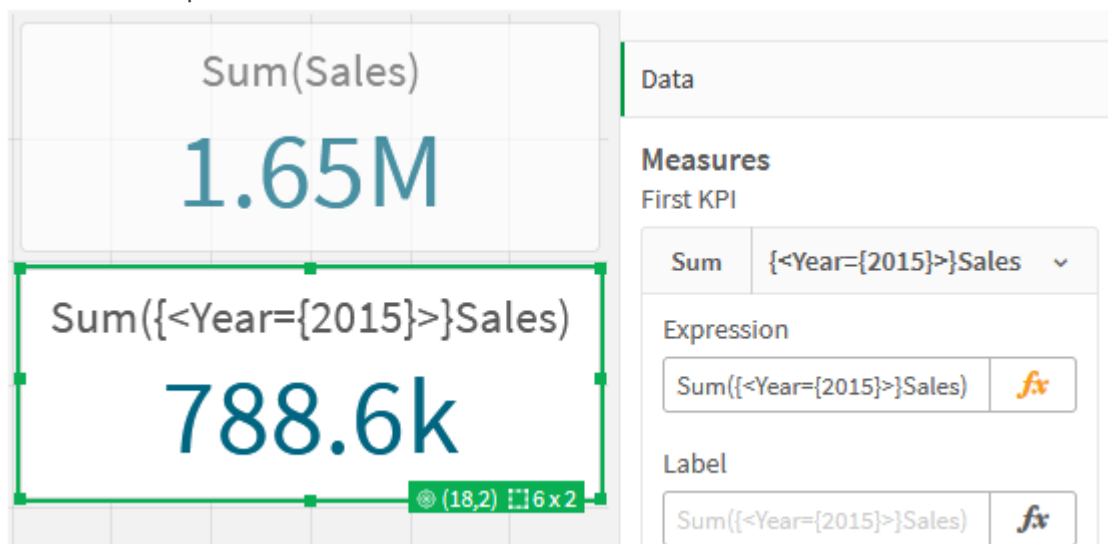
L'editor delle espressioni si apre con l'aggregazione `sum(Sales)`.



6. Nell'editor delle espressioni, creare un'espressione per sommare Sales solo per il 2015:
  - i. Aggiungere parentesi graffe per indicare un'espressione set: `sum({}Sales)`
  - i. Aggiungere parentesi angolari per indicare un modificatore set: `sum({<>}Sales)`
  - ii. Nelle parentesi angolari, aggiungere il campo da selezionare, in questo caso il campo `Year`, seguito da un segno di uguale. Quindi, racchiudere 2015 in un altro set di parentesi graffe. Il modificatore set risultante è: `{<Year={2015}>}`.  
L'intera espressione è:  
`sum({<Year={2015}>}Sales)`



- iii. Fare clic su **Applica** per salvare l'espressione e chiudere l'editor delle espressioni. La somma di Sales per 2015 è mostrata in KPI.



7. Creare altri due KPI con le seguenti espressioni:

`Sum({<Year={2015,2016}>}Sales)`

Il modificatore di cui sopra è `<Year={2015,2016}>`. L'espressione restituirà la somma di Sales per il 2015 e il 2016.

`Sum({<Year={2015},Country={'Germany'}>}Sales)`

Il modificatore di cui sopra è `<Year={2015}, Country={'Germany'}>`. L'espressione restituirà la somma di Sales per il 2015, in cui il 2015 si interseca con Germany.

KPI che utilizzano i modificatori set

### Aggiunta di identificatori set

Le espressioni set sopra utilizzeranno le selezioni correnti come base, dato che un identificatore non è stato utilizzato. Quindi, aggiungere identificatori per specificare il comportamento quando vengono effettuate selezioni.

### Procedere come indicato di seguito:

Sul proprio foglio, creare o copiare le seguenti espressioni set:

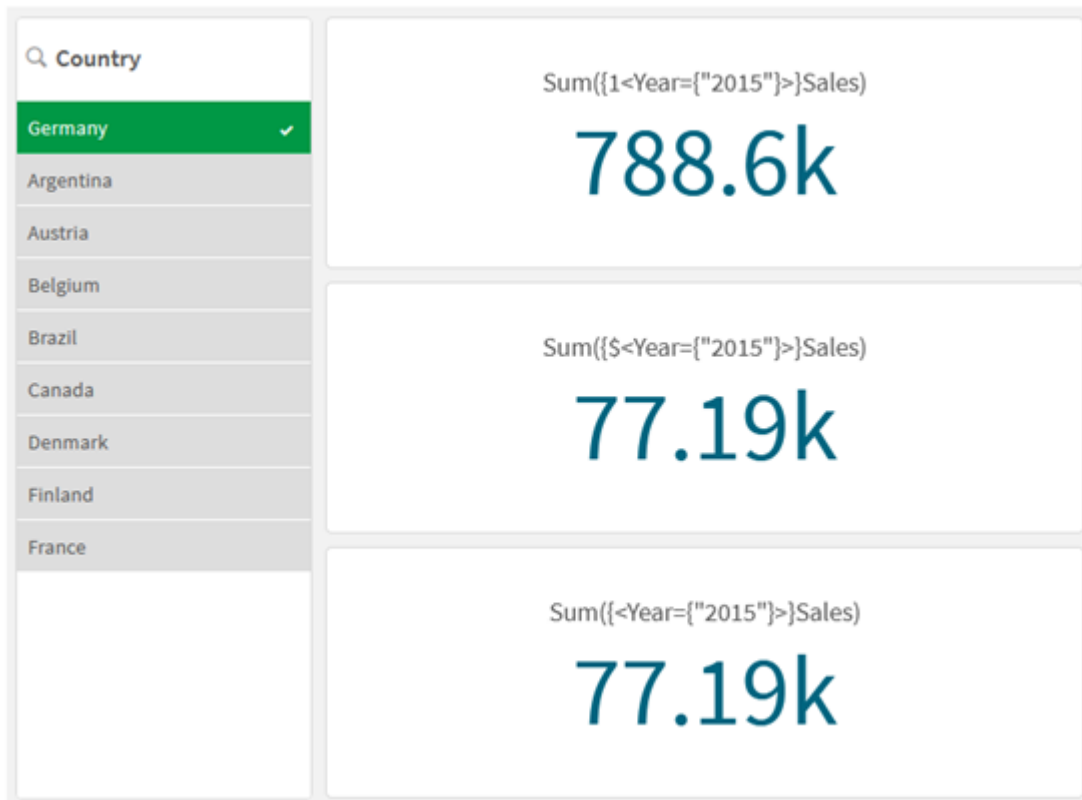
```
Sum({$<Year={"2015"}>}Sales)
```

L'identificatore \$ baserà l'espressione set sulle selezioni correnti effettuate nei dati. Ciò rappresenta anche il comportamento predefinito quando non viene utilizzato un identificatore.

```
Sum({1<Year={"2015"}>}Sales)
```

L'identificatore 1 causerà l'aggregazione di `sum(sales)` sul 2015 per ignorare la selezione corrente. Il valore dell'aggregazione non cambierà quando l'utente effettua altre selezioni. Ad esempio, quando Germany viene selezionato in basso, il valore per la somma aggregata del 2015 non cambia.

*KPI che utilizzano modificatori e identificatori set*



### Aggiunta di operatori

Gli operatori set sono utilizzati per includere, escludere o intersecare serie di dati. Tutti gli operatori utilizzano i gruppi come operandi e restituiscono un gruppo come risultato.

È possibile utilizzare gli operatori set in due situazioni diverse:

- Per eseguire un'operazione set sugli identificatori set, rappresentando i set di record nei dati.
- Per eseguire un'operazione sui set di elementi, sui valori di campo o all'interno di un modificatore set.

### Procedere come indicato di seguito:

Sul proprio foglio, creare o copiare la seguente espressione set:

```
sum({$<Year={2015}>+1<Country={'Germany'}>}Sales)
```

L'operatore segno positivo (+) produce un'unione dei set di dati per 2015 e Germany. Come spiegato con gli identificatori set sopra, l'identificatore con simbolo del dollaro (\$) indica che verranno rispettate le selezioni correnti utilizzate per il primo operando, `<Year={2015}>`. L'identificatore 1 che segnala la selezione verrà ignorato per il secondo operando, `<Country={'Germany'}>`.

KPI che utilizza l'operatore con segno positivo (+)

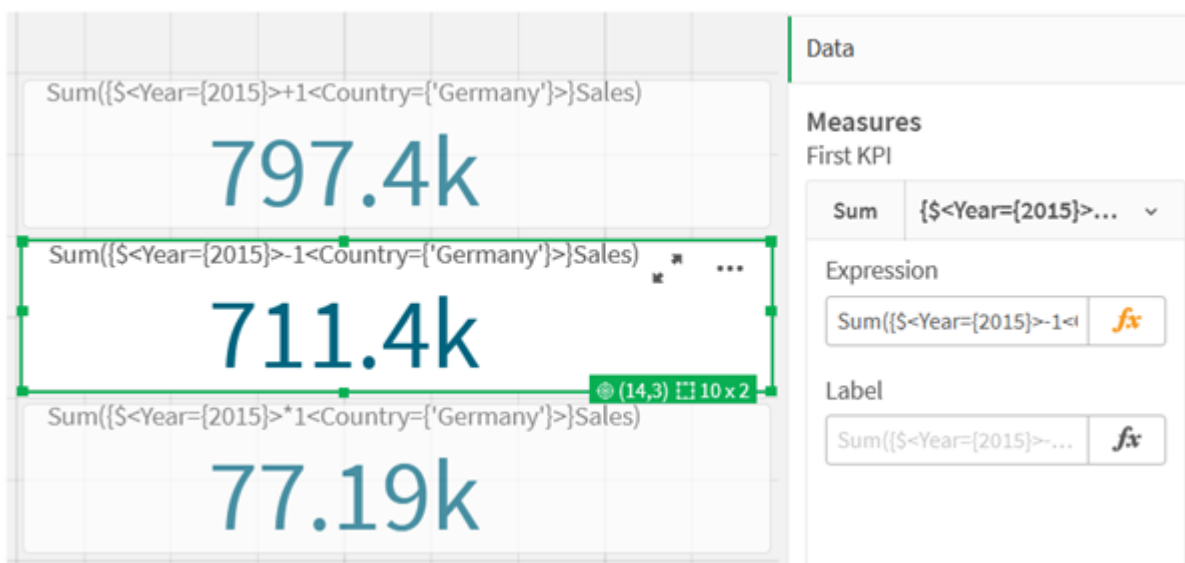


In alternativa, utilizzare un segno meno (-) per restituire una serie di dati che consiste nei record appartenenti a 2015 ma non a Germany. Oppure, utilizzare un asterisco (\*) per restituire un set composto dai record appartenenti a entrambi i set.

`Sum({$<Year={2015}>-1<Country={'Germany'}>}Sales)`

`Sum({$<Year={2015}>*1<Country={'Germany'}>}Sales)`

KPI che utilizzano operatori



### Dati tutorial espressioni set

#### Script di caricamento

Caricare i dati seguenti come caricamento inline, quindi creare le espressioni del grafico nel tutorial.

```
//Create table SalesByCountry
SalesByCountry:
Load * Inline [
Country, Year, Sales
Argentina, 2016, 66295.03
Argentina, 2015, 140037.89
```

```
Austria, 2016, 54166.09
Austria, 2015, 182739.87
Belgium, 2016, 182766.87
Belgium, 2015, 178042.33
Brazil, 2016, 174492.67
Brazil, 2015, 2104.22
Canada, 2016, 101801.33
Canada, 2015, 40288.25
Denmark, 2016, 45273.25
Denmark, 2015, 106938.41
Finland, 2016, 107565.55
Finland, 2015, 30583.44
France, 2016, 115644.26
France, 2015, 30696.98
Germany, 2016, 8775.18
Germany, 2015, 77185.68
];
```

## Sintassi per le espressioni set

La sintassi completa (non includendo l'utilizzo opzionale delle parentesi standard per definire la precedenza) viene descritta mediante la metasintassi Backus-Naur Form:

```
set_expression ::= { set_entity { set_operator set_entity } }
set_entity ::= set_identifier [ set_modifier ] | set_modifier
set_identifier ::= 1 | $ | $N | $_N | bookmark_id | bookmark_name
set_operator ::= + | - | * | /
set_modifier ::= < field_selection { , field_selection } >
field_selection ::= field_name [ = | += | -= | *= | /= ] element_set_expression
element_set_expression ::= [ - ] element_set { set_operator element_set }
element_set ::= [ field_name ] | { element_list } | element_function
element_list ::= element { , element }
element_function ::= ( P | E ) ( [set_expression] [field_name] )
element ::= field_value | " search_mask "
```

## 6.3 Sintassi generale per le espressioni grafiche

Per le espressioni grafiche, è possibile utilizzare la seguente struttura sintattica generale, con molti parametri opzionali:

```
expression ::= ( constant | expressionname | operator1 expression | expression operator2
expression | function | aggregation function | (expression ) )
dove:
```

**constant** è una stringa (un testo, una data o un'ora) racchiusa tra virgolette singole diritte o un numero. Le costanti sono scritte senza separatore delle migliaia e con un punto decimale come separatore decimale.

**expressionname** è il nome (etichetta) di un'altra espressione dello stesso grafico.

**operator1** è un operatore unario (che agisce su un'unica espressione, quella a destra).

**operator2** è un operatore binario (che agisce su due espressioni, una per ogni lato).

```
function ::= functionname ( parameters )  
parameters ::= expression { , expression }
```

Il numero e i tipi dei parametri non sono arbitrari. Dipendono dal tipo di funzione utilizzata.

```
aggregationfunction ::= aggregationfunctionname ( parameters2 )  
parameters2 ::= aggexpression { , aggexpression }
```

Il numero e i tipi dei parametri non sono arbitrari. Dipendono dal tipo di funzione utilizzata.

### 6.4 Sintassi generale per le aggregazioni

Per le aggregazioni, è possibile utilizzare la seguente struttura sintattica generale, con molti parametri opzionali:

```
aggexpression ::= ( fieldref | operator1 aggexpression | aggexpression operator2  
aggexpression | functioninaggr | ( aggexpression ) )
```

**fieldref** è un nome di campo.

```
functionaggr ::= functionname ( parameters2 )
```

Le espressioni e le funzioni possono essere nidificate liberamente e, purché **fieldref** sia sempre racchiuso esattamente da una sola funzione di aggregazione e l'espressione restituisca un valore interpretabile, Qlik Sense non genererà alcun messaggio di errore.

## 7 Operatori

In questa sezione vengono descritti gli operatori che è possibile utilizzare in Qlik Sense. Sono disponibili due tipologie di operatori:

- Operatori unari (utilizzano un solo operando)
- Operatori binari (utilizzano due operandi)

La maggior parte degli operatori è di tipo binario.

È possibile definire le seguenti tipologie di operatori:

- Operatori bit a bit
- Operatori logici
- Operatori numerici
- Operatori relazionali
- Operatori su stringa

### 7.1 Operatori bit a bit

Tutti gli operatori bit a bit convertono (troncano) gli operandi in numeri interi con segno (32 bit) e restituiscono un risultato nello stesso modo. Tutte le operazioni vengono eseguite bit a bit. Se un operando non può essere interpretato come numero, l'operazione restituirà NULL.

Operatori bit a bit

| Operatore | Nome completo | Descrizione                                                                                                                                           |
|-----------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| bitnot    | Bit inverso.  | Operatore unario. L'operazione restituisce l'inverso logico dell'operando eseguito bit a bit.<br><br><b>Esempio:</b><br><br>bitnot 17 restituisce -18 |
| bitand    | Bit AND.      | L'operazione restituisce l'AND logico degli operandi eseguito bit a bit.<br><br><b>Esempio:</b><br><br>17 bitand 7 restituisce 1                      |
| bitor     | Bit OR.       | L'operazione restituisce l'OR logico degli operandi eseguito bit a bit.<br><br><b>Esempio:</b><br><br>17 bitor 7 restituisce 23                       |

| Operatore | Nome completo                  | Descrizione                                                                                                                                                                 |
|-----------|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| bitxor    | Bit OR esclusivo.              | L'operazione restituisce l'OR esclusivo logico degli operandi eseguito bit a bit.<br><br><b>Esempio:</b><br><br>17 bitxor 7 restituisce 22                                  |
| >>        | Spostamento a destra di bit.   | L'operazione restituisce il primo operando spostato a destra. Il numero di passi viene definito nel secondo operando.<br><br><b>Esempio:</b><br><br>8 >> 2 restituisce 2    |
| <<        | Spostamento a sinistra di bit. | L'operazione restituisce il primo operando spostato a sinistra. Il numero di passi viene definito nel secondo operando.<br><br><b>Esempio:</b><br><br>8 << 2 restituisce 32 |

## 7.2 Operatori logici

Tutti gli operatori logici interpretano gli operandi in modo logico e restituiscono True (-1) o False (0) come risultato.

Operatori logici

| Operatore | Descrizione                                                                                                                                                                                                             |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| not       | Inverso logico. Uno dei pochi operatori unari. L'operazione restituisce l'inverso logico dell'operando.                                                                                                                 |
| and       | And logico. Restituisce l'operatore logico AND degli operandi.                                                                                                                                                          |
| or        | Or logico. Restituisce l'operatore logico OR degli operandi.                                                                                                                                                            |
| Xor       | Or esclusivo logico. Restituisce l'esclusivo logico OR degli operandi. È l'equivalente dell'operatore logico OR, ma con la differenza che restituisce un risultato False se entrambi gli operandi hanno un valore True. |

## 7.3 Operatori numerici

Tutti gli operatori numerici usano i valori numerici degli operandi e restituiscono un valore numerico come risultato.

Operatori numerici

| Operatore | Descrizione                                                                                                                                                                                 |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| +         | Segno per i numeri positivi (operatore unario) o addizione aritmetica. L'operatore binario restituisce la somma dei due operandi.                                                           |
| -         | Segno per i numeri negativi (operatore unario) o sottrazione aritmetica. L'operazione unaria restituisce l'operando moltiplicato per -1, e quella binaria la differenza fra i due operandi. |
| *         | Moltiplicazione aritmetica. L'operazione restituisce il prodotto dei due operandi.                                                                                                          |
| /         | Divisione aritmetica. L'operazione restituisce il rapporto fra i due operandi.                                                                                                              |

## 7.4 Operatori relazionali

Tutti gli operatori relazionali confrontano i valori degli operandi e restituiscono True (-1) o False (0) come risultato. Tutti gli operatori relazionali sono binari.

Operatori relazionali

| Operatore | Descrizione                                                                                                                                                                                                      |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <         | Minore di. Viene effettuato un confronto numerico se tutti e due gli operandi possono essere interpretati numericamente. L'operazione restituisce il valore logico della valutazione del confronto.              |
| <=        | Minore di o uguale a. Viene effettuato un confronto numerico se tutti e due gli operandi possono essere interpretati numericamente. L'operazione restituisce il valore logico della valutazione del confronto.   |
| >         | Maggiore di. Viene effettuato un confronto numerico se tutti e due gli operandi possono essere interpretati numericamente. L'operazione restituisce il valore logico della valutazione del confronto.            |
| >=        | Maggiore di o uguale a. Viene effettuato un confronto numerico se tutti e due gli operandi possono essere interpretati numericamente. L'operazione restituisce il valore logico della valutazione del confronto. |
| =         | Uguale a. Viene effettuato un confronto numerico se tutti e due gli operandi possono essere interpretati numericamente. L'operazione restituisce il valore logico della valutazione del confronto.               |

| Operatore       | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <>              | Diverso da. Viene effettuato un confronto numerico se tutti e due gli operandi possono essere interpretati numericamente. L'operazione restituisce il valore logico della valutazione del confronto.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>precedes</b> | <p>A differenza dell'operatore &lt;, non vengono eseguiti tentativi di interpretazione numerica dei valori dell'argomento prima del confronto. L'operazione restituisce true se il valore a sinistra dell'operatore ha una rappresentazione testuale che, confrontando le stringhe, precede la rappresentazione testuale del valore di destra.</p> <p><b>Esempio:</b></p> <p>'1 ' precedes ' 2 ' restituisce FALSE</p> <p>' 1 ' precedes ' 2 ' restituisce TRUE</p> <p>dato che il valore ASCII di uno spazio ( ' ') ha un valore inferiore rispetto al valore ASCII di un numero.</p> <p>Mentre:</p> <p>'1 ' &lt; ' 2 ' restituisce TRUE</p> <p>' 1 ' &lt; ' 2 ' restituisce TRUE</p> |
| <b>follows</b>  | <p>A differenza dell'operatore &gt;, non vengono eseguiti tentativi di interpretazione numerica dei valori dell'argomento prima del confronto. L'operazione restituisce true se il valore a sinistra dell'operatore ha una rappresentazione testuale che, confrontando le stringhe, segue la rappresentazione testuale del valore di destra.</p> <p><b>Esempio:</b></p> <p>' 2 ' follows '1 ' restituisce FALSE</p> <p>'2' follows ' 1' restituisce TRUE</p> <p>dato che il valore ASCII di uno spazio ( ' ') ha un valore inferiore rispetto al valore ASCII di un numero.</p> <p>Mentre:</p> <p>' 2 ' &gt; ' 1 ' restituisce TRUE</p> <p>' 2 ' &gt; '1 ' restituisce TRUE</p>        |

## 7.5 Operatori su stringa

Esistono due operatori su stringa. Uno utilizza i valori delle stringhe degli operandi e restituisce una stringa come risultato. L'altro confronta gli operandi e restituisce un valore booleano per indicare la corrispondenza.

### &

Concatenazione di stringhe. L'operazione restituisce una stringa di testo, costituita dall'unione delle due stringhe degli operandi.

#### Esempio:

`'abc' & 'xyz'` restituisce `'abcxyz'`

### like

Confronto tra stringhe con caratteri speciali. L'operazione restituisce un valore booleano True (-1) se la stringa prima dell'operatore coincide con la stringa dopo l'operatore. La seconda stringa può contenere i caratteri jolly \* (qualsiasi numero di caratteri arbitrari) oppure ? (un carattere arbitrario).

#### Esempio:

`'abc' like 'a*'` restituisce True (-1)

`'abcd' like 'a?c*'` restituisce True (-1)

`'abc' like 'a??bc'` restituisce False (0)

## 8 Funzioni per script e grafici

Trasformare e aggregare i dati mediante le funzioni negli script di caricamento dati e nelle espressioni del grafico.

È possibile utilizzare diverse funzioni nello stesso modo sia negli script di caricamento dei dati che nelle espressioni grafiche, anche se vi sono alcune eccezioni:

- Alcune funzioni possono essere utilizzate esclusivamente negli script di caricamento dei dati e si caratterizzano dal fatto che sono funzioni di script.
- Alcune funzioni possono essere utilizzate esclusivamente nelle espressioni grafiche e si caratterizzano dal fatto che sono funzioni grafiche.
- Alcune funzioni possono essere utilizzate sia negli script di caricamento dei dati che nelle espressioni grafiche, ma con alcune differenze nei parametri e nell'applicazione. Tali funzioni vengono descritte in argomenti separati relativi alla funzione di script o alla funzione grafica.

### 8.1 Connessioni di analisi per estensioni lato server (SSE, Server-Side Extension)

Le funzioni abilitate dalle connessioni di analisi saranno visibili solo una volta configurate le connessioni di analisi e avviato Qlik Sense.

Per configurare le connessioni di analisi nella QMC, vedere l'argomento "Creazione di una connessione di analisi" nella guida Gestione di siti Qlik Sense.

Per configurare le connessioni di analisi in Qlik Sense Desktop è necessario modificare il file *Settings.ini*, vedere l'argomento "Configuring analytic connections in Qlik Sense Desktop" ("Configurazione di connessioni di analisi in ") nella guida Qlik Sense Desktop.

### 8.2 Funzioni di aggregazione

La serie di funzioni conosciute come funzioni di aggregazione è costituita da funzioni che prendono più valori di campo come input e restituiscono un singolo risultato per gruppo, dove l'aggregazione viene definita con una dimensione del grafico o una clausola **group by** nell'istruzione dello script.

Le funzioni di aggregazione comprendono **Sum()**, **Count()**, **Min()**, **Max()** e molte altre ancora.

La maggior parte delle funzioni di aggregazione può essere utilizzata sia nello script di caricamento dei dati che nelle espressioni del grafico, anche se la sintassi sarà diversa.

### Limiti:

Il parametro della funzione di aggregazione non deve contenere altre funzioni di aggregazione, a meno che tali aggregazioni interne non contengano il qualificatore **TOTAL**. Nel caso di aggregazioni nidificate più complesse, utilizzare la funzione avanzata **Aggr** in combinazione con una dimensione specificata.

Quando si denomina un'entità, evitare di assegnare lo stesso nome a più campi, variabili o misure. Esiste un rigoroso ordine di precedenza per risolvere i conflitti tra entità con nomi identici. Questo ordine si riflette in qualsiasi oggetto o contesto in cui queste entità vengono utilizzate. L'ordine di precedenza è il seguente:

- All'interno di un'aggregazione, un campo ha la precedenza su una variabile. Le etichette delle misure non sono rilevanti nelle aggregazioni e non hanno priorità.
- Al di fuori di un'aggregazione, un'etichetta di misura ha la precedenza su una variabile, che a sua volta ha la precedenza su un nome di campo.
- Inoltre, al di fuori di un'aggregazione, una misura può essere riutilizzata facendo riferimento alla sua etichetta, a meno che non si tratti di un'etichetta calcolata. In tale situazione, la misura diminuisce di significato per ridurre il rischio di autoreferenzialità e in questo caso il nome sarà sempre interpretato prima come etichetta di misura, poi come nome di campo e infine come nome di variabile.

### Utilizzo delle funzioni di aggregazione in uno script di caricamento dei dati

Le funzioni di aggregazione possono essere utilizzate soltanto all'interno di istruzioni **LOAD** e **SELECT**.

### Utilizzo delle funzioni di aggregazione nelle espressioni grafiche

Il parametro della funzione di aggregazione non deve contenere altre funzioni di aggregazione, a meno che tali aggregazioni interne non contengano il qualificatore **TOTAL**. Nel caso di aggregazioni nidificate più complesse, utilizzare la funzione avanzata **Aggr** in combinazione con una dimensione specificata.

La funzione di aggregazione aggrega il set di possibili record definiti dalla selezione. Tuttavia, è possibile definire un set alternativo di record mediante un'espressione set nell'analisi di gruppo.

### Come vengono calcolate le aggregazioni

Un'aggregazione si chiude intorno ai record di una tabella specifica, aggregando i record al suo interno. Ad esempio, **Count(<Field>)** conterà il numero di record nella tabella in cui risiede <Field>. Se si desidera aggregare solo i valori di campo distinti, utilizzare la clausola **distinct**, come **Count(distinct <Field>)**.

Se la funzione di aggregazione contiene campi da tabelle diverse, la funzione di aggregazione si chiude sui record del prodotto incrociato delle tabelle dei campi costituenti. Ciò penalizza le prestazioni e per questo motivo tali aggregazioni devono essere evitate, specialmente in presenza di grandi quantità di dati.

### Aggregazione dei campi chiave

Il modo in cui vengono calcolate le aggregazioni fa sì che non sia possibile aggregare campi chiave, perché non è chiaro quale tabella debba essere utilizzata per l'aggregazione. Ad esempio, se il campo <Key> collega due tabelle, non è chiaro se **Count(<Key>)** debba restituire il numero di record dalla prima o

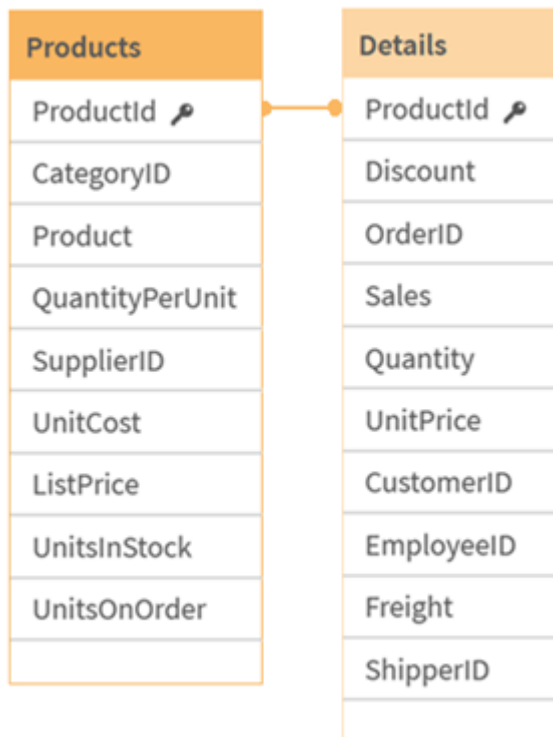
dalla seconda tabella.

Tuttavia, se si utilizza la clausola **distinct**, l'aggregazione viene definita adeguatamente e può essere calcolata per un campo chiave collegato su due tabelle.

Se si utilizza un campo chiave all'interno di una funzione di aggregazione senza la clausola **distinct**, Qlik Sense restituirà un numero che può essere privo di significato. La soluzione consiste nell'utilizzare la clausola **distinct** o una copia della chiave - una copia che risiede in una sola tabella.

Ad esempio, nelle tabelle seguenti, *ProductId* rappresenta la chiave tra le tabelle.

*Chiave ProductId tra le tabelle Products e Details*



*Count(ProductId)* può essere conteggiato nella tabella *Products* (che presenta un solo record per prodotto - *ProductId* è la chiave primaria) oppure può essere conteggiato nella tabella *Details* (che disporrà molto probabilmente di svariati record per prodotto). Se si desidera conteggiare il numero di prodotti distinti, utilizzare *Count(distinct ProductId)*. Se si desidera conteggiare il numero di righe in una tabella specifica, non utilizzare la chiave.

### Aggregazioni di campi chiave contenute in tre o più tabelle

Il prefisso **distinct** funziona solo con i campi chiave che collegano fino a due tabelle. Quando si raggruppa un'aggregazione per un campo chiave presente in tre o più tabelle, qualsiasi operazione che richieda informazioni sulla frequenza per un campo restituirà NULL. Nei casi di campi chiave che collegano tre o più tabelle, è necessario utilizzare invece una copia non chiave del campo.

## Funzioni di aggregazione di base

### Prospetto delle funzioni di aggregazione di base

Le funzioni di aggregazione di base sono un gruppo delle funzioni di aggregazione più comuni.

Ciascuna funzione viene descritta ulteriormente dopo la panoramica. È inoltre possibile fare clic sul nome della funzione nella sintassi per accedere immediatamente ai dettagli per tale funzione specifica.

### Funzioni di aggregazione di base nello script di caricamento dei dati

#### FirstSortedValue

**FirstSortedValue()** restituisce il valore dell'espressione specificata in **value** che corrisponde al risultato della classificazione dell'argomento a **sort\_weight**, ad esempio, il nome del prodotto con il prezzo unitario più basso. Il valore n nell'ordine di classificazione può essere specificato in **rank**. Se più valori risultanti condividono lo stesso **sort\_weight** per il **rank** specificato, la funzione restituisce NULL. I valori classificati vengono ripetuti su un insieme di record, come definito da una clausola **group by**, o aggregati tra la serie di dati completa, qualora la clausola **group by** non sia stata definita.

```
FirstSortedValue ([ distinct ] expression, sort_weight [, rank ])
```

#### Max

**Max()** individua il valore numerico più alto dei dati aggregati nell'espressione, come definito da una clausola **group by**. Specificando un **rank** n, è possibile trovare il valore n-esimo più alto.

```
Max ( expression[, rank])
```

#### Min

**Min()** restituisce il valore numerico più basso dei dati aggregati nell'espressione, come definito da una clausola **group by**. Specificando un **rank** n, è possibile trovare il valore n-esimo più basso.

```
Min ( expression[, rank])
```

#### Mode

**Mode()** restituisce il valore più comune, il valore mode, dei dati aggregati nell'espressione, come definito da una clausola **group by**. La funzione **Mode()** può restituire valori numerici e valori di testo.

```
Mode (expression )
```

#### Only

**Only()** restituisce un valore se esiste esclusivamente un unico risultato possibile dai dati aggregati. Se i record contengono solo un valore verrà restituito tale valore, altrimenti verrà restituito NULL. Utilizzare la clausola **group by** per valutare più record. La funzione **Only()** può restituire valori numerici e valori di testo.

```
Only (expression )
```

#### Sum

**Sum()** calcola il totale dei valori aggregati nell'espressione, come definito da una clausola **group by**.

```
Sum ([distinct]expression)
```

## Funzioni di aggregazione di base nelle espressioni grafiche

Le funzioni di aggregazione nei grafici possono essere utilizzate solo nei campi delle espressioni grafiche. L'espressione di argomento di una funzione di aggregazione non deve contenere un'altra funzione di aggregazione.

### FirstSortedValue

**FirstSortedValue()** restituisce il valore dell'espressione specificata in **value** che corrisponde al risultato della classificazione dell'argomento a **sort\_weight**, ad esempio, il nome del prodotto con il prezzo unitario più basso. Il valore n nell'ordine di classificazione può essere specificato in **rank**. Se più valori risultanti condividono lo stesso **sort\_weight** per il **rank** specificato, la funzione restituisce NULL.

```
FirstSortedValue - funzione per grafici ([{SetExpression}] [DISTINCT] [TOTAL [<fld {,fld}>]] value, sort_weight [,rank])
```

### Max

**Max()** rileva il valore più alto dei dati aggregati. Specificando **rank** n, è possibile trovare il valore n-esimo più alto.

```
Max - funzione per grafici ([{SetExpression}] [DISTINCT] [TOTAL [<fld {,fld}>]] expr [,rank])
```

### Min

**Min()** trova il valore più basso dei dati aggregati. Specificando un **rank** n, è possibile trovare il valore n-esimo più basso.

```
Min - funzione per grafici ([{SetExpression}] [DISTINCT] [TOTAL [<fld {,fld}>]] expr [,rank])
```

### Mode

**Mode()** trova il valore più comune, il valore della modalità, nei dati aggregati. La funzione **Mode()** può elaborare valori di testo e valori numerici.

```
Mode - funzione per grafici ([{SetExpression}] [TOTAL [<fld {,fld}>]] expr)
```

### Only

**Only()** restituisce un valore se esiste esclusivamente un unico risultato possibile dai dati aggregati. Ad esempio, la ricerca dell'unico prodotto con prezzo unitario = 9 restituirà NULL se più di un prodotto ha un prezzo unitario di 9.

```
Only - funzione per grafici ([{SetExpression}] [DISTINCT] [TOTAL [<fld {,fld}>]] expr)
```

### Sum

**Sum()** calcola il totale dei valori dati dall'espressione o dal campo nei dati aggregati.

```
Sum - funzione per grafici ([{SetExpression}] [DISTINCT] [TOTAL [<fld {,fld}>]] expr)
```

## FirstSortedValue

**FirstSortedValue()** restituisce il valore dell'espressione specificata in **value** che corrisponde al risultato della classificazione dell'argomento a **sort\_weight**, ad esempio, il nome del prodotto con il prezzo unitario più basso. Il valore n nell'ordine di classificazione può essere specificato in **rank**. Se più valori risultanti condividono lo stesso **sort\_weight** per il **rank** specificato, la funzione restituisce NULL. I valori classificati vengono ripetuti su un insieme di record, come definito da una clausola **group by**, o aggregati tra la serie di dati completa, qualora la clausola **group by** non sia stata definita.

### Sintassi:

```
FirstSortedValue ([ distinct ] value, sort-weight [, rank ])
```

**Tipo di dati restituiti:** duale

### Argomenti:

#### Argomenti

| Argomento                 | Descrizione                                                                                                                                                                                                                                                                                                                           |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value<br>Expression       | La funzione trova il valore dell'espressione <b>value</b> che corrisponde al risultato della classificazione di <b>sort_weight</b> .                                                                                                                                                                                                  |
| sort-weight<br>Expression | L'espressione contenente i dati da ordinare. Viene trovato il primo valore (il più basso) di <b>sort_weight</b> dal quale viene determinato il valore corrispondente dell'espressione <b>value</b> . Inserendo un segno meno davanti a <b>sort_weight</b> , la funzione restituisce invece l'ultimo valore ordinato (il più elevato). |
| rank<br>Expression        | Dichiarando un valore "n" di <b>rank</b> maggiore di 1, si otterrà il valore n-esimo nell'ordine.                                                                                                                                                                                                                                     |
| distinct                  | Se la parola <b>DISTINCT</b> è riportata prima degli argomenti della funzione, i duplicati risultanti dalla valutazione degli argomenti della funzione vengono ignorati.                                                                                                                                                              |

### Esempi e risultati:

Aggiungere lo script di esempio all'app ed eseguirlo. Aggiungere quindi a un foglio nell'app almeno i campi elencati nella colonna dei risultati per visualizzare il risultato.

Per ottenere lo stesso aspetto della colonna dei risultati mostrata di seguito, nel pannello delle proprietà, in Ordinamento passare da Automatico a Personalizza, quindi deselezionare l'ordinamento numerico e alfabetico.

### Esempi di script

| Esempio                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Risultato                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>Temp: LOAD * inline [ Customer Product OrderNumber UnitSales CustomerID Astrida AA 1 10 1 Astrida AA 7 18 1 Astrida BB 4 9 1 Astrida CC 6 2 1 Betacab AA 5 4 2 Betacab BB 2 5 2 Betacab DD 12 25 2 Canutility AA 3 8 3 Canutility CC 13 19 3 Divadip AA 9 16 4 Divadip AA 10 16 4 Divadip DD 11 10 4 ] (delimiter is ' ');  FirstSortedValue: LOAD Customer,FirstSortedValue(Product, UnitSales) as MyProductWithSmallestOrderByCustomer Resident Temp Group By Customer;</pre> | <p>Customer<br/>MyProductWithSmallestOrderByCustomer<br/>Astrida CC<br/>Betacab AA<br/>Canutility AA<br/>Divadip DD</p> <p>La funzione classifica UnitSales in ordine crescente, accedendo al Customer con il valore di UnitSales minimo, l'ordine più piccolo.</p> <p>Perché CC corrisponde all'ordine più piccolo (valore di UnitSales=2) per cliente Astrida. AA corrisponde all'ordine più piccolo (4) valore per il cliente Betacab, AA corrisponde all'ordine più piccolo (8) per il cliente Canutility e DD corrisponde all'ordine più piccolo (10) per il cliente Divadip..</p>                                     |
| <p>Presupponendo che la tabella <b>Temp</b> venga caricata come nell'esempio precedente:</p> <pre>LOAD Customer,FirstSortedValue(Product, -UnitSales) as MyProductWithLargestOrderByCustomer Resident Temp Group By Customer;</pre>                                                                                                                                                                                                                                                  | <p>Customer<br/>MyProductWithLargestOrderByCustomer<br/>Astrida AA<br/>Betacab DD<br/>Canutility CC<br/>Divadip -</p> <p>il segno meno precede l'argomento sort_weight, in modo tale che la funzione classifichi prima i più elevati.</p> <p>Perché AA corrisponde all'ordine più elevato (valore di UnitSales:18) per il cliente Astrida, DD corrisponde all'ordine più elevato (12) per il cliente Betacab e CC corrisponde all'ordine più elevato (13) per il cliente Canutility. Sono presenti due valori identici per l'ordine più elevato (16) per il cliente Divadip, pertanto viene prodotto un risultato null.</p> |

| Esempio                                                                                                                                                                                                                                          | Risultato                                                                                                                                                                                                                                                                                                                                                                                                             |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Presupponendo che la tabella <b>Temp</b> venga caricata come nell'esempio precedente:</p> <pre>LOAD Customer, FirstSortedValue(distinct Product, - UnitsSales) as MyProductWithSmallestOrderByCustomer Resident Temp Group By Customer;</pre> | <p>Customer<br/>MyProductWithLargestOrderByCustomer<br/>Astrida AA<br/>Betacab DD<br/>Canutility CC<br/>Divadip AA</p> <p>si verifica la stessa situazione dell'esempio precedente, tranne per il fatto che viene utilizzato il qualificatore <b>distinct</b>. Ciò comporta un risultato doppio per Divadip che deve non deve essere preso in considerazione, consentendo la presentazione di un valore non null.</p> |

### FirstSortedValue - funzione per grafici

**FirstSortedValue()** restituisce il valore dell'espressione specificata in **value** che corrisponde al risultato della classificazione dell'argomento a **sort\_weight**, ad esempio, il nome del prodotto con il prezzo unitario più basso. Il valore n nell'ordine di classificazione può essere specificato in **rank**. Se più valori risultanti condividono lo stesso **sort\_weight** per il **rank** specificato, la funzione restituisce NULL.

#### Sintassi:

```
FirstSortedValue([SetExpression] [DISTINCT] [TOTAL [<fld {,fld}>]] value, sort_weight [,rank])
```

Tipo di dati restituiti: duale

#### Argomenti

| Argomento            | Descrizione                                                                                                                                                                                                                                                                                                                                           |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>value</b>         | Campo di output. La funzione trova il valore dell'espressione <b>value</b> che corrisponde al risultato della classificazione di <b>sort_weight</b> .                                                                                                                                                                                                 |
| <b>sort_weight</b>   | Campo di input. L'espressione contenente i dati da ordinare. Viene trovato il primo valore (il più basso) di <b>sort_weight</b> dal quale viene determinato il valore corrispondente dell'espressione <b>value</b> . Inserendo un segno meno davanti a <b>sort_weight</b> , la funzione restituisce invece l'ultimo valore ordinato (il più elevato). |
| <b>rank</b>          | Dichiarando un valore "n" di <b>rank</b> maggiore di 1, si otterrà il valore n-esimo nell'ordine.                                                                                                                                                                                                                                                     |
| <b>SetExpression</b> | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                   |
| <b>DISTINCT</b>      | Se la parola <b>DISTINCT</b> è riportata prima degli argomenti della funzione, i duplicati risultanti dalla valutazione degli argomenti della funzione vengono ignorati.                                                                                                                                                                              |

| Argomento    | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>TOTAL</b> | <p>Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.</p> <p>Utilizzando <b>TOTAL [&lt;fld {fld}&gt;]</b>, dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili.</p> |

### Esempio: espressioni del grafico

| Esempio                                               | Risultato                                                                                                                       |
|-------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| FirstSortedValue (Product, UnitPrice)                 | Restituisce il valore Product con il più basso UnitPrice.                                                                       |
| FirstSortedValue (Product, UnitPrice, 2)              | Restituisce il valore Product con il secondo UnitPrice più basso.                                                               |
| FirstSortedValue (Customer, -UnitPrice, 2)            | Restituisce il Customer con il valore Product con il secondo UnitPrice più alto.                                                |
| FirstSortedValue (Customer, -UnitPrice*UnitsSales, 2) | Restituisce il Customer con il secondo valore di ordinamento più elevato per le vendite, UnitPrice moltiplicato per UnitsSales. |

### Esempio: principi fondamentali della funzione FirstSortedValue

#### Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata Example.
- I seguenti campi nella tabella dati:
  - Customer
  - Product
  - UnitsSales
  - UnitPrice

#### Script di caricamento

Example:

```
LOAD * inline [
Customer|Product|UnitsSales|UnitPrice
Astrida|AA|4|16
Astrida|AA|10|15
Astrida|BB|9|9
```

```
Betacab|BB|5|10
Betacab|CC|2|20
Betacab|DD||25
Canutility|AA|8|15
Canutility|CC||19
] (delimiter is '|');
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questi campi come misure:

- `=FirstSortedValue (Product, UnitPrice)`, per calcolare il prodotto con il prezzo unitario più basso.
- `=FirstSortedValue (Product, UnitPrice, 2)`, per calcolare il prodotto con il secondo prezzo unitario più basso.
- `=FirstSortedValue (Customer, -UnitPrice, 2)`, per calcolare il cliente con il prodotto che ha il secondo prezzo unitario più alto.
- `=FirstSortedValue (Customer, UnitPrice, 3)`, per calcolare il cliente con il prodotto che ha il terzo prezzo unitario più alto.
- `=FirstSortedValue (Customer, -UnitPrice*UnitSales, 2)` per calcolare il cliente con il secondo valore di ordinamento più elevato per le vendite, `UnitPrice` moltiplicato per `UnitSales`.

Results table

| FirstSortedValue<br>(Product,<br>UnitPrice) | FirstSortedValue<br>(Product,<br>UnitPrice, 2) | FirstSortedValue<br>(Customer, -<br>UnitPrice, 2) | FirstSortedValue<br>(Customer,<br>UnitPrice, 3) | FirstSortedValue<br>(Customer, -<br>UnitPrice*UnitSales, 2) |
|---------------------------------------------|------------------------------------------------|---------------------------------------------------|-------------------------------------------------|-------------------------------------------------------------|
| BB                                          | BB                                             | Betacab                                           | -                                               | Canutility                                                  |

La prima colonna restituisce il valore `BB`, poiché questo è il prodotto con il valore `UnitPrice` più basso (9).

La seconda colonna restituisce il valore `BB`, poiché questo è il prodotto con il secondo valore `UnitPrice` più basso (10).

La terza colonna restituisce il valore `Betacab`, che è il `Customer` con il `Product` che ha il secondo valore `UnitPrice` più alto (20).

La quarta colonna restituisce `NULL`, perché sono presenti due valori di `Customer` (`Astrida` e `Canutility`) con lo stesso valore di classificazione `UnitPrice`, il terzo più basso (15).

La quinta colonna restituisce `Canutility`, che è il `Customer` con il secondo valore di ordinamento più elevato per le vendite, `UnitPrice` moltiplicato per `UnitSales` (120).

### Esempio: scenario per la funzione FirstSortedValue

Espressione del grafico

#### Panoramica

Un set di dati contiene i dati sulle vendite per i primi tre mesi dell'anno. Il responsabile regionale vuole identificare i prodotti che vendono di più e di meno per pianificare le scorte future.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Date`
  - `SalesProduct`
  - `Category`
  - `Sales`

#### Script di caricamento

Example:

```
Load * inline [
Date,SalesProduct, Category, Sales
2024-01-01, Apple, Fruit, 100
2024-01-15, Banana, Fruit, 80
2024-01-30, Carrot, Vegetable, 50
2024-02-05, Apple, Fruit, 120
2024-02-20, Banana, Fruit, 90
2024-02-28, Carrot, Vegetable, 60
2024-03-10, Apple, Fruit, 110
2024-03-22, Banana, Fruit, 85
2024-03-31, Carrot, Vegetable, 55
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `Category`

Creare le seguenti misure:

- `=FirstSortedValue(SalesProduct, -Sales)`, per calcolare il prodotto più venduto in ogni categoria.
- `=FirstSortedValue(TOTAL SalesProduct, -Sales)`, per calcolare il prodotto più venduto in generale.
- `=FirstSortedValue(TOTAL SalesProduct, Sales)`, per calcolare il prodotto meno venduto in generale.

Results table

| Category  | FirstSortedValue<br>(SalesProduct, -Sales) | FirstSortedValue(TOTAL<br>SalesProduct, -Sales) | FirstSortedValue(TOTAL<br>SalesProduct, Sales) |
|-----------|--------------------------------------------|-------------------------------------------------|------------------------------------------------|
| Fruit     | Apple                                      | Apple                                           | Carrot                                         |
| Vegetable | Carrot                                     | Apple                                           | Carrot                                         |

I risultati indicano che Apple e carrot sono i prodotti più venduti nelle rispettive categorie, Fruit e vegetable. Tuttavia, in tutte le categorie, Apple è stato il prodotto più venduto e carrot è stato il prodotto meno venduto.

## Max

**Max()** individua il valore numerico più alto dei dati aggregati nell'espressione, come definito da una clausola **group by**. Specificando un **rank** n, è possibile trovare il valore n-esimo più alto.

### Sintassi:

```
Max ( expr [, rank] )
```

**Tipo di dati restituiti:** numerico

### Argomenti:

Argomenti

| Argomento          | Descrizione                                                                                                                                                                                                                                   |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| expr<br>Expression | L'espressione o il campo contenente i dati da misurare.                                                                                                                                                                                       |
| rank<br>Expression | Il valore predefinito di <b>rank</b> è 1, che corrisponde al valore più elevato. Specificando <b>rank</b> come 2 verrà restituito il secondo valore più elevato. Se <b>rank</b> è 3, verrà restituito il terzo valore più elevato e così via. |

### Esempi e risultati:

Aggiungere lo script di esempio all'app ed eseguirlo. Aggiungere quindi a un foglio nell'app almeno i campi elencati nella colonna dei risultati per visualizzare il risultato.

Per ottenere lo stesso aspetto della colonna dei risultati mostrata di seguito, nel pannello delle proprietà, in Ordinamento passare da Automatico a Personalizza, quindi deselezionare l'ordinamento numerico e alfabetico.

### Esempio:

```
Temp:
LOAD * inline [
Customer|Product|OrderNumber|UnitSales|CustomerID
Astrida|AA|1|10|1
Astrida|AA|7|18|1
Astrida|BB|4|9|1
```

```
Astrida|CC|6|2|1  
Betacab|AA|5|4|2  
Betacab|BB|2|5|2  
Betacab|DD  
Canutility|DD|3|8  
Canutility|CC  
] (delimiter is '|');
```

Max:

```
LOAD Customer, Max(UnitSales) as MyMax Resident Temp Group By Customer;
```

Tabella risultante

| Customer   | MyMax |
|------------|-------|
| Astrida    | 18    |
| Betacab    | 5     |
| Canutility | 8     |

### Esempio:

Presupponendo che la tabella **Temp** venga caricata come nell'esempio precedente:

```
LOAD Customer, Max(UnitSales,2) as MyMaxRank2 Resident Temp Group By Customer;
```

Tabella risultante

| Customer   | MyMaxRank2 |
|------------|------------|
| Astrida    | 10         |
| Betacab    | 4          |
| Canutility | -          |

### Max - funzione per grafici

**Max()** rileva il valore più alto dei dati aggregati. Specificando **rank n**, è possibile trovare il valore n-esimo più alto.



*Può essere utile consultare anche le funzioni **FirstSortedValue** e **RangeMax**, che presentano somiglianze con la funzione **Max**.*

### Sintassi:

```
Max ([{SetExpression}] [TOTAL [<fld {,fld}>]] expr [,rank])
```

**Tipo di dati restituiti:** numerico

## Argomenti

| Argomento            | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>expr</b>          | L'espressione o il campo contenente i dati da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>rank</b>          | Il valore predefinito di <b>rank</b> è 1, che corrisponde al valore più elevato. Specificando <b>rank</b> come 2 verrà restituito il secondo valore più elevato. Se <b>rank</b> è 3, verrà restituito il terzo valore più elevato e così via.                                                                                                                                                                                                                                                                                                           |
| <b>SetExpression</b> | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                                                                                                                                                                                                                     |
| <b>TOTAL</b>         | Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.<br>Utilizzando <b>TOTAL [&lt;fld {fld}&gt;]</b> , dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili. |

## Esempio: espressioni dei grafico

| Esempio                                                                                                                                                                  | Risultato                                                                                                                                                                                                                                                                                                                                                |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>Max(Unitsales)</code>                                                                                                                                              | Restituisce il valore più elevato nel campo <code>unitsales</code> .                                                                                                                                                                                                                                                                                     |
| <code>Max(Unitsales*UnitPrice)</code> Il valore di un ordine viene calcolato moltiplicando il numero di unità vendute ( <code>unitsales</code> ) per il prezzo unitario. | Restituisce il valore più elevato del risultato del calcolo di tutti i valori possibili di <code>(unitsales) * (unitPrice)</code> .                                                                                                                                                                                                                      |
| <code>Max(Unitsales, 2)</code>                                                                                                                                           | Restituisce il valore del secondo valore più alto.                                                                                                                                                                                                                                                                                                       |
| <code>Max(TOTAL Unitsales)</code>                                                                                                                                        | Il qualificatore <b>TOTAL</b> sta a indicare che viene trovato il valore più elevato possibile, ignorando le dimensioni del grafico. Per un grafico con <code>Customer</code> come dimensione, il qualificatore <b>TOTAL</b> assicurerà il valore massimo nell'intero set di dati, anziché il valore <code>unitsales</code> massimo per ciascun cliente. |
| <code>Max({1} TOTAL Unitsales)</code>                                                                                                                                    | L'espressione dell'analisi set <code>{1}</code> definisce il set di record da valutare come <b>ALL</b> , indipendentemente dalla selezione effettuata. Ad esempio, se viene selezionato un cliente specifico, verrà comunque restituito il valore <code>unitsales</code> massimo nell'intero set di dati.                                                |

### Esempio: principi fondamentali della funzione Max

Espressione del grafico

#### Panoramica

Un set di dati contiene i dati sulle vendite di prodotti suddivisi per mese. Vogliamo identificare il valore massimo delle vendite per ogni prodotto e il mese in cui si è verificato.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Customer`
  - `UnitSales`
  - `UnitPrice`

#### Script di caricamento

Example:

```
LOAD * inline [  
Customer|UnitSales|UnitPrice  
Astrida|4|16  
Astrida|10|15  
Astrida|9|9  
Betacab|5|10  
Betacab|2|20  
Betacab||25  
Canutility|8|15  
Canutility||19  
] (delimiter is '|');
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `Customer`

Creare le seguenti misure:

- `=Max(UnitSales)`, per calcolare il valore `unitSales` massimo.
- `=Max(UnitSales*UnitPrice)`, per calcolare il valore massimo dopo aver moltiplicato `unitSales` per `UnitPrice` riga per riga.
- `=Max(UnitSales, 2)`, per calcolare il secondo valore più alto.
- `=Max(TOTAL UnitSales)`, per calcolare il totale più alto, indipendentemente dalla dimensione del grafico.
- `=Max({1} TOTAL UnitSales)`, per calcolare il totale più alto ignorando tutte le selezioni.

Results table

| Customer   | Max<br>(UnitSales) | Max<br>(UnitSales*UnitPrice) | Max<br>(UnitSales,<br>2) | Max(TOTAL<br>UnitSales) | Max({1}<br>TOTAL<br>UnitSales) |
|------------|--------------------|------------------------------|--------------------------|-------------------------|--------------------------------|
| Totals     | 10                 | 150                          | 9                        | 10                      | 10                             |
| Astrida    | 10                 | 150                          | 9                        | 10                      | 10                             |
| Betacab    | 5                  | 50                           | 2                        | 10                      | 10                             |
| Canutility | 8                  | 120                          | -                        | 10                      | 10                             |

In questo esempio, fare quanto segue:

- Nella prima misura (`Max(UnitSales)`), il valore più alto per `unitSales` per il cliente Astrida è 10. La cella `Totals` di questa colonna restituisce 10 perché questo valore è anche il valore `unitSales` più alto dell'intero set di dati.
- Nella seconda misura (`Max(UnitSales*UnitPrice)`), 150 viene restituito per Astrida. Questo è il valore più alto per quel cliente quando si moltiplica `unitSales` per `unitPrice`. Anche la cella `Totals` di questa colonna restituisce 150, perché questo è anche il valore calcolato più alto nell'intero set di dati.
- Nella terza misura (`Max(UnitSales, 2)`), il secondo valore `unitSales` più alto per il cliente Astrida è 9. Questo valore è anche il secondo valore più alto per tutti e tre i clienti, quindi la riga `Totals` restituisce anche 9.
- La quarta misura (`Max(TOTAL UnitSales)`) restituisce il valore 10. Questo è il valore più alto trovato indipendentemente dalla dimensione del cliente, quindi tutte le righe restituiscono lo stesso valore. Tuttavia, se si seleziona il cliente Betacab, la tabella restituirà solo i valori per quel cliente, in questo caso 5, perché è il valore `unitSales` più alto tra tutti i prodotti per Betacab.
- La quinta misura (`Max({1} TOTAL UnitSales)`) restituisce il valore 10 indipendentemente da quale `Customer` è selezionato, perché viene utilizzato il set `{1}` modificato. Ad esempio, se si seleziona Betacab, questa colonna continuerà a restituire il valore 10.

**Esempio: scenario per la funzione Max che calcola le vendite più alte in base al mese**

Espressione del grafico

### Panoramica

Un set di dati contiene i dati sulle vendite di prodotti suddivisi per mese. Un responsabile vendite vuole identificare il valore massimo delle vendite per ogni prodotto e il mese in cui l'operazione è stata completata.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Product`
  - `Sales`
  - `Month`

### Script di caricamento

Example:

```
Load * inline [
Product,Sales,Month
A, 500, 2024-01-01
A, 700, 2024-02-01
A, 600, 2024-03-01
B, 300, 2024-01-01
B, 800, 2024-02-01
B, 450, 2024-03-01
C, 550, 2024-01-01
C, 620, 2024-02-01
C, 710, 2024-03-01
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `Product`
- `Month`

Creare le seguenti misure:

- `=If(Sales = Max(TOTAL <Product> Sales), Sales)`, per calcolare le vendite massime per ogni prodotto. Nel pannello delle proprietà della misura, selezionare **Componenti aggiuntivi > Gestione dati**. Deselezionare la casella di controllo **Includi valori zero** per rimuovere qualsiasi valore nullo.

Results table

| Product | Month      | If(Sales = Max(TOTAL <Product> Sales), Sales) |
|---------|------------|-----------------------------------------------|
| A       | 2024-02-01 | 700                                           |
| B       | 2024-02-01 | 800                                           |
| C       | 2024-03-01 | 710                                           |

I risultati della misura restituiscono il valore massimo delle vendite per ogni prodotto e il mese in cui questo valore è stato raggiunto. Per esempio, `product B` ha ottenuto le vendite più alte (800) in 2024-02-01.

### Vedere anche:

p [FirstSortedValue - funzione per grafici \(page 361\)](#)

p [RangeMax \(page 1598\)](#)

## Min

**Min()** restituisce il valore numerico più basso dei dati aggregati nell'espressione, come definito da una clausola **group by**. Specificando un **rank** n, è possibile trovare il valore n-esimo più basso.

### Sintassi:

```
Min ( expr [, rank] )
```

Tipo di dati restituiti: numerico

### Argomenti:

#### Argomenti

| Argomento          | Descrizione                                                                                                                                                                                                                               |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| expr<br>Expression | L'espressione o il campo contenente i dati da misurare.                                                                                                                                                                                   |
| rank<br>Expression | Il valore predefinito di <b>rank</b> è 1, che corrisponde al valore più basso. Specificando 2 come <b>rank</b> , verrà restituito il secondo valore più basso. Se <b>rank</b> è 3, verrà restituito il terzo valore più basso e così via. |

### Esempi e risultati:

Aggiungere lo script di esempio all'app ed eseguirlo. Aggiungere quindi a un foglio nell'app almeno i campi elencati nella colonna dei risultati per visualizzare il risultato.

Per ottenere lo stesso aspetto della colonna dei risultati mostrata di seguito, nel pannello delle proprietà, in Ordinamento passare da Automatico a Personalizza, quindi deselezionare l'ordinamento numerico e alfabetico.

### Esempio:

Temp:

```
LOAD * inline [
Customer|Product|OrderNumber|UnitSales|CustomerID
Astrida|AA|1|10|1
Astrida|AA|7|18|1
Astrida|BB|4|9|1
Astrida|CC|6|2|1
Betacab|AA|5|4|2
Betacab|BB|2|5|2
Betacab|DD
Canutility|DD|3|8
Canutility|CC
] (delimiter is '|');
Min:
LOAD Customer, Min(UnitSales) as MyMin Resident Temp Group By Customer;
```

Tabella risultante

| Customer   | MyMin |
|------------|-------|
| Astrida    | 2     |
| Betacab    | 4     |
| Canutility | 8     |

**Esempio:**

Presupponendo che la tabella **Temp** venga caricata come nell'esempio precedente:

```
LOAD Customer, Min(UnitSales,2) as MyMinRank2 Resident Temp Group By Customer;
```

Tabella risultante

| Customer   | MyMinRank2 |
|------------|------------|
| Astrida    | 9          |
| Betacab    | 5          |
| Canutility | -          |

**Min - funzione per grafici**

**Min()** trova il valore più basso dei dati aggregati. Specificando un **rank** n, è possibile trovare il valore n-esimo più basso.



*Può essere utile consultare anche le funzioni **FirstSortedValue** e **RangeMin**, che presentano somiglianze con la funzione **Min**.*

**Sintassi:**

```
Min ([SetExpression] [TOTAL [<fld {,fld}>]]) expr [,rank])
```

**Tipo di dati restituiti:** numerico

Argomenti

| Argomento            | Descrizione                                                                                                                                                                                                                               |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>expr</b>          | L'espressione o il campo contenente i dati da misurare.                                                                                                                                                                                   |
| <b>rank</b>          | Il valore predefinito di <b>rank</b> è 1, che corrisponde al valore più basso. Specificando 2 come <b>rank</b> , verrà restituito il secondo valore più basso. Se <b>rank</b> è 3, verrà restituito il terzo valore più basso e così via. |
| <b>SetExpression</b> | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                       |

| Argomento    | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>TOTAL</b> | Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico. Utilizzando <b>TOTAL [&lt;fld {fld}&gt;]</b> , dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili. |

## Esempio: espressioni del grafico

| Esempio                                                                                                                                      | Risultato                                                                                                                                                                                                                                                                                                 |
|----------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Min(Unitsales)                                                                                                                               | Restituisce il valore più basso non null nel campo unitsales.                                                                                                                                                                                                                                             |
| Min(Unitsales*UnitPrice) Il valore di un ordine viene calcolato moltiplicando il numero di unità vendute (Unitsales) per il prezzo unitario. | Restituisce il valore più basso non null del risultato del calcolo di tutti i valori possibili di (unitsales) * (unitPrice).                                                                                                                                                                              |
| Min(Unitsales, 2)                                                                                                                            | Restituisce il risultato per il secondo valore più basso in unitsales (dopo i valori NULL).                                                                                                                                                                                                               |
| Min(TOTAL Unitsales)                                                                                                                         | Il qualificatore TOTAL indica che viene identificato il valore più basso possibile, ignorando le dimensioni del grafico. Per un grafico con Customer come dimensione, il qualificatore TOTAL assicurerà il valore minimo nell'intero set di dati, anziché il valore unitsales minimo per ciascun cliente. |
| Min({1} TOTAL Unitsales)                                                                                                                     | L'espressione dell'analisi set {1} definisce il set di record da valutare come ALL, indipendentemente dalla selezione effettuata. Ad esempio, se viene selezionato un cliente specifico, verrà comunque restituito il valore unitsales minimo nell'intero set di dati.                                    |

## Esempio: principi fondamentali della funzione Min

## Espressione del grafico

## Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata Example.
- I seguenti campi nella tabella dati:

- Customer
- UnitSales
- UnitPrice

### Script di caricamento

Example:

```
LOAD * inline [
Customer|UnitSales|UnitPrice
Astrida|4|16
Astrida|10|15
Astrida|9|9
Betacab|5|10
Betacab|2|20
Betacab||25
Canutility|8|15
Canutility||19
] (delimiter is '|');
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- Customer

Creare le seguenti misure:

- =Min(UnitSales), per calcolare il valore unitSales minimo.
- =Min(UnitSales\*UnitPrice), per calcolare il valore minimo dopo aver moltiplicato unitSales per UnitPrice, riga per riga.
- =Min(UnitSales, 2), per calcolare il secondo valore più basso.
- =Min(TOTAL UnitSales), per calcolare il totale più basso, indipendentemente dalla dimensione del grafico.
- =Min({1} TOTAL UnitSales), per calcolare il totale più basso ignorando tutte le selezioni.

Results table

| Customer   | Min<br>(UnitSales) | Min<br>(UnitSales*UnitPrice) | Min<br>(UnitSales,<br>2) | Min(TOTAL<br>UnitSales) | Min({1}<br>TOTAL<br>UnitSales) |
|------------|--------------------|------------------------------|--------------------------|-------------------------|--------------------------------|
| Totals     | 2                  | 40                           | 4                        | 2                       | 2                              |
| Astrida    | 4                  | 64                           | 9                        | 2                       | 2                              |
| Betacab    | 2                  | 40                           | 5                        | 2                       | 2                              |
| Canutility | 8                  | 120                          | -                        | 2                       | 2                              |

In questo esempio, fare quanto segue:

- Nella prima misura, `Min(Unitsales)`, il valore più basso per `unitsales` per il cliente Astrida è 4. La cella `TOTALs` in questa colonna restituisce 2 perché questo valore è anche il valore `unitsales` più basso dell'intero set di dati.
- Nella seconda misura, `Min(Unitsales*UnitPrice)`, il valore 64 viene restituito per Astrida. Questo è il valore più basso per quel cliente quando si moltiplica `unitsales` per `unitPrice`. La cella `TOTALs` di questa colonna restituisce 40, perché questo è anche il valore calcolato più basso nell'intero set di dati.
- Nella terza misura, `Min(Unitsales, 2)`, il secondo valore `unitsales` più basso per il cliente Astrida è 9. Il valore 4 viene restituito nella riga `TOTALs`, perché è il secondo valore `unitsales` più basso per tutti i clienti.
- La quarta misura, `Min(TOTAL Unitsales)`, restituisce il valore 2. Questo è il valore non null più basso possibile indipendentemente dalla dimensione del cliente, quindi tutte le righe restituiscono lo stesso valore. Tuttavia, se si seleziona il cliente Betacab, la tabella restituirà solo i valori per quel cliente, in questo caso 2, perché è il valore `unitsales` più basso tra tutti i prodotti per Betacab.
- La quinta misura, `Min({1} TOTAL Unitsales)`, restituisce 2 indipendentemente dal valore selezionato nel campo `customer`, in quanto la misura calcola il totale più basso. Ad esempio, se si seleziona Betacab, questa colonna continuerà a restituire il valore 2.

### Esempio: scenario per la funzione Min che calcola le vendite più basse in base al mese

#### Espressione del grafico

#### Panoramica

Un set di dati contiene i dati sulle vendite di prodotti suddivisi per mese. Un responsabile vendite vuole identificare il valore minimo delle vendite per ogni prodotto e il mese in cui l'operazione è stata completata.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Product`
  - `Sales`
  - `Month`

#### Script di caricamento

Example:

```
Load * inline [
Product,Sales,Month
A, 500, 2024-01-01
A, 700, 2024-02-01
A, 600, 2024-03-01
B, 300, 2024-01-01
B, 800, 2024-02-01
B, 450, 2024-03-01
```

```
C, 550, 2024-01-01
C, 620, 2024-02-01
C, 710, 2024-03-01
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- Product
- Month

Creare la seguente misura:

- `=If(Sales = Min(TOTAL <Product> Sales), Sales)`, per calcolare le vendite minime per ogni prodotto. Nel pannello delle proprietà, selezionare **Componenti aggiuntivi > Gestione dati**. Deselezionare la casella di controllo **Includi valori zero**.

Results table

| Product | Month      | If(Sales = Min(TOTAL <Product> Sales), Sales) |
|---------|------------|-----------------------------------------------|
| A       | 2024-01-01 | 500                                           |
| B       | 2024-01-01 | 300                                           |
| C       | 2024-01-01 | 550                                           |

I risultati della misura restituiscono il valore minimo delle vendite per ogni prodotto e il mese in cui questo valore è stato raggiunto. Per esempio, `product B` ha ottenuto le vendite più basse (300) in 2024-01-01.

### Vedere anche:

p [FirstSortedValue - funzione per grafici \(page 361\)](#)

p [RangeMin \(page 1602\)](#)

### Mode

**Mode()** restituisce il valore più comune, il valore mode, dei dati aggregati nell'espressione, come definito da una clausola **group by**. La funzione **Mode()** può restituire valori numerici e valori di testo.

### Sintassi:

```
Mode ( expr )
```

Tipo di dati restituiti: duale

Argomenti

| Argomento       | Descrizione                                             |
|-----------------|---------------------------------------------------------|
| expr Expression | L'espressione o il campo contenente i dati da misurare. |

### Limiti:

Se più di un valore comune ricorre lo stesso numero di volte, verrà restituito NULL.

### Esempi e risultati:

Aggiungere lo script di esempio all'app ed eseguirlo. Aggiungere quindi a un foglio nell'app almeno i campi elencati nella colonna dei risultati per visualizzare il risultato.

Per ottenere lo stesso aspetto della colonna dei risultati mostrata di seguito, nel pannello delle proprietà, in Ordinamento passare da Automatico a Personalizza, quindi deselezionare l'ordinamento numerico e alfabetico.

Esempi di script

| Esempio                                                                                                                                                                                                                                                                                                                                                    | Risultato                                                                                      |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| <pre>Temp: LOAD * inline [ Customer Product OrderNumber UnitSales CustomerID Astrida AA 1 10 1 Astrida AA 7 18 1 Astrida BB 4 9 1 Astrida CC 6 2 1 Betacab AA 5 4 2 Betacab BB 2 5 2 Betacab DD Canutility DD 3 8 Canutility CC ] (delimiter is ' ');  Mode: LOAD Customer, Mode(Product) as MyMostOftenSoldProduct Resident Temp Group By Customer;</pre> | <p>MyMostOftenSoldProduct</p> <p>AA</p> <p>perché AA è l'unico prodotto venduto più volte.</p> |

### Mode - funzione per grafici

**Mode()** trova il valore più comune, il valore della modalità, nei dati aggregati. La funzione **Mode()** può elaborare valori di testo e valori numerici.

### Sintassi:

```
Mode ({[SetExpression] [TOTAL [<fld {,fld}>]]} expr)
```

Tipo di dati restituiti: duale

Argomenti

| Argomento | Descrizione                                             |
|-----------|---------------------------------------------------------|
| expr      | L'espressione o il campo contenente i dati da misurare. |

| Argomento            | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SetExpression</b> | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                                                                                                                                                                                                                     |
| <b>TOTAL</b>         | Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.<br>Utilizzando <b>TOTAL [&lt;fld {fld}&gt;]</b> , dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili. |

## Esempio: espressioni del grafico

| Esempio                      | Risultato                                                                                                                                                                                                                                                                                                     |
|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Mode(Unitsales)              | Restituisce il valore che ricorre con più frequenza in <code>unitsales</code> .<br>Restituisce NULL (-). Nessun singolo valore ricorre più frequentemente di un altro.                                                                                                                                        |
| Mode(Customer)               | Restituisce il valore che ricorre con più frequenza in <code>customer</code> .<br>Restituisce NULL (-). Nessun singolo valore ricorre più frequentemente di un altro.                                                                                                                                         |
| Mode<br>(TOTAL Unitsales)    | Il qualificatore <b>TOTAL</b> restituisce il valore che ricorre più di frequente in <code>unitsales</code> , indipendentemente dalle dimensioni del grafico.                                                                                                                                                  |
| Mode({1} TOTAL<br>Unitsales) | L'espressione dell'analisi set {1} definisce il set di record da valutare come ALL, indipendentemente dalla selezione effettuata. Ad esempio, se viene selezionato un cliente specifico, verrà comunque restituito il valore che ricorre più di frequente per <code>unitsales</code> nell'intero set di dati. |

## Esempio: principi fondamentali della funzione Mode

## Espressione del grafico

## Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Customer`

- UnitSales
- UnitPrice

### Script di caricamento

Example:

```
LOAD * inline [
Customer|UnitSales|UnitPrice
Astrida|4|16
Astrida|10|15
Astrida|9|9
Astrida|10|10
Betacab|5|10
Betacab|4|20
Betacab||25
Canutility|8|15
Canutility|4|19
] (delimiter is '|');
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- Customer

Creare le seguenti misure:

- =Mode(UnitSales), per calcolare il valore unitSales che ricorre più di frequente.
- =Mode(Customer), per calcolare il valore customer che ricorre più di frequente.
- =Mode(TOTAL UnitSales), per calcolare il valore unitSales che ricorre più di frequente nell'intero set di dati, indipendentemente dalle dimensioni dei grafici. Se viene effettuata una selezione, il valore si regolerà per riflettere tale selezione.
- =Mode({1} TOTAL UnitSales), per calcolare il valore unitSales che ricorre più di frequente, ignorando le selezioni. Se viene effettuata una selezione, il valore rimane costante.

Results table

| Customer   | Mode<br>(UnitSales) | Mode<br>(Customer) | Mode(TOTAL<br>UnitSales) | Mode({1} TOTAL<br>UnitSales) |
|------------|---------------------|--------------------|--------------------------|------------------------------|
| Totals     | 4                   | Astrida            | 4                        | 4                            |
| Astrida    | 10                  | Astrida            | 4                        | 4                            |
| Betacab    | -                   | Betacab            | 4                        | 4                            |
| Canutility | -                   | Canutility         | 4                        | 4                            |

In questo esempio, fare quanto segue:

- Nella prima misura (Mode(UnitSales)), il valore che ricorre più di frequente per il cliente Astrida è 10. La cella Totals restituisce 4, che è il valore che ricorre più di frequente per unitSales nell'intero set di dati. Non ci sono istanze dei valori che ricorrono più di frequente per Betacab e Canutility,

quindi i risultati restituiscono NULL per questi clienti.

- Nella seconda misura (`Mode(Customer)`), i risultati restituiscono il nome di ogni cliente. Tuttavia, la cella `TOTALs` restituisce `Astrida`, che è il valore che ricorre più di frequente per il cliente nell'intero set di dati.
- La terza misura (`Mode(TOTAL Unitsales)`) restituisce il valore 4. Se non viene selezionato il campo `Customer`, tutte le celle di questa colonna visualizzeranno il valore 4. Se si seleziona un cliente, come `Betacab`, questa colonna restituirà NULL, poiché non esiste un valore comune per `Betacab`.
- La quarta misura (`Mode({1} TOTAL Unitsales)`) restituisce 4 indipendentemente da qualsiasi selezione nel campo `Customer`, perché l'espressione dell'analisi del set `{1}` definisce il set di record da valutare come ALL. Ad esempio, se si seleziona `Betacab`, questa colonna continuerà a restituire il valore 4.

### Esempio: scenario per la funzione Mode per determinare le anomalie nei dati di vendita

Espressione del grafico

#### Panoramica

Un set di dati contiene i dati sulle vendite suddivisi per mese. L'azienda vuole visualizzare il valore più frequente per le vendite di ogni prodotto e nell'intero set di dati. Vuole anche confrontare le vendite per prodotto con l'intero set di dati per identificare eventuali anomalie nei dati.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Date`
  - `Product`
  - `Category`
  - `Sales`

#### Script di caricamento

Example:

```
Load * inline [
Date, Product, Category, Sales
2024-01-01, Apple, Fruit, 100
2024-01-15, Banana, Fruit, 80
2024-01-30, Carrot, Vegetable, 50
2024-02-05, Apple, Fruit, 100
2024-02-20, Banana, Fruit, 90
2024-02-28, Carrot, Vegetable, 60
2024-03-10, Apple, Fruit, 110
2024-03-22, Banana, Fruit, 80
2024-03-31, Carrot, Vegetable, 50
2024-04-10, Apple, Fruit, 125
2024-04-22, Banana, Fruit, 60
```

2024-04-31, Carrot, Vegetable, 80  
];

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- Category
- Product

Creare le seguenti misure:

- =Mode(Sales), per calcolare il valore che ricorre più di frequente per ogni prodotto.
- =Mode(TOTAL Sales), per calcolare il valore che ricorre più di frequente per l'intero set di dati.
- If(Mode(Sales) = Mode(TOTAL Sales), 'Typical', 'Atypical'), per calcolare se il valore che ricorre più di frequente per ogni prodotto è tipico rispetto ai risultati dell'intero set di dati.

Results table

| Category  | Product | Mode (Sales) | Mode(TOTAL Sales) | If(Mode(Sales) = Mode(TOTAL Sales), 'Typical', 'Atypical') |
|-----------|---------|--------------|-------------------|------------------------------------------------------------|
| Fruit     | Apple   | 100          | 80                | Atypical                                                   |
| Fruit     | Banana  | 80           | 80                | Typical                                                    |
| Vegetable | Carrot  | 50           | 80                | Atypical                                                   |

L'output della funzione Mode restituisce il valore delle vendite che ricorre più di frequente per prodotto, Mode(Sales), e per tutti i prodotti, Mode(TOTAL Sales). Questi due valori vengono poi confrontati per determinare quali risultati di vendita sono tipici o atipici.

### Vedere anche:

- p [Avg - funzione per grafici \(page 457\)](#)
- p [Median - funzione per grafici \(page 496\)](#)

### Only

Only() restituisce un valore se esiste esclusivamente un unico risultato possibile dai dati aggregati. Se i record contengono solo un valore verrà restituito tale valore, altrimenti verrà restituito NULL. Utilizzare la clausola **group by** per valutare più record. La funzione Only() può restituire valori numerici e valori di testo.

### Sintassi:

```
Only ( expr )
```

**Tipo di dati restituiti:** duale

Argomenti

| Argomento       | Descrizione                                             |
|-----------------|---------------------------------------------------------|
| expr Expression | L'espressione o il campo contenente i dati da misurare. |

### Esempi e risultati:

Aggiungere lo script di esempio all'app ed eseguirlo. Aggiungere quindi a un foglio nell'app almeno i campi elencati nella colonna dei risultati per visualizzare il risultato.

Per ottenere lo stesso aspetto della colonna dei risultati mostrata di seguito, nel pannello delle proprietà, in Ordinamento passare da Automatico a Personalizza, quindi deselezionare l'ordinamento numerico e alfabetico.

Temp:

```
LOAD * inline [
Customer|Product|OrderNumber|UnitSales|CustomerID
Astrida|AA|1|10|1
Astrida|AA|7|18|1
Astrida|BB|4|9|1
Astrida|CC|6|2|1
Betacab|AA|5|4|2
Betacab|BB|2|5|2
Betacab|DD
Canutility|DD|3|8
Canutility|CC
] (delimiter is '|');
Only:
```

```
LOAD Customer, Only(CustomerID) as MyUniqIDCheck Resident Temp Group By Customer;
```

Tabella risultante

| Cliente | MyUniqIDCheck                                                                     |
|---------|-----------------------------------------------------------------------------------|
| Astrida | 1                                                                                 |
|         | perché solo il cliente Astrida presenta record completi che includono CustomerID. |

### Only - funzione per grafici

**Only()** restituisce un valore se esiste esclusivamente un unico risultato possibile dai dati aggregati. Ad esempio, la ricerca dell'unico prodotto con prezzo unitario = 9 restituirà NULL se più di un prodotto ha un prezzo unitario di 9.

#### Sintassi:

```
Only([{{SetExpression}}] [TOTAL [<fld {,fld}>]] expr)
```

**Tipo di dati restituiti:** duale

### Argomenti

| Argomento            | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>expr</b>          | L'espressione o il campo contenente i dati da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>SetExpression</b> | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                                                                                                                                                                                                                     |
| <b>TOTAL</b>         | Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.<br>Utilizzando <b>TOTAL [&lt;fld {fld}&gt;]</b> , dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili. |



Utilizzare *Only()* quando si desidera un risultato NULL se nei dati campione sono presenti più valori possibili.

### Esempio: espressione del grafico

| Esempio                     | Risultato                                                                |
|-----------------------------|--------------------------------------------------------------------------|
| <code>only(Category)</code> | Visualizza il valore category a condizione che sia univoco nel contesto. |

### Esempio: principi fondamentali della funzione Only

#### Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Customer`
  - `Product`
  - `UnitsSales`
  - `UnitPrice`

### Script di caricamento

Example:

```
LOAD * inline [
Customer|Product|UnitsSales|UnitPrice
Astrida|AA|4|16
Astrida|AA|10|15
Astrida|BB|9|9
Betacab|BB|5|10
Betacab|CC|2|20
Betacab|DD| |25
Canutility|AA|8|15
Canutility|CC| |19
] (delimiter is '|');
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella con le seguenti misure:

- =Only({<UnitPrice={9}>} Product), per calcolare l'unico valore Product con un unitPrice uguale a 9.
- =Only({<Product={DD}>} Customer), per calcolare l'unico Customer che vende un Product chiamato DD.
- =Only({<UnitPrice={20}>} unitsales), per calcolare il numero di unitsales in cui unitPrice è 20.
- =Only({<UnitPrice={15}>} unitsales), per calcolare i valori di unitsales, dove unitPrice = 15.

Results table

| Only({<UnitPrice={9}>} Product) | Only({<Product={DD}>} Customer) | Only({<UnitPrice={20}>} UnitSales) | Only({<UnitPrice={15}>} UnitSales) |
|---------------------------------|---------------------------------|------------------------------------|------------------------------------|
| BB                              | Betacab                         | 2                                  | -                                  |

In questo esempio, fare quanto segue:

- La prima colonna restituisce il valore BB perché questo è l'unico Product che ha un unitPrice uguale a 9.
- La seconda colonna restituisce Betacab, perché è l'unico Customer a vendere un Product denominato DD.
- La terza colonna restituisce il valore 2 perché questo è l'unico valore di unitsales in cui unitPrice = 20.
- La quarta colonna restituisce NULL perché non esiste un solo risultato in cui unitPrice = 15. In questo set di dati, ci sono due valori di unitsales in cui unitPrice = 15.

### Esempio: scenario per la funzione Only

Espressione del grafico

#### Panoramica

Un set di dati contiene i dati sulle vendite di prodotti suddivisi per il periodo. L'azienda vuole determinare se un prodotto è venduto in un singolo negozio o in più negozi.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Date`
  - `RecordID`
  - `StoreProduct`
  - `Sales`

#### Script di caricamento

Example:

```
Load * inline [
Date, RecordID, StoreProduct, Sales, Store
2024-01-01, 1, Apple, 100, A
2024-01-15, 2, Banana, 80, B
2024-01-30, 3, Carrot, 50, C
2024-02-05, 4, Apple, 120, A
2024-02-20, 5, Banana, 90, B
2024-02-28, 6, Carrot, 60, D
2024-03-10, 7, Apple, 110, A
2024-03-22, 8, Banana, 85, B
2024-03-31, 9, Carrot, 55, C
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `StoreProduct`

Creare le seguenti misure:

- `=only(Store)`, per calcolare l'unico negozio in cui viene venduto il prodotto.
- `=If(IsNull(only(Store)), 'Multiple Stores', 'Single Store')`, per calcolare se il prodotto è venduto in un singolo negozio o in più negozi.
- `=Sum(Sales)`, per calcolare le vendite totali per il prodotto.

Results table

| StoreProduct | Only (Store) | If(IsNull(Only(Store)), 'Multiple Stores', 'Single Store') | Sum (Sales) |
|--------------|--------------|------------------------------------------------------------|-------------|
| Apple        | A            | Single Store                                               | 330         |
| Banana       | B            | Single Store                                               | 255         |
| Carrot       | -            | Multiple Stores                                            | 165         |

I risultati indicano che i prodotti Apples e Bananas sono venduti in un solo negozio, mentre il prodotto Carrots è venduto in più negozi.

### Esempio: scenario avanzato per creare una tabella di record di un solo mese

#### Espressione del grafico

#### Panoramica

Questo esempio utilizza la funzione `only` per mostrare una tabella di record riga per riga per un singolo mese. A tal fine, si utilizza l'analisi dei set per individuare il set di record desiderato. L'analisi dei set richiede almeno una misura, anche se, dal momento che non abbiamo effettivamente bisogno di aggregare il set di record, è possibile utilizzare efficacemente la funzione di aggregazione `only` per ospitare l'ambito dell'analisi dei set.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Un'istruzione `LOAD` precedente che deriva il campo `MonthYear` dal campo `Date` troncando il giorno del mese, utilizzando la funzione `MonthStart`, e restituendo una rappresentazione della data del 1° di ogni mese. Questo risultato viene fornito alla funzione `Date` per impostare il formato di visualizzazione su `MMM YY`, ad esempio, `Jan 24`.
- I seguenti campi nella tabella dati:
  - `Date`
  - `RecordID`
  - `StoreProduct`
  - `Sales`
  - `Store`

#### Script di caricamento

Example:

```
Load
*,
date(MonthStart (Date), 'MMM YY') AS MonthYear
inline [
Date, RecordID, StoreProduct, Sales, Store
```

```
2024-01-01, 1, Apple, 100, A
2024-01-15, 2, Banana, 80, B
2024-01-30, 3, Carrot, 50, C
2024-02-05, 4, Apple, 120, A
2024-02-20, 5, Banana, 90, B
2024-02-28, 6, Carrot, 60, D
2024-03-10, 7, Apple, 110, A
2024-03-22, 8, Banana, 85, B
2024-03-31, 9, Carrot, 55, C
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- MonthYear
- RecordID
- Date
- StoreProduct
- Store

Creare le seguenti misure:

- `=Only({$<MonthYear={'$(=MaxString(MonthYear))'}>} Sales)`, per valutare il valore sales e restituire solo quelli in cui MonthYear è l'ultimo (il più recente) MonthYear disponibile, in base alle selezioni correnti nella tabella. L'espansione con il simbolo del dollaro (\$) esegue un calcolo in linea utilizzando la funzione `MaxString`, `$(=MaxString(MonthYear))`, che restituisce la rappresentazione testuale del valore MonthYear più recente.



*Il tentativo di filtrare MonthYear in base alla rappresentazione numerica, ad esempio utilizzando la funzione `Max`, `Max(MonthYear)`, non produrrebbe alcun risultato. L'analisi dei set applica selezioni dei filtri basate solo sulla rappresentazione testuale di qualsiasi tipo di dati doppi; un tipo di dati con una rappresentazione numerica e testuale, come ad esempio `Date`.*

Results table

| MonthYear | RecordID | Date       | StoreProduct | Only({\$<MonthYear={'\$(=MaxString(MonthYear))'}>} Sales) |
|-----------|----------|------------|--------------|-----------------------------------------------------------|
| Mar 24    | 7        | 2024-03-10 | Apple        | 110                                                       |
| Mar 24    | 8        | 2024-03-22 | Banana       | 85                                                        |
| Mar 24    | 9        | 2024-03-31 | Carrot       | 55                                                        |

L'output della funzione `only` genera una tabella di record che presenta per impostazione predefinita il valore di vendita dell'ultimo, o più recente, `MonthYear` disponibile nei dati. Se si seleziona un mese precedente dalla colonna `MonthYear`, ad esempio `Feb 24`, apparirà il valore delle vendite più recente per quel mese.

### Sum

**Sum()** calcola il totale dei valori aggregati nell'espressione, come definito da una clausola **group by**.

**Sintassi:**

```
sum ( [ distinct] expr)
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

#### Argomenti

| Argomento          | Descrizione                                                                                          |
|--------------------|------------------------------------------------------------------------------------------------------|
| distinct           | Se la parola <b>distinct</b> è riportata prima dell'espressione, tutti i duplicati vengono ignorati. |
| expr<br>Expression | L'espressione o il campo contenente i dati da misurare.                                              |

**Esempi e risultati:**

Aggiungere lo script di esempio all'app ed eseguirlo. Aggiungere quindi a un foglio nell'app almeno i campi elencati nella colonna dei risultati per visualizzare il risultato.

Per ottenere lo stesso aspetto della colonna dei risultati mostrata di seguito, nel pannello delle proprietà, in Ordinamento passare da Automatico a Personalizza, quindi deselezionare l'ordinamento numerico e alfabetico.

Temp:

```
LOAD * inline [
Customer|Product|OrderNumber|UnitSales|CustomerID
Astrida|AA|1|10|1
Astrida|AA|7|18|1
Astrida|BB|4|9|1
Astrida|CC|6|2|1
Betacab|AA|5|4|2
Betacab|BB|2|5|2
Betacab|DD
Canutility|DD|3|8
Canutility|CC
] (delimiter is '|');
```

Sum:

```
LOAD Customer, Sum(UnitSales) as MySum Resident Temp Group By Customer;
```

Tabella risultante

| Customer   | MySum |
|------------|-------|
| Astrida    | 39    |
| Betacab    | 9     |
| Canutility | 8     |

## Sum - funzione per grafici

**Sum()** calcola il totale dei valori dati dall'espressione o dal campo nei dati aggregati.

**Sintassi:**

```
Sum ( [{SetExpression}] [DISTINCT] [TOTAL [<fld {,fld}>]] expr )
```

**Tipo di dati restituiti:** numerico

Argomenti

| Argomento            | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>expr</b>          | L'espressione o il campo contenente i dati da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>SetExpression</b> | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                                                                                                                                                                                                                                |
| <b>DISTINCT</b>      | <p>Se la parola <b>DISTINCT</b> è riportata prima degli argomenti della funzione, i duplicati risultanti dalla valutazione degli argomenti della funzione vengono ignorati.</p> <div>  <p><i>Sebbene il qualificatore <b>DISTINCT</b> sia supportato, si consiglia di utilizzarlo con estrema cautela perché può fuorviare l'utente inducendolo a credere che è visualizzato un valore totale quando invece alcuni dati sono stati omessi.</i></p> </div>                       |
| <b>TOTAL</b>         | <p>Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.</p> <p>Utilizzando <b>TOTAL [&lt;fld {,fld}&gt;]</b>, dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili.</p> |

Esempio: espressioni del grafico

| Esempio        | Risultato                                       |
|----------------|-------------------------------------------------|
| Sum(Unitsales) | Restituisce il totale dei valori per unitsales. |

| Esempio                                         | Risultato                                                                                                                                                                                                        |
|-------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>Sum(UnitSales*UnitPrice)</code>           | Restituisce il totale di <code>UnitPrice</code> moltiplicato per i risultati aggregati di <code>UnitSales</code> .                                                                                               |
| <code>Sum (TOTAL UnitSales*UnitPrice)</code>    | Restituisce la somma totale per tutte le righe della tabella, oltre al totale, perché il qualificatore <code>TOTAL</code> significa che la somma viene calcolata indipendentemente dalle dimensioni del grafico. |
| <code>Sum({1} TOTAL UnitSales*UnitPrice)</code> | L'espressione dell'analisi set <code>{1}</code> definisce il set di record da valutare come <code>ALL</code> , indipendentemente dalla selezione effettuata.                                                     |

### Esempio: principi fondamentali della funzione Sum

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Customer`
  - `Product`
  - `UnitSales`
  - `UnitPrice`

#### Script di caricamento

Example:

```
LOAD * inline [
Customer|Product|UnitSales|UnitPrice
Astrida|AA|4|16
Astrida|AA|10|15
Astrida|BB|9|9
Betacab|BB|5|10
Betacab|CC|2|20
Betacab|DD||25
Canutility|AA|8|15
Canutility|CC||19
] (delimiter is '|');
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- Customer
- Product

Creare le seguenti misure:

- `=Sum(Unitsales)`, per calcolare la somma di `unitsales`.
- `=Sum(Unitsales*UnitPrice)`, per calcolare il totale aggregato totale di `unitsales` moltiplicato per `UnitPrice`.
- `=Sum(TOTAL Unitsales*UnitPrice)`, per calcolare il totale per tutte le righe della tabella, oltre al totale, perché il qualificatore `TOTAL` significa che la somma è la stessa per tutte le celle, indipendentemente dalle dimensioni del grafico.
- `=Sum({1} TOTAL Unitsales*UnitPrice)`, per calcolare il totale, indipendentemente dalla selezione effettuata, perché l'espressione dell'analisi set `{1}` definisce il set di record da valutare come `ALL`, indipendentemente dalla selezione effettuata.

Results table

| Customer      | Product | Sum<br>(UnitSales) | Sum<br>(UnitSales*UnitPrice) | Sum(TOTAL<br>UnitSales*UnitPrice) | Sum({1} TOTAL<br>UnitSales*UnitPrice) |
|---------------|---------|--------------------|------------------------------|-----------------------------------|---------------------------------------|
| <b>Totals</b> | -       | <b>38</b>          | <b>505</b>                   | <b>505</b>                        | <b>505</b>                            |
| Astrida       | AA      | 14                 | 214                          | 505                               | 505                                   |
| Astrida       | BB      | 9                  | 81                           | 505                               | 505                                   |
| Betacab       | BB      | 5                  | 50                           | 505                               | 505                                   |
| Betacab       | CC      | 2                  | 40                           | 505                               | 505                                   |
| Betacab       | DD      | 0                  | 0                            | 505                               | 505                                   |
| Canutility    | AA      | 8                  | 120                          | 505                               | 505                                   |
| Canutility    | CC      | 0                  | 0                            | 505                               | 505                                   |

In questo esempio, fare quanto segue:

- Per la prima misura, `Sum(Unitsales)`, la cella `Totals` restituisce 38, che è il totale dei valori in `unitsales` per tutti i clienti e i prodotti. I risultati per ogni combinazione di clienti e prodotti vengono aggregati per ottenere la somma totale di `unitsales` per ogni combinazione. Per esempio, la somma di tutti i valori `unitsales` per il prodotto AA del cliente Astrida è 14.
- Per la seconda misura, `Sum(Unitsales*UnitPrice)`, la cella `Totals` restituisce 505. Si tratta del totale aggregato di `unitsales` moltiplicato per `unitPrice` per ogni combinazione di cliente e prodotto.
- Per la terza misura, `Sum(TOTAL Unitsales*UnitPrice)`, il valore restituito in tutte le righe, compresa `Totals`, è 505, perché il qualificatore `TOTAL` non tiene conto delle dimensioni dei grafici. Tuttavia, se viene effettuata una selezione su una dimensione come `Customer`, i valori cambieranno. Ad esempio, se si seleziona `Betacab`, la tabella risultante avrà solo tre righe, una per ogni record `Betacab`, ma la colonna restituirà lo stesso valore totale per tutte le righe (90).

- La quarta misura `Sum({1} TOTAL UnitsSales*UnitPrice)`, restituisce il valore 505 indipendentemente da qualsiasi selezione, perché l'espressione dell'analisi dei set `{1}` definisce il set di record da valutare come ALL. Se si seleziona un cliente, ad esempio `canutility`, il valore restituito è ancora 505.

### Esempio: scenario per la funzione Sum che aggrega i dati di vendita

#### Espressione del grafico

#### Panoramica

Un set di dati contiene i dati sulle vendite per i negozi suddivisi per regione. L'azienda desidera aggregare le vendite.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Month`
  - `SalesProduct`
  - `Region`
  - `Sales`

#### Script di caricamento

Example:

```
Load * inline [
Month,SalesProduct, Region, Sales
2024-01-01, T-shirt, East, 100
2024-01-01, T-shirt, West, 120
2024-01-01, Jeans, East, 200
2024-01-01, Jeans, West, 180
2024-02-01, T-shirt, East, 150
2024-02-01, T-shirt, West, 130
2024-02-01, Jeans, East, 250
2024-02-01, Jeans, West, 220
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella pivot e aggiungere questi campi come dimensioni della riga:

- `Month`
- `Region`

Aggiungere questo campo come dimensione di colonna:

- SalesProduct

Creare la seguente misura:

- =Sum(Sales), per calcolare le vendite totali per ogni prodotto.

Per espandere la tabella pivot per vedere tutte le righe, nel pannello delle proprietà, alla voce **Aspetto**, selezionare **Presentazione**. In **Righe**, selezionare **Completamente espanso**.

Results table

| Month<br>Region | SalesProduct |         |
|-----------------|--------------|---------|
|                 | Jeans        | T-shirt |
| 2024-01-01      | 380          | 220     |
| East            | 200          | 100     |
| West            | 180          | 120     |
| 2024-02-01      | 470          | 280     |
| East            | 250          | 150     |
| West            | 220          | 130     |

I risultati mostrano le vendite per ogni SalesProduct per Month e Region, con un totale aggregato per ogni Month.

## Funzioni di aggregazione contatore

Le funzioni di aggregazione contatore restituiscono vari tipi di conteggi di un'espressione su un insieme di record in uno script di caricamento dei dati oppure un numero di valori in una dimensione del grafico.

Ciascuna funzione viene descritta ulteriormente dopo la panoramica. È inoltre possibile fare clic sul nome della funzione nella sintassi per accedere immediatamente ai dettagli per tale funzione specifica.

### Funzioni di aggregazione contatore nello script di caricamento dei dati

#### Count

**Count()** restituisce il numero dei valori aggregati nell'espressione, come definito da una clausola **group by**.

```
Count ([distinct ] expression | * )
```

#### MissingCount

**MissingCount()** restituisce il numero dei valori mancanti aggregati nell'espressione, come definito da una clausola **group by**.

```
MissingCount ([ distinct ] expression)
```

### NullCount

**NullCount()** restituisce il numero dei valori NULL aggregati nell'espressione, come definito da una clausola **group by**.

```
NullCount ([ distinct ] expression)
```

### NumericCount

**NumericCount()** restituisce il numero dei valori numerici presenti nell'espressione, come definito da una clausola **group by**.

```
NumericCount ([ distinct ] expression)
```

### TextCount

**TextCount()** restituisce il numero dei valori di campo non numerici aggregati nell'espressione, come definito da una clausola **group by**.

```
TextCount ([ distinct ] expression)
```

## Funzioni di aggregazione contatore nelle espressioni grafiche

Le seguenti funzioni di aggregazione contatore possono essere utilizzate nei grafici.

### Count

**Count()** viene utilizzato per aggregare il numero di valori, di testo e numerici, in ciascuna dimensione del grafico.

```
Count - funzione per grafici ({ [SetExpression] [DISTINCT] [TOTAL [<fld  
{,fld}>]]} expr)
```

### MissingCount

**MissingCount()** viene utilizzata per aggregare il numero di valori mancanti in ciascuna dimensione del grafico. I valori mancanti sono tutti valori non numerici.

```
MissingCount - funzione per grafici ({ [SetExpression] [DISTINCT] [TOTAL [<fld  
{,fld}>]]} expr)
```

### NullCount

L'istruzione **NullCount()** viene utilizzata per aggregare il numero di valori null in ciascuna dimensione del grafico.

```
NullCount - funzione per grafici ({ [SetExpression] [DISTINCT] [TOTAL [<fld  
{,fld}>]]} expr)
```

### NumericCount

**NumericCount()** aggrega il numero di valori numerici in ciascuna dimensione del grafico.

```
NumericCount - funzione per grafici ({ [SetExpression] [DISTINCT] [TOTAL [<fld  
{,fld}>]]} expr)
```

### TextCount

**TextCount()** viene utilizzata per aggregare il numero di valori di campo che non sono numerici in ciascuna dimensione del grafico.

```
TextCount - funzione per grafici ({ [SetExpression] [DISTINCT] [TOTAL [<fld  
{,fld}>]] } expr)
```

### Count

**Count()** restituisce il numero dei valori aggregati nell'espressione, come definito da una clausola **group by**.

#### Sintassi:

```
Count( [distinct ] expr)
```

**Tipo di dati restituiti:** numero intero

#### Argomenti:

##### Argomenti

| Argomento | Descrizione                                                                                          |
|-----------|------------------------------------------------------------------------------------------------------|
| expr      | L'espressione o il campo contenente i dati da misurare.                                              |
| distinct  | Se la parola <b>distinct</b> è riportata prima dell'espressione, tutti i duplicati vengono ignorati. |

#### Esempi e risultati:

Aggiungere lo script di esempio all'app ed eseguirlo. Aggiungere quindi a un foglio nell'app almeno i campi elencati nella colonna dei risultati per visualizzare il risultato.

Per ottenere lo stesso aspetto della colonna dei risultati mostrata di seguito, nel pannello delle proprietà, in Ordinamento passare da Automatico a Personalizza, quindi deselezionare l'ordinamento numerico e alfabetico.

### Esempi di script

| Esempio                                                                                                                                                                                                                                                                                                                                                                                           | Risultato                                                                                                                                                                                                                                  |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>Temp: LOAD * inline [ Customer Product OrderNumber UnitSales UnitPrice Astrida AA 1 4 16 Astrida AA 7 10 15 Astrida BB 4 9 9 Betacab CC 6 5 10 Betacab AA 5 2 20 Betacab BB 1 25  25 Canutility AA 3 8 15 Canutility CC   19 Divadip CC 2 4 16 Divadip DD 3 1 25 ] (delimiter is ' ');  Count1:  LOAD Customer,Count(OrderNumber) as OrdersByCustomer Resident Temp Group By Customer;</pre> | <p>Customer OrdersByCustomer</p> <p>Astrida 3</p> <p>Betacab 3</p> <p>Canutility 2</p> <p>Divadip 2</p> <p>Finché la dimensione Customer risulta inclusa nella tabella sul foglio, altrimenti il risultato di OrdersByCustomer è 3, 2.</p> |
| <p>Presupponendo che la tabella <b>Temp</b> venga caricata come nell'esempio precedente:</p> <pre>LOAD Count(OrderNumber) as TotalOrderNumber Resident Temp;</pre>                                                                                                                                                                                                                                | <p>TotalOrderNumber</p> <p>10</p>                                                                                                                                                                                                          |
| <p>Presupponendo che la tabella <b>Temp</b> venga caricata come nel primo esempio:</p> <pre>LOAD Count(distinct OrderNumber) as TotalOrderNumber Resident Temp;</pre>                                                                                                                                                                                                                             | <p>TotalOrderNumber</p> <p>8</p> <p>Poiché vi sono due valori di OrderNumber con lo stesso valore, 1, e un solo valore null.</p>                                                                                                           |

### Count - funzione per grafici

**Count()** viene utilizzato per aggregare il numero di valori, di testo e numerici, in ciascuna dimensione del grafico.

#### Sintassi:

```
Count ({[SetExpression] [DISTINCT] [TOTAL [<fld {,fld}>]]} expr)
```

**Tipo di dati restituiti:** numero intero

### Argomenti

| Argomento | Descrizione                                             |
|-----------|---------------------------------------------------------|
| expr      | L'espressione o il campo contenente i dati da misurare. |

| Argomento            | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SetExpression</b> | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                                                                                                                                                                                                                     |
| <b>DISTINCT</b>      | Se la parola <b>DISTINCT</b> è riportata prima degli argomenti della funzione, i duplicati risultanti dalla valutazione degli argomenti della funzione vengono ignorati.                                                                                                                                                                                                                                                                                                                                                                                |
| <b>TOTAL</b>         | Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.<br>Utilizzando <b>TOTAL [&lt;fld {fld}&gt;]</b> , dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili. |

## Esempio: espressioni del grafico

| Esempio                                           | Risultato                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Count(OrderNumber)                                | Restituisce il numero di righe che potrebbero avere un valore per orderNumber. Tutti i record, anche quelli vuoti, vengono contati.<br><br><div>  <p><i>"0" viene conteggiato come valore e non una cella vuota. Tuttavia, se una misura viene aggregata a 0 per una dimensione, tale dimensione non verrà inclusa nei grafici.</i></p> </div> |
| Count(Customer)                                   | Valuta il numero di occorrenze in tutte le righe.                                                                                                                                                                                                                                                                                                                                                                                 |
| Count(DISTINCT [Customer])                        | Il qualificatore DISTINCT limita Count in modo che valuti solo le occorrenze uniche.                                                                                                                                                                                                                                                                                                                                              |
| Count (OrderNumber)/Count ({1} TOTAL OrderNumber) | L'espressione restituisce il numero di ordini del cliente selezionato come percentuale degli ordini di tutti i clienti.                                                                                                                                                                                                                                                                                                           |
| Count(TOTAL <Product> OrderNumber)                | Restituisce il numero di ordini effettuati per i prodotti solo per i clienti selezionati. Vengono contate anche le celle vuote.                                                                                                                                                                                                                                                                                                   |

## Esempio: principi fondamentali della funzione Count

## Espressione del grafico

## Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Customer`
  - `Product`
  - `OrderNumber`
  - `UnitSales`
  - `UnitPrice`

### Script di caricamento

Example:

```
LOAD * inline [
Customer|Product|OrderNumber|UnitSales|UnitPrice
Astrida|AA|1|4|16
Astrida|AA|7|10|15
Astrida|BB|4|9|9
Betacab|CC|6|5|10
Betacab|AA|5|2|20
Betacab|BB|1|25| 25
Canutility|AA|3|8|15
Canutility|CC|||19
Divadip|CC|2|4|16
Divadip|DD|3|1|25
](delimiter is '|');
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `Customer`

Creare le seguenti misure:

- `=Count(OrderNumber)`, per contare il numero di ordini.
- `=Count(Customer)`, per calcolare il numero di record del cliente.
- `=Count(DISTINCT Customer)`, per contare il numero di clienti distinti.
- `=Count(OrderNumber)/Count({1} TOTAL OrderNumber)`, per contare il rapporto degli ordini dei clienti selezionati rispetto al totale degli ordini dei clienti.
- `=Count(TOTAL <Product> OrderNumber)`, per contare il numero di ordini effettuati per i prodotti solo per i clienti selezionati (vengono conteggiate le celle vuote).

Results table

| Customer   | Count (OrderNumber) | Count (Customer) | Count (DISTINCT Customer) | Count (OrderNumber)/Count({1} TOTAL OrderNumber) | Count(TOTAL <Product> OrderNumber) |
|------------|---------------------|------------------|---------------------------|--------------------------------------------------|------------------------------------|
| Totals     | 10                  | 10               | 4                         | 1                                                | 10                                 |
| Astrida    | 3                   | 3                | 1                         | 0.3                                              | 10                                 |
| Betacab    | 3                   | 3                | 1                         | 0.3                                              | 10                                 |
| Canutility | 2                   | 2                | 1                         | 0.2                                              | 10                                 |
| Divadip    | 2                   | 2                | 1                         | 0.2                                              | 10                                 |

Analizzando i risultati, possiamo osservare quanto segue:

- Prima misura: ci sono 10 record di ordini nel set di dati.
- Seconda misura: ci sono 10 record di ordini nel set di dati.
- Terza misura: ci sono 4 clienti distinti (unici) nel set di dati.
- Quarta misura: restituisce 1 perché non è stata applicata alcuna selezione. Se si seleziona il cliente `canutility`, il valore cambia diventa 0.2 perché ci sono 2 record di ordini per il cliente su 10 record totali nel set di dati.
- Quinta misura: restituisce 10 perché non è stata applicata alcuna selezione. Se si selezionano i clienti `Astrida` e `canutility`, il valore diventa 5 perché si tratta del numero di ordini effettuati per i prodotti per questi clienti e (vengono conteggiate anche le celle vuote).

**Esempio: utilizzo della funzione Count per trovare il numero di vendite per prodotto**

Espressione del grafico

### Panoramica

Un set di dati contiene i dati relativi alle vendite del negozio. Bisogna trovare il numero di vendite per prodotto.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Product`
  - `Sales`
  - `Region`
  - `Status`

### Script di caricamento

Example:

```
Load * inline [
Product,Sales,Region,Status
Apple, 100, North, Completed
Banana, 80, South, Pending
Carrot, 50, East, Completed
Apple, 120, North, Completed
Banana, 90, South, Completed
Carrot, 60, West, Pending
Apple, 110, North, Completed
Banana, 85, South, Completed
Carrot, 55, East, Pending
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- Product

Creare le seguenti misure:

- =Count(Sales), per calcolare il numero di vendite per prodotto.
- =Count({<Status={'Completed'}>} Sales), per calcolare il numero di vendite con ordini completati.
- =Count({<Status={'Pending'}>} Sales), per calcolare il numero di vendite con ordini ancora in sospeso.
- =Count({<Region={'North'}>} Sales), per calcolare il numero di ordini nell'area geografica North.

Results table

| Product | Count<br>(Sales) | Count({<Status=<br>'Completed'>} Sales) | Count({<Status=<br>'Pending'>} Sales) | Count({<Region=<br>'North'>} Sales) |
|---------|------------------|-----------------------------------------|---------------------------------------|-------------------------------------|
| Totals  | 9                | 6                                       | 3                                     | 3                                   |
| Apple   | 3                | 3                                       | 0                                     | 3                                   |
| Banana  | 3                | 2                                       | 1                                     | 0                                   |
| Carrot  | 3                | 1                                       | 2                                     | 0                                   |

L'output della funzione count restituisce una scomposizione del numero totale di ordini di vendita per prodotto, oltre al numero di ordini completati e a quelli in attesa. L'ultima misura restituisce un conteggio dei record di vendita per l'area geografica North.

### MissingCount

**MissingCount()** restituisce il numero dei valori mancanti aggregati nell'espressione, come definito da una clausola **group by**.

**Sintassi:**

```
MissingCount ( [ distinct ] expr)
```

**Tipo di dati restituiti:** numero intero

**Argomenti:**

## Argomenti

| Argomento          | Descrizione                                                                                          |
|--------------------|------------------------------------------------------------------------------------------------------|
| expr<br>Expression | L'espressione o il campo contenente i dati da misurare.                                              |
| distinct           | Se la parola <b>distinct</b> è riportata prima dell'espressione, tutti i duplicati vengono ignorati. |

**Esempi e risultati:**

Aggiungere lo script di esempio all'app ed eseguirlo. Aggiungere quindi a un foglio nell'app almeno i campi elencati nella colonna dei risultati per visualizzare il risultato.

Per ottenere lo stesso aspetto della colonna dei risultati mostrata di seguito, nel pannello delle proprietà, in Ordinamento passare da Automatico a Personalizza, quindi deselezionare l'ordinamento numerico e alfabetico.

## Esempi di script

| Esempio                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Risultato                                                                                                                                                                                                                                 |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>Temp: LOAD * inline [ Customer Product OrderNumber UnitSales UnitPrice Astrida AA 1 4 16 Astrida AA 7 10 15 Astrida BB 4 9 9 Betacab CC 6 5 10 Betacab AA 5 2 20 Betacab BB    25 Canutility AA   15 Canutility CC    19 Divadip CC 2 4 16 Divadip DD 3 1 25 ] (delimiter is ' '); MissCount1:  LOAD Customer,MissingCount(OrderNumber) as MissingOrdersByCustomer Resident Temp Group By Customer;  Load MissingCount(OrderNumber) as TotalMissingCount Resident Temp;</pre> | <p>Customer<br/>MissingOrdersByCustomer</p> <p>Astrida 0</p> <p>Betacab 1</p> <p>Canutility 2</p> <p>Divadip 0</p> <p>La seconda istruzione restituisce:</p> <p>TotalMissingCount</p> <p>3</p> <p>in una tabella con tale dimensione.</p> |

| Esempio                                                                                                                                                                             | Risultato                                                                                     |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|
| Presupponendo che la tabella <b>Temp</b> venga caricata come nell'esempio precedente:<br><br>LOAD MissingCount(distinct OrderNumber) as<br>TotalMissingCountDistinct Resident Temp; | TotalMissingCountDistinct<br><br>1<br>Perché è presente un unico valore OrderNumber mancante. |

## MissingCount - funzione per grafici

**MissingCount()** viene utilizzata per aggregare il numero di valori mancanti in ciascuna dimensione del grafico. I valori mancanti sono tutti valori non numerici.

**Sintassi:**

**MissingCount** ({ [SetExpression] [DISTINCT] [TOTAL [<fld {,fld}>]] } expr)

**Tipo di dati restituiti:** numero intero

### Argomenti

| Argomento            | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>expr</b>          | L'espressione o il campo contenente i dati da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>SetExpression</b> | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                                                                                                                                                                                                                      |
| <b>DISTINCT</b>      | Se la parola <b>DISTINCT</b> è riportata prima degli argomenti della funzione, i duplicati risultanti dalla valutazione degli argomenti della funzione vengono ignorati.                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>TOTAL</b>         | Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.<br>Utilizzando <b>TOTAL [&lt;fld {,fld}&gt;]</b> , dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili. |

### Esempio: espressioni del grafico

| Esempio                   | Risultato                                                                                                                                                                                                                                                                                                                                                                     |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MissingCount(OrderNumber) | Restituisce il numero di valori mancanti in cui i campi OrderNumber sono vuoti.<br><br><div>  <p><i>"0" viene conteggiato come valore e non una cella vuota. Tuttavia, se una misura viene aggregata a 0 per una dimensione, tale dimensione non verrà inclusa nei grafici.</i></p> </div> |

| Esempio                                                               | Risultato                                                                                                                                  |
|-----------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| MissingCount<br>(OrderNumber)/MissingCount<br>({1} TOTAL OrderNumber) | L'espressione restituisce il numero di ordini incompleti del cliente selezionato come frazione degli ordini incompleti di tutti i clienti. |

### Esempio: principi fondamentali della funzione MissingCount

#### Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata Example.
- I seguenti campi nella tabella dati:
  - Customer
  - Product
  - OrderNumber
  - UnitsSales
  - UnitPrice

#### Script di caricamento

Example:

```
LOAD * inline [
Customer, Product, OrderNumber, UnitsSales, UnitPrice
Astrida,AA, 1,4,16
Astrida, AA,7,10,15
Astrida, BB,4,9,9
Betacab, BB,6,5,10
Betacab, CC,5,2,20
Betacab ,DD,,25
Canutility, AA,,15
Canutility, CC,,19
Divadip, AA,2,4,16
Divadip, BB,,4,16
Divadip, DD,3,,25
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- Customer

Creare le seguenti misure:

- `=MissingCount(OrderNumber)`, per calcolare il numero di ordini senza il valore `OrderNumber`.
- `=MissingCount(OrderNumber)/MissingCount ({1} Total [OrderNumber])`, per restituire il numero di ordini incompleti del cliente selezionato come frazione degli ordini incompleti di tutti i clienti. Per mostrare questo valore come una percentuale, sotto la voce **Formattazione numero**, selezionare **Numero > Formattazione semplice > 12,34%**.

Results table

| Customer      | MissingCount (OrderNumber) | MissingCount(OrderNumber)/MissingCount ({1} Total [OrderNumber]) |
|---------------|----------------------------|------------------------------------------------------------------|
| <b>Totals</b> | <b>4</b>                   | <b>100.00%</b>                                                   |
| Astrida       | 0                          | 0.00%                                                            |
| Betacab       | 1                          | 25.00%                                                           |
| Canutility    | 2                          | 50.00%                                                           |
| Divadip       | 1                          | 25.00%                                                           |

Nella prima misura, la funzione `MissingCount` restituisce 4 perché 4 dei record 11 nel campo `OrderNumber` sono vuoti.



*"0" viene conteggiato come valore e non una cella vuota. Tuttavia, se una misura viene aggregata a 0 per una dimensione, tale dimensione non verrà inclusa nei grafici.*

Nella seconda misura, l'output della funzione `MissingCount` restituisce il numero di ordini incompleti per il cliente selezionato come una percentuale degli ordini incompleti per tutti i clienti. Ad esempio, se si seleziona `Canutility` dalla dimensione `Customer`, la misura restituisce 50% perché la metà di tutti gli ordini incompleti (2 di 4) sono con questo cliente.

**Esempio: principi fondamentali della funzione `MissingCount` per identificare i dati mancanti su un prodotto**

Espressione del grafico

### Panoramica

Un set di dati contiene i dati mensili degli indicatori chiave per il prodotto (KPI). Un product manager vuole scoprire in quali KPI vi sono dei dati mancanti.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:

- Month
- KPI
- Value

### Script di caricamento

Example:

```
Load * inline [
Month, KPI, Value
2024-01-01, Revenue, 5000
2024-02-01, Revenue,
2024-03-01, Revenue, 5500
2024-04-01, Revenue,
2024-01-01, Expenses, 3000
2024-02-01, Expenses, 3100
2024-03-01, Expenses,
2024-04-01, Expenses, 3200
2024-01-01, Customer Count, 200
2024-02-01, Customer Count, 210
2024-03-01, Customer Count,
2024-04-01, Customer Count,
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- KPI

Creare la seguente misura:

- =MissingCount(Value), per calcolare il numero di valori mancanti.

Results table

| KPI            | MissingCount(Value) |
|----------------|---------------------|
| Totals         | 5                   |
| Customer Count | 2                   |
| Expenses       | 1                   |
| Revenue        | 2                   |

I risultati mostrano che nei record 5 nel set di dati vi sono dei dati mancanti, così come le categorie KPI a cui questi record si riferiscono.

### NullCount

**NullCount()** restituisce il numero dei valori NULL aggregati nell'espressione, come definito da una clausola **group by**.

**Sintassi:**

```
NullCount ( [ distinct ] expr)
```

**Tipo di dati restituiti:** numero intero

**Argomenti:**

## Argomenti

| Argomento          | Descrizione                                                                                          |
|--------------------|------------------------------------------------------------------------------------------------------|
| expr<br>Expression | L'espressione o il campo contenente i dati da misurare.                                              |
| distinct           | Se la parola <b>distinct</b> è riportata prima dell'espressione, tutti i duplicati vengono ignorati. |

**Esempi e risultati:**

Aggiungere lo script di esempio all'app ed eseguirlo. Aggiungere quindi a un foglio nell'app almeno i campi elencati nella colonna dei risultati per visualizzare il risultato.

Per ottenere lo stesso aspetto della colonna dei risultati mostrata di seguito, nel pannello delle proprietà, in Ordinamento passare da Automatico a Personalizza, quindi deselezionare l'ordinamento numerico e alfabetico.

## Esempi di script

| Esempio                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Risultato                                                                                                                                                                                                                                                     |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>Set NULLINTERPRET = NULL; Temp: LOAD * inline [ Customer Product OrderNumber UnitSales CustomerID Astrida AA 1 10 1 Astrida AA 7 18 1 Astrida BB 4 9 1 Astrida CC 6 2 1 Betacab AA 5 4 2 Betacab BB 2 5 2 Betacab DD    Canutility AA 3 8  Canutility CC NULL   ] (delimiter is ' '); Set NULLINTERPRET=; NullCount1:  LOAD Customer,NullCount(OrderNumber) as NullOrdersByCustomer Resident Temp Group By Customer;  LOAD NullCount(OrderNumber) as TotalNullCount Resident Temp;</pre> | <p>Customer NullOrdersByCustomer</p> <p>Astrida 0</p> <p>Betacab 0</p> <p>Canutility 1</p> <p>La seconda istruzione restituisce:</p> <p>TotalNullCount</p> <p>1</p> <p>in una tabella con tale dimensione, poiché solo un record contiene un valore null.</p> |

## NullCount - funzione per grafici

L'istruzione **NullCount()** viene utilizzata per aggregare il numero di valori null in ciascuna dimensione del grafico.

### Sintassi:

```
NullCount ({ [SetExpression] [DISTINCT] [TOTAL [<fld {,fld}>]] } expr)
```

**Tipo di dati restituiti:** numero intero

### Argomenti

| Argomento            | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>expr</b>          | L'espressione o il campo contenente i dati da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>SetExpression</b> | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                                                                                                                                                                                                                      |
| <b>DISTINCT</b>      | Se la parola <b>DISTINCT</b> è riportata prima degli argomenti della funzione, i duplicati risultanti dalla valutazione degli argomenti della funzione vengono ignorati.                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>TOTAL</b>         | Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.<br>Utilizzando <b>TOTAL [&lt;fld {,fld}&gt;]</b> , dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili. |

### Esempio: espressioni del grafico

| Esempio                                                                        | Risultato                                                                                                                                                                                                                                                |
|--------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>NullCount(OrderNumber)</code>                                            | Restituisce il numero di valori null nel campo <code>orderNumber</code> .                                                                                                                                                                                |
| <code>NullCount<br/>(OrderNumber)/NullCount<br/>({1} Total OrderNumber)</code> | L'espressione restituisce il numero di ordini incompleti con <code>NULL</code> come il valore <code>orderNumber</code> del cliente selezionato come frazione di tutti gli ordini dei clienti con valore <code>NULL</code> per <code>orderNumber</code> . |

### Esempio: principi fondamentali della funzione NullCount

Espressione del grafico

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- La variabile per la gestione del valore `NullInterpret` che è impostata su `NULL`. La variabile interpreterà tutti i record con il testo `NULL` come valori nulli.
- I seguenti campi nella tabella dati:
  - Customer
  - Product
  - OrderNumber

### Script di caricamento

```
Set NULLINTERPRET = NULL;  
Example:  
LOAD * inline [  
Customer|Product|OrderNumber  
Astrida|AA|1  
Astrida|AA|7  
Astrida|BB|4  
Astrida|CC|6  
Betacab|AA|5  
Betacab|BB|2  
Betacab|DD|NULL  
Canutility|AA|3  
Canutility|AA|NULL  
Canutility|CC|NULL  
] (delimiter is '|');
```



*Per reimpostare la variabile `NullInterpret` al valore predefinito, includere la seguente istruzione di variabile alla fine dello script di caricamento: `Set NULLINTERPRET =;`*

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- Customer

Creare le seguenti misure:

- `=NullCount(OrderNumber)`, per calcolare il numero di ordini con il valore `NULL` per `OrderNumber`.
- `=NullCount (OrderNumber)/NullCount ({1} Total OrderNumber)`, per restituire il numero di ordini (di clienti selezionati) come il valore `NULL` per `OrderNumber` come frazione di tutti gli ordini dei clienti con valore `NULL` per `OrderNumber`. Per mostrare questo valore come una percentuale, sotto la voce **Formattazione numero**, selezionare **Numero > Formattazione semplice > 12,34%**.

Results table

| Customer   | NullCount (OrderNumber) | NullCount (OrderNumber)/NullCount ({1} Total OrderNumber) |
|------------|-------------------------|-----------------------------------------------------------|
| Totals     | 3                       | 100.00%                                                   |
| Astrida    | 0                       | 0.00%                                                     |
| Betacab    | 1                       | 33.33%                                                    |
| Canutility | 2                       | 66.67%                                                    |

La prima misura restituisce 3 perché ci sono 3 record totali per orderNumber con valore NULL (1 record con il cliente Betacab e 2 record con Canutility).

La seconda misura restituisce il numero di ordini con valore NULL come frazione degli ordini NULL di tutti i clienti. Ad esempio, 66.67% di tutti i numeri degli ordini con valore NULL sono attribuiti al cliente Canutility.

**Esempio: utilizzo della funzione NullCount per identificare i valori null nei dati KPI.**

Espressione del grafico

### Panoramica

Un set di dati contiene i dati mensili degli indicatori chiave per il prodotto (KPI). Un product manager vuole scoprire quali KPI hanno valori null nei dati.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata Example.
- La variabile per la gestione del valore NULL interpret che è impostata su NULL. La variabile interpreterà tutti i record con il testo NULL come valori nulli.
- I seguenti campi nella tabella dati:
  - Month
  - KPI
  - Value

### Script di caricamento

```
Set NULLINTERPRET = NULL;
```

Example:

```
LOAD * inline [
Month, KPI, Value
2024-01-01, Revenue, 5000
2024-02-01, Revenue, NULL
2024-03-01, Revenue, 5500
2024-04-01, Revenue, NULL
2024-01-01, Expenses, 3000
2024-02-01, Expenses, 3100
```

```
2024-03-01, Expenses, NULL
2024-04-01, Expenses, 3200
2024-01-01, Customer Count, 200
2024-02-01, Customer Count, 210
2024-03-01, Customer Count, NULL
2024-04-01, Customer Count, NULL
];
```



*Per reimpostare la variabile `NullInterpret` al valore predefinito, includere la seguente istruzione di variabile alla fine dello script di caricamento: `Set NULLINTERPRET =;`*

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella pivot e aggiungere questo campo come dimensione di riga:

- KPI

Creare la seguente misura:

- `=NullCount(Value)`, per calcolare il numero di valori mancanti.

Results table

| KPI            | NullCount(Value) |
|----------------|------------------|
| Customer Count | 2                |
| Expenses       | 1                |
| Revenue        | 2                |

I risultati mostrano che in 5 record nel set di dati vi sono dei valori `NULL`, così come le categorie KPI a cui sono correlati questi record.

### NumericCount

**NumericCount()** restituisce il numero dei valori numerici presenti nell'espressione, come definito da una clausola **group by**.

#### Sintassi:

```
NumericCount ( [ distinct ] expr)
```

**Tipo di dati restituiti:** numero intero

**Argomenti:**

Argomenti

| Argomento          | Descrizione                                                                                          |
|--------------------|------------------------------------------------------------------------------------------------------|
| expr<br>Expression | L'espressione o il campo contenente i dati da misurare.                                              |
| distinct           | Se la parola <b>distinct</b> è riportata prima dell'espressione, tutti i duplicati vengono ignorati. |

**Esempi e risultati:**

Aggiungere lo script di esempio all'app ed eseguirlo. Aggiungere quindi a un foglio nell'app almeno i campi elencati nella colonna dei risultati per visualizzare il risultato.

Per ottenere lo stesso aspetto della colonna dei risultati mostrata di seguito, nel pannello delle proprietà, in Ordinamento passare da Automatico a Personalizza, quindi deselezionare l'ordinamento numerico e alfabetico.

Esempio di script

| Esempio                                                                                                                                                                                | Risultato                                                                                                                                    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| LOAD NumericCount(OrderNumber) as<br>TotalNumericCount Resident Temp;                                                                                                                  | La seconda istruzione restituisce:<br>TotalNumericCount<br>7<br>in una tabella con tale dimensione.                                          |
| Presupponendo che la tabella <b>Temp</b> venga<br>caricata come nell'esempio precedente:<br><br>LOAD NumericCount(distinct OrderNumber) as<br>TotalNumericCountDistinct Resident Temp; | TotalNumericCountDistinct<br>6<br>Poiché è presente un valore OrderNumber che ne<br>duplica un altro, il risultato è 6 valori non duplicati. |

**Esempio:**

Temp:

```
LOAD * inline [
```

```
Customer|Product|OrderNumber|UnitSales|UnitPrice
```

```
Astrida|AA|1|4|16
```

```
Astrida|AA|7|10|15
```

```
Astrida|BB|4|9|9
```

```
Betacab|CC|6|5|10
```

```
Betacab|AA|5|2|20
```

```
Betacab|BB||| 25
```

```
Canutility|AA|||15
```

```
Canutility|CC| ||19
```

```
Divadip|CC|2|4|16
```

```
Divadip|DD|7|1|25
```

```
] (delimiter is '|');
```

```
NumCount1:
```

```
LOAD Customer,NumericCount(LineNumber) as NumericCountByCustomer Resident Temp Group By Customer;
```

Tabella risultante

| Customer   | NumericCountByCustomer |
|------------|------------------------|
| Astrida    | 3                      |
| Betacab    | 2                      |
| Canutility | 0                      |
| Divadip    | 2                      |

### NumericCount - funzione per grafici

**NumericCount()** aggrega il numero di valori numerici in ciascuna dimensione del grafico.

**Sintassi:**

```
NumericCount ({ [SetExpression] [DISTINCT] [TOTAL [<fld {,fld}>]] } expr)
```

**Tipo di dati restituiti:** numero intero

Argomenti

| Argomento            | Descrizione                                                                                                                                                                                                         |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>expr</b>          | L'espressione o il campo contenente i dati da misurare.                                                                                                                                                             |
| <b>SetExpression</b> | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis. |
| <b>DISTINCT</b>      | Se la parola <b>DISTINCT</b> è riportata prima degli argomenti della funzione, i duplicati risultanti dalla valutazione degli argomenti della funzione vengono ignorati.                                            |

| Argomento    | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>TOTAL</b> | <p>Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.</p> <p>Utilizzando <b>TOTAL</b> [<code>&lt;fld {fld}&gt;</code>], dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili.</p> |

### Esempio: espressioni del grafico

| Esempio                                                          | Risultato                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NumericCount (OrderNumber)                                       | <p>Restituisce il numero di righe del campo orderNumber che hanno valori numerici.</p> <div>  <p><i>"0" viene conteggiato come valore e non una cella vuota. Tuttavia, se una misura viene aggregata a 0 per una dimensione, tale dimensione non verrà inclusa nei grafici. Le celle vuote vengono considerate come campi non numerici e non vengono conteggiate da NumericCount.</i></p> </div> |
| NumericCount (Product)                                           | Restituisce il numero di nomi di prodotti che sono valori numerici. In genere questa funzione può essere utilizzata per verificare che a nessun campo di testo sia stato assegnato contenuto numerico.                                                                                                                                                                                                                                                                            |
| NumericCount (DISTINCT OrderNumber)/Count (DISTINCT OrderNumber) | Conta il numero di numeri distinti di ordine numerico e lo divide per il numero dei numeri di ordine numerico e non numerico. Questo sarà 1 se tutti i valori di campo sono numerici.                                                                                                                                                                                                                                                                                             |

### Esempio: principi fondamentali della funzione NumericCount

#### Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - Customer
  - Product
  - OrderNumber

### Script di caricamento

Example:

```
LOAD * inline [
Customer|Product|OrderNumber
Astrida|AA|1
Astrida|AA|7
Astrida|BB|4
Betacab|CC|6
Betacab|AA|5
Betacab|BB|
Canutility|AA|
Canutility|CC|
Divadip|CC|2
Divadip|DD|3
](delimiter is '|');
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- OrderNumber

Creare le seguenti misure:

- =NumericCount(OrderNumber), per calcolare il numero di ordini contenenti valori numerici.
- =NumericCount(Product), per restituire il numero di nomi di prodotti che sono esclusivamente valori numerici.
- =NumericCount(DISTINCT OrderNumber)/Count(DISTINCT OrderNumber), per calcolare il numero totale di numeri di ordini con valori numerici distinti e dividere questa quantità per il numero totale di numeri di ordini con valori numerici e non numerici distinti. Questo sarà 1 se tutti i valori di campo sono numerici.

Results table

| OrderNumber   | NumericCount<br>(OrderNumber) | NumericCount<br>(Product) | NumericCount (DISTINCT<br>OrderNumber)/Count(DISTINCT<br>OrderNumber) |
|---------------|-------------------------------|---------------------------|-----------------------------------------------------------------------|
| <b>Totals</b> | <b>7</b>                      | <b>0</b>                  | <b>0.875</b>                                                          |
| 1             | 1                             | 0                         | 1                                                                     |
| 2             | 1                             | 0                         | 1                                                                     |
| 3             | 1                             | 0                         | 1                                                                     |
| 4             | 1                             | 0                         | 1                                                                     |
| 5             | 1                             | 0                         | 1                                                                     |
| 6             | 1                             | 0                         | 1                                                                     |
| 7             | 1                             | 0                         | 1                                                                     |
|               | 0                             | 0                         | 0                                                                     |

L'output della prima misura restituisce 7 perché tre dei dieci record in `orderNumber` sono vuoti e i rimanenti sono numerici.

La seconda misura restituisce 0 perché non ci sono valori numerici; tutti i nomi dei prodotti sono testuali. In genere, questo tipo di misura può essere utilizzato per verificare che nessun campo di testo abbia contenuti numerici.

La terza misura conta il numero di numeri distinti di ordine numerico e lo divide per il numero dei numeri sia di ordine numerico che non numerico. Questo sarà 1 se tutti i valori di campo sono numerici. È in genere possibile utilizzare questa operazione per verificare che tutti i valori di campo siano numerici. Nel seguente esempio, sono presenti 7 valori numerici distinti per `orderNumber` di 8 valori distinti numerici e non numerici, pertanto l'espressione restituisce 0.875.

### Esempio: scenario per la funzione `NumericCount`

Espressione del grafico

#### Panoramica

Un set di dati contiene i dati di vendita mensili per più negozi. Il responsabile dei negozi vuole scoprire quali negozi e prodotti hanno valori di vendita che non sono numerici. In questo modo, è possibile trovare i record di vendita con valori non validi, ad esempio se il record era vuoto o contiene dati non numerici.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Store`
  - `StoreProduct`
  - `Sales`

#### Script di caricamento

Example:

```
LOAD * inline [
Store, StoreProduct, Sales
A, Widget A, 1200
B, Widget A, No data
C, Widget A, 1300
A, Widget B, 800
B, Widget B,
C, Widget B, "Error"
A, Widget C, 500
B, Widget C, 600
C, Widget C, 700
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella pivot e aggiungere questo campo come dimensione di riga:

- Store

Aggiungere questo campo come dimensione di colonna:

- StoreProduct

Creare la seguente misura:

- =NumericCount(Sales), per calcolare le vendite del negozio che hanno dati numerici.

Results table

| Store | StoreProduct |          |          |
|-------|--------------|----------|----------|
|       | Widget A     | Widget B | Widget C |
| A     | 1            | 1        | 1        |
| B     | 0            | 0        | 1        |
| C     | 1            | 0        | 1        |

I risultati mostrano che ci sono 3 record in cui il valore numerico è 0. Questi sono i record di vendita che devono essere analizzati perché contengono valori non numerici.

### TextCount

**TextCount()** restituisce il numero dei valori di campo non numerici aggregati nell'espressione, come definito da una clausola **group by**.

Sintassi:

```
TextCount ( [ distinct ] expr)
```

**Tipo di dati restituiti:** numero intero

**Argomenti:**

Argomenti

| Argomento          | Descrizione                                                                                          |
|--------------------|------------------------------------------------------------------------------------------------------|
| expr<br>Expression | L'espressione o il campo contenente i dati da misurare.                                              |
| distinct           | Se la parola <b>distinct</b> è riportata prima dell'espressione, tutti i duplicati vengono ignorati. |

### Esempi e risultati:

Aggiungere lo script di esempio all'app ed eseguirlo. Aggiungere quindi a un foglio nell'app almeno i campi elencati nella colonna dei risultati per visualizzare il risultato.

Per ottenere lo stesso aspetto della colonna dei risultati mostrata di seguito, nel pannello delle proprietà, in Ordinamento passare da Automatico a Personalizza, quindi deselezionare l'ordinamento numerico e alfabetico.

### Esempio:

```
Temp:
LOAD * inline [
Customer|Product|OrderNumber|UnitSales|UnitPrice
Astrida|AA|1|4|16
Astrida|AA|7|10|15
Astrida|BB|4|9|9
Betacab|CC|6|5|10
Betacab|AA|5|2|20
Betacab|BB||| 25
Canutility|AA|||15
Canutility|CC| ||19
Divadip|CC|2|4|16
Divadip|DD|3|1|25
] (delimiter is '|');
TextCount1:
LOAD Customer,TextCount(Product) as ProductTextCount Resident Temp Group By Customer;
```

Tabella risultante

| Customer   | ProductTextCount |
|------------|------------------|
| Astrida    | 3                |
| Betacab    | 3                |
| Canutility | 2                |
| Divadip    | 2                |

### Esempio:

```
LOAD Customer,TextCount(OrderNumber) as OrderNumberTextCount Resident Temp Group By Customer;
```

Tabella risultante

| Customer   | OrderNumberTextCount |
|------------|----------------------|
| Astrida    | 0                    |
| Betacab    | 1                    |
| Canutility | 2                    |
| Divadip    | 0                    |

## TextCount - funzione per grafici

**TextCount()** viene utilizzata per aggregare il numero di valori di campo che non sono numerici in ciascuna dimensione del grafico.

### Sintassi:

```
TextCount ([SetExpression] [DISTINCT] [TOTAL [<fld {,fld}>]] expr)
```

**Tipo di dati restituiti:** numero intero

### Argomenti

| Argomento            | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>expr</b>          | L'espressione o il campo contenente i dati da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>SetExpression</b> | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                                                                                                                                                                                                                      |
| <b>DISTINCT</b>      | Se la parola <b>DISTINCT</b> è riportata prima degli argomenti della funzione, i duplicati risultanti dalla valutazione degli argomenti della funzione vengono ignorati.                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>TOTAL</b>         | Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.<br>Utilizzando <b>TOTAL [&lt;fld {,fld}&gt;]</b> , dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili. |

### Esempio: espressioni del grafico

| Esempio                                      | Risultato                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TextCount (Product)                          | Restituisce il numero di nomi di prodotti che hanno valori testuali.                                                                                                                                                                                                                                                                                                                                                                                                             |
| TextCount (OrderNumber)                      | Restituisce il numero di righe del campo orderNumber che hanno valori testuali.<br><br><div>  <p><i>"0" viene conteggiato come valore e non una cella vuota. Tuttavia, se una misura viene aggregata a 0 per una dimensione, tale dimensione non verrà inclusa nei grafici. Le celle vuote vengono considerate come campi non di testo e non vengono conteggiate da TextCount.</i></p> </div> |
| TextCount (DISTINCT Product)/Count (Product) | Calcola l'intero numero di valori testuali distinti di Product e lo divide per il numero totale di valori in Product.                                                                                                                                                                                                                                                                                                                                                            |

### Esempio: principi fondamentali della funzione TextCount

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Customer`
  - `Product`
  - `OrderNumber`

#### Script di caricamento

```
Set NULLINTERPRET = ' ';
Example:
LOAD * inline [
Customer|Product|OrderNumber
Astrida|AA|1
Astrida|AA|7
Astrida|BB|4
Betacab|CC|6
Betacab|AA|5
Betacab|BB|
Canutility|AA|
Canutility|CC|
Divadip|CC|2
Divadip|DD|3
](delimiter is '|');
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere le seguenti misure:

- `=TextCount(Product)`, per restituire il numero di nomi di prodotti che sono valori testuali.
- `=TextCount(OrderNumber)`, per calcolare il numero dei numeri di ordine che sono valori di testo.
- `=TextCount(DISTINCT Product)/Count(Product)`, per calcolare il numero totale di valori testuali distinti di `Product` e lo divide per il numero totale di valori in `Product`.

Results table

| <code>TextCount(Product)</code> | <code>TextCount(OrderNumber)</code> | <code>TextCount(DISTINCT Product)/Count(Product)</code> |
|---------------------------------|-------------------------------------|---------------------------------------------------------|
| 10                              | 3                                   | 0.4                                                     |

La prima misura restituisce 10 perché tutti e dieci i valori dei record in `Product` sono di testo.

La seconda misura restituisce 3 perché le celle vuote vengono trattate come testo e conteggiate. In genere, è possibile utilizzare questo tipo di misura per verificare che a nessun campo numerico siano assegnati valori testuali o siano vuoti.

La terza misura restituisce 0.4. Questo risultato viene calcolato dividendo il numero totale di valori testuali distinti di `Product` (4) per il numero totale di valori in `Product` (10).

### Esempio: scenario per la funzione `TextCount`

#### Espressione del grafico

#### Panoramica

Un set di dati contiene i dati delle risposte al sondaggio, con i reclami e i feedback generali per prodotto. Bisogna esaminare solo le risposte con valori di testo.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `CustomerID`
  - `ProductPurchased`
  - `Complaint`
  - `Feedback`

#### Script di caricamento

```
Set NullInterpret = '';
Example:
Load * Inline [
CustomerID, ProductPurchased, Complaint, Feedback
1, Laptop, Slow performance, "The laptop is slow, poor battery, slow graphics"
2, Mobile, Heating issue, "Mobile heats up, average camera"
3, Laptop, Battery problem, "Battery not working, slow to charge"
4, Tablet, Screen issue, "Screen is broken, unresponsive touch"
5, Mobile, No issues, "Excellent performance"
6, Mobile, 123, "123"
7, Laptop, ,
8, Tablet, Warranty expired,
9, Mobile, , "Contact customer service"
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `ProductPurchased`

Creare le seguenti misure:

- `=TextCount(ProductPurchased)`, per calcolare il numero di nomi di prodotti con valori non numerici.
- `=TextCount(Complaint)`, per calcolare il numero di nomi di reclami con valori non numerici.
- `=TextCount(Feedback)`, per calcolare il numero di valori di feedback con valori non numerici.

Results table

| ProductPurchased | TextCount<br>(ProductPurchased) | TextCount<br>(Complaint) | TextCount<br>(Feedback) |
|------------------|---------------------------------|--------------------------|-------------------------|
| Laptop           | 3                               | 2                        | 2                       |
| Mobile           | 4                               | 2                        | 3                       |
| Tablet           | 2                               | 2                        | 1                       |

I risultati mostrano che per il prodotto Laptop ci sono 3 record di cui 2 con reclami e 2 con feedback. Allo stesso modo, per il prodotto Mobile, ci sono 4 record, inclusi 2 reclami e 3 risposte di feedback. I valori numerici o i valori vuoti non vengono conteggiati perché si prevede che i campi ProductPurchased, Complaint e Feedback siano valori di testo.

## Funzioni di aggregazione finanziaria

In questa sezione vengono descritte le funzioni di aggregazione per le operazioni finanziarie relative ai pagamenti e al flusso di cassa.

Ciascuna funzione viene descritta ulteriormente dopo la panoramica. È inoltre possibile fare clic sul nome della funzione nella sintassi per accedere immediatamente ai dettagli per tale funzione specifica.

### Funzioni di aggregazione finanziaria nello script di caricamento dei dati

#### IRR

**IRR()** restituisce il tasso di rendimento interno aggregato per una serie di flussi di cassa rappresentati dai numeri nell'espressione ripetuti su un numero di record, come definito da una clausola group by.

**IRR** (expression)

#### XIRR

**XIRR()** restituisce il tasso di rendimento interno aggregato (annuale) per una programmazione di flussi di cassa (non necessariamente periodici) rappresentati da coppie di numeri in **pmt** e **date** ripetuti su un insieme di record, come definito da una clausola group by. Tutti i pagamenti sono scontati in base ad un anno composto da 365 giorni.

**XIRR** (valueexpression, dateexpression )

#### NPV

La funzione script **NPV** applica un tasso di sconto e più valori ordinati in base al periodo. I flussi in entrata (redditi) sono positivi e i flussi in uscita (pagamenti futuri) sono considerati valori negativi per questi calcoli. Si verificano alla fine di ogni periodo.

**VAN** (rate, expression)

### XNPV

**XNPV()** restituisce il valore attuale netto aggregato per una programmazione di flussi di cassa (non necessariamente periodici) rappresentati da coppie di numeri in **pmt** e **date**. Tutti i pagamenti sono scontati in base ad un anno composto da 365 giorni.

```
XNPV (rate, valueexpression, dateexpression)
```

### Funzioni di aggregazione finanziaria nelle espressioni grafiche

Queste funzioni di aggregazione possono essere utilizzate nei grafici.

### IRR

**IRR()** restituisce il tasso di rendimento interno aggregato di una serie di flussi di cassa rappresentati dai numeri dell'espressione data da **value** ripetuti sulle dimensioni del grafico.

```
IRR - funzione per grafici ([TOTAL [<fld {,fld}>]] value)
```

### NPV

**NPV()** restituisce il valore attuale netto aggregato di un investimento basato su un **discount\_rate** per periodo e una serie di pagamenti futuri (valori negativi) ed entrate (valori positivi) rappresentati dai numeri in **value** ripetuti sulle dimensioni del grafico. Si presuppone che i pagamenti e le entrate avvengano alla fine di ciascun periodo.

```
NPV - funzione per grafici ([TOTAL [<fld {,fld}>]] discount_rate, value)
```

### XIRR

**XIRR()** restituisce il tasso di rendimento interno aggregato (annuale) per una programmazione di flussi di cassa (non necessariamente periodici) rappresentati da coppie di numeri nelle espressioni date da **pmt** e **date** ripetuti sulle dimensioni dei grafici. Tutti i pagamenti sono scontati in base ad un anno composto da 365 giorni.

```
XIRR - funzione per grafici ([TOTAL [<fld {,fld}>]] pmt, date)
```

### XNPV

**XNPV()** restituisce il valore netto attuale aggregato per una programmazione di flussi di cassa (non necessariamente periodica) rappresentati da coppie di numeri nelle espressioni date da **pmt** e **date**, ripetuti sulle dimensioni del grafico. Tutti i pagamenti sono scontati in base ad un anno composto da 365 giorni.

```
XNPV - funzione per grafici ([TOTAL [<fld{,fld}>]] discount_rate, pmt, date)
```

### IRR

**IRR()** restituisce il tasso di rendimento interno aggregato per una serie di flussi di cassa rappresentati dai numeri nell'espressione ripetuti su un numero di record, come definito da una clausola group by.

Questi flussi di cassa non devono essere necessariamente pari, come dovrebbero essere per una annualità. Tuttavia, i flussi di cassa devono ricorrere a intervalli regolari, ad esempio ogni mese o ogni

anno. Il tasso di rendimento interno è il tasso di interesse ricevuto per un investimento che consiste in pagamenti (valori negativi) ed entrate (valori positivi) che ricorrono ad intervalli regolari. La funzione necessita di almeno un valore positivo e un valore negativo da calcolare.

Questa funzione utilizza una versione semplificata del metodo Newton per il calcolo del tasso di rendimento interno (IRR).

### Sintassi:

**IRR** (value)

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento | Descrizione                                             |
|-----------|---------------------------------------------------------|
| value     | L'espressione o il campo contenente i dati da misurare. |

### Limiti:

I valori di testo, i valori NULL e i valori mancanti vengono ignorati.

### Esempi e risultati:

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

### Esempi e risultati:

#### Esempi e risultati

| Esempio                                                                                                                                                                                                                                                            | Anno | IRR2013 |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------|
| <pre>Cashflow: LOAD 2013 as Year, * inline [ Date Discount Payments 2013-01-01 0.1 -10000 2013-03-01 0.1 3000 2013-10-30 0.1 4200 2014-02-01 0.2 6800 ] (delimiter is ' ');  Cashflow1:  LOAD Year,IRR(Payments) as IRR2013 Resident Cashflow Group By Year;</pre> | 2013 | 0.1634  |

## IRR - funzione per grafici

**IRR()** restituisce il tasso di rendimento interno aggregato di una serie di flussi di cassa rappresentati dai numeri dell'espressione data da **value** ripetuti sulle dimensioni del grafico.

Questi flussi di cassa non devono essere necessariamente pari, come dovrebbero essere per una annualità. Tuttavia, i flussi di cassa devono ricorrere a intervalli regolari, ad esempio ogni mese o ogni anno. Il tasso di rendimento interno è il tasso di interesse ricevuto per un investimento che consiste in pagamenti (valori negativi) ed entrate (valori positivi) che ricorrono ad intervalli regolari. La funzione necessita di almeno un valore positivo e un valore negativo da calcolare.

Questa funzione utilizza una versione semplificata del metodo Newton per il calcolo del tasso di rendimento interno (IRR).

### Sintassi:

```
IRR([TOTAL [<fld {,fld}>]] value)
```

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | L'espressione o il campo contenente i dati da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| TOTAL     | <p>Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.</p> <p>Utilizzando <b>TOTAL [&lt;fld {,fld}&gt;]</b>, dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili.</p> |

### Limiti:

Il parametro della funzione di aggregazione non deve contenere altre funzioni di aggregazione, a meno che tali aggregazioni interne non contengano il qualificatore **TOTAL**. Nel caso di aggregazioni nidificate più complesse, utilizzare la funzione avanzata **Aggr** in combinazione con una dimensione specificata.

I valori di testo, i valori NULL e i valori mancanti vengono ignorati.

### Esempi e risultati:

#### Esempi e risultati

| Esempio           | Risultato                                                                                                                                                                                                                                                                                                                                                              |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IRR<br>(Payments) | <p>0.1634</p> <p>Si presuppone che i pagamenti siano di natura periodica, ad esempio mensili.</p> <div>  <p><i>Il campo Data viene utilizzato nell'esempio XIRR dove i pagamenti possono essere non periodici purché vengano fornite le date in cui sono effettuati.</i></p> </div> |

Dati utilizzati negli esempi:

```
Cashflow:
LOAD 2013 as Year, * inline [
Date|Discount|Payments
2013-01-01|0.1|-10000
2013-03-01|0.1|3000
2013-10-30|0.1|4200
2014-02-01|0.2|6800
] (delimiter is '|');
```

---

**Vedere anche:**

p [XIRR - funzione per grafici \(page 437\)](#)

p [Aggr - funzione per grafici \(page 605\)](#)

### VAN

La funzione script **NPV** applica un tasso di sconto e più valori ordinati in base al periodo. I flussi in entrata (redditi) sono positivi e i flussi in uscita (pagamenti futuri) sono considerati valori negativi per questi calcoli. Si verificano alla fine di ogni periodo.

Il Valore Attuale Netto, o VAN, viene utilizzato per calcolare il valore totale attuale di un flusso di cassa futuro. Per calcolare il VAN, dobbiamo stimare i flussi di cassa futuri per ogni periodo e determinare il tasso di sconto corretto. La funzione script **NPV** applica un tasso di sconto e più valori ordinati in base al periodo. I flussi in entrata (redditi) sono positivi e i flussi in uscita (pagamenti futuri) sono considerati valori negativi per questi calcoli. Si verificano alla fine di ogni periodo.

**Sintassi:**

```
NPV(discount_rate, value)
```

**Tipo di dati restituiti:** numerico. Per impostazione predefinita, il risultato sarà formattato come valuta.

La formula per calcolare il valore attuale netto è:

$$NPV = \sum_{t=1}^n \frac{R_t}{(1+i)^t}$$

dove:

- $R_t$  = Flussi di cassa netti in entrata e in uscita durante un singolo periodo  $t$
- $i$  = Tasso di sconto o rendimento che si potrebbe ottenere con investimenti alternativi
- $t$  = Numero di periodi del timer

### Argomenti

| Argomento     | Descrizione                                                                                                                                                        |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| discount_rate | <b>discount_rate</b> è il tasso percentuale di sconto applicato.<br><br>Un valore di 0,1 indica un tasso di sconto del 10%.                                        |
| value         | Questo campo contiene i valori di più periodi ordinati in base al periodo. Il primo valore viene assunto come flusso di cassa alla fine del periodo 1, e così via. |

### Limiti:

La funzione NPV() presenta le seguenti limitazioni:

- I valori di testo, i valori NULL e i valori mancanti vengono ignorati.
- I valori del flusso di cassa devono essere in ordine di periodo crescente.

### Casi di utilizzo

NPV() è una funzione finanziaria utilizzata per verificare la redditività del progetto e per ricavare altre misure. Questa funzione è utile quando i flussi di cassa sono disponibili come dati non elaborati.

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione SET DateFormat nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - Pagamento singolo (script)

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati di un progetto e del suo flusso di cassa per un periodo, che viene caricato in una tabella denominata CashFlow.

- Un caricamento residente dalla tabella `CashFlow`, utilizzato per calcolare il campo NPV del progetto in una tabella denominata `NPV`.
- Un tasso di sconto fisso del 10%, utilizzato per il calcolo del VAN.
- Un'istruzione `Group By`, utilizzata per raggruppare tutti i pagamenti del progetto.

### Script di caricamento

```
CashFlow:
Load
*
Inline
[
PrjId,PeriodId,Values
1,1,1000
];

NPV:
Load
    PrjId,
    NPV(0.1,Values) as NPV //Discount Rate of 10%
Resident CashFlow
Group By PrjId;
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `PrjId`
- `NPV`

Tabella dei risultati

| ID progetto | NPV      |
|-------------|----------|
| 1           | \$909.09 |

Per un singolo pagamento di \$1000 da ricevere alla fine di un periodo, con un tasso di sconto del 10% per periodo, il VAN è pari a \$1000 diviso per  $(1 + \text{tasso di sconto})$ . Il VAN effettivo è pari a \$909,09.

### Esempio 2 - Pagamenti multipli (script)

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati di un progetto e del suo flusso di cassa per più periodi, che viene caricato in una tabella denominata `CashFlow`.
- Un caricamento residente dalla tabella `CashFlow`, utilizzato per calcolare il campo VAN del progetto in una tabella denominata `NPV`.
- Nel calcolo del VAN viene utilizzato un tasso di sconto fisso del 10% (0,1).
- Un'istruzione `Group By`, utilizzata per raggruppare tutti i pagamenti del progetto.

### Script di caricamento

```
CashFlow:
Load
*
Inline
[
PrjId,PeriodId,Values
1,1,1000
1,2,1000
];

NPV:
Load
    PrjId,
    NPV(0.1,Values) as NPV //Discount Rate of 10%
Resident CashFlow
Group By PrjId;
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `PrjId`
- `NPV`

Tabella dei risultati

| ID progetto | NPV       |
|-------------|-----------|
| 1           | \$1735.54 |

Per pagamenti di \$1000 dollari da ricevere alla fine di due periodi, con un tasso di sconto del 10% per periodo, il VAN effettivo è pari a \$1735,54.

### Esempio 3 - Pagamenti multipli (script)

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Tassi di sconto per due progetti, che vengono caricati in una tabella denominata `Project`.
- Flussi di cassa per più periodi per ogni progetto per ID progetto e ID periodo. Questo ID periodo può essere utilizzato per ordinare i record nel caso in cui i dati non siano ordinati.
- La combinazione di `NoConcatenate`, caricamenti residenti e la funzione `Left Join` per creare una tabella temporanea, `tmpNPV`. La tabella combina i record di `Project` e delle tabelle `CashFlow` in una tabella piatta. In questa tabella i tassi di sconto saranno ripetuti per ogni periodo.
- Un caricamento residente dalla tabella `tmpNPV`, utilizzato per calcolare il campo VAN per ciascun progetto in una tabella denominata `NPV`.
- Il tasso di sconto a valore singolo associato a ciascun progetto. Questo dato viene recuperato con la funzione `only()` e viene utilizzato nel calcolo del VAN per ogni progetto.
- Un'istruzione `Group By`, utilizzata per raggruppare tutti i pagamenti per ciascun progetto in base all'ID progetto.

Per evitare che vengano caricati dati sintetici o ridondanti nel modello di dati, la tabella `tmpNPV` viene eliminata alla fine dello script.

### Script di caricamento

```
Project:
Load * inline [
PrjId,Discount_Rate
1,0.1
2,0.15
];

CashFlow:
Load
*
Inline
[
PrjId,PeriodId,Values
1,1,1000
1,2,1000
1,3,1000
2,1,500
2,2,500
2,3,1000
2,4,1000
];

tmpNPV:
NoConcatenate Load *
Resident Project;
Left Join
Load *
Resident CashFlow;

NPV:
Load
    PrjId,
    NPV(Only(Discount_Rate),Values) as NPV //Discount Rate will be 10% for Project 1 and 15% for
```

```
Project 2  
Resident tmpNPV  
Group By PrjId;
```

```
Drop table tmpNPV;
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- PrjId
- NPV

Tabella dei risultati

| ID progetto | NPV       |
|-------------|-----------|
| 1           | \$2486.85 |
| 2           | \$2042.12 |

L'ID progetto 1 prevede di ricevere pagamenti di \$1000 alla fine di tre periodi, con un tasso di sconto del 10% per periodo. Pertanto, il VAN effettivo è pari a 2486,85 dollari.

L'ID progetto 2 prevede due pagamenti di \$500 e altri due pagamenti di \$1000 in quattro periodi con un tasso di sconto del 15%. Pertanto, il VAN effettivo è pari a \$2042.12.

### Esempio 4 - Esempio di redditività del progetto (script)

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Tassi di sconto e investimenti iniziali (periodo 0) per due progetti, caricati in una tabella denominata `Project`.
- Flussi di cassa per più periodi per ogni progetto per ID progetto e ID periodo. Questo ID periodo può essere utilizzato per ordinare i record nel caso in cui i dati non siano ordinati.
- La combinazione di `NoConcatenate`, caricamenti residenti e la funzione `Left Join` per creare una tabella temporanea, `tmpNPV`. La tabella combina i record di `Project` e delle tabelle `CashFlow` in una tabella piatta. In questa tabella i tassi di sconto saranno ripetuti per ogni periodo.
- Il tasso di sconto a valore singolo associato a ciascun progetto, che viene recuperato con la funzione `only()` e utilizzato nel calcolo del VAN per ciascun progetto.
- Un caricamento residente dalla tabella `tmpNPV` viene utilizzato per calcolare il campo VAN per ciascun progetto in una tabella denominata `NPV`.

- Per calcolare l'indice di redditività del progetto viene creato un campo aggiuntivo che divide il VAN per l'investimento iniziale di ogni progetto.
- Una dichiarazione di raggruppamento per ID progetto viene utilizzata per raggruppare tutti i pagamenti per ogni progetto.

Per evitare che vengano caricati dati sintetici o ridondanti nel modello di dati, la tabella tmpNPV viene eliminata alla fine dello script.

### Script di caricamento

```
Project:
Load * inline [
PrjId,Discount_Rate, Initial_Investment
1,0.1,100000
2,0.15,100000
];

CashFlow:
Load
*
Inline
[
PrjId,PeriodId,Values,
1,1,35000
1,2,35000
1,3,35000
2,1,30000
2,2,40000
2,3,50000
2,4,60000
];

tmpNPV:
NoConcatenate Load *
Resident Project;
Left Join
Load *
Resident CashFlow;

NPV:
Load
PrjId,
NPV(Only(Discount_Rate),Values) as NPV, //Discount Rate will be 10% for Project 1 and
15% for Project 2
NPV(Only(Discount_Rate),Values)/ Only(Initial_Investment) as Profitability_Index
Resident tmpNPV
Group By PrjId;

Drop table tmpNPV;
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- PrjId
- NPV

Creare la seguente misura:

```
=only(Profitability_Index)
```

Tabella dei risultati

| ID progetto | NPV         | =only(Indice_di_redditività) |
|-------------|-------------|------------------------------|
| 1           | \$87039.82  | 0.87                         |
| 2           | \$123513.71 | 1.24                         |

Il progetto ID 1 ha un VAN effettivo di \$87039,82 e un investimento iniziale di \$100.000. Pertanto, l'indice di redditività è pari a 0,87. Poiché è inferiore a 1, il progetto non è redditizio.

Il progetto ID 2 ha un VAN effettivo di \$123513,71 e un investimento iniziale di \$100.000. Pertanto, l'indice di redditività è pari a 1.24. Poiché è superiore a 1, il progetto è redditizio.

### NPV - funzione per grafici

**NPV()** restituisce il valore attuale netto aggregato di un investimento basato su un **discount\_rate** per periodo e una serie di pagamenti futuri (valori negativi) ed entrate (valori positivi) rappresentati dai numeri in **value** ripetuti sulle dimensioni del grafico. Si presuppone che i pagamenti e le entrate avvengano alla fine di ciascun periodo.

#### Sintassi:

```
NPV([TOTAL [<fld {,fld}>]] discount_rate, value)
```

**Tipo di dati restituiti:** numerico Per impostazione predefinita, il risultato sarà formattato come valuta.

#### Argomenti:

Argomenti

| Argomento     | Descrizione                                                      |
|---------------|------------------------------------------------------------------|
| discount_rate | <b>discount_rate</b> è il tasso percentuale di sconto applicato. |
| value         | L'espressione o il campo contenente i dati da misurare.          |

| Argomento | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TOTAL     | <p>Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.</p> <p>Utilizzando <b>TOTAL [&lt;fld {fld}&gt;]</b>, dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili.</p> <p>Il qualificatore <b>TOTAL</b> può essere seguito da un elenco di uno o più nomi di campo tra parentesi acute. Questi nomi di campo devono essere un sottogruppo delle variabili di dimensione del grafico. In questo caso, il calcolo verrà effettuato ignorando tutte le variabili di dimensione del grafico eccetto quelle elencate, ad esempio un valore verrà restituito per ogni combinazione di valori di campo nei campi delle dimensioni elencate. Anche i campi che non sono attualmente una dimensione in un grafico possono essere inclusi nell'elenco. Questo può essere utile in caso di dimensioni di gruppo, dove i campi di dimensione non sono fissi. Elencando tutte le variabili nel gruppo viene attivata la funzione in corrispondenza delle modifiche del livello di drill-down.</p> |

**Limiti:**

**discount\_rate** e **value** non devono contenere funzioni di aggregazione, a meno che queste aggregazioni interne non contengano il qualificatore **TOTAL**. Nel caso di aggregazioni nidificate più complesse, utilizzare la funzione avanzata **Aggr** in combinazione con una dimensione specificata.

I valori di testo, i valori NULL e i valori mancanti vengono ignorati.

**Esempi e risultati:**

## Esempi e risultati

| Esempio                 | Risultato |
|-------------------------|-----------|
| NPV(Discount, Payments) | -\$540.12 |

Dati utilizzati negli esempi:

```
Cashflow:
LOAD 2013 as Year, * inline [
Date|Discount|Payments
2013-01-01|0.1|-10000
2013-03-01|0.1|3000
2013-10-30|0.1|4200
2014-02-01|0.2|6800
] (delimiter is '|');
```

**Vedere anche:**

p [XNPV - funzione per grafici \(page 447\)](#)

p [Aggr - funzione per grafici \(page 605\)](#)

## XIRR

**XIRR()** restituisce il tasso di rendimento interno aggregato (annuale) per una programmazione di flussi di cassa (non necessariamente periodici) rappresentati da coppie di numeri in **pmt** e **date** ripetuti su un insieme di record, come definito da una clausola group by. Tutti i pagamenti sono scontati in base ad un anno composto da 365 giorni.

La funzionalità XIRR di Qlik (funzioni **XIRR()** e **RangeXIRR()**) utilizza la seguente equazione, risolvendo il valore Rate, per determinare il valore XIRR corretto:

$$XNPV(Rate, pmt, date) = 0$$

L'equazione viene risolta usando una versione semplificata del metodo Newton.

### Sintassi:

**XIRR** (pmt, date )

**Tipo di dati restituiti:** numerico

### Argomenti

| Argomento | Descrizione                                                                                                                               |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------|
| pmt       | Pagamenti. L'espressione o il campo contenente i flussi di cassa corrispondenti alla programmazione di pagamento fornita in <b>date</b> . |
| date      | L'espressione o il campo contenente la programmazione di date corrispondente ai pagamenti con flusso di cassa forniti in <b>pmt</b> .     |

Quando si lavora con questa funzione, si applicano le seguenti limitazioni:

- Se una o entrambe le parti di una coppia di dati include o includono valori di testo, valori NULL e valori mancanti, l'intera coppia di dati verrà ignorata.
- Questa funzione richiede almeno un pagamento negativo valido e un pagamento positivo valido (con date corrispondenti valide). Se questi pagamenti non vengono forniti, sarà restituito un valore NULL.

I seguenti argomenti possono risultare utili per lavorare con questa funzione:

- [XNPV \(page 440\)](#): usare questa funzione per calcolare il valore attuale netto aggregato per una programmazione di flussi di cassa.
- [RangeXIRR \(page 1621\)](#): **RangeXIRR()** è la funzione di scala equivalente per la funzione **XIRR()**.



Nelle diverse versioni di Qlik Sense Client-Managed, sono presenti delle variazioni nell'algoritmo sottostante usato da questa funzione. Per maggiori informazioni sugli aggiornamenti recenti dell'algoritmo, vedere l'articolo del supporto [Correzione e aggiornamento della funzione XIRR](#).

### Esempio

Script di caricamento e risultati

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Dati di transazioni per una serie di flussi di cassa.
- L'utilizzo della funzione **XIRR()** per calcolare il tasso di rendimento annuale interno per questi flussi di cassa.

### Script di caricamento

Cashflow:

```
LOAD 2013 as Year, * inline [
Date|Payments
2013-01-01|-10000
2013-03-01|3000
2013-10-30|4200
2014-02-01|6800
] (delimiter is '|');
```

Cashflow1:

```
LOAD Year,XIRR(Payments, Date) as XIRR2013 Resident Cashflow Group By Year;
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- Year
- XIRR2013

Tabella dei risultati

| Anno | XIRR2013 |
|------|----------|
| 2013 | 0.5385   |

### Interpretazione del valore restituito di XIRR

La funzione XIRR viene solitamente usata per analizzare un investimento, che presenta un pagamento (negativo) in uscita all'inizio e successivamente una serie di pagamenti (positivi) in entrata più piccoli. Di seguito è riportato un esempio semplificato con un pagamento negativo e un pagamento positivo:

Cashflow:

```
LOAD * inline [
Date|Payments
2023-01-01|-100
2024-01-01|110
] (delimiter is '|');
```

Effettuiamo un pagamento iniziale pari a 100 e riceviamo 110 dopo un anno esatto. Ciò rappresenta un tasso di rendimento del 10% all'anno. `XIRR(Payments, Date)` restituisce un valore pari a 0,1.

Il valore restituito della funzione `XIRR` può essere positivo o negativo. Nel caso di un investimento, un risultato negativo indica che l'investimento è una perdita. L'importo del guadagno o della perdita può essere calcolato semplicemente tramite un'aggregazione `Sum` sul campo dei pagamenti.

Nell'esempio precedente, prestiamo denaro per un anno. Il tasso di rendimento può essere considerato come l'interesse. È inoltre possibile usare la funzione `XIRR` quando si è dall'altro lato della transazione (ad esempio, quando si è il debitore invece del prestatore).

Si consideri questo esempio:

```
Cashflow:
LOAD * inline [
Date|Payments
2023-01-01|100
2024-01-01|-110
] (delimiter is '|');
```

È uguale al primo esempio, ma invertito. Ora prendiamo in prestito 100 per un anno e ripaghiamo il prestito con un interesse del 10%. In questo esempio, il calcolo `XIRR` restituisce 0,1 (10%), lo stesso valore del primo esempio.

Si noti che, nel primo esempio, abbiamo ricevuto un profitto di 10 e, nel secondo esempio, abbiamo subito una perdita di 10, ma il valore restituito della funzione `XIRR` è positivo in entrambi i casi. Ciò è dovuto al fatto che la funzione `XIRR` calcola l'interesse nascosto nella transazione, a prescindere dal ruolo che si riveste in essa.

### Limitazioni con più soluzioni

La funzione `XIRR` di Qlik è definita dalla seguente equazione, nella quale il valore `Rate` è risolto:

$$\text{XNPV}(\text{Rate}, \text{pmt}, \text{date}) = 0$$

A volte è possibile che questa equazione abbia più di una soluzione. Questo fenomeno è noto come "problema dell'IRR multiplo" ed è causato da una serie di flussi di cassa non normali (detti anche flussi di cassa non convenzionali). Il seguente script di caricamento mostra un esempio di tutto ciò:

```
Cashflow:
LOAD * inline [
Date|Payments
2021-01-01|-200
2022-01-01|500
2023-01-01|-250
] (delimiter is '|');
```

In questo esempio vi sono una soluzione negativa e una soluzione positiva (`Rate = -0,3` e `Rate = 0,8`). `XIRR()` restituirà 0,8.

Quando la funzione `XIRR` di Qlik cerca una soluzione, inizia da `Rate = 0` e aumenta gradualmente il tasso finché non trova una soluzione. Se esiste più di una soluzione positiva, restituirà la prima che trova. Se non riesce a trovare una soluzione positiva, azzererà `Rate` e inizierà a cercare una soluzione nella direzione negativa.

Si noti che una serie di flussi di cassa "normali" avrà sempre un'unica soluzione. Una serie di flussi di cassa "normali" implica che tutti i pagamenti con lo stesso segno (positivo o negativo) sono in un gruppo continuo.

### Vedere anche:

p [XNPV \(page 440\)](#)

p [RangeXIRR \(page 1621\)](#)

≤ [Correzione e aggiornamento della funzione XIRR](#)

### XIRR - funzione per grafici

**XIRR()** restituisce il tasso di rendimento interno aggregato (annuale) per una programmazione di flussi di cassa (non necessariamente periodici) rappresentati da coppie di numeri nelle espressioni date da **pmt** e **date** ripetuti sulle dimensioni dei grafici. Tutti i pagamenti sono scontati in base ad un anno composto da 365 giorni.

La funzionalità XIRR di Qlik (funzioni **XIRR()** e **RangeXIRR()**) utilizza la seguente equazione, risolvendo il valore rate, per determinare il valore XIRR corretto:

$$\text{XNPV}(\text{Rate}, \text{pmt}, \text{date}) = 0$$

l'equazione viene risolta usando una versione semplificata del metodo Newton.

### Sintassi:

```
XIRR ([TOTAL [<fld {,fld}>]] pmt, date)
```

**Tipo di dati restituiti:** numerico

#### Argomenti

| Argomento | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| pmt       | Pagamenti. L'espressione o il campo contenente i flussi di cassa corrispondenti alla programmazione di pagamento fornita in <b>date</b> .                                                                                                                                                                                                                                                                                                                                                                                                                          |
| date      | L'espressione o il campo contenente la programmazione di date corrispondente ai pagamenti con flusso di cassa forniti in <b>pmt</b> .                                                                                                                                                                                                                                                                                                                                                                                                                              |
| TOTAL     | <p>Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.</p> <p>Utilizzando <b>TOTAL [&lt;fld {,fld}&gt;]</b>, dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili.</p> |

Quando si lavora con questa funzione, si applicano le seguenti limitazioni:

- **pmt** e **date** non devono contenere funzioni di aggregazione, a meno che queste aggregazioni interne non contengano il qualificatore **TOTAL**. Nel caso di aggregazioni nidificate più complesse, utilizzare la funzione avanzata **Aggr** in combinazione con una dimensione specificata.
- Se una o entrambe le parti di una coppia di dati includono valori di testo, valori NULL e valori mancanti, l'intera coppia di dati verrà ignorata.
- Questa funzione richiede almeno un pagamento negativo valido e un pagamento positivo valido (con date corrispondenti valide). Se questi pagamenti non vengono forniti, sarà restituito un valore NULL.

I seguenti argomenti possono risultare utili per lavorare con questa funzione:

- [XNPV - funzione per grafici \(page 447\)](#): usare questa funzione per calcolare il valore attuale netto aggregato per una programmazione di flussi di cassa.
- [RangeXIRR \(page 1621\)](#): **RangeXIRR()** è la funzione di scala equivalente per la funzione **XIRR()**.



*Nelle diverse versioni di Qlik Sense Client-Managed, sono presenti delle variazioni nell'algoritmo sottostante usato da questa funzione. Per maggiori informazioni sugli aggiornamenti recenti dell'algoritmo, vedere l'articolo del supporto [Correzione e aggiornamento della funzione XIRR](#).*

### Esempio

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente le transazioni del flusso di cassa.
- Informazioni archiviate in una tabella chiamata `CashFlow`.

#### Script di caricamento

`CashFlow:`

```
LOAD 2013 as Year, * inline [
Date|Payments
2013-01-01|-10000
2013-03-01|3000
2013-10-30|4200
2014-02-01|6800
] (delimiter is '|');
```

### Risultati

Procedere come indicato di seguito:

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere il seguente calcolo come misura:

```
=XIRR(Payments, Date)
```

Tabella dei risultati

| =XIRR(Payments, Date) |
|-----------------------|
| 0.5385                |

### Interpretazione del valore restituito di XIRR

La funzione XIRR viene solitamente usata per analizzare un investimento, che presenta un pagamento (negativo) in uscita all'inizio e successivamente una serie di pagamenti (positivi) in entrata più piccoli. Di seguito è riportato un esempio semplificato con un pagamento negativo e un pagamento positivo:

```
Cashflow:  
LOAD * inline [  
Date|Payments  
2023-01-01|-100  
2024-01-01|110  
] (delimiter is '|');
```

Effettuiamo un pagamento iniziale pari a 100 e riceviamo 110 dopo un anno esatto. Ciò rappresenta un tasso di rendimento del 10% all'anno. XIRR(Payments, Date) restituisce un valore pari a 0,1.

Il valore restituito della funzione XIRR può essere positivo o negativo. Nel caso di un investimento, un risultato negativo indica che l'investimento è una perdita. L'importo del guadagno o della perdita può essere calcolato semplicemente tramite un'aggregazione Sum sul campo dei pagamenti.

Nell'esempio precedente, prestiamo denaro per un anno. Il tasso di rendimento può essere considerato come l'interesse. È inoltre possibile usare la funzione XIRR quando si è dall'altro lato della transazione (ad esempio, quando si è il debitore invece del prestatore).

Si consideri questo esempio:

```
Cashflow:  
LOAD * inline [  
Date|Payments  
2023-01-01|100  
2024-01-01|-110  
] (delimiter is '|');
```

È uguale al primo esempio, ma invertito. Ora prendiamo in prestito 100 per un anno e ripaghiamo il prestito con un interesse del 10%. In questo esempio, il calcolo XIRR restituisce 0,1 (10%), lo stesso valore del primo esempio.

Si noti che, nel primo esempio, abbiamo ricevuto un profitto di 10 e, nel secondo esempio, abbiamo subito una perdita di 10, ma il valore restituito della funzione XIRR è positivo in entrambi i casi. Ciò è dovuto al fatto che la funzione XIRR calcola l'interesse nascosto nella transazione, a prescindere dal ruolo che si riveste in essa.

### Limitazioni con più soluzioni

La funzione XIRR di Qlik è definita dalla seguente equazione, nella quale il valore Rate è risolto:

$$\text{XNPV}(\text{Rate}, \text{pmt}, \text{date}) = 0$$

A volte è possibile che questa equazione abbia più di una soluzione. Questo fenomeno è noto come "problema dell'IRR multiplo" ed è causato da una serie di flussi di cassa non normali (detti anche flussi di cassa non convenzionali). Il seguente script di caricamento mostra un esempio di tutto ciò:

```
Cashflow:
LOAD * inline [
Date|Payments
2021-01-01|-200
2022-01-01|500
2023-01-01|-250
] (delimiter is '|');
```

In questo esempio vi sono una soluzione negativa e una soluzione positiva (Rate = -0,3 e Rate = 0,8). **XIRR()** restituirà 0,8.

Quando la funzione XIRR di Qlik cerca una soluzione, inizia da Rate = 0 e aumenta gradualmente il tasso finché non trova una soluzione. Se esiste più di una soluzione positiva, restituirà la prima che trova. Se non riesce a trovare una soluzione positiva, azzererà Rate e inizierà a cercare una soluzione nella direzione negativa.

Si noti che una serie di flussi di cassa "normali" avrà sempre un'unica soluzione. Una serie di flussi di cassa "normali" implica che tutti i pagamenti con lo stesso segno (positivo o negativo) sono in un gruppo continuo.

---

### Vedere anche:

- p [IRR - funzione per grafici \(page 423\)](#)
- p [Aggr - funzione per grafici \(page 605\)](#)
- ≤ [Correzione e aggiornamento della funzione XIRR](#)

### XNPV

**XNPV()** restituisce il valore attuale netto aggregato per una programmazione di flussi di cassa (non necessariamente periodici) rappresentati da coppie di numeri in **pmt** e **date**. Tutti i pagamenti sono scontati in base ad un anno composto da 365 giorni.

### Sintassi:

```
XNPV(discount_rate, pmt, date)
```

**Tipo di dati restituiti:** numerico



Per impostazione predefinita, il risultato sarà formattato come valuta.

La formula per calcolare l'XNPV è mostrata di seguito:

Formula di aggregazione XNPV

$$XNPV = \sum_{i=1}^n \frac{P_i}{(1+rate)^{(d_i-d_1)/365}}$$

dove:

- $P_i$  = Flussi di cassa netti in entrata e in uscita durante un singolo periodo  $i$
- $d_1$  = la data del primo pagamento
- $d_i$  = la data del  $i^o$  pagamento
- $rate$  = tasso di sconto

Il valore attuale netto, o VAN, viene utilizzato per calcolare il valore totale attuale di una serie futura di flussi di cassa tenendo in considerazione un tasso di sconto. Per calcolare l'XNPV, è necessario stimare flussi di cassa futuri con date corrispondenti. In seguito, per ogni pagamento, si applica il tasso di sconto composto in base alla data del pagamento.

Eseguire un'aggregazione XNPV su una serie di pagamenti è simile a eseguire un'aggregazione Sum sugli stessi pagamenti. La differenza è che ogni importo viene modificato (o "scontato") in base al tasso di sconto scelto (simile al tasso di interesse) e a quanto tempo nel futuro avviene il pagamento. Eseguire XNPV con il parametro **discount\_rate** impostato su zero renderà XNPV equivalente a un'operazione Sum (i pagamenti non verranno modificati prima di essere sommati). In generale, più **discount\_rate** si avvicina allo zero, più il risultato di XNPV assomiglierà a quello di un'aggregazione Sum.

#### Argomenti

| Argomento     | Descrizione                                                                                                                                                        |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| discount_rate | <p><b>discount_rate</b> è il tasso annuale in base al quale dovrebbero essere scontati i pagamenti.</p> <p>Un valore di 0,1 indica un tasso di sconto del 10%.</p> |

| Argomento | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| pmt       | <p>Pagamenti. L'espressione o il campo contenente i flussi di cassa corrispondenti alla programmazione di pagamento fornita in <b>date</b>. I valori positivi sono considerati flussi in entrata, mentre quelli negativi sono considerati flussi in uscita.</p> <div>  <p><i><b>XNPV()</b> non sconta il flusso di cassa iniziale, dato che avrà sempre luogo alla data di inizio. I pagamenti successivi sono scontati in base ad un anno composto da 365 giorni. È differente da <b>NPV()</b>, dove anche il primo pagamento viene scontato.</i></p> </div> |
| date      | L'espressione o il campo contenente la programmazione di date corrispondente ai pagamenti con flusso di cassa forniti in <b>pmt</b> . Il primo valore viene utilizzato come data di inizio per il calcolo dei differimenti temporali per i flussi di cassa futuri.                                                                                                                                                                                                                                                                                                                                                                             |

Quando si lavora con questa funzione, si applicano le seguenti limitazioni:

- Se una o entrambe le parti di una coppia di dati includono valori di testo, valori NULL e valori mancanti, l'intera coppia di dati verrà ignorata.

### Casi di utilizzo

- **XNPV()** è utilizzato nella modellistica finanziaria per calcolare il valore attuale netto (VAN) di un'opportunità di investimento.
- Grazie alla sua maggiore precisione, l'**XNPV** è preferito al VAN per tutti i tipi di modelli finanziari.

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - Pagamento singolo (script)

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati di un progetto e del suo flusso di cassa per un anno, in una tabella denominata `CashFlow`. La data iniziale per il calcolo è fissata al 1° luglio 2022, con un flusso di cassa netto pari a 0. Dopo un anno, si verifica un flusso di cassa di 1000 dollari.
- Un caricamento residente dalla tabella `CashFlow`, utilizzato per calcolare il campo `XNPV` del progetto in una tabella denominata `XNPV`.
- Nel calcolo del `XNPV` viene utilizzato un tasso di sconto fisso del 10% (0,1).
- Un'istruzione `Group By` viene utilizzata per raggruppare tutti i pagamenti del progetto.

### Script di caricamento

`CashFlow:`

`Load`

`*`

`Inline`

`[`

`PrjId,Dates,Values`

`1,'07/01/2022',0`

`1,'07/01/2023',1000`

`];`

`XNPV:`

`Load`

`PrjId,`

`XNPV(0.1,Values,Dates) as XNPV //Discount Rate of 10%`

`Resident CashFlow`

`Group By PrjId;`

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `PrjId`
- `XNPV`

Tabella dei risultati

| PrjId | XNPV     |
|-------|----------|
| 1     | \$909.09 |

Secondo la formula, il valore `XNPV` per il primo record è 0, mentre per il secondo record il valore `XNPV` è di \$909,09. Pertanto, il valore `XNPV` totale è \$909.09.

### Esempio 2 - Pagamenti multipli (script)

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati di un progetto e del suo flusso di cassa per un anno, in una tabella denominata `CashFlow`.
- Un caricamento residente dalla tabella `CashFlow`, utilizzato per calcolare il campo `XNPV` del progetto in una tabella denominata `XNPV`.
- Nel calcolo del `XNPV` viene utilizzato un tasso di sconto fisso del 10% (0,1).
- Un'istruzione `Group By` viene utilizzata per raggruppare tutti i pagamenti del progetto.

#### Script di caricamento

`CashFlow:`

`Load`

`*`

`Inline`

`[`

`PrjId,Dates,Values`

`1,'07/01/2022',0`

`1,'07/01/2024',500`

`1,'07/01/2023',1000`

`];`

`XNPV:`

`Load`

`PrjId,`

`XNPV(0.1,Values,Dates) as XNPV //Discount Rate of 10%`

`Resident CashFlow`

`Group By PrjId;`

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `PrjId`
- `XNPV`

Tabella dei risultati

| PrjId | XNPV      |
|-------|-----------|
| 1     | \$1322.21 |

In questo esempio, alla fine del primo anno si riceve un pagamento di \$1000 dollari e alla fine del secondo anno se ne riceve uno di \$500. Con un tasso di sconto del 10% per periodo, l'XNPV effettivo è pari a \$1322,21.

Notare che solo la prima riga di dati deve fare riferimento alla data base per i calcoli. Per il resto delle righe, l'ordine non è importante, poiché il parametro della data viene utilizzato per calcolare il periodo trascorso.

### Esempio 3 - Pagamenti multipli e flussi di cassa irregolari (script)

#### Script di caricamento e risultati

##### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Tassi di sconto per due progetti in una tabella denominata `Project`.
- Flussi di cassa per più periodi per ogni progetto per ID progetto e Date. Il campo `Dates` viene utilizzato per calcolare la durata per la quale il tasso di sconto viene applicato al flusso di cassa. A parte il primo record (flusso di cassa iniziale e data), l'ordine dei record non è importante e la sua modifica non dovrebbe influire sui calcoli.
- Utilizzando una combinazione di `NoConcatenate`, caricamenti residenti e la funzione `Left Join`, viene creata una tabella temporanea, `tmpNPV`, che combina i record delle tabelle `Project` e `CashFlow` in una tabella piatta. In questa tabella i tassi di sconto saranno ripetuti per ogni flusso di cassa.
- Un caricamento residente dalla tabella `tmpNPV`, utilizzato per calcolare il campo `XNPV` per ciascun progetto in una tabella denominata `XNPV`.
- Il tasso di sconto a valore singolo associato a ciascun progetto viene recuperato tramite la funzione `only()` e utilizzato nel calcolo `XNPV` per ciascun progetto.
- Un'istruzione `Group By`, raggruppata per ID progetto, viene utilizzata per raggruppare tutti i pagamenti e le date corrispondenti per ogni progetto.
- Per evitare che vengano caricati dati sintetici o ridondanti nel modello di dati, la tabella `tmpXNPV` viene eliminata alla fine dello script.

#### Script di caricamento

```
Project:
Load * inline [
PrjId,Discount_Rate
1,0.1
2,0.15
];
```

```
CashFlow:
Load
*
Inline
[
PrjId,Dates,Values
```

```
1, '07/01/2021', 0
1, '07/01/2022', 1000
1, '07/01/2023', 1000
2, '07/01/2020', 0
2, '07/01/2023', 500
2, '07/01/2024', 1000
2, '07/01/2022', 500
];

tmpXNPV:
NoConcatenate Load *
Resident Project;
Left Join
Load *
Resident CashFlow;

XNPV:
Load
    PrjId,
    XNPV(Only(Discount_Rate),Values,Dates) as XNPV //Discount Rate will be 10% for Project 1 and
15% for Project 2
Resident tmpXNPV
Group By PrjId;

Drop table tmpXNPV;
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- PrjId
- XNPV

Tabella dei risultati

| PrjId | XNPV      |
|-------|-----------|
| 1     | \$1735.54 |
| 2     | \$278.36  |

Il progetto ID 1 ha un flusso di cassa iniziale di 0 dollari il 1° luglio 2021. Sono previsti due pagamenti di 1.000 dollari da ricevere alla fine di due anni successivi, a un tasso di sconto del 10% per periodo. Pertanto, il valore XNPV effettivo è pari a \$1735.54.

Il progetto ID 2 ha un'uscita iniziale di \$1.000 (quindi il segno negativo) il 1° luglio 2020. Dopo due anni è previsto un pagamento di \$500. Dopo 3 anni è previsto un altro pagamento di \$500. Infine, il 1° luglio 2024 è previsto un pagamento di \$1.000. Con un tasso di sconto del 15%, l'XNPV effettivo è pari a \$278,36.

### Vedere anche:

p [Drop table \(page 157\)](#)  
p [group by \(page 169\)](#)

- p [Join \(page 74\)](#)
- p [Max \(page 365\)](#)
- p [NoConcatenate \(page 93\)](#)
- p [NPV - funzione per grafici \(page 432\)](#)
- p [Only \(page 381\)](#)

### XNPV - funzione per grafici

**XNPV()** restituisce il valore netto attuale aggregato per una programmazione di flussi di cassa (non necessariamente periodica) rappresentati da coppie di numeri nelle espressioni date da **pmt** e **date**, ripetuti sulle dimensioni del grafico. Tutti i pagamenti sono scontati in base ad un anno composto da 365 giorni.

#### Sintassi:

```
XNPV ([TOTAL [<fld{,fld}>]] discount_rate, pmt, date)
```

**Tipo di dati restituiti:** numerico



Per impostazione predefinita, il risultato sarà formattato come valuta.

La formula per calcolare l'XNPV è mostrata di seguito:

Formula di aggregazione XNPV

$$XNPV = \sum_{i=1}^n \frac{P_i}{(1+rate)^{(d_i-d_1)/365}}$$

dove:

- $P_i$  = Flussi di cassa netti in entrata e in uscita durante un singolo periodo  $i$
- $d_1$  = la data del primo pagamento
- $d_i$  = la data del  $i^o$  pagamento
- $rate$  = tasso di sconto

Il valore attuale netto, o VAN, viene utilizzato per calcolare il valore totale attuale di una serie futura di flussi di cassa tenendo in considerazione un tasso di sconto. Per calcolare l'XNPV, è necessario stimare flussi di cassa futuri con date corrispondenti. In seguito, per ogni pagamento, si applica il tasso di sconto composto in base alla data del pagamento.

Eseguire un'aggregazione XNPV su una serie di pagamenti è simile a eseguire un'aggregazione Sum sugli stessi pagamenti. La differenza è che ogni importo viene modificato (o "scontato") in base al tasso di sconto scelto (simile al tasso di interesse) e a quanto tempo nel futuro avviene il pagamento. Eseguire XNPV con il parametro **discount\_rate** impostato su zero renderà XNPV equivalente a un'operazione Sum (i pagamenti non verranno modificati prima di essere sommati). In generale, più **discount\_rate** si avvicina allo zero, più il risultato di XNPV assomiglierà a quello di un'aggregazione Sum.

## Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| discount_rate | <p><b>discount_rate</b> è il tasso annuale in base al quale dovrebbero essere scontati i pagamenti.</p> <p>Un valore di 0,1 indica un tasso di sconto del 10%.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| pmt           | <p>Pagamenti. L'espressione o il campo contenente i flussi di cassa corrispondenti alla programmazione di pagamento fornita in <b>date</b>. I valori positivi sono considerati flussi in entrata, mentre quelli negativi sono considerati flussi in uscita.</p> <div>  <p><b>XNPV()</b> non sconta il flusso di cassa iniziale, dato che avrà sempre luogo alla data di inizio. I pagamenti successivi sono scontati in base ad un anno composto da 365 giorni. È differente da <b>NPV()</b>, dove anche il primo pagamento viene scontato.</p> </div> |
| date          | <p>L'espressione o il campo contenente la programmazione di date corrispondente ai pagamenti con flusso di cassa forniti in <b>pmt</b>. Il primo valore viene utilizzato come data di inizio per il calcolo dei differimenti temporali per i flussi di cassa futuri.</p>                                                                                                                                                                                                                                                                                                                                                                |
| TOTAL         | <p>Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.</p> <p>Utilizzando <b>TOTAL [&lt;fld {fld}&gt;]</b>, dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili.</p>                                                                       |

Quando si lavora con questa funzione, si applicano le seguenti limitazioni:

- **discount\_rate**, **pmt** e **date** non devono contenere funzioni di aggregazione, a meno che queste aggregazioni interne non contengano il qualificatore **TOTAL** o **ALL**. Nel caso di aggregazioni nidificate più complesse, utilizzare la funzione avanzata **Aggr** in combinazione con una dimensione specificata.
- Se una o entrambe le parti di una coppia di dati includono valori di testo, valori NULL e valori mancanti, l'intera coppia di dati verrà ignorata.

## Casi di utilizzo

- **XNPV()** è utilizzato nella modellistica finanziaria per calcolare il valore attuale netto (VAN) di un'opportunità di investimento.
- Grazie alla sua maggiore precisione, l'**XNPV** è preferito al VAN per tutti i tipi di modelli finanziari.

## Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione **SET DateFormat** nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema,

a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente le transazioni del flusso di cassa.
- Informazioni archiviate in una tabella chiamata `CashFlow`.

#### Script di caricamento

```
CashFlow:
LOAD 2013 as Year, * inline [
Date|Payments
2013-01-01|-10000
2013-03-01|3000
2013-10-30|4200
2014-02-01|6800
] (delimiter is '|');
```

#### Risultati

Procedere come indicato di seguito:

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere il seguente calcolo come misura:

`=XNPV(0.09, Payments, Date)`

Tabella dei risultati

| <code>=XNPV(0.09, Payments, Date)</code> |
|------------------------------------------|
| \$3062.49                                |

Vedere anche:

p [NPV - funzione per grafici \(page 432\)](#)

p [Aggr - funzione per grafici \(page 605\)](#)

## Funzioni di aggregazione statistica

Ciascuna funzione viene descritta ulteriormente dopo la panoramica. È inoltre possibile fare clic sul nome della funzione nella sintassi per accedere immediatamente ai dettagli per tale funzione specifica.

### Funzioni di aggregazione statistica nello script di caricamento dei dati

Le seguenti funzioni di aggregazione statistica possono essere utilizzate negli script.

#### Avg

**Avg()** restituisce il valore medio dei dati aggregati nell'espressione su un insieme di record, come definito da una clausola **group by**.

```
Avg ([distinct] expression)
```

#### Correl

**Correl()** restituisce il coefficiente di correlazione aggregato per una serie di coordinate rappresentata da coppie di numeri in x-expression e y-expression ripetute su un insieme di record definito da una clausola **group by**.

```
Correl (x-expression, y-expression)
```

#### Fractile

**Fractile()** restituisce il valore corrispondente al frattale inclusivo (quantile) dei dati aggregati nell'espressione su un insieme di record, come definito da una clausola **group by**.

```
Fractile (expression, fractile)
```

#### FractileExc

**FractileExc()** restituisce il valore corrispondente al frattale esclusivo (quantile) dei dati aggregati nell'espressione su un insieme di record, come definito da una clausola **group by**.

```
FractileExc (expression, fractile)
```

#### Kurtosis

**Kurtosis()** restituisce il kurtosis dei dati nell'espressione su un insieme di record, come definito da una clausola **group by**.

```
Kurtosis ([distinct ] expression )
```

#### LINEST\_B

**LINEST\_B()** restituisce il valore b aggregato (intercettazione sull'asse y) di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentata da coppie di numeri in x-expression e y-expression ripetute su un insieme di record, come definito da una clausola **group by**.

```
LINEST B (y-expression, x-expression [, y0 [, x0 ]])
```

### LINEST\_df

**LINEST\_DF()** restituisce i gradi di libertà aggregati di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentata da coppie di numeri in x-expression e y-expression ripetute su un insieme di record, come definito da una clausola **group by**.

```
LINEST_DF (y-expression, x-expression [, y0 [, x0 ]])
```

### LINEST\_f

Questa funzione di script restituisce la statistica F aggregata ( $r^2/(1-r^2)$ ) di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentata da coppie di numeri in x-expression e y-expression ripetute su un insieme di record definito da una clausola **group by**.

```
LINEST_F (y-expression, x-expression [, y0 [, x0 ]])
```

### LINEST\_m

**LINEST\_M()** restituisce il valore m aggregato (pendenza) di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentata da coppie di numeri in x-expression e y-expression ripetute su un insieme di record, come definito da una clausola **group by**.

```
LINEST_M (y-expression, x-expression [, y0 [, x0 ]])
```

### LINEST\_r2

**LINEST\_R2()** restituisce il valore  $r^2$  aggregato (coefficiente di determinazione) di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentata da coppie di numeri in x-expression e y-expression ripetute su un insieme di record, come definito da una clausola **group by**.

```
LINEST_R2 (y-expression, x-expression [, y0 [, x0 ]])
```

### LINEST\_seb

**LINEST\_SEB()** restituisce l'errore standard aggregato del valore b di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentata da coppie di numeri in x-expression e y-expression ripetute su un insieme di record, come definito da una clausola **group by**.

```
LINEST_SEB (y-expression, x-expression [, y0 [, x0 ]])
```

### LINEST\_sem

**LINEST\_SEM()** restituisce l'errore standard aggregato del valore m di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentata da coppie di numeri in x-expression e y-expression ripetute su un insieme di record, come definito da una clausola **group by**.

```
LINEST_SEM (y-expression, x-expression [, y0 [, x0 ]])
```

### LINEST\_sey

**LINEST\_SEY()** restituisce l'errore standard aggregato della stima del valore y di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentata da coppie di numeri in x-expression e y-expression ripetute su un insieme di record, come definito da una clausola **group by**.

```
LINEST_SEY (y-expression, x-expression [, y0 [, x0 ]])
```

### **LINEST\_ssreg**

**LINEST\_SSREG()** restituisce la somma di una regressione aggregata dei quadrati di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentata da coppie di numeri in x-expression e y-expression ripetute su un insieme di record, come definito da una clausola **group by**.

```
LINEST_SSREG (y-expression, x-expression [, y0 [, x0 ]])
```

### **Linest\_ssresid**

**LINEST\_SSRESID()** restituisce la somma residua aggregata dei quadrati di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentata da coppie di numeri in x-expression e y-expression ripetute su un insieme di record, come definito da una clausola **group by**.

```
LINEST_SSRESID (y-expression, x-expression [, y0 [, x0 ]])
```

### **Median**

**Median()** restituisce la mediana aggregata dei valori nell'espressione su un insieme di record, come definito da una clausola **group by**.

```
Median (expression)
```

### **Skew**

**Skew()** restituisce l'asimmetria dell'espressione su un insieme di record, come definito da una clausola **group by**.

```
Skew ([ distinct] expression)
```

### **Stdev**

**Stdev()** restituisce la deviazione standard dei valori dati dall'espressione su un insieme di record, come definito da una clausola **group by**.

```
Stdev ([distinct] expression)
```

### **Sterr**

**Sterr()** restituisce l'errore standard aggregato ( $stdev/\sqrt{n}$ ) per una serie di valori rappresentata da un'espressione ripetuta su un insieme di record, come definito da una clausola **group by**.

```
Sterr ([distinct] expression)
```

### **STEYX**

**STEYX()** restituisce l'errore standard aggregato del valore y previsto per ogni valore x nella regressione per una serie di coordinate rappresentata da coppie di numeri in x-expression e y-expression ripetute su un insieme di record, come definito da una clausola **group by**.

```
STEYX (y-expression, x-expression)
```

## Funzioni di aggregazione statistica nelle espressioni grafiche

Le seguenti funzioni di aggregazione statistica possono essere utilizzate nei grafici:

### **Avg**

**Avg()** restituisce la media aggregata dell'espressione o del campo ripetuto sulle dimensioni del grafico.

```
Avg - funzione per grafici {[SetExpression] [DISTINCT] [TOTAL [<fld{, fld}>]]} expr)
```

### Correl

**Correl()** restituisce il coefficiente di correlazione aggregato per due serie di dati. La funzione di correlazione è una misura della relazione tra le serie di dati e viene aggregata per coppie di valori (x,y) ripetute sulle dimensioni del grafico.

```
Correl - funzione per grafici {[SetExpression] [TOTAL [<fld {, fld}>]]}  
value1, value2 )
```

### Fractile

**Fractile()** trova il valore che corrisponde al frattale inclusivo (quantile) dei dati aggregati nella scala data dall'espressione ripetuta sulle dimensioni del grafico.

```
Fractile - funzione per grafici {[SetExpression] [TOTAL [<fld {, fld}>]]}  
expr, fraction)
```

### FractileExc

**FractileExc()** trova il valore che corrisponde al frattale esclusivo (quantile) dei dati aggregati nella scala data dall'espressione ripetuta sulle dimensioni del grafico.

```
FractileExc - funzione per grafici {[SetExpression] [TOTAL [<fld {, fld}>]]}  
expr, fraction)
```

### Kurtosis

**Kurtosis()** trova il kurtosis della scala di dati aggregati nell'espressione o nel campo ripetuto sulle dimensioni del grafico.

```
Kurtosis - funzione per grafici {[SetExpression] [DISTINCT] [TOTAL [<fld{, fld}>]]} expr)
```

### LINEST\_b

**LINEST\_B()** restituisce il valore b aggregato (intersezione con l'asse y) di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentate da coppie di numeri nelle espressioni date dalle espressioni **x\_value** e **y\_value**, ripetute sulle dimensioni del grafico.

```
LINEST_R2 - funzione per grafici {[SetExpression] [TOTAL [<fld{, fld}>]] }y_  
value, x_value[, y0_const[, x0_const]])
```

### LINEST\_df

**LINEST\_DF()** restituisce i gradi di libertà aggregato di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentate da coppie di numeri nelle espressioni date da **x\_value** e **y\_value**, ripetute sulle dimensioni del grafico.

```
LINEST_DF - funzione per grafici {[SetExpression] [TOTAL [<fld{, fld}>]]} y_  
value, x_value [, y0_const [, x0_const]])
```

### LINEST\_f

**LINEST\_F()** restituisce la statistica F aggregata ( $r^2/(1-r^2)$ ) di una regressione lineare definita dall'equazione  $y=mx+b$  di una serie di coordinate rappresentate da coppie di numeri nell'espressione data da **x\_value** e da **y\_value**, ripetute sulle dimensioni del grafico.

```
LINEST F - funzione per grafici ([SetExpression] [TOTAL [<fld{, fld}>]] } y_value, x_value [, y0_const [, x0_const]])
```

### LINEST\_m

**LINEST\_M()** restituisce il valore m aggregato (pendenza) di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentate da coppie di numeri date dalle espressioni **x\_value** e **y\_value**, ripetute sulle dimensioni del grafico.

```
LINEST M - funzione per grafici ([SetExpression] [TOTAL [<fld{, fld}>]] } y_value, x_value [, y0_const [, x0_const]])
```

### LINEST\_r2

**LINEST\_R2()** restituisce il valore r2 aggregato (coefficiente di determinazione) di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentate da coppie di numeri date dalle espressioni **x\_value** e **y\_value**, ripetute sulle dimensioni del grafico.

```
LINEST R2 - funzione per grafici ([SetExpression] [TOTAL [<fld{, fld}>]] } y_value, x_value [, y0_const [, x0_const]])
```

### LINEST\_seb

**LINEST\_SEB()** restituisce l'errore standard aggregato del valore b di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentate da coppie di numeri fornite dalle espressioni **x\_value** e **y\_value**, ripetute sulle dimensioni del grafico.

```
LINEST SEB - funzione per grafici ([SetExpression] [TOTAL [<fld{, fld}>]] } y_value, x_value [, y0_const [, x0_const]])
```

### LINEST\_sem

**LINEST\_SEM()** restituisce l'errore standard aggregato del valore m di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentate da coppie di numeri fornite dalle espressioni **x\_value** e **y\_value**, ripetute sulle dimensioni del grafico.

```
LINEST SEM - funzione per grafici ([set_expression] [distinct ] [total [<fld{, fld}>]] y-expression, x-expression [, y0 [, x0 ]])
```

### LINEST\_sey

**LINEST\_SEY()** restituisce l'errore standard aggregato della stima y di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentate da coppie di numeri date dalle espressioni **x\_value** e **y\_value**, ripetute sulle dimensioni del grafico.

```
LINEST SEY - funzione per grafici ([SetExpression] [TOTAL [<fld{, fld}>]] } y_value, x_value [, y0_const [, x0_const]])
```

### LINEST\_ssreg

**LINEST\_SSREG()** restituisce la somma di regressione aggregata dei quadrati di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentate da coppie di numeri date dalle espressioni **x\_value** e **y\_value**, ripetute sulle dimensioni del grafico.

```
LINEST SSREG - funzione per grafici {[SetExpression] [TOTAL [<fld{ ,fld}>]]  
{y_value, x_value[, y0_const[, x0_const]]}
```

### LINEST\_ssresid

**LINEST\_SSRESID()** restituisce la somma residua aggregata dei quadrati di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentate da coppie di numeri nelle espressioni fornite da **x\_value** e **y\_value**, ripetute sulle dimensioni del grafico.

```
LINEST SSRESID - funzione per grafici {[SetExpression] [TOTAL [<fld{ ,fld}>]]  
{y_value, x_value[, y0_const[, x0_const]]}
```

### Median

**Median()** restituisce il valore mediano della scala di valori aggregati nell'espressione ripetuti sulle dimensioni del grafico.

```
Median - funzione per grafici {[SetExpression] [TOTAL [<fld{, fld}>]]} expr)
```

### MutualInfo

**MutualInfo** calcola le informazioni reciproche (MI, Mutual Information) tra due campi o tra valori aggregati in **Aggr()**.

```
MutualInfo - funzione per grafici {[SetExpression] [DISTINCT] [TOTAL target,  
driver [, datatype [, breakdownbyvalue [, samplesize ]]]}
```

### Skew

**Skew()** restituisce l'asimmetria aggregata dell'espressione o del campo ripetuta sulle dimensioni del grafico.

```
Skew - funzione per grafici {[SetExpression] [DISTINCT] [TOTAL [<fld{  
,fld}>]]} expr)
```

### Stdev

**Stdev()** trova la deviazione standard della scala di dati aggregati nell'espressione o nel campo ripetuta sulle dimensioni del grafico.

```
Stdev - funzione per grafici {[SetExpression] [DISTINCT] [TOTAL [<fld{,  
fld}>]]} expr)
```

### Sterr

**Sterr()** trova il valore dell'errore standard della media, ( $stdev/\sqrt{n}$ ), per la serie di valori aggregati nell'espressione ripetuta sulle dimensioni del grafico.

```
Sterr - funzione per grafici {[SetExpression] [DISTINCT] [TOTAL [<fld{,  
fld}>]]} expr)
```

**STEYX**

**STEYX()** restituisce l'errore standard aggregato quando si prevedono i valori y per ciascun valore x in una regressione lineare data da una serie di coordinate rappresentate da coppie di numeri nelle espressioni date da **y\_value** e **x\_value**.

**STEYX - funzione per grafici** {[SetExpression] [TOTAL [<fld{, fld}>]]} y\_value, x\_value)

**Avg**

**Avg()** restituisce il valore medio dei dati aggregati nell'espressione su un insieme di record, come definito da una clausola **group by**.

**Sintassi:**

**Avg** ( [DISTINCT] expr)

**Tipo di dati restituiti:** numerico

**Argomenti:**

## Argomenti

| Argomento | Descrizione                                                                                          |
|-----------|------------------------------------------------------------------------------------------------------|
| expr      | L'espressione o il campo contenente i dati da misurare.                                              |
| DISTINCT  | Se la parola <b>distinct</b> è riportata prima dell'espressione, tutti i duplicati vengono ignorati. |

**Esempi e risultati:**

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

## Dati risultanti

| Esempio                                                                                                                                                                                                                                                                                                                              | Risultato                                                                                                                 |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| Temp:<br>crosstable (Month, Sales) load * inline [<br>Customer Jan Feb Mar  Apr May Jun Jul Aug Sep Oct Nov Dec<br>Astrida 46 60 70 13 78 20 45 65 78 12 78 22<br>Betacab 65 56 22 79 12 56 45 24 32 78 55 15<br>Canutility 77 68 34 91 24 68 57 36 44 90 67 27<br>Divadip 36 44 90 67 27 57 68 47 90 80 94<br>] (delimiter is ' '); | Customer<br>MyAverageSalesByCustomer<br><br>Astrida 48.916667<br><br>Betacab 44.916667<br><br>Canutility 56.916667        |
| Avg1:<br><br>LOAD Customer, Avg(Sales) as MyAverageSalesByCustomer<br>Resident Temp Group By Customer;                                                                                                                                                                                                                               | Divadip 63.083333<br>Questo può essere controllato nel foglio creando una tabella che includa la misura.<br>Sum(Sales)/12 |

| Esempio                                                                                                                                                                                          | Risultato                                                                                                                                                                                                                                        |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Presupponendo che la tabella <b>Temp</b> venga caricata come nell'esempio precedente:</p> <pre>LOAD Customer,Avg(DISTINCT Sales) as MyAvgSalesDistinct Resident Temp Group By Customer;</pre> | <p>Customer<br/>MyAverageSalesByCustomer</p> <p>Astrida 43.1</p> <p>Betacab 43.909091</p> <p>Canutility 55.909091</p> <p>Divadip 61</p> <p>Vengono contati solo i valori distinti. Dividere il totale per il numero di valori non duplicati.</p> |

## Avg - funzione per grafici

**Avg()** restituisce la media aggregata dell'espressione o del campo ripetuto sulle dimensioni del grafico.

**Sintassi:**

```
Avg ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] expr)
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

### Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| expr          | L'espressione o il campo contenente i dati da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| SetExpression | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                                                                                                                                                                                                                                 |
| DISTINCT      | Se la parola <b>DISTINCT</b> è riportata prima degli argomenti della funzione, i duplicati risultanti dalla valutazione degli argomenti della funzione vengono ignorati.                                                                                                                                                                                                                                                                                                                                                                                            |
| TOTAL         | <p>Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.</p> <p>Utilizzando <b>TOTAL [&lt;fld {, fld}&gt;]</b>, dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili.</p> |

### Limiti:

Il parametro della funzione di aggregazione non deve contenere altre funzioni di aggregazione, a meno che tali aggregazioni interne non contengano il qualificatore **TOTAL**. Nel caso di aggregazioni nidificate più complesse, utilizzare la funzione avanzata **Aggr** in combinazione con una dimensione specificata.

### Esempi e risultati:

Example table

| Customer   | Jan | Feb | Mar | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec |
|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Astrida    | 46  | 60  | 70  | 13  | 78  | 20  | 45  | 65  | 78  | 12  | 78  | 22  |
| Betacab    | 65  | 56  | 22  | 79  | 12  | 56  | 45  | 24  | 32  | 78  | 55  | 15  |
| Canutility | 77  | 68  | 34  | 91  | 24  | 68  | 57  | 36  | 44  | 90  | 67  | 27  |
| Divadip    | 57  | 36  | 44  | 90  | 67  | 27  | 57  | 68  | 47  | 90  | 80  | 94  |

Esempi di funzioni

| Esempio                | Risultato                                                                                                                                                  |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Avg(Sales)             | Per un tabella che include la dimensione Customer e la misura Avg([Sales]), se vengono mostrati i <b>Totali</b> , il risultato sarà 2566.                  |
| Avg([TOTAL (Sales)])   | 53,458333 per tutti i valori di Customer, perché il qualificatore TOTAL sta a indicare che le dimensioni vengono ignorate.                                 |
| Avg (DISTINCT (Sales)) | 51,862069 per il totale perché l'utilizzo del qualificatore Distinct sta a indicare che vengono valutati solo i valori univoci in sales per ogni Customer. |

### Dati utilizzati negli esempi:

Monthnames:

```
LOAD *, Dual(MonthText,MonthNumber) as Month INLINE [
MonthText, MonthNumber
Jan, 1
Feb, 2
Mar, 3
Apr, 4
May, 5
Jun, 6
Jul, 7
Aug, 8
Sep, 9
Oct, 10
Nov, 11
Dec, 12
];
```

Sales2013:

```
Crosstable (MonthText, Sales) LOAD * inline [
```

```
Customer|Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|Sep|Oct|Nov|Dec  
Astrida|46|60|70|13|78|20|45|65|78|12|78|22  
Betacab|65|56|22|79|12|56|45|24|32|78|55|15  
Canutility|77|68|34|91|24|68|57|36|44|90|67|27  
Divadip|57|36|44|90|67|27|57|68|47|90|80|94  
] (delimiter is '|');
```

### Vedere anche:

p [Aggr - funzione per grafici \(page 605\)](#)

### Correl

**Correl()** restituisce il coefficiente di correlazione aggregato per una serie di coordinate rappresentata da coppie di numeri in x-expression e y-expression ripetute su un insieme di record definito da una clausola **group by**.

### Sintassi:

```
Correl (value1, value2)
```

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento         | Descrizione                                                                                                                 |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------|
| value1,<br>value2 | Le espressioni o i campi contenenti i due gruppi campione per i quali deve essere misurato il coefficiente di correlazione. |

### Limiti:

Se una o entrambe le parti di una coppia di dati includono valori di testo, valori NULL e valori mancanti, l'intera coppia di dati verrà ignorata.

### Esempi e risultati:

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

### Dati risultanti

| Esempio                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Risultato                                                                                                                                                   |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>Salary:  Load *, 1 as Grp;  LOAD * inline [  "Employee name" Gender Age Salary  Aiden Charles Male 20 25000  Brenda Davies Male 25 32000  Charlotte Edberg Female 45 56000  Daroush Ferrara Male 31 29000  Eunice Goldblum Female 31 32000  Freddy Halvorsen Male 25 26000  Gauri Indu Female 36 46000  Harry Jones Male 38 40000  Ian Underwood Male 40 45000  Jackie Kingsley Female 23 28000  ] (delimiter is ' ');  Correl1: LOAD Grp, Correl(Age,Salary) as Correl_ Salary Resident Salary Group By Grp;</pre> | <p>In una tabella con dimensione corre1_salary, verrà mostrato il risultato del calcolo Correl() nello script di caricamento dei dati:</p> <p>0,9270611</p> |

### Correl - funzione per grafici

**Correl()** restituisce il coefficiente di correlazione aggregato per due serie di dati. La funzione di correlazione è una misura della relazione tra le serie di dati e viene aggregata per coppie di valori (x,y) ripetute sulle dimensioni del grafico.

#### Sintassi:

```
Correl ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] value1, value2 )
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

#### Argomenti

| Argomento      | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value1, value2 | Le espressioni o i campi contenenti i due gruppi campione per i quali deve essere misurato il coefficiente di correlazione.                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| SetExpression  | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                                                                                                                                                                                                                               |
| DISTINCT       | Se la parola <b>DISTINCT</b> è riportata prima degli argomenti della funzione, i duplicati risultanti dalla valutazione degli argomenti della funzione vengono ignorati.                                                                                                                                                                                                                                                                                                                                                                                          |
| TOTAL          | <p>Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.</p> <p>Utilizzando <b>TOTAL [&lt;fld {fld}&gt;]</b>, dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili.</p> |

**Limiti:**

Il parametro della funzione di aggregazione non deve contenere altre funzioni di aggregazione, a meno che tali aggregazioni interne non contengano il qualificatore **TOTAL**. Nel caso di aggregazioni nidificate più complesse, utilizzare la funzione avanzata **Aggr** in combinazione con una dimensione specificata.

Se una o entrambe le parti di una coppia di dati includono valori di testo, valori NULL e valori mancanti, l'intera coppia di dati verrà ignorata.

**Esempi e risultati:**

#### Esempi di funzioni

| Esempio                    | Risultato                                                                                                                                                                       |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| correl<br>(Age,<br>salary) | Per un tabella che include la dimensione Employee name e la misura correl(Age, salary), il risultato è 0,9270611. Il risultato viene visualizzato solo per la cella dei totali. |

| Esempio                                                 | Risultato                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|---------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>Correl (TOTAL Age, Salary))</code>                | 0.927. Questo risultato e quelli successivi sono visualizzati con tre posizioni decimali per favorire la leggibilità.<br><br>Se si crea una casella di filtro con la dimensione Gender e da essa si effettuano delle selezioni, si otterrà il risultato 0,951 quando viene selezionato il valore Female e 0,939 quando viene selezionato il valore Male. Ciò si verifica perché la selezione esclude tutti i risultati che non appartengono all'altro valore di Gender. |
| <code>Correl({1} TOTAL Age, Salary))</code>             | 0.927. Indipendente dalle selezioni. Ciò si verifica perché l'espressione set {1} ignora tutte le selezioni e le dimensioni.                                                                                                                                                                                                                                                                                                                                            |
| <code>Correl (TOTAL &lt;Gender&gt; Age, Salary))</code> | 0,927 nella cella del totale, 0,939 per tutti i valori di Male e 0,951 per tutti i valori di Female. Questo corrisponde ai risultati ottenuti eseguendo selezioni in una casella di filtro in base a Gender.                                                                                                                                                                                                                                                            |

Dati utilizzati negli esempi:

Salary:

```
LOAD * inline [
"Employee name"|Gender|Age|Salary
Aiden Charles|Male|20|25000
Brenda Davies|Male|25|32000
Charlotte Edberg|Female|45|56000
Daroush Ferrara|Male|31|29000
Eunice Goldblum|Female|31|32000
Freddy Halvorsen|Male|25|26000
Gauri Indu|Female|36|46000
Harry Jones|Male|38|40000
Ian Underwood|Male|40|45000
Jackie Kingsley|Female|23|28000
] (delimiter is '|');
```

**Vedere anche:**

p [Aggr - funzione per grafici \(page 605\)](#)

p [Avg - funzione per grafici \(page 457\)](#)

p [RangeCorrel \(page 1589\)](#)

## Fractile

**Fractile()** restituisce il valore corrispondente al frattale inclusivo (quantile) dei dati aggregati nell'espressione su un insieme di record, come definito da una clausola **group by**.



È possibile utilizzare [FractileExc \(page 467\)](#) per calcolare il frattale esclusivo.

### Sintassi:

```
Fractile(expr, fraction)
```

**Tipo di dati restituiti:** numerico

La funzione restituisce il valore corrispondente alla classificazione come definito da  $\text{rank} = \text{fraction} * (N-1) + 1$ , in cui  $N$  è il numero dei valori in *expr*. Se *rank* è un numero non intero, viene effettuata un'interpolazione tra i due valori più vicini.

### Argomenti:

#### Argomenti

| Argomento       | Descrizione                                                                                             |
|-----------------|---------------------------------------------------------------------------------------------------------|
| <i>expr</i>     | L'espressione o campo contenente i dati da utilizzare durante il calcolo del frattale.                  |
| <i>fraction</i> | Un numero compreso tra 0 e 1 corrispondente al frattale (quantile espresso come frazione) da calcolare. |

### Esempi e risultati:

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

| Dati risultanti                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                           |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Esempio                                                                                                                                                                                                                                                                                                                     | Risultato                                                                                                                                                                                                 |
| <pre>Table1: Crosstable (Type, value) Load recno() as ID, * inline [ Observation Comparison 35 2 40 27 12 38 15 31 21 1 14 19 46 1 10 34 28 3 48 1 16 2 30 3 32 2 48 1 31 2 22 1 12 3 39 29 19 37 25 2 ] (delimiter is ' ');  Fractile1: LOAD Type, Fractile(Value,0.75) as MyFractile Resident Table1 Group By Type;</pre> | <p>In una tabella con dimensioni Type e MyFractile, i risultati dei calcoli Fractile() nello script di caricamento dei dati sono:</p> <p>Type MyFractile</p> <p>Comparison 27.5</p> <p>Observation 36</p> |

## Fractile - funzione per grafici

**Fractile()** trova il valore che corrisponde al frattale inclusivo (quantile) dei dati aggregati nella scala data dall'espressione ripetuta sulle dimensioni del grafico.



È possibile utilizzare [FractileExc - funzione per grafici \(page 468\)](#) per calcolare il frattale esclusivo.

### Sintassi:

```
Fractile([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] expr, fraction)
```

**Tipo di dati restituiti:** numerico

La funzione restituisce il valore corrispondente alla classificazione come definito  $\text{darank} = \text{fraction} * (N-1) + 1$ , in cui  $N$  è il numero dei valori in `expr`. Se `rank` è un numero non intero, viene effettuata un'interpolazione tra i due valori più vicini.

### Argomenti:

| Argomenti     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| expr          | L'espressione o campo contenente i dati da utilizzare durante il calcolo del frattale.                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| fraction      | Un numero compreso tra 0 e 1 corrispondente al frattale (quantile espresso come frazione) da calcolare.                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| SetExpression | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                                                                                                                                                                                                                               |
| DISTINCT      | Se la parola <b>DISTINCT</b> è riportata prima degli argomenti della funzione, i duplicati risultanti dalla valutazione degli argomenti della funzione vengono ignorati.                                                                                                                                                                                                                                                                                                                                                                                          |
| TOTAL         | <p>Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.</p> <p>Utilizzando <b>TOTAL [&lt;fld {fld}&gt;]</b>, dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili.</p> |

### Limiti:

Il parametro della funzione di aggregazione non deve contenere altre funzioni di aggregazione, a meno che tali aggregazioni interne non contengano il qualificatore **TOTAL**. Nel caso di aggregazioni nidificate più complesse, utilizzare la funzione avanzata **Aggr** in combinazione con una dimensione specificata.

### Esempi e risultati:

| Example table |     |     |     |     |     |     |     |     |     |     |     |     |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Customer      | Jan | Feb | Mar | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec |
| Astrida       | 46  | 60  | 70  | 13  | 78  | 20  | 45  | 65  | 78  | 12  | 78  | 22  |
| Betacab       | 65  | 56  | 22  | 79  | 12  | 56  | 45  | 24  | 32  | 78  | 55  | 15  |
| Canutility    | 77  | 68  | 34  | 91  | 24  | 68  | 57  | 36  | 44  | 90  | 67  | 27  |
| Divadip       | 57  | 36  | 44  | 90  | 67  | 27  | 57  | 68  | 47  | 90  | 80  | 94  |

### Esempi di funzioni

| Esempio                                  | Risultato                                                                                                                                                                                                                                          |
|------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Fractile<br>(Sales,<br>0.75)             | Per un tabella che include la dimensione Customer e la misura Fractile([Sales]), se vengono mostrati i <b>Totali</b> , il risultato sarà 71,75. Questo è il punto nella distribuzione dei valori di Sales sotto al quale ricade il 75% dei valori. |
| Fractile<br>(TOTAL<br>Sales,<br>0.75))   | 71,75 per tutti i valori di Customer, perché il qualificatore TOTAL sta a indicare che le dimensioni vengono ignorate.                                                                                                                             |
| Fractile<br>(DISTINCT<br>Sales,<br>0.75) | 70 per il totale perché l'utilizzo del qualificatore DISTINCT sta a indicare che vengono valutati solo i valori univoci in Sales per ogni Customer.                                                                                                |

Dati utilizzati negli esempi:

Monthnames:

```
LOAD *, Dual(MonthText,MonthNumber) as Month INLINE [
MonthText, MonthNumber
Jan, 1
Feb, 2
Mar, 3
Apr, 4
May, 5
Jun, 6
Jul, 7
Aug, 8
Sep, 9
Oct, 10
Nov, 11
Dec, 12
];
```

Sales2013:

```
Crosstable (MonthText, Sales) LOAD * inline [
Customer|Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|Sep|Oct|Nov|Dec
Astrida|46|60|70|13|78|20|45|65|78|12|78|22
Betacab|65|56|22|79|12|56|45|24|32|78|55|15
Canutility|77|68|34|91|24|68|57|36|44|90|67|27
Divadip|57|36|44|90|67|27|57|68|47|90|80|94
] (delimiter is '|');
```

**Vedere anche:**

p [Aggr - funzione per grafici \(page 605\)](#)

## FractileExc

**FractileExc()** restituisce il valore corrispondente al frattale esclusivo (quantile) dei dati aggregati nell'espressione su un insieme di record, come definito da una clausola **group by**.



È possibile utilizzare [Fractile \(page 463\)](#) per calcolare il frattale inclusivo.

### Sintassi:

```
FractileExc(expr, fraction)
```

**Tipo di dati restituiti:** numerico

La funzione restituisce il valore corrispondente alla classificazione come definito da  $\text{rank} = \text{fraction} * (N+1)$ , in cui  $N$  è il numero dei valori in *expr*. Se *rank* è un numero non intero, viene effettuata un'interpolazione tra i due valori più vicini.

### Argomenti:

#### Argomenti

| Argomento | Descrizione                                                                                             |
|-----------|---------------------------------------------------------------------------------------------------------|
| expr      | L'espressione o campo contenente i dati da utilizzare durante il calcolo del frattale.                  |
| fraction  | Un numero compreso tra 0 e 1 corrispondente al frattale (quantile espresso come frazione) da calcolare. |

### Esempi e risultati:

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

| Dati risultanti                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                              |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Esempio                                                                                                                                                                                                                                                                                                                        | Risultato                                                                                                                                                                                                    |
| <pre>Table1: Crosstable (Type, value) Load recno() as ID, * inline [ Observation Comparison 35 2 40 27 12 38 15 31 21 1 14 19 46 1 10 34 28 3 48 1 16 2 30 3 32 2 48 1 31 2 22 1 12 3 39 29 19 37 25 2 ] (delimiter is ' ');  Fractile1: LOAD Type, FractileExc(Value,0.75) as MyFractile Resident Table1 Group By Type;</pre> | <p>In una tabella con dimensioni Type e MyFractile, i risultati dei calcoli FractileExc() nello script di caricamento dei dati sono:</p> <p>Type MyFractile</p> <p>Comparison 28.5</p> <p>Observation 38</p> |

## FractileExc - funzione per grafici

**FractileExc()** trova il valore che corrisponde al frattale esclusivo (quantile) dei dati aggregati nella scala data dall'espressione ripetuta sulle dimensioni del grafico.



È possibile utilizzare [Fractile - funzione per grafici \(page 464\)](#) per calcolare il frattale inclusivo.

### Sintassi:

```
FractileExc ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] expr,
fraction)
```

### Tipo di dati restituiti: numerico

La funzione restituisce il valore corrispondente alla classificazione come definito  $\text{darank} = \text{fraction} * (N+1)$ , in cui  $N$  è il numero dei valori in `expr`. Se `rank` è un numero non intero, viene effettuata un'interpolazione tra i due valori più vicini.

### Argomenti:

| Argomenti     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| expr          | L'espressione o campo contenente i dati da utilizzare durante il calcolo del frattale.                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| fraction      | Un numero compreso tra 0 e 1 corrispondente al frattale (quantile espresso come frazione) da calcolare.                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| SetExpression | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                                                                                                                                                                                                                               |
| DISTINCT      | Se la parola <b>DISTINCT</b> è riportata prima degli argomenti della funzione, i duplicati risultanti dalla valutazione degli argomenti della funzione vengono ignorati.                                                                                                                                                                                                                                                                                                                                                                                          |
| TOTAL         | <p>Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.</p> <p>Utilizzando <b>TOTAL [&lt;fld {fld}&gt;]</b>, dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili.</p> |

### Limiti:

Il parametro della funzione di aggregazione non deve contenere altre funzioni di aggregazione, a meno che tali aggregazioni interne non contengano il qualificatore **TOTAL**. Nel caso di aggregazioni nidificate più complesse, utilizzare la funzione avanzata **Aggr** in combinazione con una dimensione specificata.

### Esempi e risultati:

| Example table |     |     |     |     |     |     |     |     |     |     |     |     |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Customer      | Jan | Feb | Mar | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec |
| Astrida       | 46  | 60  | 70  | 13  | 78  | 20  | 45  | 65  | 78  | 12  | 78  | 22  |
| Betacab       | 65  | 56  | 22  | 79  | 12  | 56  | 45  | 24  | 32  | 78  | 55  | 15  |
| Canutility    | 77  | 68  | 34  | 91  | 24  | 68  | 57  | 36  | 44  | 90  | 67  | 27  |
| Divadip       | 57  | 36  | 44  | 90  | 67  | 27  | 57  | 68  | 47  | 90  | 80  | 94  |

### Esempi di funzioni

| Esempio                                  | Risultato                                                                                                                                                                                                                                             |
|------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FractileExc<br>(Sales, 0.75)             | Per un tabella che include la dimensione Customer e la misura FractileExc([Sales]), se vengono mostrati i <b>Totali</b> , il risultato sarà 75.25. Questo è il punto nella distribuzione dei valori di Sales sotto al quale ricade il 75% dei valori. |
| FractileExc<br>(TOTAL Sales,<br>0.75))   | 75,25 per tutti i valori di Customer, perché il qualificatore TOTAL sta a indicare che le dimensioni vengono ignorate.                                                                                                                                |
| FractileExc<br>(DISTINCT<br>Sales, 0.75) | 73,50 per il totale perché l'utilizzo del qualificatore DISTINCT sta a indicare che vengono valutati solo i valori univoci in Sales per ogni Customer.                                                                                                |

Dati utilizzati negli esempi:

Monthnames:

```
LOAD *, Dual(MonthText,MonthNumber) as Month INLINE [
MonthText, MonthNumber
Jan, 1
Feb, 2
Mar, 3
Apr, 4
May, 5
Jun, 6
Jul, 7
Aug, 8
Sep, 9
Oct, 10
Nov, 11
Dec, 12
];
```

Sales2013:

```
Crosstable (MonthText, Sales) LOAD * inline [
Customer|Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|Sep|Oct|Nov|Dec
Astrida|46|60|70|13|78|20|45|65|78|12|78|22
Betacab|65|56|22|79|12|56|45|24|32|78|55|15
Canutility|77|68|34|91|24|68|57|36|44|90|67|27
Divadip|57|36|44|90|67|27|57|68|47|90|80|94
] (delimiter is '|');
```

**Vedere anche:**

p [Aggr - funzione per grafici \(page 605\)](#)

### Kurtosis

**Kurtosis()** restituisce il kurtosis dei dati nell'espressione su un insieme di record, come definito da una clausola **group by**.

### Sintassi:

```
Kurtosis([distinct ] expr )
```

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento | Descrizione                                                                                          |
|-----------|------------------------------------------------------------------------------------------------------|
| expr      | L'espressione o il campo contenente i dati da misurare.                                              |
| distinct  | Se la parola <b>distinct</b> è riportata prima dell'espressione, tutti i duplicati vengono ignorati. |

### Esempi e risultati:

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

Dati risultanti

| Esempio                                                                                                                                                                                                                                                                                                                                                          | Risultato                                                                                                                                                                                                                                                       |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>Table1: Crosstable (Type, value) Load recno() as ID, * inline [ Observation Comparison 35 2 40 27 12 38 15 31 21 1 14 19 46 1 10 34 28 3 48 1 16 2 30 3 32 2 48 1 31 2 22 1 12 3 39 29 19 37 25 2 ] (delimiter is ' ');  Kurtosis1: LOAD Type, Kurtosis(Value) as MyKurtosis1, Kurtosis(DISTINCT value) as MyKurtosis2 Resident Table1 Group By Type;</pre> | <p>In una tabella con dimensioni Type, MyKurtosis1 e MyKurtosis2, i risultati dei calcoli Kurtosis() nello script di caricamento dei dati sono:</p> <pre>Type MyKurtosis1 MyKurtosis2 Comparison -1.1612957 -1.4982366 Observation -1.1148768 -0.93540144</pre> |

## Kurtosis - funzione per grafici

**Kurtosis()** trova il kurtosis della scala di dati aggregati nell'espressione o nel campo ripetuto sulle dimensioni del grafico.

### Sintassi:

```
Kurtosis ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] expr)
```

**Tipo di dati restituiti:** numerico

### Argomenti:

Argomenti

| Argomento | Descrizione                                             |
|-----------|---------------------------------------------------------|
| expr      | L'espressione o il campo contenente i dati da misurare. |

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SetExpression | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                                                                                                                                                                                                                               |
| DISTINCT      | Se la parola <b>DISTINCT</b> è riportata prima degli argomenti della funzione, i duplicati risultanti dalla valutazione degli argomenti della funzione vengono ignorati.                                                                                                                                                                                                                                                                                                                                                                                          |
| TOTAL         | <p>Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.</p> <p>Utilizzando <b>TOTAL [&lt;fld {fld}&gt;]</b>, dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili.</p> |

**Limiti:**

Il parametro della funzione di aggregazione non deve contenere altre funzioni di aggregazione, a meno che tali aggregazioni interne non contengano il qualificatore **TOTAL**. Nel caso di aggregazioni nidificate più complesse, utilizzare la funzione avanzata **Aggr** in combinazione con una dimensione specificata.

**Esempi e risultati:**

Example table

| Type        | Value |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------------|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Comparison  | 2     | 27 | 38 | 31 | 1  | 19 | 1  | 34 | 3  | 1  | 2  | 3  | 2  | 1  | 2  | 1  | 3  | 29 | 37 | 2  |
| Observation | 35    | 40 | 12 | 15 | 21 | 14 | 46 | 10 | 28 | 48 | 16 | 30 | 32 | 48 | 31 | 22 | 12 | 39 | 19 | 25 |

Esempi di funzioni

| Esempio                | Risultato                                                                                                                                                                                                                                                                      |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| kurtosis (value)       | Per una tabella che include la dimensione type e la misura kurtosis(value), se vengono visualizzati i <b>Totali</b> per la tabella e la formattazione del numero è impostata su 3 cifre significative, il risultato è 1,252. Per comparison è 1,161 e per observation è 1,115. |
| kurtosis (TOTAL value) | 1,252 per tutti i valori di type, perché il qualificatore TOTAL sta a indicare che le dimensioni vengono ignorate.                                                                                                                                                             |

Dati utilizzati negli esempi:

Table1:  
Crosstable (Type, value)

```
Load recno() as ID, * inline [
Observation|Comparison
35|2
40|27
12|38
15|31
21|1
14|19
46|1
10|34
28|3
48|1
16|2
30|3
32|2
48|1
31|2
22|1
12|3
39|29
19|37
25|2 ] (delimiter is '|');
```

### Vedere anche:

p [Avg - funzione per grafici \(page 457\)](#)

## LINEST\_B

**LINEST\_B()** restituisce il valore b aggregato (intercettazione sull'asse y) di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentata da coppie di numeri in x-expression e y-expression ripetute su un insieme di record, come definito da una clausola **group by**.

### Sintassi:

```
LINEST_B (y_value, x_value[, y0 [, x0 ]])
```

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento | Descrizione                                                           |
|-----------|-----------------------------------------------------------------------|
| y_value   | L'espressione o il campo contenente la scala di valori y da misurare. |
| x_value   | L'espressione o il campo contenente la scala di valori x da misurare. |

| Argomento  | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| y(0), x(0) | <p>È possibile dichiarare un valore facoltativo y0 forzando il passaggio della linea di regressione attraverso l'asse delle y in un determinato punto. Dichiarando sia y0 che x0, è possibile forzare il passaggio della linea di regressione attraverso una coordinata singola fissa.</p> <p>A meno che non vengano dichiarati sia y0 che x0, la funzione richiede almeno due coppie di dati valide per il calcolo. Se vengono dichiarati i valori y0 e x0, sarà sufficiente una singola coppia di dati.</p> |

**Limiti:**

Se una o entrambe le parti di una coppia di dati includono valori di testo, valori NULL e valori mancanti, l'intera coppia di dati verrà ignorata.

**Vedere anche:**

p [Esempi di utilizzo delle funzioni linest \(page 517\)](#)

**LINEST\_B - funzione per grafici**

**LINEST\_B()** restituisce il valore b aggregato (intersezione con l'asse y) di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentate da coppie di numeri nelle espressioni date dalle espressioni **x\_value** e **y\_value**, ripetute sulle dimensioni del grafico.

**Sintassi:**

```
LINEST_B ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value, x_value
[, y0_const [ , x0_const]])
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

## Argomenti

| Argomento | Descrizione                                                           |
|-----------|-----------------------------------------------------------------------|
| y_value   | L'espressione o il campo contenente la scala di valori y da misurare. |
| x_value   | L'espressione o il campo contenente la scala di valori x da misurare. |

| Argomento          | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| y0_const, x0_const | <p>È possibile dichiarare un valore facoltativo y0 forzando il passaggio della linea di regressione attraverso l'asse delle y in un determinato punto. Dichiarando sia y0 che x0, è possibile forzare il passaggio della linea di regressione attraverso una coordinata singola fissa.</p> <div>  <p><i>A meno che non vengano dichiarati sia y0 che x0, la funzione richiede almeno due coppie di dati valide per il calcolo. Se vengono dichiarati i valori y0 e x0, sarà sufficiente una singola coppia di dati.</i></p> </div> |
| SetExpression      | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                                                                                                                                                                                                                                                                                 |
| DISTINCT           | Se la parola <b>DISTINCT</b> è riportata prima degli argomenti della funzione, i duplicati risultanti dalla valutazione degli argomenti della funzione vengono ignorati.                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| TOTAL              | <p>Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.</p> <p>Utilizzando <b>TOTAL [&lt;fld {fld}&gt;]</b>, dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili.</p>                                                   |

**Limiti:**

Il parametro della funzione di aggregazione non deve contenere altre funzioni di aggregazione, a meno che tali aggregazioni interne non contengano il qualificatore **TOTAL**. Nel caso di aggregazioni nidificate più complesse, utilizzare la funzione avanzata **Aggr** in combinazione con una dimensione specificata.

Se una o entrambe le parti di una coppia di dati includono valori di testo, valori NULL e valori mancanti, l'intera coppia di dati verrà ignorata.

**Vedere anche:**

- p [Esempi di utilizzo delle funzioni linest \(page 517\)](#)
- p [Avg - funzione per grafici \(page 457\)](#)

**LINEST\_DF**

**LINEST\_DF()** restituisce i gradi di libertà aggregati di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentata da coppie di numeri in x-expression e y-expression ripetute su un insieme di record, come definito da una clausola **group by**.

### Sintassi:

```
LINEST_DF (y_value, x_value[, y0 [, x0 ]])
```

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento  | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| y_value    | L'espressione o il campo contenente la scala di valori y da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| x_value    | L'espressione o il campo contenente la scala di valori x da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| y(0), x(0) | <p>È possibile dichiarare un valore facoltativo y0 forzando il passaggio della linea di regressione attraverso l'asse delle y in un determinato punto. Dichiarando sia y0 che x0, è possibile forzare il passaggio della linea di regressione attraverso una coordinata singola fissa.</p> <p>A meno che non vengano dichiarati sia y0 che x0, la funzione richiede almeno due coppie di dati valide per il calcolo. Se vengono dichiarati i valori y0 e x0, sarà sufficiente una singola coppia di dati.</p> |

### Limiti:

Se una o entrambe le parti di una coppia di dati includono valori di testo, valori NULL e valori mancanti, l'intera coppia di dati verrà ignorata.

### Vedere anche:

p [Esempi di utilizzo delle funzioni linest \(page 517\)](#)

## LINEST\_DF - funzione per grafici

**LINEST\_DF()** restituisce i gradi di libertà aggregato di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentate da coppie di numeri nelle espressioni date da **x\_value** e **y\_value**, ripetute sulle dimensioni del grafico.

### Sintassi:

```
LINEST_DF ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value, x_value [, y0_const [, x0_const]])
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| y_value       | L'espressione o il campo contenente la scala di valori y da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| x_value       | L'espressione o il campo contenente la scala di valori x da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| y0, x0        | <p>È possibile dichiarare un valore facoltativo y0 forzando il passaggio della linea di regressione attraverso l'asse delle y in un determinato punto. Dichiarando sia y0 che x0, è possibile forzare il passaggio della linea di regressione attraverso una coordinata singola fissa.</p> <div>  <p><i>A meno che non vengano dichiarati sia y0 che x0, la funzione richiede almeno due coppie di dati valide per il calcolo. Se vengono dichiarati i valori y0 e x0, sarà sufficiente una singola coppia di dati.</i></p> </div> |
| SetExpression | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                                                                                                                                                                                                                                                                                 |
| DISTINCT      | Se la parola <b>DISTINCT</b> è riportata prima degli argomenti della funzione, i duplicati risultanti dalla valutazione degli argomenti della funzione vengono ignorati.                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| TOTAL         | <p>Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.</p> <p>Utilizzando <b>TOTAL [&lt;fld {fld}&gt;]</b>, dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili.</p>                                                   |

**Limiti:**

Il parametro della funzione di aggregazione non deve contenere altre funzioni di aggregazione, a meno che tali aggregazioni interne non contengano il qualificatore **TOTAL**. Nel caso di aggregazioni nidificate più complesse, utilizzare la funzione avanzata **Aggr** in combinazione con una dimensione specificata.

Se una o entrambe le parti di una coppia di dati includono valori di testo, valori NULL e valori mancanti, l'intera coppia di dati verrà ignorata.

**Vedere anche:**

p [Esempi di utilizzo delle funzioni linest \(page 517\)](#)

[p Avg - funzione per grafici \(page 457\)](#)

## LINEST\_F

Questa funzione di script restituisce la statistica F aggregata ( $r^2/(1-r^2)$ ) di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentata da coppie di numeri in x-expression e y-expression ripetute su un insieme di record definito da una clausola **group by**.

### Sintassi:

```
LINEST_F (y_value, x_value[, y0 [, x0 ]])
```

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento  | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| y_value    | L'espressione o il campo contenente la scala di valori y da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| x_value    | L'espressione o il campo contenente la scala di valori x da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| y(0), x(0) | È possibile dichiarare un valore facoltativo y0 forzando il passaggio della linea di regressione attraverso l'asse delle y in un determinato punto. Dichiarando sia y0 che x0, è possibile forzare il passaggio della linea di regressione attraverso una coordinata singola fissa.<br><br>A meno che non vengano dichiarati sia y0 che x0, la funzione richiede almeno due coppie di dati valide per il calcolo. Se vengono dichiarati i valori y0 e x0, sarà sufficiente una singola coppia di dati. |

### Limiti:

Se una o entrambe le parti di una coppia di dati includono valori di testo, valori NULL e valori mancanti, l'intera coppia di dati verrà ignorata.

### Vedere anche:

[p Esempi di utilizzo delle funzioni linest \(page 517\)](#)

## LINEST\_F - funzione per grafici

**LINEST\_F()** restituisce la statistica F aggregata ( $r^2/(1-r^2)$ ) di una regressione lineare definita dall'equazione  $y=mx+b$  di una serie di coordinate rappresentate da coppie di numeri nell'espressione data da **x\_value** e da **y\_value**, ripetute sulle dimensioni del grafico.

### Sintassi:

```
LINEST_F ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value, x_value  
[, y0_const [, x0_const]])
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| y_value       | L'espressione o il campo contenente la scala di valori y da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| x_value       | L'espressione o il campo contenente la scala di valori x da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| y0, x0        | <p>È possibile dichiarare un valore facoltativo y0 forzando il passaggio della linea di regressione attraverso l'asse delle y in un determinato punto. Dichiarando sia y0 che x0, è possibile forzare il passaggio della linea di regressione attraverso una coordinata singola fissa.</p> <div>  <p><i>A meno che non vengano dichiarati sia y0 che x0, la funzione richiede almeno due coppie di dati valide per il calcolo. Se vengono dichiarati i valori y0 e x0, sarà sufficiente una singola coppia di dati.</i></p> </div> |
| SetExpression | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                                                                                                                                                                                                                                                                                 |
| DISTINCT      | Se la parola <b>DISTINCT</b> è riportata prima degli argomenti della funzione, i duplicati risultanti dalla valutazione degli argomenti della funzione vengono ignorati.                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| TOTAL         | <p>Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.</p> <p>Utilizzando <b>TOTAL [&lt;fld {fld}&gt;]</b>, dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili.</p>                                                   |

**Limiti:**

Il parametro della funzione di aggregazione non deve contenere altre funzioni di aggregazione, a meno che tali aggregazioni interne non contengano il qualificatore **TOTAL**. Nel caso di aggregazioni nidificate più complesse, utilizzare la funzione avanzata **Aggr** in combinazione con una dimensione specificata.

Se una o entrambe le parti di una coppia di dati includono valori di testo, valori NULL e valori mancanti, l'intera coppia di dati verrà ignorata.

**Vedere anche:**

p [Esempi di utilizzo delle funzioni linest \(page 517\)](#)

[p Avg - funzione per grafici \(page 457\)](#)

## LINEST\_M

**LINEST\_M()** restituisce il valore m aggregato (pendenza) di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentata da coppie di numeri in x-expression e y-expression ripetute su un insieme di record, come definito da una clausola **group by**.

### Sintassi:

```
LINEST_M (y_value, x_value[, y0 [, x0 ]])
```

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento  | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| y_value    | L'espressione o il campo contenente la scala di valori y da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| x_value    | L'espressione o il campo contenente la scala di valori x da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| y(0), x(0) | È possibile dichiarare un valore facoltativo y0 forzando il passaggio della linea di regressione attraverso l'asse delle y in un determinato punto. Dichiarando sia y0 che x0, è possibile forzare il passaggio della linea di regressione attraverso una coordinata singola fissa.<br><br>A meno che non vengano dichiarati sia y0 che x0, la funzione richiede almeno due coppie di dati valide per il calcolo. Se vengono dichiarati i valori y0 e x0, sarà sufficiente una singola coppia di dati. |

### Limiti:

Se una o entrambe le parti di una coppia di dati includono valori di testo, valori NULL e valori mancanti, l'intera coppia di dati verrà ignorata.

### Vedere anche:

[p Esempi di utilizzo delle funzioni linest \(page 517\)](#)

## LINEST\_M - funzione per grafici

**LINEST\_M()** restituisce il valore m aggregato (pendenza) di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentate da coppie di numeri date dalle espressioni **x\_value** e **y\_value**, ripetute sulle dimensioni del grafico.

### Sintassi:

```
LINEST_M ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value, x_value  
[, y0_const [, x0_const]])
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| y_value       | L'espressione o il campo contenente la scala di valori y da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| x_value       | L'espressione o il campo contenente la scala di valori x da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| y0, x0        | <p>È possibile dichiarare un valore facoltativo y0 forzando il passaggio della linea di regressione attraverso l'asse delle y in un determinato punto. Dichiarando sia y0 che x0, è possibile forzare il passaggio della linea di regressione attraverso una coordinata singola fissa.</p> <div>  <p><i>A meno che non vengano dichiarati sia y0 che x0, la funzione richiede almeno due coppie di dati valide per il calcolo. Se vengono dichiarati i valori y0 e x0, sarà sufficiente una singola coppia di dati.</i></p> </div> |
| SetExpression | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                                                                                                                                                                                                                                                                                 |
| DISTINCT      | Se la parola <b>DISTINCT</b> è riportata prima degli argomenti della funzione, i duplicati risultanti dalla valutazione degli argomenti della funzione vengono ignorati.                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| TOTAL         | <p>Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.</p> <p>Utilizzando <b>TOTAL [&lt;fld {fld}&gt;]</b>, dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili.</p>                                                   |

**Limiti:**

Il parametro della funzione di aggregazione non deve contenere altre funzioni di aggregazione, a meno che tali aggregazioni interne non contengano il qualificatore **TOTAL**. Nel caso di aggregazioni nidificate più complesse, utilizzare la funzione avanzata **Aggr** in combinazione con una dimensione specificata.

Se una o entrambe le parti di una coppia di dati includono valori di testo, valori NULL e valori mancanti, l'intera coppia di dati verrà ignorata.

**Vedere anche:**

p [Esempi di utilizzo delle funzioni lineat \(page 517\)](#)

[p Avg - funzione per grafici \(page 457\)](#)

## LINEST\_R2

**LINEST\_R2()** restituisce il valore  $r^2$  aggregato (coefficiente di determinazione) di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentata da coppie di numeri in x-expression e y-expression ripetute su un insieme di record, come definito da una clausola **group by**.

### Sintassi:

```
LINEST_R2 (y_value, x_value[, y0 [, x0 ]])
```

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento  | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| y_value    | L'espressione o il campo contenente la scala di valori y da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| x_value    | L'espressione o il campo contenente la scala di valori x da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| y(0), x(0) | È possibile dichiarare un valore facoltativo y0 forzando il passaggio della linea di regressione attraverso l'asse delle y in un determinato punto. Dichiarando sia y0 che x0, è possibile forzare il passaggio della linea di regressione attraverso una coordinata singola fissa.<br><br>A meno che non vengano dichiarati sia y0 che x0, la funzione richiede almeno due coppie di dati valide per il calcolo. Se vengono dichiarati i valori y0 e x0, sarà sufficiente una singola coppia di dati. |

### Limiti:

Se una o entrambe le parti di una coppia di dati includono valori di testo, valori NULL e valori mancanti, l'intera coppia di dati verrà ignorata.

### Vedere anche:

[p Esempi di utilizzo delle funzioni linest \(page 517\)](#)

## LINEST\_R2 - funzione per grafici

**LINEST\_R2()** restituisce il valore  $r^2$  aggregato (coefficiente di determinazione) di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentate da coppie di numeri date dalle espressioni **x\_value** e **y\_value**, ripetute sulle dimensioni del grafico.

### Sintassi:

```
LINEST_R2 ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value, x_value[, y0_const[, x0_const]])
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| y_value       | L'espressione o il campo contenente la scala di valori y da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| x_value       | L'espressione o il campo contenente la scala di valori x da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| y0, x0        | <p>È possibile dichiarare un valore facoltativo y0 forzando il passaggio della linea di regressione attraverso l'asse delle y in un determinato punto. Dichiarando sia y0 che x0, è possibile forzare il passaggio della linea di regressione attraverso una coordinata singola fissa.</p> <div>  <p><i>A meno che non vengano dichiarati sia y0 che x0, la funzione richiede almeno due coppie di dati valide per il calcolo. Se vengono dichiarati i valori y0 e x0, sarà sufficiente una singola coppia di dati.</i></p> </div> |
| SetExpression | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                                                                                                                                                                                                                                                                                 |
| DISTINCT      | Se la parola <b>DISTINCT</b> è riportata prima degli argomenti della funzione, i duplicati risultanti dalla valutazione degli argomenti della funzione vengono ignorati.                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| TOTAL         | <p>Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.</p> <p>Utilizzando <b>TOTAL [&lt;fld {fld}&gt;]</b>, dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili.</p>                                                   |

**Limiti:**

Il parametro della funzione di aggregazione non deve contenere altre funzioni di aggregazione, a meno che tali aggregazioni interne non contengano il qualificatore **TOTAL**. Nel caso di aggregazioni nidificate più complesse, utilizzare la funzione avanzata **Aggr** in combinazione con una dimensione specificata.

Se una o entrambe le parti di una coppia di dati includono valori di testo, valori NULL e valori mancanti, l'intera coppia di dati verrà ignorata.

**Vedere anche:**

p [Esempi di utilizzo delle funzioni lineat \(page 517\)](#)

[p Avg - funzione per grafici \(page 457\)](#)

## LINEST\_SEB

**LINEST\_SEB()** restituisce l'errore standard aggregato del valore b di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentata da coppie di numeri in x-expression e y-expression ripetute su un insieme di record, come definito da una clausola **group by**.

### Sintassi:

```
LINEST_SEB (y_value, x_value[, y0 [, x0 ]])
```

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento  | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| y_value    | L'espressione o il campo contenente la scala di valori y da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| x_value    | L'espressione o il campo contenente la scala di valori x da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| y(0), x(0) | È possibile dichiarare un valore facoltativo y0 forzando il passaggio della linea di regressione attraverso l'asse delle y in un determinato punto. Dichiarando sia y0 che x0, è possibile forzare il passaggio della linea di regressione attraverso una coordinata singola fissa.<br><br>A meno che non vengano dichiarati sia y0 che x0, la funzione richiede almeno due coppie di dati valide per il calcolo. Se vengono dichiarati i valori y0 e x0, sarà sufficiente una singola coppia di dati. |

### Limiti:

Se una o entrambe le parti di una coppia di dati includono valori di testo, valori NULL e valori mancanti, l'intera coppia di dati verrà ignorata.

### Vedere anche:

[p Esempi di utilizzo delle funzioni linest \(page 517\)](#)

## LINEST\_SEB - funzione per grafici

**LINEST\_SEB()** restituisce l'errore standard aggregato del valore b di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentate da coppie di numeri fornite dalle espressioni **x\_value** e **y\_value**, ripetute sulle dimensioni del grafico.

### Sintassi:

```
LINEST_SEB ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value, x_value[, y0_const[, x0_const]])
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| y_value       | L'espressione o il campo contenente la scala di valori y da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| x_value       | L'espressione o il campo contenente la scala di valori x da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| y0, x0        | <p>È possibile dichiarare un valore facoltativo y0 forzando il passaggio della linea di regressione attraverso l'asse delle y in un determinato punto. Dichiarando sia y0 che x0, è possibile forzare il passaggio della linea di regressione attraverso una coordinata singola fissa.</p> <div>  <p><i>A meno che non vengano dichiarati sia y0 che x0, la funzione richiede almeno due coppie di dati valide per il calcolo. Se vengono dichiarati i valori y0 e x0, sarà sufficiente una singola coppia di dati.</i></p> </div> |
| SetExpression | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                                                                                                                                                                                                                                                                                 |
| DISTINCT      | Se la parola <b>DISTINCT</b> è riportata prima degli argomenti della funzione, i duplicati risultanti dalla valutazione degli argomenti della funzione vengono ignorati.                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| TOTAL         | <p>Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.</p> <p>Utilizzando <b>TOTAL [&lt;fld {fld}&gt;]</b>, dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili.</p>                                                   |

**Limiti:**

Il parametro della funzione di aggregazione non deve contenere altre funzioni di aggregazione, a meno che tali aggregazioni interne non contengano il qualificatore **TOTAL**. Nel caso di aggregazioni nidificate più complesse, utilizzare la funzione avanzata **Aggr** in combinazione con una dimensione specificata.

Se una o entrambe le parti di una coppia di dati includono valori di testo, valori NULL e valori mancanti, l'intera coppia di dati verrà ignorata.

**Vedere anche:**

p [Esempi di utilizzo delle funzioni linest \(page 517\)](#)

[p Avg - funzione per grafici \(page 457\)](#)

## LINEST\_SEM

**LINEST\_SEM()** restituisce l'errore standard aggregato del valore  $m$  di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentata da coppie di numeri in  $x$ -expression e  $y$ -expression ripetute su un insieme di record, come definito da una clausola **group by**.

### Sintassi:

```
LINEST_SEM (y_value, x_value[, y0 [, x0 ]])
```

**Tipo di dati restituiti:** numerico

### Argomenti:

| Argomento  | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| y_value    | L'espressione o il campo contenente la scala di valori y da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| x_value    | L'espressione o il campo contenente la scala di valori x da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| y(0), x(0) | È possibile dichiarare un valore facoltativo y0 forzando il passaggio della linea di regressione attraverso l'asse delle y in un determinato punto. Dichiarando sia y0 che x0, è possibile forzare il passaggio della linea di regressione attraverso una coordinata singola fissa.<br><br>A meno che non vengano dichiarati sia y0 che x0, la funzione richiede almeno due coppie di dati valide per il calcolo. Se vengono dichiarati i valori y0 e x0, sarà sufficiente una singola coppia di dati. |

### Limiti:

Se una o entrambe le parti di una coppia di dati includono valori di testo, valori NULL e valori mancanti, l'intera coppia di dati verrà ignorata.

### Vedere anche:

[p Esempi di utilizzo delle funzioni linest \(page 517\)](#)

## LINEST\_SEM - funzione per grafici

**LINEST\_SEM()** restituisce l'errore standard aggregato del valore  $m$  di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentate da coppie di numeri fornite dalle espressioni **x\_value** e **y\_value**, ripetute sulle dimensioni del grafico.

### Sintassi:

```
LINEST_SEM ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value, x_value[, y0_const[, x0_const]])
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| y_value       | L'espressione o il campo contenente la scala di valori y da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| x_value       | L'espressione o il campo contenente la scala di valori x da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| y0, x0        | <p>È possibile dichiarare un valore facoltativo y0 forzando il passaggio della linea di regressione attraverso l'asse delle y in un determinato punto. Dichiarando sia y0 che x0, è possibile forzare il passaggio della linea di regressione attraverso una coordinata singola fissa.</p> <div>  <p><i>A meno che non vengano dichiarati sia y0 che x0, la funzione richiede almeno due coppie di dati valide per il calcolo. Se vengono dichiarati i valori y0 e x0, sarà sufficiente una singola coppia di dati.</i></p> </div> |
| SetExpression | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                                                                                                                                                                                                                                                                                 |
| DISTINCT      | Se la parola <b>DISTINCT</b> è riportata prima degli argomenti della funzione, i duplicati risultanti dalla valutazione degli argomenti della funzione vengono ignorati.                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| TOTAL         | <p>Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.</p> <p>Utilizzando <b>TOTAL [&lt;fld {fld}&gt;]</b>, dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili.</p>                                                   |

**Limiti:**

Il parametro della funzione di aggregazione non deve contenere altre funzioni di aggregazione, a meno che tali aggregazioni interne non contengano il qualificatore **TOTAL**. Nel caso di aggregazioni nidificate più complesse, utilizzare la funzione avanzata **Aggr** in combinazione con una dimensione specificata.

Se una o entrambe le parti di una coppia di dati includono valori di testo, valori NULL e valori mancanti, l'intera coppia di dati verrà ignorata.

**Vedere anche:**

p [Esempi di utilizzo delle funzioni lineat \(page 517\)](#)

[p Avg - funzione per grafici \(page 457\)](#)

## LINEST\_SEY

**LINEST\_SEY()** restituisce l'errore standard aggregato della stima del valore y di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentata da coppie di numeri in x-expression e y-expression ripetute su un insieme di record, come definito da una clausola **group by**.

### Sintassi:

```
LINEST_SEY (y_value, x_value[, y0 [, x0 ]])
```

**Tipo di dati restituiti:** numerico

### Argomenti:

| Argomento  | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| y_value    | L'espressione o il campo contenente la scala di valori y da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| x_value    | L'espressione o il campo contenente la scala di valori x da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| y(0), x(0) | È possibile dichiarare un valore facoltativo y0 forzando il passaggio della linea di regressione attraverso l'asse delle y in un determinato punto. Dichiarando sia y0 che x0, è possibile forzare il passaggio della linea di regressione attraverso una coordinata singola fissa.<br><br>A meno che non vengano dichiarati sia y0 che x0, la funzione richiede almeno due coppie di dati valide per il calcolo. Se vengono dichiarati i valori y0 e x0, sarà sufficiente una singola coppia di dati. |

### Limiti:

Se una o entrambe le parti di una coppia di dati includono valori di testo, valori NULL e valori mancanti, l'intera coppia di dati verrà ignorata.

### Vedere anche:

[p Esempi di utilizzo delle funzioni linest \(page 517\)](#)

## LINEST\_SEY - funzione per grafici

**LINEST\_SEY()** restituisce l'errore standard aggregato della stima y di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentate da coppie di numeri date dalle espressioni **x\_value** e **y\_value**, ripetute sulle dimensioni del grafico.

### Sintassi:

```
LINEST_SEY ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value, x_value[, y0_const[, x0_const]])
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| y_value       | L'espressione o il campo contenente la scala di valori y da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| x_value       | L'espressione o il campo contenente la scala di valori x da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| y0, x0        | <p>È possibile dichiarare un valore facoltativo y0 forzando il passaggio della linea di regressione attraverso l'asse delle y in un determinato punto. Dichiarando sia y0 che x0, è possibile forzare il passaggio della linea di regressione attraverso una coordinata singola fissa.</p> <div>  <p><i>A meno che non vengano dichiarati sia y0 che x0, la funzione richiede almeno due coppie di dati valide per il calcolo. Se vengono dichiarati i valori y0 e x0, sarà sufficiente una singola coppia di dati.</i></p> </div> |
| SetExpression | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                                                                                                                                                                                                                                                                                 |
| DISTINCT      | Se la parola <b>DISTINCT</b> è riportata prima degli argomenti della funzione, i duplicati risultanti dalla valutazione degli argomenti della funzione vengono ignorati.                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| TOTAL         | <p>Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.</p> <p>Utilizzando <b>TOTAL [&lt;fld {fld}&gt;]</b>, dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili.</p>                                                   |

**Limiti:**

Il parametro della funzione di aggregazione non deve contenere altre funzioni di aggregazione, a meno che tali aggregazioni interne non contengano il qualificatore **TOTAL**. Nel caso di aggregazioni nidificate più complesse, utilizzare la funzione avanzata **Aggr** in combinazione con una dimensione specificata.

Se una o entrambe le parti di una coppia di dati includono valori di testo, valori NULL e valori mancanti, l'intera coppia di dati verrà ignorata.

**Vedere anche:**

p [Esempi di utilizzo delle funzioni lineat \(page 517\)](#)

[p Avg - funzione per grafici \(page 457\)](#)

## LINEST\_SSREG

**LINEST\_SSREG()** restituisce la somma di una regressione aggregata dei quadrati di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentata da coppie di numeri in x-expression e y-expression ripetute su un insieme di record, come definito da una clausola **group by**.

### Sintassi:

```
LINEST_SSREG (y_value, x_value[, y0 [, x0 ]])
```

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento  | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| y_value    | L'espressione o il campo contenente la scala di valori y da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| x_value    | L'espressione o il campo contenente la scala di valori x da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| y(0), x(0) | È possibile dichiarare un valore facoltativo y0 forzando il passaggio della linea di regressione attraverso l'asse delle y in un determinato punto. Dichiarando sia y0 che x0, è possibile forzare il passaggio della linea di regressione attraverso una coordinata singola fissa.<br><br>A meno che non vengano dichiarati sia y0 che x0, la funzione richiede almeno due coppie di dati valide per il calcolo. Se vengono dichiarati i valori y0 e x0, sarà sufficiente una singola coppia di dati. |

### Limiti:

Se una o entrambe le parti di una coppia di dati includono valori di testo, valori NULL e valori mancanti, l'intera coppia di dati verrà ignorata.

### Vedere anche:

[p Esempi di utilizzo delle funzioni linest \(page 517\)](#)

## LINEST\_SSREG - funzione per grafici

**LINEST\_SSREG()** restituisce la somma di regressione aggregata dei quadrati di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentate da coppie di numeri date dalle espressioni **x\_value** e **y\_value**, ripetute sulle dimensioni del grafico.

### Sintassi:

```
LINEST_SSREG ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value, x_value[, y0_const[, x0_const]])
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| y_value       | L'espressione o il campo contenente la scala di valori y da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| x_value       | L'espressione o il campo contenente la scala di valori x da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| y0, x0        | <p>È possibile dichiarare un valore facoltativo y0 forzando il passaggio della linea di regressione attraverso l'asse delle y in un determinato punto. Dichiarando sia y0 che x0, è possibile forzare il passaggio della linea di regressione attraverso una coordinata singola fissa.</p> <div>  <p><i>A meno che non vengano dichiarati sia y0 che x0, la funzione richiede almeno due coppie di dati valide per il calcolo. Se vengono dichiarati i valori y0 e x0, sarà sufficiente una singola coppia di dati.</i></p> </div> |
| SetExpression | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                                                                                                                                                                                                                                                                                 |
| DISTINCT      | Se la parola <b>DISTINCT</b> è riportata prima degli argomenti della funzione, i duplicati risultanti dalla valutazione degli argomenti della funzione vengono ignorati.                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| TOTAL         | <p>Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.</p> <p>Utilizzando <b>TOTAL [&lt;fld {fld}&gt;]</b>, dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili.</p>                                                   |

**Limiti:**

Il parametro della funzione di aggregazione non deve contenere altre funzioni di aggregazione, a meno che tali aggregazioni interne non contengano il qualificatore **TOTAL**. Nel caso di aggregazioni nidificate più complesse, utilizzare la funzione avanzata **Aggr** in combinazione con una dimensione specificata.

Se una o entrambe le parti di una coppia di dati includono valori di testo, valori NULL e valori mancanti, l'intera coppia di dati verrà ignorata.

**Vedere anche:**

p [Esempi di utilizzo delle funzioni linest \(page 517\)](#)

[p Avg - funzione per grafici \(page 457\)](#)

## LINEST\_SSRESID

**LINEST\_SSRESID()** restituisce la somma residua aggregata dei quadrati di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentata da coppie di numeri in x-expression e y-expression ripetute su un insieme di record, come definito da una clausola **group by**.

### Sintassi:

```
LINEST_SSRESID (y_value, x_value[, y0 [, x0 ]])
```

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento  | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| y_value    | L'espressione o il campo contenente la scala di valori y da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| x_value    | L'espressione o il campo contenente la scala di valori x da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| y(0), x(0) | È possibile dichiarare un valore facoltativo y0 forzando il passaggio della linea di regressione attraverso l'asse delle y in un determinato punto. Dichiarando sia y0 che x0, è possibile forzare il passaggio della linea di regressione attraverso una coordinata singola fissa.<br><br>A meno che non vengano dichiarati sia y0 che x0, la funzione richiede almeno due coppie di dati valide per il calcolo. Se vengono dichiarati i valori y0 e x0, sarà sufficiente una singola coppia di dati. |

### Limiti:

Se una o entrambe le parti di una coppia di dati includono valori di testo, valori NULL e valori mancanti, l'intera coppia di dati verrà ignorata.

### Vedere anche:

[p Esempi di utilizzo delle funzioni linest \(page 517\)](#)

## LINEST\_SSRESID - funzione per grafici

**LINEST\_SSRESID()** restituisce la somma residua aggregata dei quadrati di una regressione lineare definita dall'equazione  $y=mx+b$  per una serie di coordinate rappresentate da coppie di numeri nelle espressioni fornite da **x\_value** e **y\_value**, ripetute sulle dimensioni del grafico.

### Sintassi:

```
LINEST_SSRESID ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value,  
x_value[, y0_const[, x0_const]])
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| y_value       | L'espressione o il campo contenente la scala di valori y da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| x_value       | L'espressione o il campo contenente la scala di valori x da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| y0, x0        | <p>È possibile dichiarare un valore facoltativo y0 forzando il passaggio della linea di regressione attraverso l'asse delle y in un determinato punto. Dichiarando sia y0 che x0, è possibile forzare il passaggio della linea di regressione attraverso una coordinata singola fissa.</p> <div>  <p><i>A meno che non vengano dichiarati sia y0 che x0, la funzione richiede almeno due coppie di dati valide per il calcolo. Se vengono dichiarati i valori y0 e x0, sarà sufficiente una singola coppia di dati.</i></p> </div> |
| SetExpression | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                                                                                                                                                                                                                                                                                 |
| DISTINCT      | Se la parola <b>DISTINCT</b> è riportata prima degli argomenti della funzione, i duplicati risultanti dalla valutazione degli argomenti della funzione vengono ignorati.                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| TOTAL         | <p>Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.</p> <p>Utilizzando <b>TOTAL [&lt;fld {fld}&gt;]</b>, dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili.</p>                                                   |

È possibile dichiarare un valore facoltativo y0 forzando il passaggio della linea di regressione attraverso l'asse delle y in un determinato punto. Dichiarando sia y0 che x0, è possibile forzare il passaggio della linea di regressione attraverso una coordinata singola fissa.

**Limiti:**

Il parametro della funzione di aggregazione non deve contenere altre funzioni di aggregazione, a meno che tali aggregazioni interne non contengano il qualificatore **TOTAL**. Nel caso di aggregazioni nidificate più complesse, utilizzare la funzione avanzata **Aggr** in combinazione con una dimensione specificata.

Se una o entrambe le parti di una coppia di dati includono valori di testo, valori NULL e valori mancanti, l'intera coppia di dati verrà ignorata.

**Vedere anche:**

p [Esempi di utilizzo delle funzioni linest \(page 517\)](#)

p [Avg - funzione per grafici \(page 457\)](#)

**Median**

**Median()** restituisce la mediana aggregata dei valori nell'espressione su un insieme di record, come definito da una clausola **group by**.

**Sintassi:**

**Median** (expr)

**Tipo di dati restituiti:** numerico

**Argomenti:**

## Argomenti

| Argomento | Descrizione                                             |
|-----------|---------------------------------------------------------|
| expr      | L'espressione o il campo contenente i dati da misurare. |

**Esempio: espressione dello script che utilizza una mediana****Esempio - espressione script****Script di caricamento**

Caricare i seguenti dati inline e l'espressione di script nell'editor caricamento dati per questo esempio.

Table 1:

```
Load RecNo() as ROWNo, Letter, Number Inline
```

```
[Letter, Number
```

```
A,1
```

```
A,3
```

```
A,4
```

```
A,9
```

```
B,2
```

```
B,8
```

```
B,9];
```

```
Median:
```

```
LOAD Letter,
```

```
Median(Number) as MyMedian
```

```
Resident Table1 Group By Letter;
```

**Creazione di una visualizzazione**

Creare una visualizzazione tabella in un foglio Qlik Sense con **Lettera** e **MyMedian** come dimensioni.

**Risultato**

| Letter | MyMedian |
|--------|----------|
| A      | 3.5      |
| B      | 8        |

**Spiegazione**

La mediana è considerata il numero "medio" quando i numeri sono stati ordinati dal più piccolo al più grande. Se la serie di dati presenta un numero pari di valori, la funzione restituisce la media dei due valori centrali. In questo esempio, la mediana è calcolata per ogni serie di valori di **A** e **B**, che è 3,5 e 8, rispettivamente.

**Median - funzione per grafici**

**Median()** restituisce il valore mediano della scala di valori aggregati nell'espressione ripetuti sulle dimensioni del grafico.

**Sintassi:**

```
Median ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] expr)
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

## Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| expr          | L'espressione o il campo contenente i dati da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| SetExpression | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                                                                                                                                                                                                                                 |
| DISTINCT      | Se la parola <b>DISTINCT</b> è riportata prima degli argomenti della funzione, i duplicati risultanti dalla valutazione degli argomenti della funzione vengono ignorati.                                                                                                                                                                                                                                                                                                                                                                                            |
| TOTAL         | <p>Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.</p> <p>Utilizzando <b>TOTAL [&lt;fld {, fld}&gt;]</b>, dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili.</p> |

### Limiti:

Il parametro della funzione di aggregazione non deve contenere altre funzioni di aggregazione, a meno che tali aggregazioni interne non contengano il qualificatore **TOTAL**. Nel caso di aggregazioni nidificate più complesse, utilizzare la funzione avanzata **Aggr** in combinazione con una dimensione specificata.

### Esempio: espressione del grafico che utilizza una mediana

Esempio: espressione del grafico

#### Script di caricamento

Caricare i dati seguenti come un caricamento inline nell'editor di caricamento dati per creare l'esempio di espressione del grafico in basso.

```
Load RecNo() as RowNo, Letter, Number Inline  
[Letter, Number  
A,1  
A,3  
A,4  
A,9  
B,2  
B,8  
B,9];
```

#### Creazione di una visualizzazione

Creare la visualizzazione di una tabella in un foglio Qlik Sense con **Lettera** come dimensione.

#### Espressione del grafico

Aggiungere la seguente espressione alla tabella come misura:

```
Median(Number)
```

#### Risultato

| Letter | Median(Number) |
|--------|----------------|
| Totals | 4              |
| A      | 3.5            |
| B      | 8              |

#### Spiegazione

La mediana è considerata il numero "medio" quando i numeri sono stati ordinati dal più piccolo al più grande. Se la serie di dati presenta un numero pari di valori, la funzione restituisce la media dei due valori centrali. In questo esempio, la mediana è calcolata per ogni serie di valori di **A** e **B**, che è 3,5 e 8, rispettivamente.

La mediana per i **Totali** è calcolata da tutti i valori, che equivale a 4.

---

### Vedere anche:

p [Avg - funzione per grafici \(page 457\)](#)

### MutualInfo - funzione per grafici

**MutualInfo** calcola le informazioni reciproche (MI, Mutual Information) tra due campi o tra valori aggregati in **Aggr()**.

**MutualInfo** restituisce le informazioni reciproche aggregate per due set di dati. Ciò consente un'analisi dei driver chiave tra un campo e un driver potenziale. Le informazioni reciproche misurano il rapporto tra i set di dati e vengono aggregate per i valori di coppia (x,y) ripetuti sulle dimensioni del grafico. Le informazioni reciproche vengono misurate tra 0 e 1 e possono essere formattate come valore percentile. **MutualInfo** viene definito mediante selezioni o un'espressione set.

**MutualInfo** consente tipi diversi di analisi MI:

- MI pairwise: calcola il valore MI tra un campo driver e un campo target.
- Scomposizione driver per valore: il valore MI viene calcolato tra i singoli valori di campo nei campi driver e target.
- Selezione funzionalità: utilizzare **MutualInfo** in un grafico a griglia per creare una matrice in cui tutti i campi vengono confrontati l'uno con l'altro in base alle informazioni reciproche (MI).

**MutualInfo** non indica necessariamente casualità tra i campi che condividono informazioni reciproche. Due campi possono condividere informazioni reciproche, ma potrebbero non essere driver reciprocamente uguali. Ad esempio, al momento di confrontare le vendite di gelati e la temperatura esterna, **MutualInfo** mostrerà le informazioni reciproche tra i due. Non indicherà se è la temperatura esterna a spingere le vendite di gelati, che rappresenta una motivazione probabile, o se sono le vendite dei gelati a spingere la temperatura esterna, ovvero una motivazione piuttosto improbabile.

Quando si calcolano le informazioni reciproche, le associazioni influiscono sulla corrispondenza tra e la frequenza dei valori dai campi che provengono da tabelle diverse.

I valori restituiti per gli stessi campi o selezioni possono variare leggermente. Ciò è dovuto al fatto che ciascun richiamo **MutualInfo** opera in base a un campione selezionato in modo casuale e alla casualità intrinseca dell'algoritmo **MutualInfo**.

**MutualInfo** può essere applicato alla funzione **Aggr()**.

### Sintassi:

```
MutualInfo ({SetExpression}} [DISTINCT] [TOTAL] field1, field2 , datatype [,  
breakdownbyvalue [, sampleize ]])
```

Tipo di dati restituiti: numerico

Argomenti:

Argomenti

| Argomento        | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| field1, field2   | Le espressioni o i campi contenenti i due set di campioni per i quali vengono misurate le informazioni mutue.                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| datatype         | <p>I tipi di dati contenuti nel target e nel driver,</p> <p>1 o 'dd' per discrete:discrete</p> <p>2 o 'cc' per continuous:continuous</p> <p>3 o 'cd' per continuous:discrete</p> <p>4 o 'dc' per discrete:continuous</p> <p>I tipi di dati non fanno distinzioni tra maiuscole e minuscole.</p>                                                                                                                                                                                                                                                   |
| breakdownbyvalue | <p>Un valore statico corrispondente a un valore nel driver. Se fornito, il calcolo calcolerà il contributo MI per tale valore. È possibile utilizzare <b>ValueList()</b> o <b>ValueLoop()</b>. Se viene aggiunto <b>Null()</b>, il calcolo calcolerà il valore MI complessivo per tutti i valori nel driver.</p> <p>La scomposizione per valore richiede che il driver contenga dati discreti.</p>                                                                                                                                                |
| samplesize       | <p>Il numero di valori da campionare dal target e dal driver. Il campionamento è casuale. <b>MutualInfo</b> richiede una dimensione di campionamento minima di 80. Per impostazione predefinita, <b>MutualInfo</b> campiona solo fino a 10.000 coppie di dati, dato che <b>MutualInfo</b> può presentare un grosso impatto sulle risorse. È possibile specificare un numero maggiore di coppie dati nelle dimensioni del campione. In caso di esaurimento del tempo a disposizione per <b>MutualInfo</b>, ridurre le dimensioni del campione.</p> |
| SetExpression    | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                                                                                                                                                                                                               |
| DISTINCT         | Se la parola <b>DISTINCT</b> è riportata prima degli argomenti della funzione, i duplicati risultanti dalla valutazione degli argomenti della funzione vengono ignorati.                                                                                                                                                                                                                                                                                                                                                                          |

| Argomento | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TOTAL     | <p>Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.</p> <p>Utilizzando <b>TOTAL [&lt;fld {fld}&gt;]</b>, dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili.</p> |

### Limiti:

Se una o entrambe le parti di una coppia di dati includono valori di testo, valori NULL e valori mancanti, l'intera coppia di dati verrà ignorata.

### Esempi e risultati:

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

#### Esempi di funzioni

| Esempio                                                 | Risultato                                                                                                                                                                                                                                                                                                                                      |
|---------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| mutualinfo<br>(Age, salary, 1)                          | Per un tabella che include la dimensione Employee name e la misura mutualinfo(Age, salary, 1), il risultato è 0.99820986. Il risultato viene visualizzato solo per la cella dei totali.                                                                                                                                                        |
| mutualinfo<br>(TOTAL Age, salary, 1, null(), 81)        | Se si crea una casella di filtro con la dimensione Gender e si eseguono selezioni da questa, si otterrà il risultato 0.99805677 quando viene selezionato il valore Female e 0.99847373 quando viene selezionato il valore Male. Ciò si verifica perché la selezione esclude tutti i risultati che non appartengono all'altro valore di Gender. |
| mutualinfo<br>(TOTAL Age, Gender, 1, ValueLoop (25,35)) | 0.68196996. La selezione di qualsiasi valore da Gender porterà il risultato a 0.                                                                                                                                                                                                                                                               |
| mutualinfo<br>({1}<br>TOTAL Age, salary, 1, null())     | 0.99820986. Ciò risulta indipendente dalle selezioni. L'espressione set {1} ignora tutte le selezioni e le dimensioni.                                                                                                                                                                                                                         |

Dati utilizzati negli esempi:

salary:

```
LOAD * inline [
```

```
"Employee name"|Age|Gender|Salary
```

Aiden Charles|20|Male|25000  
Ann Lindquist|69|Female|58000  
Anna Johansen|37|Female|36000  
Anna Karlsson|42|Female|23000  
Antonio Garcia|20|Male|61000  
Benjamin Smith|42|Male|27000  
Bill Yang|49|Male|50000  
Binh Protzmann|69|Male|21000  
Bob Park|51|Male|54000  
Brenda Davies|25|Male|32000  
Celine Gagnon|48|Female|38000  
Cezar Sandu|50|Male|46000  
Charles Ingvar Jönsson|27|Male|58000  
Charlotte Edberg|45|Female|56000  
Cindy Lynn|69|Female|28000  
Clark Wayne|63|Male|31000  
Daroush Ferrara|31|Male|29000  
David Cooper|37|Male|64000  
David Leg|58|Male|57000  
Eunice Goldblum|31|Female|32000  
Freddy Halvorsen|25|Male|26000  
Gauri Indu|36|Female|46000  
George van Zaant|59|Male|47000  
Glenn Brown|58|Male|40000  
Harry Jones|38|Male|40000  
Helen Brolin|52|Female|66000  
Hiroshi Ito|24|Male|42000

Ian Underwood|40|Male|45000  
Ingrid Hendrix|63|Female|27000  
Ira Baume|39|Female|39000  
Jackie Kingsley|23|Female|28000  
Jennica Williams|36|Female|48000  
Jerry Tessel|31|Male|57000  
Jim Bond|50|Male|58000  
Joan Callins|60|Female|65000  
Joan Cleaves|25|Female|61000  
Joe Cheng|61|Male|41000  
John Doe|36|Male|59000  
John Lemon|43|Male|21000  
Karen Helmkey|54|Female|25000  
Karl Berger|38|Male|68000  
Karl Straubbaum|30|Male|40000  
Kaya Alpan|32|Female|60000  
Kenneth Finley|21|Male|25000  
Leif Shine|63|Male|70000  
Lennart Skoglund|63|Male|24000  
Leona Korhonen|46|Female|50000  
Lina André|50|Female|65000  
Louis Presley|29|Male|36000  
Luke Langston|50|Male|63000  
Marcus Salvatori|31|Male|46000  
Marie Simon|57|Female|23000  
Mario Rossi|39|Male|62000  
Markus Danzig|26|Male|48000

Michael Carlen|21|Male|45000

Michelle Tyson|44|Female|69000

Mike Ashkenaz|45|Male|68000

Miro Ito|40|Male|39000

Nina Mihn|62|Female|57000

Olivia Nguyen|35|Female|51000

Olivier Simenon|44|Male|31000

Östen Ärlig|68|Male|57000

Pamala Garcia|69|Female|29000

Paolo Romano|34|Male|45000

Pat Taylor|67|Female|69000

Paul Dupont|34|Male|38000

Peter Smith|56|Male|53000

Pierre Clouseau|21|Male|37000

Preben Jørgensen|35|Male|38000

Rey Jones|65|Female|20000

Ricardo Gucci|55|Male|65000

Richard Ranieri|30|Male|64000

Rob Carsson|46|Male|54000

Rolf Wesselund|25|Male|51000

Ronaldo Costa|64|Male|39000

Sabrina Richards|57|Female|40000

Sato Hiromu|35|Male|21000

Sehoon Daw|57|Male|24000

Stefan Lind|67|Male|35000

Steve Cioazzi|58|Male|23000

Sunil Gupta|45|Male|40000

Sven Svensson|45|Male|55000

Tom Lindwall|46|Male|24000

Tomas Nilsson|27|Male|22000

Trinity Rizzo|52|Female|48000

Vanessa Lambert|54|Female|27000

] (delimiter is '|');

### Skew

**Skew()** restituisce l'asimmetria dell'espressione su un insieme di record, come definito da una clausola **group by**.

#### Sintassi:

```
Skew([ distinct] expr)
```

**Tipo di dati restituiti:** numerico

#### Argomenti:

##### Argomenti

| Argomento | Descrizione                                                                                          |
|-----------|------------------------------------------------------------------------------------------------------|
| expr      | L'espressione o il campo contenente i dati da misurare.                                              |
| DISTINCT  | Se la parola <b>distinct</b> è riportata prima dell'espressione, tutti i duplicati vengono ignorati. |

#### Esempi e risultati:

Aggiungere lo script di esempio all'app ed eseguirlo. Creare quindi una tabella lineare utilizzando `Type` e `MySkew` come dimensioni.

Dati risultanti

| Esempio                                                                                                                                                                                                                                                                                                    | Risultato                                                                                                                                                                               |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>Table1: Crosstable (Type, value) Load recno() as ID, * inline [ Observation Comparison 35 2 40 27 12 38 15 31 21 1 14 19 46 1 10 34 28 3 48 1 16 2 30 3 32 2 48 1 31 2 22 1 12 3 39 29 19 37 25 2 ] (delimiter is ' ');  Skew1: LOAD Type, Skew(Value) as MySkew Resident Table1 Group By Type;</pre> | <p>I risultati del calcolo di Skew() sono:</p> <ul style="list-style-type: none"> <li>• Type è Myskew</li> <li>• Comparison è 0.86414768</li> <li>• observation è 0.32625351</li> </ul> |

### Skew - funzione per grafici

**Skew()** restituisce l'asimmetria aggregata dell'espressione o del campo ripetuta sulle dimensioni del grafico.

#### Sintassi:

```
Skew ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] expr)
```

**Tipo di dati restituiti:** numerico

#### Argomenti:

Argomenti

| Argomento | Descrizione                                             |
|-----------|---------------------------------------------------------|
| expr      | L'espressione o il campo contenente i dati da misurare. |

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SetExpression | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                                                                                                                                                                                                                               |
| DISTINCT      | Se la parola <b>DISTINCT</b> è riportata prima degli argomenti della funzione, i duplicati risultanti dalla valutazione degli argomenti della funzione vengono ignorati.                                                                                                                                                                                                                                                                                                                                                                                          |
| TOTAL         | <p>Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.</p> <p>Utilizzando <b>TOTAL [&lt;fld {fld}&gt;]</b>, dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili.</p> |

**Limiti:**

Il parametro della funzione di aggregazione non deve contenere altre funzioni di aggregazione, a meno che tali aggregazioni interne non contengano il qualificatore **TOTAL**. Nel caso di aggregazioni nidificate più complesse, utilizzare la funzione avanzata **Aggr** in combinazione con una dimensione specificata.

**Esempi e risultati:**

Aggiungere lo script di esempio all'app ed eseguirlo. Creare quindi una tabella lineare utilizzando `type` come dimensione e `skew(Value)` come misura.

Si consiglia di abilitare `TOTALs` nelle proprietà della tabella.

| Esempio                                                                                                                                                                                                                            | Risultato                                                                                                                                                                                         |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>Table1: Crosstable (Type, value) Load recno() as ID, * inline [ Observation Comparison 35 2 40 27 12 38 15 31 21 1 14 19 46 1 10 34 28 3 48 1 16 2 30 3 32 2 48 1 31 2 22 1 12 3 39 29 19 37 25 2 ] (delimiter is ' ');</pre> | <p>I risultati del calcolo di Skew(Value) sono:</p> <ul style="list-style-type: none"> <li>• Total è 0.23522195</li> <li>• Comparison è 0.86414768</li> <li>• Observation è 0.32625351</li> </ul> |

**Vedere anche:**

p [Avg - funzione per grafici \(page 457\)](#)

**Stdev**

**Stdev()** restituisce la deviazione standard dei valori dati dall'espressione su un insieme di record, come definito da una clausola **group by**.

**Sintassi:**

```
Stdev ([distinct] expr)
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

## Argomenti

| Argomento | Descrizione                                                                                          |
|-----------|------------------------------------------------------------------------------------------------------|
| expr      | L'espressione o il campo contenente i dati da misurare.                                              |
| distinct  | Se la parola <b>distinct</b> è riportata prima dell'espressione, tutti i duplicati vengono ignorati. |

### Esempi e risultati:

Aggiungere lo script di esempio all'app ed eseguirlo. Creare quindi una tabella lineare utilizzando `Type` e `MyStdev` come dimensioni.

Dati risultanti

| Esempio                                                                                                                                                                                                                                                                                                       | Risultato                                                                                                                                                                                                                                           |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>Table1: Crosstable (Type, value) Load recno() as ID, * inline [ Observation Comparison 35 2 40 27 12 38 15 31 21 1 14 19 46 1 10 34 28 3 48 1 16 2 30 3 32 2 48 1 31 2 22 1 12 3 39 29 19 37 25 2 ] (delimiter is ' ');  Stdev1: LOAD Type, Stdev(Value) as MyStdev Resident Table1 Group By Type;</pre> | <p>I risultati del calcolo di <code>Stdev()</code> sono:</p> <ul style="list-style-type: none"><li>• <code>Type</code> è <code>MyStdev</code></li><li>• <code>Comparison</code> è 14.61245</li><li>• <code>observation</code> è 12.507997</li></ul> |

### Stdev - funzione per grafici

**Stdev()** trova la deviazione standard della scala di dati aggregati nell'espressione o nel campo ripetuta sulle dimensioni del grafico.

### Sintassi:

```
Stdev ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] expr)
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

| Argomenti     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| expr          | L'espressione o il campo contenente i dati da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| SetExpression | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                                                                                                                                                                                                                               |
| DISTINCT      | Se la parola <b>DISTINCT</b> è riportata prima degli argomenti della funzione, i duplicati risultanti dalla valutazione degli argomenti della funzione vengono ignorati.                                                                                                                                                                                                                                                                                                                                                                                          |
| TOTAL         | <p>Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.</p> <p>Utilizzando <b>TOTAL [&lt;fld {fld}&gt;]</b>, dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili.</p> |

**Limiti:**

Il parametro della funzione di aggregazione non deve contenere altre funzioni di aggregazione, a meno che tali aggregazioni interne non contengano il qualificatore **TOTAL**. Nel caso di aggregazioni nidificate più complesse, utilizzare la funzione avanzata **Aggr** in combinazione con una dimensione specificata.

**Esempi e risultati:**

Aggiungere lo script di esempio all'app ed eseguirlo. Creare quindi una tabella lineare utilizzando `type` come dimensione e `stdev(value)` come misura.

Si consiglia di abilitare `TOTALS` nelle proprietà della tabella.

| Esempio                                                                                                                                                                                                                                           | Risultato                                                                                                                                                                                     |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> stdev(value) Table1: Crosstable (Type, value) Load recno() as ID, * inline [ Observation Comparison 35 2 40 27 12 38 15 31 21 1 14 19 46 1 10 34 28 3 48 1 16 2 30 3 32 2 48 1 31 2 22 1 12 3 39 29 19 37 25 2 ] (delimiter is ' '); </pre> | <p>I risultati del calcolo di Stdev(Value) sono:</p> <ul style="list-style-type: none"> <li>• Total è 15.47529</li> <li>• Comparison è 14.61245</li> <li>• Observation è 12.507997</li> </ul> |

**Vedere anche:**

p [Avg - funzione per grafici \(page 457\)](#)

p [STEYX - funzione per grafici \(page 515\)](#)

**Sterr**

**Sterr()** restituisce l'errore standard aggregato ( $\text{stdev}/\sqrt{n}$ ) per una serie di valori rappresentata da un'espressione ripetuta su un insieme di record, come definito da una clausola **group by**.

**Sintassi:**

```
Sterr ([distinct] expr)
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

## Argomenti

| Argomento | Descrizione                                                                                          |
|-----------|------------------------------------------------------------------------------------------------------|
| expr      | L'espressione o il campo contenente i dati da misurare.                                              |
| distinct  | Se la parola <b>distinct</b> è riportata prima dell'espressione, tutti i duplicati vengono ignorati. |

### Limiti:

I valori di testo, i valori NULL e i valori mancanti vengono ignorati.

### Esempi e risultati:

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

| Dati risultanti                                                                                                                                                                                                                                                                                               |                                                                                                                                                                                                                 |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Esempio                                                                                                                                                                                                                                                                                                       | Risultato                                                                                                                                                                                                       |
| <pre>Table1: Crosstable (Type, value) Load recno() as ID, * inline [ Observation Comparison 35 2 40 27 12 38 15 31 21 1 14 19 46 1 10 34 28 3 48 1 16 2 30 3 32 2 48 1 31 2 22 1 12 3 39 29 19 37 25 2 ] (delimiter is ' ');  Sterr1: LOAD Type, Sterr(Value) as MySterr Resident Table1 Group By Type;</pre> | <p>In una tabella con le dimensioni Type e MySterr, i risultati del calcolo Sterr() nello script di caricamento dei dati sono:</p> <p>Type MySterr</p> <p>Comparison 3.2674431</p> <p>Observation 2.7968733</p> |

### Sterr - funzione per grafici

**Sterr()** trova il valore dell'errore standard della media, ( $\text{stdev}/\sqrt{n}$ ), per la serie di valori aggregati nell'espressione ripetuta sulle dimensioni del grafico.

### Sintassi:

```
Sterr ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] expr)
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

### Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| expr          | L'espressione o il campo contenente i dati da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| SetExpression | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                                                                                                                                                                                                                               |
| DISTINCT      | Se la parola <b>DISTINCT</b> è riportata prima degli argomenti della funzione, i duplicati risultanti dalla valutazione degli argomenti della funzione vengono ignorati.                                                                                                                                                                                                                                                                                                                                                                                          |
| TOTAL         | <p>Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.</p> <p>Utilizzando <b>TOTAL [&lt;fld {fld}&gt;]</b>, dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili.</p> |

**Limiti:**

Il parametro della funzione di aggregazione non deve contenere altre funzioni di aggregazione, a meno che tali aggregazioni interne non contengano il qualificatore **TOTAL**. Nel caso di aggregazioni nidificate più complesse, utilizzare la funzione avanzata **Aggr** in combinazione con una dimensione specificata.

I valori di testo, i valori NULL e i valori mancanti vengono ignorati.

**Esempi e risultati:**

Aggiungere lo script di esempio all'app ed eseguirlo. Creare quindi una tabella lineare utilizzando `type` come dimensione e `sterr(value)` come misura.

Si consiglia di abilitare `total`s nelle proprietà della tabella.

| Esempio                                                                                                                                                                                                                            | Risultato                                                                                                                                                                                       |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>Table1: Crosstable (Type, value) Load recno() as ID, * inline [ Observation Comparison 35 2 40 27 12 38 15 31 21 1 14 19 46 1 10 34 28 3 48 1 16 2 30 3 32 2 48 1 31 2 22 1 12 3 39 29 19 37 25 2 ] (delimiter is ' ');</pre> | <p>I risultati del calcolo di Sterr(Value) sono:</p> <ul style="list-style-type: none"> <li>• Total è 2.4468583</li> <li>• Comparison è 3.2674431</li> <li>• Observation è 2.7968733</li> </ul> |

**Vedere anche:**

[p Avg - funzione per grafici \(page 457\)](#)

[p STEYX - funzione per grafici \(page 515\)](#)

**STEYX**

**STEYX()** restituisce l'errore standard aggregato del valore y previsto per ogni valore x nella regressione per una serie di coordinate rappresentata da coppie di numeri in x-expression e y-expression ripetute su un insieme di record, come definito da una clausola **group by**.

**Sintassi:**

```
STEYX (y_value, x_value)
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

## Argomenti

| Argomento | Descrizione                                                           |
|-----------|-----------------------------------------------------------------------|
| y_value   | L'espressione o il campo contenente la scala di valori y da misurare. |
| x_value   | L'espressione o il campo contenente la scala di valori x da misurare. |

### **Limiti:**

Se una o entrambe le parti di una coppia di dati includono valori di testo, valori NULL e valori mancanti, l'intera coppia di dati verrà ignorata.

### **Esempi e risultati:**

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

### Dati risultanti

| Esempio                                                                                                                                                                                                                                                                                                                                                                                                                                      | Risultato                                                                                                                                                     |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Trend:</p> <p>Load *, 1 as Grp;</p> <p>LOAD * inline [</p> <p>Month KnownY KnownX</p> <p>Jan 2 6</p> <p>Feb 3 5</p> <p>Mar 9 11</p> <p>Apr 6 7</p> <p>May 8 5</p> <p>Jun 7 4</p> <p>Jul 5 5</p> <p>Aug 10 8</p> <p>Sep 9 10</p> <p>Oct 12 14</p> <p>Nov 15 17</p> <p>Dec 14 16</p> <p>] (delimiter is ' ');</p> <p>STEYX1:</p> <p>LOAD Grp,</p> <p>STEYX(KnownY, KnownX)</p> <p>as MySTEYX</p> <p>Resident Trend Group</p> <p>By Grp;</p> | <p>In una tabella con la dimensione <code>mysteyx</code>, il risultato del calcolo <code>STEYX()</code> nello script di caricamento dei dati è 2,0714764.</p> |

### STEYX - funzione per grafici

**STEYX()** restituisce l'errore standard aggregato quando si prevedono i valori y per ciascun valore x in una regressione lineare data da una serie di coordinate rappresentate da coppie di numeri nelle espressioni date da **y\_value** e **x\_value**.

#### Sintassi:

```
STEYX ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value, x_value)
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| y_value       | L'espressione o il campo contenente la scala di valori y noti da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| x_value       | L'espressione o il campo contenente la scala di valori x conosciuti da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| SetExpression | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                                                                                                                                                                                                                               |
| DISTINCT      | Se la parola <b>DISTINCT</b> è riportata prima degli argomenti della funzione, i duplicati risultanti dalla valutazione degli argomenti della funzione vengono ignorati.                                                                                                                                                                                                                                                                                                                                                                                          |
| TOTAL         | <p>Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.</p> <p>Utilizzando <b>TOTAL [&lt;fld {fld}&gt;]</b>, dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili.</p> |

**Limiti:**

Il parametro della funzione di aggregazione non deve contenere altre funzioni di aggregazione, a meno che tali aggregazioni interne non contengano il qualificatore **TOTAL**. Nel caso di aggregazioni nidificate più complesse, utilizzare la funzione avanzata **Aggr** in combinazione con una dimensione specificata.

Se una o entrambe le parti di una coppia di dati includono valori di testo, valori NULL e valori mancanti, l'intera coppia di dati verrà ignorata.

**Esempi e risultati:**

Aggiungere lo script di esempio all'app ed eseguirlo. Creare quindi una tabella lineare utilizzando `KnownY` e `KnownX` come dimensione e `Steyx(KnownY, KnownX)` come misura.

Si consiglia di abilitare `TOTALS` nelle proprietà della tabella.

| Esempio                                                                                                                                                                              | Risultato                                                                                                                         |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| <pre>Trend: LOAD * inline [ Month KnownY KnownX Jan 2 6 Feb 3 5 Mar 9 11 Apr 6 7 May 8 5 Jun 7 4 Jul 5 5 Aug 10 8 Sep 9 10 Oct 12 14 Nov 15 17 Dec 14 16 ] (delimiter is ' ');</pre> | <p>Il risultato del calcolo di STEYX(KnownY,KnownX) è 2,071 (se la formattazione del numero è impostata su 3 cifre decimali).</p> |

**Vedere anche:**

p [Avg - funzione per grafici \(page 457\)](#)

p [Sterr - funzione per grafici \(page 511\)](#)

**Esempi di utilizzo delle funzioni linest**

Le funzioni linest vengono utilizzate per trovare valori associati con l'analisi della regressione lineare. In questa sezione viene descritta la procedura di creazione delle visualizzazioni mediante dati campione per trovare i valori delle funzioni linest disponibili in Qlik Sense. Le funzioni linest possono essere utilizzate nello script di caricamento dei dati e nelle espressioni grafiche.

Per le descrizioni della sintassi e degli argomenti, fare riferimento ai singoli argomenti delle funzioni grafiche e di script linest.

**Espressioni dati e script utilizzati negli esempi**

Caricare i seguenti dati inline ed espressioni script nell'editor caricamento dati per gli esempi linest() in basso.

```
T1:
LOAD *, 1 as Grp;
LOAD * inline [
X|Y
1|0
2|1
3|3
4|8
5|14
6|20
7|0
8|50
9|25
10|60
11|38
12|19
13|26
14|143
15|98
16|27
17|59
18|78
19|158
20|279 ] (delimiter is '|');

R1:
LOAD
Grp,
linest_B(Y,X) as Linest_B,
linest_DF(Y,X) as Linest_DF,
linest_F(Y,X) as Linest_F,
linest_M(Y,X) as Linest_M,
linest_R2(Y,X) as Linest_R2,
linest_SEB(Y,X,1,1) as Linest_SEB,
linest_SEM(Y,X) as Linest_SEM,
linest_SEY(Y,X) as Linest_SEY,
linest_SSREG(Y,X) as Linest_SSREG,
linest_SSRESID(Y,X) as Linest_SSRESID
resident T1 group by Grp;
```

### Esempio 1: espressioni script che utilizzano linest

Esempio: espressioni script

#### Creare una visualizzazione dai calcoli script di caricamento dei dati

Creare la visualizzazione di una tabella in un foglio di Qlik Sense con i seguenti campi come colonne:

- Linest\_B
- Linest\_DF
- Linest\_F

- Linest\_M
- Linest\_R2
- Linest\_SEB
- Linest\_SEM
- Linest\_SEY
- Linest\_SSREG
- Linest\_SSRESID

### Risultato

La tabella contenente i risultati dei calcoli linest effettuati nello script di caricamento dei dati dovrebbe presentare l'aspetto seguente:

Tabella dei risultati

| Linest_B | Linest_DF | Linest_F | Linest_M | Linest_R2 | Linest_SEB |
|----------|-----------|----------|----------|-----------|------------|
| -35.047  | 18        | 20.788   | 8.605    | 0.536     | 22.607     |

Tabella dei risultati

| Linest_SEM | Linest_SEY | Linest_SSREG | Linest_SSRESID |
|------------|------------|--------------|----------------|
| 1.887      | 48.666     | 49235.014    | 42631.186      |

### Espressioni del grafico che utilizzano linest

#### Esempio: espressioni del grafico

Creare la visualizzazione di una tabella in un foglio Qlik Sense con i seguenti campi come dimensioni:

```
ValueList('Linest_b', 'Linest_df', 'Linest_f', 'Linest_m', 'Linest_r2', 'Linest_SEB', 'Linest_SEM', 'Linest_SEY', 'Linest_SSREG', 'Linest_SSRESID')
```

Questa espressione utilizza la funzione delle dimensioni sintetiche per creare etichette con i nomi delle funzioni linest. È possibile modificare l'etichetta in **Linest functions** per risparmiare spazio.

Aggiungere la seguente espressione alla tabella come misura:

```
Pick(Match(ValueList('Linest_b', 'Linest_df', 'Linest_f', 'Linest_m', 'Linest_r2', 'Linest_SEB', 'Linest_SEM', 'Linest_SEY', 'Linest_SSREG', 'Linest_SSRESID'), 'Linest_b', 'Linest_df', 'Linest_f', 'Linest_m', 'Linest_r2', 'Linest_SEB', 'Linest_SEM', 'Linest_SEY', 'Linest_SSREG', 'Linest_SSRESID'), Linest_b(Y,X), Linest_df(Y,X), Linest_f(Y,X), Linest_m(Y,X), Linest_r2(Y,X), Linest_SEB(Y,X,1,1), Linest_SEM(Y,X), Linest_SEY(Y,X), Linest_SSREG(Y,X), Linest_SSRESID(Y,X) )
```

Questa espressione consente di visualizzare il risultato di ciascuna funzione `linest` in relazione al nome corrispondente nella dimensione sintetica. Il risultato di `Linest_b(Y,X)` viene visualizzato accanto a **linest\_b** e così via.

### Risultato

Tabella dei risultati

| Linest functions | Linest function results |
|------------------|-------------------------|
| Linest_b         | -35.047                 |
| Linest_df        | 18                      |
| Linest_f         | 20.788                  |
| Linest_m         | 8.605                   |
| Linest_r2        | 0.536                   |
| Linest_SEB       | 22.607                  |
| Linest_SEM       | 1.887                   |
| Linest_SEY       | 48.666                  |
| Linest_SSREG     | 49235.014               |
| Linest_SSRESID   | 42631.186               |

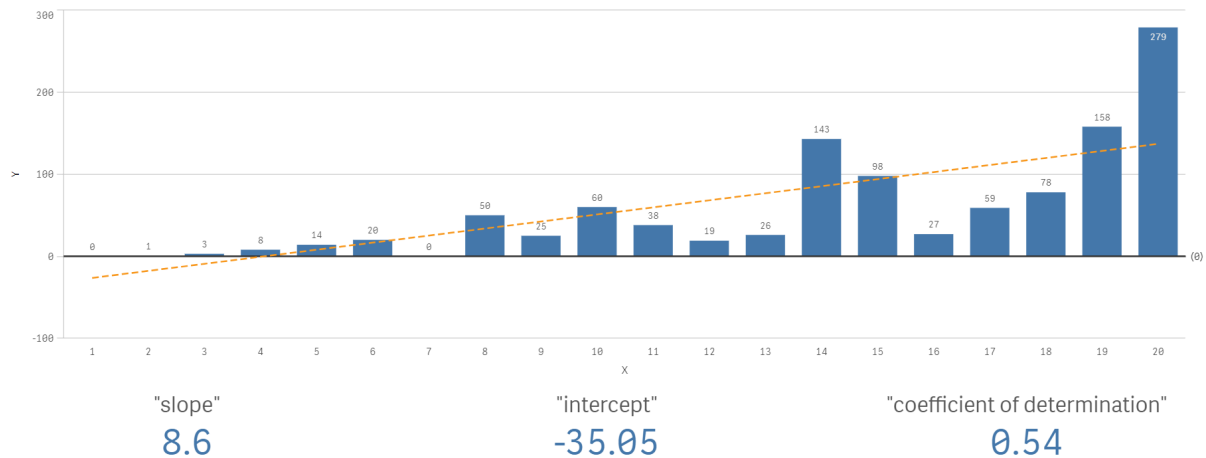
### Esempio 3: espressioni del grafico che utilizzano `linest`

Esempio: espressioni del grafico

1. Creare una visualizzazione grafico a barre in un foglio Qlik Sense con **x** come dimensione e **Y** come misura.
2. Aggiungere una linea di tendenza lineare alla misura Y.
3. Aggiungere una visualizzazione KPI al foglio.
  1. Aggiungere *pendenza* come etichetta per il KPI.
  2. Aggiungere `sum(Linest_M)` come espressione per il KPI.
4. Aggiungere una seconda visualizzazione KPI al foglio.
  1. Aggiungere *Intercetta* come etichetta per il KPI.
  2. Aggiungere `sum(Linest_B)` come espressione per il KPI.
5. Aggiungere una terza visualizzazione KPI al foglio.
  1. Aggiungere *coefficiente di determinazione* come etichetta per il KPI.
  2. Aggiungere `sum(Linest_R2)` come espressione per il KPI.

### Risultato

LinestFuncInGraph



### Spiegazione

Il grafico a barre mostra il tracciato dei dati X e Y. Le funzioni `linest()` pertinenti forniscono valori per l'equazione di regressione lineare su cui si basa la linea di tendenza, cioè  $y = m * x + b$ . L'equazione usa il metodo dei "minimi quadrati" per calcolare una linea retta (linea di tendenza) restituendo un array che descrive una linea che meglio si adatta ai dati.

I KPI mostrano i risultati delle funzioni `linest()` `sum(Linest_M)` per la pendenza e `sum(Linest_B)` per l'intercetta Y, che sono variabili nell'equazione di regressione lineare, e il corrispondente valore aggregato R2 per il coefficiente di determinazione.

## Funzioni di test statistici

Le funzioni di test statistici possono essere utilizzate sia nello script di caricamento dei dati che nelle espressioni grafiche, anche se la sintassi sarà diversa.

### Funzioni di test del chi quadrato

Di solito vengono utilizzate per lo studio delle variabili qualitative. È possibile confrontare le frequenze osservate in una tabella di frequenze a senso unico con le frequenze previste oppure studiare il collegamento tra due variabili in una tabella di contingenza.

### Funzioni di t-test

Le funzioni t-test vengono utilizzate per un'analisi statistica di due popolazioni medie. Un t-test a due campioni analizza se i due campioni sono diversi e viene generalmente utilizzato quando due distribuzioni standard presentano varianze sconosciute e quando un esperimento utilizza dimensioni ridotte per il campione.

### Funzioni di z-test

Un'analisi statistica di due popolazioni medie. Uno z-test con due campioni analizza se i due campioni sono differenti e viene comunemente utilizzato quando due distribuzioni standard presentano varianze conosciute e quando un esperimento utilizza dimensioni notevoli per il campione.

### Funzioni chi2-test

Di solito vengono utilizzate per lo studio delle variabili qualitative. È possibile confrontare le frequenze osservate in una tabella di frequenze a senso unico con le frequenze previste oppure studiare il collegamento tra due variabili in una tabella di contingenza. Chi-squared test functions are used to determine whether there is a statistically significant difference between the expected frequencies and the observed frequencies in one or more groups. Often a histogram is used, and the different bins are compared to an expected distribution.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

#### Chi2Test\_chi2

**Chi2Test\_chi2()** restituisce il valore aggregato di chi<sup>2</sup>-test per una o due serie di valori.

```
Chi2Test_chi2(col, row, actual_value[, expected_value])
```

#### Chi2Test\_df

**Chi2Test\_df()** restituisce il valore df (gradi di libertà) aggregato di chi<sup>2</sup>-test per una o due serie di valori.

```
Chi2Test_df(col, row, actual_value[, expected_value])
```

#### Chi2Test\_p

**Chi2Test\_p()** restituisce il valore p (significatività) aggregato di chi<sup>2</sup>-test per una o due serie di valori.

```
Chi2Test_p - funzione per grafici(col, row, actual_value[, expected_value])
```

#### Vedere anche:

p [Funzioni di t-test \(page 525\)](#)

p [Funzioni di z-test \(page 561\)](#)

#### Chi2Test\_chi2

**Chi2Test\_chi2()** restituisce il valore aggregato di chi<sup>2</sup>-test per una o due serie di valori.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.



*Tutte le funzioni Qlik Sense del chi<sup>2</sup>-test presentano gli stessi argomenti.*

**Sintassi:**

```
Chi2Test_chi2(col, row, actual_value[, expected_value])
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

## Argomenti

| Argomento      | Descrizione                                                                                       |
|----------------|---------------------------------------------------------------------------------------------------|
| col, row       | La riga e la colonna specificate nella matrice di valori testati.                                 |
| actual_value   | Il valore osservato dei dati in corrispondenza di <b>col</b> e <b>row</b> specificati.            |
| expected_value | Il valore previsto per la distribuzione in corrispondenza di <b>col</b> e <b>row</b> specificati. |

**Limiti:**

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

**Esempi:**

```
Chi2Test_chi2( Grp, Grade, Count )
```

```
Chi2Test_chi2( Gender, Description, Observed, Expected )
```

**Vedere anche:**

p [Esempi di utilizzo delle funzioni chi2-test nei grafici \(page 577\)](#)

p [Esempi di utilizzo delle funzioni chi2-test negli script di caricamento dei dati \(page 581\)](#)

**Chi2Test\_df**

**Chi2Test\_df()** restituisce il valore df (gradi di libertà) aggregato di chi<sup>2</sup>-test per una o due serie di valori.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.



*Tutte le funzioni Qlik Sense del chi<sup>2</sup>-test presentano gli stessi argomenti.*

**Sintassi:**

```
Chi2Test_df(col, row, actual_value[, expected_value])
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

Argomenti

| Argomento      | Descrizione                                                                                       |
|----------------|---------------------------------------------------------------------------------------------------|
| col, row       | La riga e la colonna specificate nella matrice di valori testati.                                 |
| actual_value   | Il valore osservato dei dati in corrispondenza di <b>col</b> e <b>row</b> specificati.            |
| expected_value | Il valore previsto per la distribuzione in corrispondenza di <b>col</b> e <b>row</b> specificati. |

**Limiti:**

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

**Esempi:**

```
Chi2Test_df( Grp, Grade, Count )
```

```
Chi2Test_df( Gender, Description, Observed, Expected )
```

**Vedere anche:**

[p \*Esempi di utilizzo delle funzioni chi2-test nei grafici \(page 577\)\*](#)

[p \*Esempi di utilizzo delle funzioni chi2-test negli script di caricamento dei dati \(page 581\)\*](#)

**Chi2Test\_p** - funzione per grafici

**Chi2Test\_p()** restituisce il valore p (significatività) aggregato di chi<sup>2</sup>-test per una o due serie di valori. Il test può essere eseguito o sui valori in **actual\_value**, per testare le variazioni all'interno della matrice **col** e **row** specificata o confrontando i valori in **actual\_value** con quelli corrispondenti in **expected\_value**, se specificato.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.



*Tutte le funzioni Qlik Sense del chi<sup>2</sup>-test presentano gli stessi argomenti.*

**Sintassi:**

```
Chi2Test_p(col, row, actual_value[, expected_value])
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

Argomenti

| Argomento      | Descrizione                                                                                       |
|----------------|---------------------------------------------------------------------------------------------------|
| col, row       | La riga e la colonna specificate nella matrice di valori testati.                                 |
| actual_value   | Il valore osservato dei dati in corrispondenza di <b>col</b> e <b>row</b> specificati.            |
| expected_value | Il valore previsto per la distribuzione in corrispondenza di <b>col</b> e <b>row</b> specificati. |

**Limiti:**

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

**Esempi:**

```
Chi2Test_p( Grp, Grade, Count )  
Chi2Test_p( Gender, Description, Observed, Expected )
```

---

**Vedere anche:**

p [Esempi di utilizzo delle funzioni chi2-test nei grafici \(page 577\)](#)

p [Esempi di utilizzo delle funzioni chi2-test negli script di caricamento dei dati \(page 581\)](#)

### Funzioni di t-test

Le funzioni t-test vengono utilizzate per un'analisi statistica di due popolazioni medie. Un t-test a due campioni analizza se i due campioni sono diversi e viene generalmente utilizzato quando due distribuzioni standard presentano varianze sconosciute e quando un esperimento utilizza dimensioni ridotte per il campione.

Nelle sezioni seguenti le funzioni di test statistici t-test vengono raggruppate in base al test per studenti con campioni adatto a ogni tipo di funzione.

[Creazione di un report t-test tipico \(page 583\)](#)

#### T-test con due campioni indipendenti

Le seguenti funzioni possono essere utilizzate per due t-test per studenti con campioni indipendenti.

**ttest\_conf**

**TTest\_conf** restituisce il valore aggregato dell'intervallo di confidenza di t-test per due campioni indipendenti.

TTest\_conf restituisce il valore aggregato dell'intervallo di confidenza di t-test per due campioni indipendenti. ( grp, value [, sig[, eq\_var]])

ttest\_df

**TTest\_df()** restituisce il valore aggregato (gradi di libertà) del t-test di Student per due serie indipendenti di valori.

TTest\_df() restituisce il valore aggregato (gradi di libertà) del t-test di Student per due serie indipendenti di valori. (grp, value [, eq\_var])

ttest\_dif

**TTest\_dif()** è una funzione numerica che restituisce la differenza media aggregata del t-test di Student per due serie indipendenti di valori.

TTest\_dif() è una funzione numerica che restituisce la differenza media aggregata del t-test di Student per due serie indipendenti di valori. (grp, value)

ttest\_lower

**TTest\_lower()** restituisce il valore aggregato per il limite inferiore dell'intervallo di confidenza per due serie indipendenti di valori.

TTest\_lower() restituisce il valore aggregato per il limite inferiore dell'intervallo di confidenza per due serie indipendenti di valori. (grp, value [, sig[, eq\_var]])

ttest\_sig

**TTest\_sig()** restituisce il livello di significatività a due code aggregato del t-test di Student per due serie indipendenti di valori.

TTest\_sig() restituisce il livello di significatività a due code aggregato del t-test di Student per due serie indipendenti di valori. (grp, value [, eq\_var])

ttest\_sterr

**TTest\_sterr()** restituisce l'errore standard aggregato del t-test di Student della differenza media per due serie indipendenti di valori.

TTest\_sterr() restituisce l'errore standard aggregato del t-test di Student della differenza media per due serie indipendenti di valori. (grp, value [, eq\_var])

ttest\_t

**TTest\_t()** restituisce il valore t aggregato per due serie indipendenti di valori.

TTest\_t() restituisce il valore t aggregato per due serie indipendenti di valori. (grp, value [, eq\_var])

**ttest\_upper**

**TTest\_upper()** restituisce il valore aggregato per il limite superiore dell'intervallo di confidenza per due serie indipendenti di valori.

TTest\_upper() restituisce il valore aggregato per il limite superiore dell'intervallo di confidenza per due serie indipendenti di valori. (grp, value [, sig [, eq\_var]])

**Due t-test con campioni indipendenti pesati**

Le seguenti funzioni si applicano a due t-test per studenti con campioni indipendenti in cui la serie di dati di input viene fornita in formato a due colonne pesate.

**ttestw\_conf**

**TTestw\_conf()** restituisce il valore t aggregato per due serie indipendenti di valori.

TTestw\_conf() restituisce il valore t aggregato per due serie indipendenti di valori. (weight, grp, value [, sig[, eq\_var]])

**ttestw\_df**

**TTestw\_df()** restituisce il valore df (gradi di libertà) aggregato del t-test di Student per due serie indipendenti di valori.

TTestw\_df() restituisce il valore df (gradi di libertà) aggregato del t-test di Student per due serie indipendenti di valori. (weight, grp, value [, eq\_var])

**ttestw\_dif**

**TTestw\_dif()** restituisce la differenza media aggregata del t-test di Student per due serie indipendenti di valori.

TTestw\_dif() restituisce la differenza media aggregata del t-test di Student per due serie indipendenti di valori. ( weight, grp, value)

**ttestw\_lower**

**TTestw\_lower()** restituisce il valore aggregato per il limite inferiore dell'intervallo di confidenza per due serie indipendenti di valori.

TTestw\_lower() restituisce il valore aggregato per il limite inferiore dell'intervallo di confidenza per due serie indipendenti di valori. (weight, grp, value [, sig[, eq\_var]])

**ttestw\_sig**

**TTestw\_sig()** restituisce il livello di significatività a due code aggregato del t-test di Student per due serie indipendenti di valori.

TTestw\_sig() restituisce il livello di significatività a due code aggregato del t-test di Student per due serie indipendenti di valori. ( weight, grp, value [, eq\_var])

**ttestw\_sterr**

**TTestw\_sterr()** restituisce l'errore standard aggregato del t-test di Student della differenza media per due serie indipendenti di valori.

TTestw\_sterr() restituisce l'errore standard aggregato del t-test di Student della differenza media per due serie indipendenti di valori. (weight, grp, value [, eq\_var])

**ttestw\_t**

**TTestw\_t()** restituisce il valore t aggregato per due serie indipendenti di valori.

TTestw\_t() restituisce il valore t aggregato per due serie indipendenti di valori. (weight, grp, value [, eq\_var])

**ttestw\_upper**

**TTestw\_upper()** restituisce il valore aggregato per il limite superiore dell'intervallo di confidenza per due serie indipendenti di valori.

TTestw\_upper() restituisce il valore aggregato per il limite superiore dell'intervallo di confidenza per due serie indipendenti di valori. (weight, grp, value [, sig [, eq\_var]])

### T-test con un unico campione

Le seguenti funzioni possono essere utilizzate per t-test per studenti con un unico campione.

**ttest1\_conf**

**TTest1\_conf()** restituisce il valore aggregato dell'intervallo di confidenza per una serie di valori.

TTest1\_conf() restituisce il valore aggregato dell'intervallo di confidenza per una serie di valori. (value [, sig])

**ttest1\_df**

**TTest1\_df()** restituisce il valore df (gradi di libertà) aggregato del t-test di Student per una serie di valori.

TTest1\_df() restituisce il valore df (gradi di libertà) aggregato del t-test di Student per una serie di valori. (value)

**ttest1\_dif**

**TTest1\_dif()** restituisce la differenza media aggregata del t-test di Student per una serie di valori.

TTest1\_dif() restituisce la differenza media aggregata del t-test di Student per una serie di valori. (value)

**ttest1\_lower**

**TTest1\_lower()** restituisce il valore aggregato per il limite inferiore dell'intervallo di confidenza per una serie di valori.

TTest1\_lower() restituisce il valore aggregato per il limite inferiore dell'intervallo di confidenza per una serie di valori. (value [, sig])

**ttest1\_sig**

**TTest1\_sig()** restituisce il livello di significatività a due code aggregato del t-test di Student per una serie di valori.

TTest1\_sig() restituisce il livello di significatività a due code aggregato del t-test di Student per una serie di valori. (value)

**ttest1\_sterr**

**TTest1\_sterr()** restituisce l'errore standard aggregato del t-test di Student della differenza media per una serie di valori.

TTest1\_sterr() restituisce l'errore standard aggregato del t-test di Student della differenza media per una serie di valori. (value)

**ttest1\_t**

**TTest1\_t()** restituisce il valore t aggregato per una serie di valori.

TTest1\_t() restituisce il valore t aggregato per una serie di valori. (value)

**ttest1\_upper**

**TTest1\_upper()** restituisce il valore aggregato per il limite superiore dell'intervallo di confidenza per una serie di valori.

TTest1\_upper() restituisce il valore aggregato per il limite superiore dell'intervallo di confidenza per una serie di valori. (value [, sig])

### T-test con un unico campione pesato

Le seguenti funzioni possono essere utilizzate per t-test per studenti con un unico campione in cui la serie di dati di input viene fornita in formato a due colonne pesate.

**ttest1w\_conf**

**TTest1w\_conf()** è una funzione **numerica** che restituisce il valore aggregato dell'intervallo di attendibilità per una serie di valori.

TTest1w\_conf() è una funzione numerica che restituisce il valore aggregato dell'intervallo di attendibilità per una serie di valori. (weight, value [, sig])

**ttest1w\_df**

**TTest1w\_df()** restituisce il valore df (gradi di libertà) aggregato del t-test di Student per una serie di valori.

TTest1w\_df() restituisce il valore df (gradi di libertà) aggregato del t-test di Student per una serie di valori. (weight, value)

**ttest1w\_dif**

**TTest1w\_dif()** restituisce la differenza media aggregata del t-test di Student per una serie di valori.

TTest1w\_dif() restituisce la differenza media aggregata del t-test di Student per una serie di valori. (weight, value)

### ttest1w\_lower

**TTest1w\_lower()** restituisce il valore aggregato per il limite inferiore dell'intervallo di confidenza per una serie di valori.

TTest1w\_lower() restituisce il valore aggregato per il limite inferiore dell'intervallo di confidenza per una serie di valori. (weight, value [, sig])

### ttest1w\_sig

**TTest1w\_sig()** restituisce il livello di significatività a due code aggregato del t-test di Student per una serie di valori.

TTest1w\_sig() restituisce il livello di significatività a due code aggregato del t-test di Student per una serie di valori. (weight, value)

### ttest1w\_sterr

**TTest1w\_sterr()** restituisce l'errore standard aggregato del t-test di Student della differenza media per una serie di valori.

TTest1w\_sterr() restituisce l'errore standard aggregato del t-test di Student della differenza media per una serie di valori. (weight, value)

### ttest1w\_t

**TTest1w\_t()** restituisce il valore t aggregato per una serie di valori.

TTest1w\_t() restituisce il valore t aggregato per una serie di valori. (weight, value)

### ttest1w\_upper

**TTest1w\_upper()** restituisce il valore aggregato per il limite superiore dell'intervallo di confidenza per una serie di valori.

TTest1w\_upper() restituisce il valore aggregato per il limite superiore dell'intervallo di confidenza per una serie di valori. (weight, value [, sig])

### TTest\_conf

**TTest\_conf** restituisce il valore aggregato dell'intervallo di confidenza di t-test per due campioni indipendenti.

Questa funzione viene applicata ai t-test di student con campioni indipendenti.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

### Sintassi:

**TTest\_conf** ( grp, value [, sig [, eq\_var]])

**Tipo di dati restituiti:** numerico

**Argomenti:**

Argomenti

| Argomento | Descrizione                                                                                                                                                                                                                                                                                         |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I valori campione da valutare. I valori campione devono essere raggruppati n modo logico come specificato esattamente dai due valori in <b>group</b> . Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> . |
| grp       | Il campo contenente i nomi di ciascuno dei due gruppi di esempi. Se nello script Load non è fornito un nome di campo per il gruppo, al campo verrà automaticamente assegnato il nome <b>Type</b> .                                                                                                  |
| sig       | Il livello di significatività a due code può essere specificato in <b>sig</b> . Se omesso, <b>sig</b> viene impostato su 0,025, per un intervallo di confidenza pari al 95%.                                                                                                                        |
| eq_var    | Se <b>eq_var</b> è specificato come False (0), verranno presupposte varianze separate dei due esempi. Se <b>eq_var</b> è specificato come True (1), verranno presupposte varianze identiche dei due esempi.                                                                                         |

**Limiti:**

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

**Esempi:**

```
TTest_conf( Group, value )  
TTest_conf( Group, value, sig, false )
```

**Vedere anche:**

p [Creazione di un report t-test tipico \(page 583\)](#)

**TTest\_df**

**TTest\_df()** restituisce il valore aggregato (gradi di libertà) del t-test di Student per due serie indipendenti di valori.

Questa funzione viene applicata ai t-test di student con campioni indipendenti.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

### Sintassi:

```
TTest_df (grp, value [, eq_var])
```

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento | Descrizione                                                                                                                                                                                                                                                                                         |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I valori campione da valutare. I valori campione devono essere raggruppati n modo logico come specificato esattamente dai due valori in <b>group</b> . Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> . |
| grp       | Il campo contenente i nomi di ciascuno dei due gruppi di esempi. Se nello script Load non è fornito un nome di campo per il gruppo, al campo verrà automaticamente assegnato il nome <b>Type</b> .                                                                                                  |
| eq_var    | Se <b>eq_var</b> è specificato come False (0), verranno presupposte varianze separate dei due esempi. Se <b>eq_var</b> è specificato come True (1), verranno presupposte varianze identiche dei due esempi.                                                                                         |

### Limiti:

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

### Esempi:

```
TTest_df( Group, Value )  
TTest_df( Group, Value, false )
```

### Vedere anche:

p [Creazione di un report t-test tipico \(page 583\)](#)

### TTest\_dif

**TTest\_dif()** è una funzione numerica che restituisce la differenza media aggregata del t-test di Student per due serie indipendenti di valori.

Questa funzione viene applicata ai t-test di student con campioni indipendenti.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

### Sintassi:

```
TTest_dif (grp, value [, eq_var] )
```

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento | Descrizione                                                                                                                                                                                                                                                                                         |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I valori campione da valutare. I valori campione devono essere raggruppati n modo logico come specificato esattamente dai due valori in <b>group</b> . Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> . |
| grp       | Il campo contenente i nomi di ciascuno dei due gruppi di esempi. Se nello script Load non è fornito un nome di campo per il gruppo, al campo verrà automaticamente assegnato il nome <b>Type</b> .                                                                                                  |
| eq_var    | Se <b>eq_var</b> è specificato come False (0), verranno presupposte varianze separate dei due esempi. Se <b>eq_var</b> è specificato come True (1), verranno presupposte varianze identiche dei due esempi.                                                                                         |

### Limiti:

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

### Esempi:

```
TTest_dif( Group, value )  
TTest_dif( Group, value, false )
```

### Vedere anche:

p [Creazione di un report t-test tipico \(page 583\)](#)

### TTest\_lower

**TTest\_lower()** restituisce il valore aggregato per il limite inferiore dell'intervallo di confidenza per due serie indipendenti di valori.

Questa funzione viene applicata ai t-test di student con campioni indipendenti.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

### Sintassi:

```
TTest_lower (grp, value [, sig [, eq_var]])
```

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento | Descrizione                                                                                                                                                                                                                                                                                          |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I valori campione da valutare. I valori campione devono essere raggruppati in modo logico come specificato esattamente dai due valori in <b>group</b> . Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> . |
| grp       | Il campo contenente i nomi di ciascuno dei due gruppi di esempi. Se nello script Load non è fornito un nome di campo per il gruppo, al campo verrà automaticamente assegnato il nome <b>Type</b> .                                                                                                   |
| sig       | Il livello di significatività a due code può essere specificato in <b>sig</b> . Se omissso, <b>sig</b> viene impostato su 0,025, per un intervallo di confidenza pari al 95%.                                                                                                                        |
| eq_var    | Se <b>eq_var</b> è specificato come False (0), verranno presupposte varianze separate dei due esempi. Se <b>eq_var</b> è specificato come True (1), verranno presupposte varianze identiche dei due esempi.                                                                                          |

### Limiti:

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

### Esempi:

```
TTest_lower( Group, value )  
TTest_lower( Group, value, sig, false )
```

### Vedere anche:

p [Creazione di un report t-test tipico \(page 583\)](#)

### TTest\_sig

**TTest\_sig()** restituisce il livello di significatività a due code aggregato del t-test di Student per due serie indipendenti di valori.

Questa funzione viene applicata ai t-test di student con campioni indipendenti.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

### Sintassi:

```
TTest_sig (grp, value [, eq_var])
```

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento | Descrizione                                                                                                                                                                                                                                                                                          |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I valori campione da valutare. I valori campione devono essere raggruppati in modo logico come specificato esattamente dai due valori in <b>group</b> . Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> . |
| grp       | Il campo contenente i nomi di ciascuno dei due gruppi di esempi. Se nello script Load non è fornito un nome di campo per il gruppo, al campo verrà automaticamente assegnato il nome <b>Type</b> .                                                                                                   |
| eq_var    | Se <b>eq_var</b> è specificato come False (0), verranno presupposte varianze separate dei due esempi. Se <b>eq_var</b> è specificato come True (1), verranno presupposte varianze identiche dei due esempi.                                                                                          |

### Limiti:

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

### Esempi:

```
TTest_sig( Group, value )  
TTest_sig( Group, value, false )
```

### Vedere anche:

p [Creazione di un report t-test tipico \(page 583\)](#)

### TTest\_sterr

**TTest\_sterr()** restituisce l'errore standard aggregato del t-test di Student della differenza media per due serie indipendenti di valori.

Questa funzione viene applicata ai t-test di student con campioni indipendenti.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

### Sintassi:

```
TTest_sterr (grp, value [, eq_var])
```

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento | Descrizione                                                                                                                                                                                                                                                                                          |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I valori campione da valutare. I valori campione devono essere raggruppati in modo logico come specificato esattamente dai due valori in <b>group</b> . Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> . |
| grp       | Il campo contenente i nomi di ciascuno dei due gruppi di esempi. Se nello script Load non è fornito un nome di campo per il gruppo, al campo verrà automaticamente assegnato il nome <b>Type</b> .                                                                                                   |
| eq_var    | Se <b>eq_var</b> è specificato come False (0), verranno presupposte varianze separate dei due esempi. Se <b>eq_var</b> è specificato come True (1), verranno presupposte varianze identiche dei due esempi.                                                                                          |

### Limiti:

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

### Esempi:

```
TTest_sterr( Group, value )  
TTest_sterr( Group, value, false )
```

### Vedere anche:

p [Creazione di un report t-test tipico \(page 583\)](#)

### TTest\_t

**TTest\_t()** restituisce il valore t aggregato per due serie indipendenti di valori.

Questa funzione viene applicata ai t-test di student con campioni indipendenti.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

**Sintassi:**

```
TTest_t(grp, value[, eq_var])
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

| Argomenti |                                                                                                                                                                                                                                                                                                     |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Argomento | Descrizione                                                                                                                                                                                                                                                                                         |
| value     | I valori campione da valutare. I valori campione devono essere raggruppati n modo logico come specificato esattamente dai due valori in <b>group</b> . Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> . |
| grp       | Il campo contenente i nomi di ciascuno dei due gruppi di esempi. Se nello script Load non è fornito un nome di campo per il gruppo, al campo verrà automaticamente assegnato il nome <b>Type</b> .                                                                                                  |
| eq_var    | Se <b>eq_var</b> è specificato come False (0), verranno presupposte varianze separate dei due esempi. Se <b>eq_var</b> è specificato come True (1), verranno presupposte varianze identiche dei due esempi.                                                                                         |

**Limiti:**

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

**Esempio:**

```
TTest_t( Group, Value, false )
```

**Vedere anche:**

p [Creazione di un report t-test tipico \(page 583\)](#)

**TTest\_upper**

**TTest\_upper()** restituisce il valore aggregato per il limite superiore dell'intervallo di confidenza per due serie indipendenti di valori.

Questa funzione viene applicata ai t-test di student con campioni indipendenti.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

### Sintassi:

```
TTest_upper (grp, value [, sig [, eq_var]])
```

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento | Descrizione                                                                                                                                                                                                                                                                                         |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I valori campione da valutare. I valori campione devono essere raggruppati n modo logico come specificato esattamente dai due valori in <b>group</b> . Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> . |
| grp       | Il campo contenente i nomi di ciascuno dei due gruppi di esempi. Se nello script Load non è fornito un nome di campo per il gruppo, al campo verrà automaticamente assegnato il nome <b>Type</b> .                                                                                                  |
| sig       | Il livello di significatività a due code può essere specificato in <b>sig</b> . Se omesso, <b>sig</b> viene impostato su 0,025, per un intervallo di confidenza pari al 95%.                                                                                                                        |
| eq_var    | Se <b>eq_var</b> è specificato come False (0), verranno presupposte varianze separate dei due esempi. Se <b>eq_var</b> è specificato come True (1), verranno presupposte varianze identiche dei due esempi.                                                                                         |

### Limiti:

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

### Esempi:

```
TTest_upper( Group, value )  
TTest_upper( Group, value, sig, false )
```

### Vedere anche:

p [Creazione di un report t-test tipico \(page 583\)](#)

### TTestw\_conf

**TTestw\_conf()** restituisce il valore t aggregato per due serie indipendenti di valori.

Questa funzione viene applicata ai t-test di student con due campioni indipendenti in cui la serie di dati di input viene fornita in formato a due colonne pesate.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

### Sintassi:

```
TTestw_conf (weight, grp, value [, sig [, eq_var]])
```

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento | Descrizione                                                                                                                                                                                                                                                                                          |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I valori campione da valutare. I valori campione devono essere raggruppati in modo logico come specificato esattamente dai due valori in <b>group</b> . Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> . |
| weight    | Ogni valore in <b>value</b> può essere contato una o più volte in base al corrispondente valore del peso in <b>weight</b> .                                                                                                                                                                          |
| grp       | Il campo contenente i nomi di ciascuno dei due gruppi di esempi. Se nello script Load non è fornito un nome di campo per il gruppo, al campo verrà automaticamente assegnato il nome <b>Type</b> .                                                                                                   |
| sig       | Il livello di significatività a due code può essere specificato in <b>sig</b> . Se omissso, <b>sig</b> viene impostato su 0,025, per un intervallo di confidenza pari al 95%.                                                                                                                        |
| eq_var    | Se <b>eq_var</b> è specificato come False (0), verranno presupposte varianze separate dei due esempi. Se <b>eq_var</b> è specificato come True (1), verranno presupposte varianze identiche dei due esempi.                                                                                          |

### Limiti:

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

### Esempi:

```
TTestw_conf( weight, Group, Value )  
TTestw_conf( weight, Group, Value, sig, false )
```

### Vedere anche:

p [Creazione di un report t-test tipico \(page 583\)](#)

### TTestw\_df

**TTestw\_df()** restituisce il valore df (gradi di libertà) aggregato del t-test di Student per due serie indipendenti di valori.

Questa funzione viene applicata ai t-test di student con due campioni indipendenti in cui la serie di dati di input viene fornita in formato a due colonne pesate.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

### Sintassi:

```
TTestw_df (weight, grp, value [, eq_var])
```

**Tipo di dati restituiti:** numerico

### Argomenti:

| Argomenti |                                                                                                                                                                                                                                                                                                     |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Argomento | Descrizione                                                                                                                                                                                                                                                                                         |
| weight    | Ogni valore in <b>value</b> può essere contato una o più volte in base al corrispondente valore del peso in <b>weight</b> .                                                                                                                                                                         |
| grp       | Il campo contenente i nomi di ciascuno dei due gruppi di esempi. Se nello script Load non è fornito un nome di campo per il gruppo, al campo verrà automaticamente assegnato il nome <b>Type</b> .                                                                                                  |
| value     | I valori campione da valutare. I valori campione devono essere raggruppati n modo logico come specificato esattamente dai due valori in <b>group</b> . Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> . |
| eq_var    | Se <b>eq_var</b> è specificato come False (0), verranno presupposte varianze separate dei due esempi. Se <b>eq_var</b> è specificato come True (1), verranno presupposte varianze identiche dei due esempi.                                                                                         |

### Limiti:

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

### Esempi:

```
TTestw_df( weight, Group, Value )  
TTestw_df( weight, Group, Value, false )
```

### Vedere anche:

p [Creazione di un report t-test tipico \(page 583\)](#)

### TTestw\_dif

**TTestw\_dif()** restituisce la differenza media aggregata del t-test di Student per due serie indipendenti di valori.

Questa funzione viene applicata ai t-test di student con due campioni indipendenti in cui la serie di dati di input viene fornita in formato a due colonne pesate.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

#### Sintassi:

```
TTestw_dif (weight, grp, value)
```

**Tipo di dati restituiti:** numerico

#### Argomenti:

##### Argomenti

| Argomento | Descrizione                                                                                                                                                                                                                                                                                         |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| weight    | Ogni valore in <b>value</b> può essere contato una o più volte in base al corrispondente valore del peso in <b>weight</b> .                                                                                                                                                                         |
| grp       | Il campo contenente i nomi di ciascuno dei due gruppi di esempi. Se nello script Load non è fornito un nome di campo per il gruppo, al campo verrà automaticamente assegnato il nome <b>Type</b> .                                                                                                  |
| value     | I valori campione da valutare. I valori campione devono essere raggruppati n modo logico come specificato esattamente dai due valori in <b>group</b> . Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> . |

#### Limiti:

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

#### Esempi:

```
TTestw_dif( weight, Group, Value )  
TTestw_dif( weight, Group, Value, false )
```

#### Vedere anche:

p [Creazione di un report t-test tipico \(page 583\)](#)

## TTestw\_lower

**TTestw\_lower()** restituisce il valore aggregato per il limite inferiore dell'intervallo di confidenza per due serie indipendenti di valori.

Questa funzione viene applicata ai t-test di student con due campioni indipendenti in cui la serie di dati di input viene fornita in formato a due colonne pesate.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

### Sintassi:

```
TTestw_lower (weight, grp, value [, sig [, eq_var]])
```

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento | Descrizione                                                                                                                                                                                                                                                                                          |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| weight    | Ogni valore in <b>value</b> può essere contato una o più volte in base al corrispondente valore del peso in <b>weight</b> .                                                                                                                                                                          |
| grp       | Il campo contenente i nomi di ciascuno dei due gruppi di esempi. Se nello script Load non è fornito un nome di campo per il gruppo, al campo verrà automaticamente assegnato il nome <b>Type</b> .                                                                                                   |
| value     | I valori campione da valutare. I valori campione devono essere raggruppati in modo logico come specificato esattamente dai due valori in <b>group</b> . Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> . |
| sig       | Il livello di significatività a due code può essere specificato in <b>sig</b> . Se omesso, <b>sig</b> viene impostato su 0,025, per un intervallo di confidenza pari al 95%.                                                                                                                         |
| eq_var    | Se <b>eq_var</b> è specificato come False (0), verranno presupposte varianze separate dei due esempi. Se <b>eq_var</b> è specificato come True (1), verranno presupposte varianze identiche dei due esempi.                                                                                          |

### Limiti:

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

### Esempi:

```
TTestw_lower( weight, Group, value )
TTestw_lower( weight, Group, value, sig, false )
```

**Vedere anche:**

p [Creazione di un report t-test tipico \(page 583\)](#)

**TTestw\_sig**

**TTestw\_sig()** restituisce il livello di significatività a due code aggregato del t-test di Student per due serie indipendenti di valori.

Questa funzione viene applicata ai t-test di student con due campioni indipendenti in cui la serie di dati di input viene fornita in formato a due colonne pesate.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

**Sintassi:**

```
TTestw_sig ( weight, grp, value [, eq_var])
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

## Argomenti

| Argomento | Descrizione                                                                                                                                                                                                                                                                                         |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| weight    | Ogni valore in <b>value</b> può essere contato una o più volte in base al corrispondente valore del peso in <b>weight</b> .                                                                                                                                                                         |
| grp       | Il campo contenente i nomi di ciascuno dei due gruppi di esempi. Se nello script Load non è fornito un nome di campo per il gruppo, al campo verrà automaticamente assegnato il nome <b>Type</b> .                                                                                                  |
| value     | I valori campione da valutare. I valori campione devono essere raggruppati n modo logico come specificato esattamente dai due valori in <b>group</b> . Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> . |
| eq_var    | Se <b>eq_var</b> è specificato come False (0), verranno presupposte varianze separate dei due esempi. Se <b>eq_var</b> è specificato come True (1), verranno presupposte varianze identiche dei due esempi.                                                                                         |

**Limiti:**

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

**Esempi:**

```
TTestw_sig( weight, Group, Value )
TTestw_sig( weight, Group, Value, false )
```

**Vedere anche:**

[p Creazione di un report t-test tipico \(page 583\)](#)

**TTestw\_sterr**

**TTestw\_sterr()** restituisce l'errore standard aggregato del t-test di Student della differenza media per due serie indipendenti di valori.

Questa funzione viene applicata ai t-test di student con due campioni indipendenti in cui la serie di dati di input viene fornita in formato a due colonne pesate.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

**Sintassi:**

```
TTestw_sterr (weight, grp, value [, eq_var])
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

## Argomenti

| Argomento | Descrizione                                                                                                                                                                                                                                                                                          |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| weight    | Ogni valore in <b>value</b> può essere contato una o più volte in base al corrispondente valore del peso in <b>weight</b> .                                                                                                                                                                          |
| grp       | Il campo contenente i nomi di ciascuno dei due gruppi di esempi. Se nello script Load non è fornito un nome di campo per il gruppo, al campo verrà automaticamente assegnato il nome <b>Type</b> .                                                                                                   |
| value     | I valori campione da valutare. I valori campione devono essere raggruppati in modo logico come specificato esattamente dai due valori in <b>group</b> . Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> . |
| eq_var    | Se <b>eq_var</b> è specificato come False (0), verranno presupposte varianze separate dei due esempi. Se <b>eq_var</b> è specificato come True (1), verranno presupposte varianze identiche dei due esempi.                                                                                          |

### Limiti:

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

### Esempi:

```
TTestw_sterr( weight, Group, value )  
TTestw_sterr( weight, Group, value, false )
```

### Vedere anche:

p [Creazione di un report t-test tipico \(page 583\)](#)

### TTestw\_t

**TTestw\_t()** restituisce il valore t aggregato per due serie indipendenti di valori.

Questa funzione viene applicata ai t-test di student con due campioni indipendenti in cui la serie di dati di input viene fornita in formato a due colonne pesate.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

### Sintassi:

```
ttestw_t (weight, grp, value [, eq_var])
```

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento | Descrizione                                                                                                                                                                                                                                                                                         |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I valori campione da valutare. I valori campione devono essere raggruppati n modo logico come specificato esattamente dai due valori in <b>group</b> . Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> . |
| weight    | Ogni valore in <b>value</b> può essere contato una o più volte in base al corrispondente valore del peso in <b>weight</b> .                                                                                                                                                                         |
| grp       | Il campo contenente i nomi di ciascuno dei due gruppi di esempi. Se nello script Load non è fornito un nome di campo per il gruppo, al campo verrà automaticamente assegnato il nome <b>Type</b> .                                                                                                  |

| Argomento | Descrizione                                                                                                                                                                                                 |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| eq_var    | Se <b>eq_var</b> è specificato come False (0), verranno presupposte varianze separate dei due esempi. Se <b>eq_var</b> è specificato come True (1), verranno presupposte varianze identiche dei due esempi. |

**Limiti:**

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

**Esempi:**

```
TTestw_t( weight, Group, value )
TTestw_t( weight, Group, value, false )
```

**Vedere anche:**

p [Creazione di un report t-test tipico \(page 583\)](#)

**TTestw\_upper**

**TTestw\_upper()** restituisce il valore aggregato per il limite superiore dell'intervallo di confidenza per due serie indipendenti di valori.

Questa funzione viene applicata ai t-test di student con due campioni indipendenti in cui la serie di dati di input viene fornita in formato a due colonne pesate.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

**Sintassi:**

```
TTestw_upper (weight, grp, value [, sig [, eq_var]])
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

## Argomenti

| Argomento | Descrizione                                                                                                                 |
|-----------|-----------------------------------------------------------------------------------------------------------------------------|
| weight    | Ogni valore in <b>value</b> può essere contato una o più volte in base al corrispondente valore del peso in <b>weight</b> . |

| Argomento | Descrizione                                                                                                                                                                                                                                                                                          |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| grp       | Il campo contenente i nomi di ciascuno dei due gruppi di esempi. Se nello script Load non è fornito un nome di campo per il gruppo, al campo verrà automaticamente assegnato il nome <b>Type</b> .                                                                                                   |
| value     | I valori campione da valutare. I valori campione devono essere raggruppati in modo logico come specificato esattamente dai due valori in <b>group</b> . Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> . |
| sig       | Il livello di significatività a due code può essere specificato in <b>sig</b> . Se omissso, <b>sig</b> viene impostato su 0,025, per un intervallo di confidenza pari al 95%.                                                                                                                        |
| eq_var    | Se <b>eq_var</b> è specificato come False (0), verranno presupposte varianze separate dei due esempi. Se <b>eq_var</b> è specificato come True (1), verranno presupposte varianze identiche dei due esempi.                                                                                          |

### Limiti:

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

### Esempi:

```
TTestw_upper( weight, Group, value )  
TTestw_upper( weight, Group, value, sig, false )
```

### Vedere anche:

p [Creazione di un report t-test tipico \(page 583\)](#)

### TTest1\_conf

**TTest1\_conf()** restituisce il valore aggregato dell'intervallo di confidenza per una serie di valori.

Questa funzione viene applicata ai t-test di student con un campione.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

### Sintassi:

```
TTest1_conf (value [, sig ])
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

Argomenti

| Argomento | Descrizione                                                                                                                                                                   |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I campioni da valutare. Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> .          |
| sig       | Il livello di significatività a due code può essere specificato in <b>sig</b> . Se omissso, <b>sig</b> viene impostato su 0,025, per un intervallo di confidenza pari al 95%. |

**Limiti:**

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

**Esempi:**

```
TTest1_conf( value )  
TTest1_conf( value, 0.005 )
```

---

**Vedere anche:**

p [Creazione di un report t-test tipico \(page 583\)](#)

**TTest1\_df**

**TTest1\_df()** restituisce il valore df (gradi di libertà) aggregato del t-test di Student per una serie di valori.

Questa funzione viene applicata ai t-test di student con un campione.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

**Sintassi:**

```
TTest1_df (value)
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

Argomenti

| Argomento | Descrizione                                                                                                                                                          |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I campioni da valutare. Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> . |

**Limiti:**

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

**Esempio:**

```
TTest1_df( value )
```

---

**Vedere anche:**

p [Creazione di un report t-test tipico \(page 583\)](#)

**TTest1\_dif**

**TTest1\_dif()** restituisce la differenza media aggregata del t-test di Student per una serie di valori.

Questa funzione viene applicata ai t-test di student con un campione.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

**Sintassi:**

```
TTest1_dif (value)
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

Argomenti

| Argomento | Descrizione                                                                                                                                                          |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I campioni da valutare. Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> . |

### Limiti:

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

### Esempio:

```
TTest1_dif( value )
```

---

### Vedere anche:

p [Creazione di un report t-test tipico \(page 583\)](#)

### TTest1\_lower

**TTest1\_lower()** restituisce il valore aggregato per il limite inferiore dell'intervallo di confidenza per una serie di valori.

Questa funzione viene applicata ai t-test di student con un campione.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

### Sintassi:

```
TTest1_lower (value [, sig])
```

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento | Descrizione                                                                                                                                                                   |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I campioni da valutare. Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> .          |
| sig       | Il livello di significatività a due code può essere specificato in <b>sig</b> . Se omissso, <b>sig</b> viene impostato su 0,025, per un intervallo di confidenza pari al 95%. |

### Limiti:

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

### Esempi:

```
TTest1_lower( value )  
TTest1_lower( value, 0.005 )
```

### Vedere anche:

p [Creazione di un report t-test tipico \(page 583\)](#)

### TTest1\_sig

**TTest1\_sig()** restituisce il livello di significatività a due code aggregato del t-test di Student per una serie di valori.

Questa funzione viene applicata ai t-test di student con un campione.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

### Sintassi:

```
TTest1_sig (value)
```

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento | Descrizione                                                                                                                                                          |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I campioni da valutare. Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> . |

### Limiti:

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

### Esempio:

```
TTest1_sig( value )
```

### Vedere anche:

p [Creazione di un report t-test tipico \(page 583\)](#)

### TTest1\_sterr

**TTest1\_sterr()** restituisce l'errore standard aggregato del t-test di Student della differenza media per una serie di valori.

Questa funzione viene applicata ai t-test di student con un campione.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

### Sintassi:

```
TTest1_sterr (value)
```

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento | Descrizione                                                                                                                                                          |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I campioni da valutare. Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> . |

### Limiti:

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

### Esempio:

```
TTest1_sterr( value )
```

### Vedere anche:

p [Creazione di un report t-test tipico \(page 583\)](#)

### TTest1\_t

**TTest1\_t()** restituisce il valore t aggregato per una serie di valori.

Questa funzione viene applicata ai t-test di student con un campione.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

### Sintassi:

```
TTest1_t (value)
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

Argomenti

| Argomento | Descrizione                                                                                                                                                          |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I campioni da valutare. Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> . |

**Limiti:**

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

**Esempio:**

```
TTest1_t( value )
```

**Vedere anche:**

p [Creazione di un report t-test tipico \(page 583\)](#)

**TTest1\_upper**

**TTest1\_upper()** restituisce il valore aggregato per il limite superiore dell'intervallo di confidenza per una serie di valori.

Questa funzione viene applicata ai t-test di student con un campione.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

**Sintassi:**

```
TTest1_upper (value [, sig])
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

Argomenti

| Argomento | Descrizione                                                                                                                                                          |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I campioni da valutare. Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> . |

| Argomento | Descrizione                                                                                                                                                                   |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| sig       | Il livello di significatività a due code può essere specificato in <b>sig</b> . Se omissso, <b>sig</b> viene impostato su 0,025, per un intervallo di confidenza pari al 95%. |

### Limiti:

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

### Esempi:

```
TTest1_upper( value )
TTest1_upper( value, 0.005 )
```

### Vedere anche:

p [Creazione di un report t-test tipico \(page 583\)](#)

### TTest1w\_conf

**TTest1w\_conf()** è una funzione **numerica** che restituisce il valore aggregato dell'intervallo di attendibilità per una serie di valori.

Questa funzione viene applicata ai t-test di student con un campione in cui la serie di dati di input viene fornita in formato a due colonne pesate.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

### Sintassi:

```
TTest1w_conf (weight, value [, sig ])
```

**Tipo di dati restituiti:** numerico

### Argomenti:

| Argomenti |                                                                                                                                                                               |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Argomento | Descrizione                                                                                                                                                                   |
| value     | I campioni da valutare. Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> .          |
| weight    | Ogni valore in <b>value</b> può essere contato una o più volte in base al corrispondente valore del peso in <b>weight</b> .                                                   |
| sig       | Il livello di significatività a due code può essere specificato in <b>sig</b> . Se omissso, <b>sig</b> viene impostato su 0,025, per un intervallo di confidenza pari al 95%. |

### Limiti:

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

### Esempi:

```
TTest1w_conf( weight, value )  
TTest1w_conf( weight, value, 0.005 )
```

### Vedere anche:

p [Creazione di un report t-test tipico \(page 583\)](#)

### TTest1w\_df

**TTest1w\_df()** restituisce il valore df (gradi di libertà) aggregato del t-test di Student per una serie di valori.

Questa funzione viene applicata ai t-test di student con un campione in cui la serie di dati di input viene fornita in formato a due colonne pesate.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

### Sintassi:

```
TTest1w_df (weight, value)
```

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento | Descrizione                                                                                                                                                          |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I campioni da valutare. Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> . |
| weight    | Ogni valore in <b>value</b> può essere contato una o più volte in base al corrispondente valore del peso in <b>weight</b> .                                          |

### Limiti:

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

### Esempio:

```
TTest1w_df( weight, value )
```

### Vedere anche:

p [Creazione di un report t-test tipico \(page 583\)](#)

### TTest1w\_dif

**TTest1w\_dif()** restituisce la differenza media aggregata del t-test di Student per una serie di valori.

Questa funzione viene applicata ai t-test di student con un campione in cui la serie di dati di input viene fornita in formato a due colonne pesate.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

### Sintassi:

```
TTest1w_dif (weight, value)
```

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento | Descrizione                                                                                                                                                          |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I campioni da valutare. Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> . |
| weight    | Ogni valore in <b>value</b> può essere contato una o più volte in base al corrispondente valore del peso in <b>weight</b> .                                          |

### Limiti:

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

### Esempio:

```
TTest1w_dif( weight, value )
```

### Vedere anche:

p [Creazione di un report t-test tipico \(page 583\)](#)

### TTest1w\_lower

**TTest1w\_lower()** restituisce il valore aggregato per il limite inferiore dell'intervallo di confidenza per una serie di valori.

Questa funzione viene applicata ai t-test di student con un campione in cui la serie di dati di input viene fornita in formato a due colonne pesate.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

### Sintassi:

```
TTest1w_lower (weight, value [, sig ])
```

**Tipo di dati restituiti:** numerico

### Argomenti:

| Argomenti |                                                                                                                                                                               |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Argomento | Descrizione                                                                                                                                                                   |
| value     | I campioni da valutare. Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> .          |
| weight    | Ogni valore in <b>value</b> può essere contato una o più volte in base al corrispondente valore del peso in <b>weight</b> .                                                   |
| sig       | Il livello di significatività a due code può essere specificato in <b>sig</b> . Se omissso, <b>sig</b> viene impostato su 0,025, per un intervallo di confidenza pari al 95%. |

### Limiti:

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

### Esempi:

```
TTest1w_lower( weight, value )  
TTest1w_lower( weight, value, 0.005 )
```

### Vedere anche:

p [Creazione di un report t-test tipico \(page 583\)](#)

### TTest1w\_sig

**TTest1w\_sig()** restituisce il livello di significatività a due code aggregato del t-test di Student per una serie di valori.

Questa funzione viene applicata ai t-test di student con un campione in cui la serie di dati di input viene fornita in formato a due colonne pesate.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

### Sintassi:

```
TTest1w_sig (weight, value)
```

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento | Descrizione                                                                                                                                                          |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I campioni da valutare. Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> . |
| weight    | Ogni valore in <b>value</b> può essere contato una o più volte in base al corrispondente valore del peso in <b>weight</b> .                                          |

### Limiti:

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

### Esempio:

```
TTest1w_sig( weight, value )
```

### Vedere anche:

p [Creazione di un report t-test tipico \(page 583\)](#)

### TTest1w\_sterr

**TTest1w\_sterr()** restituisce l'errore standard aggregato del t-test di Student della differenza media per una serie di valori.

Questa funzione viene applicata ai t-test di student con un campione in cui la serie di dati di input viene fornita in formato a due colonne pesate.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

### Sintassi:

```
TTest1w_sterr (weight, value)
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

Argomenti

| Argomento | Descrizione                                                                                                                                                          |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I campioni da valutare. Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> . |
| weight    | Ogni valore in <b>value</b> può essere contato una o più volte in base al corrispondente valore del peso in <b>weight</b> .                                          |

**Limiti:**

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

**Esempio:**

```
TTest1w_sterr( weight, value )
```

---

**Vedere anche:**

p [Creazione di un report t-test tipico \(page 583\)](#)

**TTest1w\_t**

**TTest1w\_t()** restituisce il valore t aggregato per una serie di valori.

Questa funzione viene applicata ai t-test di student con un campione in cui la serie di dati di input viene fornita in formato a due colonne pesate.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

**Sintassi:**

```
TTest1w_t ( weight, value)
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

Argomenti

| Argomento | Descrizione                                                                                                                                                          |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I campioni da valutare. Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> . |
| weight    | Ogni valore in <b>value</b> può essere contato una o più volte in base al corrispondente valore del peso in <b>weight</b> .                                          |

**Limiti:**

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

**Esempio:**

```
TTest1w_t( weight, value )
```

---

**Vedere anche:**

p [Creazione di un report t-test tipico \(page 583\)](#)

**TTest1w\_upper**

**TTest1w\_upper()** restituisce il valore aggregato per il limite superiore dell'intervallo di confidenza per una serie di valori.

Questa funzione viene applicata ai t-test di student con un campione in cui la serie di dati di input viene fornita in formato a due colonne pesate.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

**Sintassi:**

```
TTest1w_upper (weight, value [, sig])
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

Argomenti

| Argomento | Descrizione                                                                                                                                                                   |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I campioni da valutare. Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> .          |
| weight    | Ogni valore in <b>value</b> può essere contato una o più volte in base al corrispondente valore del peso in <b>weight</b> .                                                   |
| sig       | Il livello di significatività a due code può essere specificato in <b>sig</b> . Se omissso, <b>sig</b> viene impostato su 0,025, per un intervallo di confidenza pari al 95%. |

**Limiti:**

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

**Esempi:**

```
TTest1w_upper( weight, value )  
TTest1w_upper( weight, value, 0.005 )
```

---

**Vedere anche:**

p [Creazione di un report t-test tipico \(page 583\)](#)

### Funzioni di z-test

Un'analisi statistica di due popolazioni medie. Uno z-test con due campioni analizza se i due campioni sono differenti e viene comunemente utilizzato quando due distribuzioni standard presentano varianze conosciute e quando un esperimento utilizza dimensioni notevoli per il campione.

Le funzioni di test statistici z-test vengono raggruppate in base al tipo di serie di dati di input adatto alla funzione.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

[Esempi di utilizzo delle funzioni z-test \(page 587\)](#)

### Funzioni per il formato a colonna singola

Le funzioni seguenti vengono utilizzate per z-test con serie di dati di input semplici.

#### ztest\_conf

**ZTest\_conf()** restituisce il valore z aggregato per una serie di valori.

ZTest\_conf() restituisce il valore z aggregato per una serie di valori.  
(value [, sigma [, sig ]])

#### ztest\_dif

**ZTest\_dif()** restituisce la differenza media aggregata di z-test per una serie di valori.

ZTest dif() restituisce la differenza media aggregata di z-test per una serie di valori. (value [, sigma])

#### ztest\_sig

**ZTest\_sig()** restituisce il livello di significatività a due code aggregato di z-test per una serie di valori.

ZTest sig() restituisce il livello di significatività a due code aggregato di z-test per una serie di valori. (value [, sigma])

#### ztest\_sterr

**ZTest\_sterr()** restituisce l'errore standard aggregato di z-test della differenza media per una serie di valori.

ZTest sterr() restituisce l'errore standard aggregato di z-test della differenza media per una serie di valori. (value [, sigma])

#### ztest\_z

**ZTest\_z()** restituisce il valore z aggregato per una serie di valori.

ZTest z() restituisce il valore z aggregato per una serie di valori. (value [, sigma])

#### ztest\_lower

**ZTest\_lower()** restituisce il valore aggregato per il limite inferiore dell'intervallo di confidenza per due serie indipendenti di valori.

ZTest lower() restituisce il valore aggregato per il limite inferiore dell'intervallo di confidenza per due serie indipendenti di valori. (grp, value [, sig [, eq\_var]])

#### ztest\_upper

**ZTest\_upper()** restituisce il valore aggregato per il limite superiore dell'intervallo di confidenza per due serie indipendenti di valori.

ZTest upper() restituisce il valore aggregato per il limite superiore dell'intervallo di confidenza per due serie indipendenti di valori. (grp, value [, sig [, eq\_var]])

### Funzioni per il formato a due colonne pesate

Le funzioni seguenti vengono utilizzate per z-test in cui la serie di dati di input viene fornita in un formato a due colonne pesate.

#### ztestw\_conf

**ZTestw\_conf()** restituisce il valore z aggregato dell'intervallo di confidenza per una serie di valori.

**ZTestw\_conf() restituisce il valore z aggregato dell'intervallo di confidenza per una serie di valori.** (weight, value [, sigma [, sig]])

#### ztestw\_dif

**ZTestw\_dif()** restituisce la differenza media aggregata di z-test per una serie di valori.

**ZTestw\_dif() restituisce la differenza media aggregata di z-test per una serie di valori.** (weight, value [, sigma])

#### ztestw\_lower

**ZTestw\_lower()** restituisce il valore aggregato per il limite inferiore dell'intervallo di confidenza per due serie indipendenti di valori.

**ZTestw\_lower() restituisce il valore aggregato per il limite inferiore dell'intervallo di confidenza per due serie indipendenti di valori.** (weight, value [, sigma])

#### ztestw\_sig

**ZTestw\_sig()** restituisce il livello di significatività a due code aggregato di z-test per una serie di valori.

**ZTestw\_sig() restituisce il livello di significatività a due code aggregato di z-test per una serie di valori.** (weight, value [, sigma])

#### ztestw\_sterr

**ZTestw\_sterr()** restituisce l'errore standard aggregato di z-test della differenza media per una serie di valori.

**ZTestw\_sterr() restituisce l'errore standard aggregato di z-test della differenza media per una serie di valori.** (weight, value [, sigma])

#### ztestw\_upper

**ZTestw\_upper()** restituisce il valore aggregato per il limite superiore dell'intervallo di confidenza per due serie indipendenti di valori.

**ZTestw\_upper() restituisce il valore aggregato per il limite superiore dell'intervallo di confidenza per due serie indipendenti di valori.** (weight, value [, sigma])

#### ztestw\_z

**ZTestw\_z()** restituisce il valore z aggregato per una serie di valori.

**ZTestw\_z() restituisce il valore z aggregato per una serie di valori.** (weight, value [, sigma])

### ZTest\_z

**ZTest\_z()** restituisce il valore z aggregato per una serie di valori.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

#### Sintassi:

```
ZTest_z(value[, sigma])
```

**Tipo di dati restituiti:** numerico

#### Argomenti:

##### Argomenti

| Argomento | Descrizione                                                                                                                                                                     |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I valori campione da valutare. Si assume una popolazione media pari a 0. Se si desidera che il test sia eseguito su un'altra media, sottrarre quella media dai valori campione. |
| sigma     | Se conosciuta, la deviazione standard può essere dichiarata in <b>sigma</b> . Se <b>sigma</b> viene omesso, verrà utilizzata la deviazione standard del campione effettiva.     |

#### Limiti:

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

#### Esempio:

```
ZTest_z( value-TestValue )
```

#### Vedere anche:

p [Esempi di utilizzo delle funzioni z-test \(page 587\)](#)

### ZTest\_sig

**ZTest\_sig()** restituisce il livello di significatività a due code aggregato di z-test per una serie di valori.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

### Sintassi:

```
ZTest_sig(value[, sigma])
```

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento | Descrizione                                                                                                                                                                     |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I valori campione da valutare. Si assume una popolazione media pari a 0. Se si desidera che il test sia eseguito su un'altra media, sottrarre quella media dai valori campione. |
| sigma     | Se conosciuta, la deviazione standard può essere dichiarata in <b>sigma</b> . Se <b>sigma</b> viene omissso, verrà utilizzata la deviazione standard del campione effettiva.    |

### Limiti:

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

### Esempio:

```
ZTest_sig(Value-TestValue)
```

### Vedere anche:

p [Esempi di utilizzo delle funzioni z-test \(page 587\)](#)

### ZTest\_dif

**ZTest\_dif()** restituisce la differenza media aggregata di z-test per una serie di valori.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

### Sintassi:

```
ZTest_dif(value[, sigma])
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

Argomenti

| Argomento | Descrizione                                                                                                                                                                     |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I valori campione da valutare. Si assume una popolazione media pari a 0. Se si desidera che il test sia eseguito su un'altra media, sottrarre quella media dai valori campione. |
| sigma     | Se conosciuta, la deviazione standard può essere dichiarata in <b>sigma</b> . Se <b>sigma</b> viene omissso, verrà utilizzata la deviazione standard del campione effettiva.    |

**Limiti:**

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

**Esempio:**

```
ZTest_dif(Value-TestValue)
```

---

**Vedere anche:**

p [Esempi di utilizzo delle funzioni z-test \(page 587\)](#)

**ZTest\_sterr**

**ZTest\_sterr()** restituisce l'errore standard aggregato di z-test della differenza media per una serie di valori.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

**Sintassi:**

```
ZTest_sterr(value[, sigma])
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

Argomenti

| Argomento | Descrizione                                                                                                                                                                     |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I valori campione da valutare. Si assume una popolazione media pari a 0. Se si desidera che il test sia eseguito su un'altra media, sottrarre quella media dai valori campione. |
| sigma     | Se conosciuta, la deviazione standard può essere dichiarata in <b>sigma</b> . Se <b>sigma</b> viene omesso, verrà utilizzata la deviazione standard del campione effettiva.     |

**Limiti:**

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

**Esempio:**

```
ZTest_sterr(Value-TestValue)
```

---

**Vedere anche:**

p [Esempi di utilizzo delle funzioni z-test \(page 587\)](#)

**ZTest\_conf**

**ZTest\_conf()** restituisce il valore z aggregato per una serie di valori.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

**Sintassi:**

```
ZTest_conf (value[, sigma[, sig]])
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

Argomenti

| Argomento | Descrizione                                                                                                                                                                     |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I valori campione da valutare. Si assume una popolazione media pari a 0. Se si desidera che il test sia eseguito su un'altra media, sottrarre quella media dai valori campione. |
| sigma     | Se conosciuta, la deviazione standard può essere dichiarata in <b>sigma</b> . Se <b>sigma</b> viene omissso, verrà utilizzata la deviazione standard del campione effettiva.    |
| sig       | Il livello di significatività a due code può essere specificato in <b>sig</b> . Se omissso, <b>sig</b> viene impostato su 0,025, per un intervallo di confidenza pari al 95%.   |

**Limiti:**

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

**Esempio:**

```
ZTest_conf(Value-TestValue)
```

---

**Vedere anche:**

p [Esempi di utilizzo delle funzioni z-test \(page 587\)](#)

**ZTest\_lower**

**ZTest\_lower()** restituisce il valore aggregato per il limite inferiore dell'intervallo di confidenza per due serie indipendenti di valori.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

**Sintassi:**

```
ZTest_lower (grp, value [, sig [, eq_var]])
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

Argomenti

| Argomento | Descrizione                                                                                                                                                                                                                                                                                         |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I valori campione da valutare. I valori campione devono essere raggruppati n modo logico come specificato esattamente dai due valori in <b>group</b> . Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> . |
| grp       | Il campo contenente i nomi di ciascuno dei due gruppi di esempi. Se nello script Load non è fornito un nome di campo per il gruppo, al campo verrà automaticamente assegnato il nome <b>Type</b> .                                                                                                  |
| sig       | Il livello di significatività a due code può essere specificato in <b>sig</b> . Se omesso, <b>sig</b> viene impostato su 0,025, per un intervallo di confidenza pari al 95%.                                                                                                                        |
| eq_var    | Se <b>eq_var</b> è specificato come False (0), verranno presupposte varianze separate dei due esempi. Se <b>eq_var</b> è specificato come True (1), verranno presupposte varianze identiche dei due esempi.                                                                                         |

**Limiti:**

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

**Esempi:**

```
ZTest_lower( Group, Value )  
ZTest_lower( Group, Value, sig, false )
```

**Vedere anche:**

p [Esempi di utilizzo delle funzioni z-test \(page 587\)](#)

**ZTest\_upper**

**ZTest\_upper()** restituisce il valore aggregato per il limite superiore dell'intervallo di confidenza per due serie indipendenti di valori.

Questa funzione viene applicata ai t-test di student con campioni indipendenti.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

### Sintassi:

```
ZTest_upper (grp, value [, sig [, eq_var]])
```

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento | Descrizione                                                                                                                                                                                                                                                                                          |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I valori campione da valutare. I valori campione devono essere raggruppati in modo logico come specificato esattamente dai due valori in <b>group</b> . Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> . |
| grp       | Il campo contenente i nomi di ciascuno dei due gruppi di esempi. Se nello script Load non è fornito un nome di campo per il gruppo, al campo verrà automaticamente assegnato il nome <b>Type</b> .                                                                                                   |
| sig       | Il livello di significatività a due code può essere specificato in <b>sig</b> . Se omissso, <b>sig</b> viene impostato su 0,025, per un intervallo di confidenza pari al 95%.                                                                                                                        |
| eq_var    | Se <b>eq_var</b> è specificato come False (0), verranno presupposte varianze separate dei due esempi. Se <b>eq_var</b> è specificato come True (1), verranno presupposte varianze identiche dei due esempi.                                                                                          |

### Limiti:

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

### Esempi:

```
ZTest_upper( Group, value )  
ZTest_upper( Group, value, sig, false )
```

### Vedere anche:

p [Esempi di utilizzo delle funzioni z-test \(page 587\)](#)

### ZTestw\_z

**ZTestw\_z()** restituisce il valore z aggregato per una serie di valori.

Questa funzione viene applicata agli z-test in cui la serie di dati di input viene fornita in formato a due colonne pesate.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

### Sintassi:

```
ZTestw_z (weight, value [, sigma])
```

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento | Descrizione                                                                                                                                                                                      |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I valori devono essere restituiti da <b>value</b> . Si assume una media campione pari a 0. Se si desidera che il test sia eseguito su un'altra media, sottrarre quel valore dai valori campione. |
| weight    | Ogni valore campione in <b>value</b> può essere conteggiato una o più volte in base al corrispondente valore del peso in <b>weight</b> .                                                         |
| sigma     | Se conosciuta, la deviazione standard può essere dichiarata in <b>sigma</b> . Se <b>sigma</b> viene omissso, verrà utilizzata la deviazione standard del campione effettiva.                     |

### Limiti:

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

### Esempio:

```
ZTestw_z( weight, value-TestValue)
```

### Vedere anche:

p [Esempi di utilizzo delle funzioni z-test \(page 587\)](#)

### ZTestw\_sig

**ZTestw\_sig()** restituisce il livello di significatività a due code aggregato di z-test per una serie di valori.

Questa funzione viene applicata agli z-test in cui la serie di dati di input viene fornita in formato a due colonne pesate.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

### Sintassi:

```
ZTestw_sig (weight, value [, sigma])
```

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento | Descrizione                                                                                                                                                                                      |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I valori devono essere restituiti da <b>value</b> . Si assume una media campione pari a 0. Se si desidera che il test sia eseguito su un'altra media, sottrarre quel valore dai valori campione. |
| weight    | Ogni valore campione in <b>value</b> può essere conteggiato una o più volte in base al corrispondente valore del peso in <b>weight</b> .                                                         |
| sigma     | Se conosciuta, la deviazione standard può essere dichiarata in <b>sigma</b> . Se <b>sigma</b> viene omissa, verrà utilizzata la deviazione standard del campione effettiva.                      |

### Limiti:

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

### Esempio:

```
ZTestw_sig( weight, value-Testvalue)
```

### Vedere anche:

p [Esempi di utilizzo delle funzioni z-test \(page 587\)](#)

### ZTestw\_dif

**ZTestw\_dif()** restituisce la differenza media aggregata di z-test per una serie di valori.

Questa funzione viene applicata agli z-test in cui la serie di dati di input viene fornita in formato a due colonne pesate.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

### Sintassi:

```
ZTestw_dif ( weight, value [, sigma])
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

Argomenti

| Argomento | Descrizione                                                                                                                                                                                      |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I valori devono essere restituiti da <b>value</b> . Si assume una media campione pari a 0. Se si desidera che il test sia eseguito su un'altra media, sottrarre quel valore dai valori campione. |
| weight    | Ogni valore campione in <b>value</b> può essere conteggiato una o più volte in base al corrispondente valore del peso in <b>weight</b> .                                                         |
| sigma     | Se conosciuta, la deviazione standard può essere dichiarata in <b>sigma</b> . Se <b>sigma</b> viene omissso, verrà utilizzata la deviazione standard del campione effettiva.                     |

**Limiti:**

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

**Esempio:**

```
ZTestw_dif( weight, value-TestValue)
```

---

**Vedere anche:**

p [Esempi di utilizzo delle funzioni z-test \(page 587\)](#)

**ZTestw\_sterr**

**ZTestw\_sterr()** restituisce l'errore standard aggregato di z-test della differenza media per una serie di valori.

Questa funzione viene applicata agli z-test in cui la serie di dati di input viene fornita in formato a due colonne pesate.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

**Sintassi:**

```
ZTestw_sterr (weight, value [, sigma])
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

Argomenti

| Argomento | Descrizione                                                                                                                                                                                      |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I valori devono essere restituiti da <b>value</b> . Si assume una media campione pari a 0. Se si desidera che il test sia eseguito su un'altra media, sottrarre quel valore dai valori campione. |
| weight    | Ogni valore campione in <b>value</b> può essere conteggiato una o più volte in base al corrispondente valore del peso in <b>weight</b> .                                                         |
| sigma     | Se conosciuta, la deviazione standard può essere dichiarata in <b>sigma</b> . Se <b>sigma</b> viene omissa, verrà utilizzata la deviazione standard del campione effettiva.                      |

**Limiti:**

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

**Esempio:**

```
ZTestw_sterr( weight, Value-TestValue)
```

**Vedere anche:**

p [Esempi di utilizzo delle funzioni z-test \(page 587\)](#)

**ZTestw\_conf**

**ZTestw\_conf()** restituisce il valore z aggregato dell'intervallo di confidenza per una serie di valori.

Questa funzione viene applicata agli z-test in cui la serie di dati di input viene fornita in formato a due colonne pesate.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

**Sintassi:**

```
ZTest_conf( weight, value[, sigma[, sig]])
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

Argomenti

| Argomento | Descrizione                                                                                                                                                                     |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I valori campione da valutare. Si assume una popolazione media pari a 0. Se si desidera che il test sia eseguito su un'altra media, sottrarre quella media dai valori campione. |
| weight    | Ogni valore campione in <b>value</b> può essere conteggiato una o più volte in base al corrispondente valore del peso in <b>weight</b> .                                        |
| sigma     | Se conosciuta, la deviazione standard può essere dichiarata in <b>sigma</b> . Se <b>sigma</b> viene omissso, verrà utilizzata la deviazione standard del campione effettiva.    |
| sig       | Il livello di significatività a due code può essere specificato in <b>sig</b> . Se omissso, <b>sig</b> viene impostato su 0,025, per un intervallo di confidenza pari al 95%.   |

**Limiti:**

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

**Esempio:**

```
ZTestw_conf( weight, value-TestValue)
```

**Vedere anche:**

p [Esempi di utilizzo delle funzioni z-test \(page 587\)](#)

**ZTestw\_lower**

**ZTestw\_lower()** restituisce il valore aggregato per il limite inferiore dell'intervallo di confidenza per due serie indipendenti di valori.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

**Sintassi:**

```
ZTestw_lower (grp, value [, sig [, eq_var]])
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

| Argomento | Descrizione                                                                                                                                                                                                                                                                                          |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I valori campione da valutare. I valori campione devono essere raggruppati in modo logico come specificato esattamente dai due valori in <b>group</b> . Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> . |
| grp       | Il campo contenente i nomi di ciascuno dei due gruppi di esempi. Se nello script Load non è fornito un nome di campo per il gruppo, al campo verrà automaticamente assegnato il nome <b>Type</b> .                                                                                                   |
| sig       | Il livello di significatività a due code può essere specificato in <b>sig</b> . Se omesso, <b>sig</b> viene impostato su 0,025, per un intervallo di confidenza pari al 95%.                                                                                                                         |
| eq_var    | Se <b>eq_var</b> è specificato come False (0), verranno presupposte varianze separate dei due esempi. Se <b>eq_var</b> è specificato come True (1), verranno presupposte varianze identiche dei due esempi.                                                                                          |

**Limiti:**

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

**Esempi:**

```
ZTestw_lower( Group, Value )  
ZTestw_lower( Group, Value, sig, false )
```

---

**Vedere anche:**

p [Esempi di utilizzo delle funzioni z-test \(page 587\)](#)

**ZTestw\_upper**

**ZTestw\_upper()** restituisce il valore aggregato per il limite superiore dell'intervallo di confidenza per due serie indipendenti di valori.

Questa funzione viene applicata ai t-test di student con campioni indipendenti.

Se la funzione viene utilizzata nello script di caricamento dei dati, i valori verranno ripetuti su un insieme di record, come definito da una clausola group by.

Se la funzione viene utilizzata in un'espressione grafica, i valori verranno ripetuti sulle dimensioni del grafico.

### Sintassi:

```
ZTestw_upper (grp, value [, sig [, eq_var]])
```

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento | Descrizione                                                                                                                                                                                                                                                                                         |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | I valori campione da valutare. I valori campione devono essere raggruppati n modo logico come specificato esattamente dai due valori in <b>group</b> . Se nello script Load non è fornito un nome di campo per i valori di esempio, al campo verrà automaticamente assegnato il nome <b>Value</b> . |
| grp       | Il campo contenente i nomi di ciascuno dei due gruppi di esempi. Se nello script Load non è fornito un nome di campo per il gruppo, al campo verrà automaticamente assegnato il nome <b>Type</b> .                                                                                                  |
| sig       | Il livello di significatività a due code può essere specificato in <b>sig</b> . Se omesso, <b>sig</b> viene impostato su 0,025, per un intervallo di confidenza pari al 95%.                                                                                                                        |
| eq_var    | Se <b>eq_var</b> è specificato come False (0), verranno presupposte varianze separate dei due esempi. Se <b>eq_var</b> è specificato come True (1), verranno presupposte varianze identiche dei due esempi.                                                                                         |

### Limiti:

I valori di testo, i valori NULL e i valori mancanti nel valore dell'espressione determinano la restituzione di NULL.

### Esempi:

```
ZTestw_upper( Group, Value )  
ZTestw_upper( Group, Value, sig, false )
```

### Vedere anche:

p [Esempi di utilizzo delle funzioni z-test \(page 587\)](#)

## Esempi delle funzioni di test statistici

In questa sezione sono inclusi esempi di funzioni di test statistici applicati ai grafici e allo script di caricamento dei dati.

### Esempi di utilizzo delle funzioni chi2-test nei grafici

Le funzioni chi2-test vengono utilizzate per trovare i valori associati all'analisi statistica del chi quadrato.

In questa sezione viene descritto come creare visualizzazioni utilizzando dati campione per trovare i valori delle funzioni del test di distribuzione del chi quadrato disponibili in Qlik Sense. Per le descrizioni della sintassi e degli argomenti, fare riferimento ai singoli argomenti delle funzioni dei grafici di chi2-test.

### Caricamento dei dati per i campioni

Esistono tre set di dati campione che descrivono tre differenti campioni statistici da caricare nello script.

Procedere come indicato di seguito:

1. Creare una nuova app.

Nell'editor caricamento dati immettere quanto segue:

```
// Sample_1 data is pre-aggregated... Note: make sure you set your DecimalSep='.' at the top of the script.
```

- 2.

Sample\_1:

```
LOAD * inline [
```

```
Grp,Grade,Count
```

```
I,A,15
```

```
I,B,7
```

```
I,C,9
```

```
I,D,20
```

```
I,E,26
```

```
I,F,19
```

```
II,A,10
```

```
II,B,11
```

```
II,C,7
```

```
II,D,15
```

```
II,E,21
```

```
II,F,16
```

```
];
```

```
// Sample_2 data is pre-aggregated: If raw data is used, it must be aggregated using count()...
```

Sample\_2:

```
LOAD * inline [
```

```
Sex,Opinion,OpCount

1,2,58

1,1,11

1,0,10

2,2,35

2,1,25

2,0,23 ] (delimiter is ',' );

// Sample_3a data is transformed using the crosstable statement...

Sample_3a:

crosstable(Gender, Actual) LOAD

Description,

[Men (Actual)] as Men,

[Women (Actual)] as women;

LOAD * inline [

Men (Actual),women (Actual),Description

58,35,Agree

11,25,Neutral

10,23,Disagree ] (delimiter is ',' );

// Sample_3b data is transformed using the crosstable statement...

Sample_3b:

crosstable(Gender, Expected) LOAD

Description,

[Men (Expected)] as Men,

[Women (Expected)] as Women;

LOAD * inline [

Men (Expected),Women (Expected),Description

45.35,47.65,Agree
```

```
17.56,18.44,Neutral
```

```
16.09,16.91,Disagree ] (delimiter is ',');
```

```
// Sample_3a and Sample_3b will result in a (fairly harmless) Synthetic Key...
```

3. Fare clic su  per caricare dati.

### Creazione di visualizzazioni delle funzioni grafiche chi2-test

#### Esempio: Campione 1

Procedere come indicato di seguito:

1. Nell'editor caricamento dati fare clic su  per accedere alla visualizzazione dell'app, quindi fare clic sul foglio creato in precedenza.  
Viene aperta la vista foglio.
2. Fare clic su  **Modifica foglio** per modificare il foglio.
3. Da **Grafici**, aggiungere una tabella, mentre da **Campi** aggiungere Grp, Grade e Count come dimensioni.  
In questa tabella sono mostrati i dati campione.
4. Aggiungere un'altra tabella con la seguente espressione come dimensione.  
`valueList('p', 'df', 'chi2')`  
Viene utilizzata la funzione delle dimensioni sintetiche per creare etichette per le dimensioni con i nomi delle tre funzioni chi2-test.  
Aggiungere la seguente espressione alla tabella come misura.  
`IF(ValueList('p', 'df', 'chi2')='p', Chi2Test_p(Grp, Grade, Count),`  
5. `IF(ValueList('p', 'df', 'chi2')='df', Chi2Test_df(Grp, Grade, Count),`  
`Chi2Test_chi2(Grp, Grade, Count)))`  
Ciò ha l'effetto di inserire il valore risultante di ciascuna funzione chi2-test presente nella tabella accanto alla dimensione sintetica associata.
6. Impostare la **Formattazione numero** della misura su **Numero** e **3Cifre significative**.



*Nell'espressione per la misura, è possibile utilizzare invece la seguente espressione: `Pick(Match(ValueList('p', 'df', 'chi2'), 'p', 'df', 'chi2'), Chi2Test_p(Grp, Grade, Count), Chi2Test_df(Grp, Grade, Count), Chi2Test_chi2(Grp, Grade, Count))`*

#### Risultato:

La tabella risultante per le funzioni chi2-test per i dati del Campione 1 conterrà i seguenti valori:

Tabella dei risultati

| p     | df | Chi2 |
|-------|----|------|
| 0.820 | 5  | 2.21 |

### Esempio: Campione 2

Procedere come indicato di seguito:

1. Nel foglio che si stava modificando nell'esempio Campione 1, da **Grafici** aggiungere una tabella e da **Campi** aggiungere Sex, Opinion e OpCount come dimensioni.
2. Creare una copia della tabella dei risultati del Campione 1 utilizzando i comandi **Copia** e **Incolla**. Modificare l'espressione nella misura e sostituire gli argomenti in tutte e tre le funzioni chi2-test con i nomi dei campi utilizzati nei dati del campione 2, ad esempio: `chi2Test_p(Sex,Opinion,OpCount)`.

#### Risultato:

La tabella risultante per le funzioni chi2-test per i dati del Campione 2 conterrà i seguenti valori:

| Tabella dei risultati |    |      |
|-----------------------|----|------|
| p                     | df | Chi2 |
| 0.000309              | 2  | 16.2 |

### Esempio: Campione 3

Procedere come indicato di seguito:

1. Creare altre due tabelle nello stesso modo degli esempi per i dati del Campione 1 e del Campione 2. Nella tabella delle dimensioni utilizzare i seguenti campi come dimensioni: Gender, Description, Actual e Expected.
2. Nella tabella dei risultati utilizzare i nomi dei campi utilizzati nei dati del Campione 3, ad esempio: `chi2Test_p(Gender,Description,Actual,Expected)`.

#### Risultato:

La tabella risultante per le funzioni chi2-test per i dati del Campione 3 conterrà i seguenti valori:

| Tabella dei risultati |    |      |
|-----------------------|----|------|
| p                     | df | Chi2 |
| 0.000308              | 2  | 16.2 |

### Esempi di utilizzo delle funzioni chi2-test negli script di caricamento dei dati

Le funzioni chi2-test vengono utilizzate per trovare i valori associati all'analisi statistica del chi quadrato. In questa sezione viene descritto come utilizzare le funzioni del test di distribuzione del chi quadrato disponibili nello script di caricamento dei dati in Qlik Sense. Per le descrizioni della sintassi e degli argomenti, fare riferimento ai singoli argomenti delle funzioni di script di chi2-test.

In questo esempio viene utilizzata una tabella contenente il numero di studenti che ha ottenuto un voto (da A a F) per due gruppi di studenti (I e II).

Data table

| Group | A  | B  | C | D  | E  | F  |
|-------|----|----|---|----|----|----|
| I     | 15 | 7  | 9 | 20 | 26 | 19 |
| II    | 10 | 11 | 7 | 15 | 21 | 16 |

### Caricamento dei dati campione

Procedere come indicato di seguito:

1. Creare una nuova app.

Nell'editor caricamento dati immettere quanto segue:

```
// Sample_1 data is pre-aggregated... Note: make sure you set your DecimalSep='.' at the top of the script.
```

- 2.

Sample\_1:

```
LOAD * inline [
```

```
Grp,Grade,Count
```

```
I,A,15
```

```
I,B,7
```

```
I,C,9
```

```
I,D,20
```

```
I,E,26
```

```
I,F,19
```

```
II,A,10
```

```
II,B,11
```

```
II,C,7
```

```
II,D,15
```

```
II,E,21
```

```
II,F,16
```

```
];
```

3. Fare clic su  per caricare dati.

I dati campione sono stati ora caricati.

### Caricamento dei valori delle funzioni chi2-test

Verranno ora caricati i valori di chi2-test basati sui dati campione in una nuova tabella, raggruppati per Grp.

Procedere come indicato di seguito:

Nell'editor caricamento dati aggiungere alla fine dello script il codice seguente:

```
// Sample_1 data is pre-aggregated... Note: make sure you set your DecimalSep='.' at the top of the script.
```

1.

chi2\_table:

```
LOAD Grp,
```

```
Chi2Test_chi2(Grp, Grade, Count) as chi2,
```

```
Chi2Test_df(Grp, Grade, Count) as df,
```

```
Chi2Test_p(Grp, Grade, Count) as p
```

```
resident Sample_1 group by Grp;
```

2. Fare clic su  per caricare dati.

Sono stati ora caricati i valori di chi2-test in una tabella denominata Chi2\_table.

### Risultati

È possibile ora visualizzare i valori di chi2-test risultanti nel sistema di visualizzazione modello dati in **Anteprima**. Dovrebbero ora avere l'aspetto seguente:

| Results |       |    |       |
|---------|-------|----|-------|
| Grp     | chi2  | df | p     |
| I       | 16.00 | 5  | 0.007 |
| II      | 9.40  | 5  | 0.094 |

### Creazione di un report t-test tipico

Un tipico report t-test per studenti può includere tabelle con risultati **Group Statistics** e **Independent Samples Test**.

Nelle sezioni successive verrà trattata la creazione di queste tabelle utilizzando le funzioni Qlik Sense t-test applicate a due gruppi di campioni indipendenti, Observation e Comparison. Le tabelle corrispondenti per questi campioni avranno l'aspetto seguente:

| Statistiche di gruppo |    |       |                    |                     |
|-----------------------|----|-------|--------------------|---------------------|
| Type                  | N  | Mean  | Standard Deviation | Standard Error Mean |
| Comparison            | 20 | 11.95 | 14.61245           | 3.2674431           |
| Observation           | 20 | 27.15 | 12.507997          | 2.7968933           |

Independent Sample Test

| Type                       | conf     | t     | df              | Sig. (2-tailed) | Mean Difference | Standard Error Difference | 95% Confidence Interval (Lower) | 95% Confidence Interval (Upper) |
|----------------------------|----------|-------|-----------------|-----------------|-----------------|---------------------------|---------------------------------|---------------------------------|
| Equal Variance not Assumed | 0        | 3.534 | 37.116717335823 | 0.001           | 15.2            | 4.30101                   | 6.48625                         | 23.9137                         |
| Equal Variance Assumed     | 8.706939 | 3.534 | 38              | 0.001           | 15.2            | 4.30101                   | 6.49306                         | 23.9069                         |

### Caricamento dei dati campione

Procedere come indicato di seguito:

1. Creare una nuova app utilizzando un nuovo foglio.
2. Nell'editor caricamento dati immettere quanto segue:

```
Table1:
Crosstable (Type, value)
Load recno() as ID, * inline [
Observation|Comparison
35|2
40|27
12|38
15|31
21|1
14|19
46|1
10|34
28|3
48|1
16|2
30|3
32|2
48|1
31|2
22|1
12|3
39|29
19|37
25|2 ] (delimiter is '|');
```

Nello script Load **recno()** è incluso perché **crosstable** richiede tre argomenti. Pertanto, **recno()** fornisce semplicemente un argomento aggiuntivo, in questo caso un ID per ciascuna riga. Senza di esso i valori di esempio di **Comparison** non verrebbero caricati.

3. Fare clic su  per caricare dati.

### Creazione della tabella Group statistics

Procedere come indicato di seguito:

1. Nell'editor caricamento dati, fare clic su  per accedere alla panoramica dell'app, quindi fare clic sul foglio creato in precedenza.  
Viene visualizzata la vista foglio.
2. Fare clic su  **Modifica foglio** per modificare il foglio.
3. Da **Grafici**, aggiungere una tabella, mentre da **Campi** aggiungere Type come dimensione per la tabella.
4. Aggiungere le seguenti espressioni come misure:

Espressioni di esempio

| Etichetta           | Espressione  |
|---------------------|--------------|
| N                   | Count(Value) |
| Mean                | Avg(Value)   |
| Standard Deviation  | Stdev(Value) |
| Standard Error Mean | Sterr(Value) |

5. Fare clic su **Ordinamento** e assicurarsi che Type sia nella parte superiore dell'elenco dell'ordinamento.

### Risultato:

Una tabella Group statistics per questi campioni avrà l'aspetto seguente:

Statistiche di gruppo

| Type        | N  | Mean  | Standard Deviation | Standard Error Mean |
|-------------|----|-------|--------------------|---------------------|
| Comparison  | 20 | 11.95 | 14.61245           | 3.2674431           |
| Observation | 20 | 27.15 | 12.507997          | 2.7968933           |

### Creazione della tabella Independent sample test

Procedere come indicato di seguito:

1. Fare clic su  **Modifica foglio** per modificare il foglio.
2. Da **Grafici** aggiungere una tabella con la seguente espressione come dimensione per la tabella, `=ValueList (Dual('Equal Variance not Assumed', 0), Dual('Equal Variance Assumed', 1))` e assegnarle l'etichetta Tipo.

## 3. Aggiungere le seguenti espressioni come misure:

Espressioni di esempio

| Etichetta                       | Espressione                                                                                                                                                                 |
|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| conf                            | if(ValueList (Dual('Equal Variance not Assumed', 0), Dual('Equal Variance Assumed', 1)),TTest_conf(Type, Value),TTest_conf(Type, Value, 0))                                 |
| t                               | if(ValueList (Dual('Equal Variance not Assumed', 0), Dual('Equal Variance Assumed', 1)),TTest_t(Type, Value),TTest_t(Type, Value, 0))                                       |
| df                              | if(ValueList (Dual('Equal Variance not Assumed', 0), Dual('Equal Variance Assumed', 1)),TTest_df(Type, Value),TTest_df(Type, Value, 0))                                     |
| Sig. (2-tailed)                 | if(ValueList (Dual('Equal Variance not Assumed', 0), Dual('Equal Variance Assumed', 1)),TTest_sig(Type, Value),TTest_sig(Type, Value, 0))                                   |
| Mean Difference                 | TTest_dif(Type, Value)                                                                                                                                                      |
| Standard Error Difference       | if(ValueList (Dual('Equal Variance not Assumed', 0), Dual('Equal Variance Assumed', 1)),TTest_sterr(Type, Value),TTest_sterr(Type, Value, 0))                               |
| 95% Confidence Interval (Lower) | if(ValueList (Dual('Equal Variance not Assumed', 0), Dual('Equal Variance Assumed', 1)),TTest_lower(Type, Value,(1-(95)/100)/2),TTest_lower(Type, Value,(1-(95)/100)/2, 0)) |
| 95% Confidence Interval (Upper) | if(ValueList (Dual('Equal Variance not Assumed', 0), Dual('Equal Variance Assumed', 1)),TTest_upper(Type, Value,(1-(95)/100)/2),TTest_upper(Type, Value,(1-(95)/100)/2, 0)) |

## Risultato:

## Independent Sample Test

| Type                       | conf | t     | df              | Sig. (2-tailed) | Mean Difference | Standard Error Difference | 95% Confidence Interval (Lower) | 95% Confidence Interval (Upper) |
|----------------------------|------|-------|-----------------|-----------------|-----------------|---------------------------|---------------------------------|---------------------------------|
| Equal Variance not Assumed | 0    | 3.534 | 37.116717335823 | 0.001           | 15.2            | 4.30101                   | 6.48625                         | 23.9137                         |

| Type                   | conf     | t     | df | Sig. (2-tailed) | Mean Difference | Standard Error Difference | 95% Confidence Interval (Lower) | 95% Confidence Interval (Upper) |
|------------------------|----------|-------|----|-----------------|-----------------|---------------------------|---------------------------------|---------------------------------|
| Equal Variance Assumed | 8.706939 | 3.534 | 38 | 0.001           | 15.2            | 4.30101                   | 6.49306                         | 23.9069                         |

### Esempi di utilizzo delle funzioni z-test

Le funzioni z-test vengono utilizzate per individuare valori associati con l'analisi statistica z-test per campioni con grandi quantità di dati, in generale superiori a 30, e di cui si conosce la varianza.

In questa sezione viene descritta la procedura di creazione delle visualizzazioni mediante dati campione per trovare i valori delle funzioni z-test disponibili in Qlik Sense. Per le descrizioni della sintassi e degli argomenti, fare riferimento ai singoli argomenti delle funzioni dei grafici di z-test.

### Caricamento dei dati campione

I dati campione utilizzati in questo esempio sono identici a quelli utilizzati negli esempi delle funzioni t-test. Di norma, le dimensioni dei dati campione risulterebbero troppo ridotte per le analisi z-test, tuttavia sono sufficienti per illustrare l'utilizzo delle diverse funzioni z-test in Qlik Sense.

Procedere come indicato di seguito:

1. Creare una nuova app utilizzando un nuovo foglio.



*Se è stata creata un'app per le funzioni t-test, è possibile utilizzarla e creare un nuovo foglio per queste funzioni.*

2. Nell'editor caricamento dati immettere quanto segue:

```
Table1:
Crosstable (Type, Value)
Load recno() as ID, * inline [
Observation|Comparison
35|2
40|27
12|38
15|31
21|1
14|19
46|1
10|34
28|3
48|1
```

```

16|2
30|3
32|2
48|1
31|2
22|1
12|3
39|29
19|37
25|2 ] (delimiter is '|');

```

Nello script Load **recno()** è incluso perché **crosstable** richiede tre argomenti. Pertanto, **recno()** fornisce semplicemente un argomento aggiuntivo, in questo caso un ID per ciascuna riga. Senza di esso i valori di esempio di **Comparison** non verrebbero caricati.

3. Fare clic su  per caricare dati.

### Creazione della tabella z-test

Procedere come indicato di seguito:

1. Nell'editor caricamento dati fare clic su  per accedere alla panoramica App, quindi fare clic sul foglio creato sopra.  
Viene aperta la vista foglio.
2. Fare clic su  **Modifica foglio** per modificare il foglio.
3. Da **Grafici** aggiungere una tabella e da **Campi** aggiungere Type come dimensione.
4. Aggiungere le seguenti espressioni come misure alla tabella.

Espressioni di esempio

| Etichetta   | Espressione        |
|-------------|--------------------|
| ZTest Conf  | ZTest_conf(Value)  |
| ZTest Dif   | ZTest_dif(Value)   |
| ZTest Sig   | ZTest_sig(Value)   |
| ZTest Sterr | ZTest_sterr(Value) |
| ZTest Z     | ZTest_z(Value)     |



*Per visualizzare valori validi, potrebbe essere necessario regolare la formattazione numero delle misure. La tabella sarebbe più facilmente leggibile se la formattazione dei numeri sulla maggior parte delle misure fosse impostata su **Numero>Semplice**, anziché su **Auto**. Ma per ZTest Sig, ad esempio, utilizzare la formattazione numerica: **Personalizzato** e quindi modificare il modello di formattazione in **#.#####**.*

### Risultato:

La tabella risultante per le funzioni z-test per i dati campione conterrà i valori seguenti:

z-test tabella dei risultati

| Type        | ZTest Conf | ZTest Dif | ZTest Sig | ZTest Sterr | ZTest Z |
|-------------|------------|-----------|-----------|-------------|---------|
| Comparison  | 6.40       | 11.95     | 0.000123  | 3.27        | 3.66    |
| Observation | 5.48       | 27.15     | 0.000000  | 2.80        | 9.71    |

### Creazione della tabella z-testw

Le funzioni z-testw devono essere utilizzate quando la serie di dati di input presenta un formato a due colonne pesate. Le espressioni richiedono un valore per l'argomento weight.

Gli esempi qui riportati utilizzano il valore 2 dappertutto, tuttavia si potrebbe utilizzare un'espressione, che definisce un valore weight per ciascuna osservazione.

Procedere come indicato di seguito:

1. Nell'editor caricamento dati fare clic su  per accedere alla panoramica App, quindi fare clic sul foglio creato sopra.  
Viene aperta la vista foglio.
2. Fare clic su  **Modifica foglio** per modificare il foglio.
3. Da **Grafici** aggiungere una tabella e da **Campi** aggiungere Type come dimensione.
4. Aggiungere alla tabella le espressioni seguenti come misure.

Espressioni di esempio

| Etichetta    | Espressione           |
|--------------|-----------------------|
| ZTestw Conf  | ZTestw_conf(2,Value)  |
| ZTestw Dif   | ZTestw_dif(2,Value)   |
| ZTestw Sig   | ZTestw_sig(2,Value)   |
| ZTestw Sterr | ZTestw_sterr(2,Value) |
| ZTestw Z     | ZTestw_z(2,Value)     |

Usare la stessa formattazione numero dell'esempio sulle funzioni z-test.

### Risultato:

La tabella risultante per le funzioni z-testw conterrà i valori seguenti:

z-testw tabella dei risultati

| Type        | ZTestw Conf | ZTestw Dif | ZTestw Sig   | ZTestw Sterr | ZTestw Z |
|-------------|-------------|------------|--------------|--------------|----------|
| Comparison  | 4.47        | 11.95      | 8.037185e-08 | 2.28         | 5.24     |
| Observation | 3.83        | 27.15      | 0            | 1.95         | 13.91    |

## Funzioni di aggregazione delle stringhe

In questa sezione vengono descritte le funzioni di aggregazione relative alle stringhe.

Ciascuna funzione viene descritta ulteriormente dopo la panoramica. È inoltre possibile fare clic sul nome della funzione nella sintassi per accedere immediatamente ai dettagli per tale funzione specifica.

### Funzioni di aggregazione delle stringhe nello script di caricamento dei dati

#### Concat

**Concat()** viene utilizzato per unire le stringhe. Questa funzione di script restituisce la concatenazione di stringhe aggregata di tutti i valori dell'espressione ripetuti su un insieme di record, come definito da una clausola **group by**.

```
Concat ([ distinct ] expression [, delimiter [, sort-weight]])
```

#### FirstValue

**FirstValue()** restituisce il valore che è stato caricato per primo dai record definiti dall'espressione, ordinato in base a una clausola **group by**.



*Questa funzione è disponibile solo come funzione di script.*

```
FirstValue (expression)
```

#### LastValue

**LastValue()** restituisce il valore che è stato caricato per ultimo dai record definiti dall'espressione, ordinato in base a una clausola **group by**.



*Questa funzione è disponibile solo come funzione di script.*

```
LastValue (expression)
```

#### MaxString

**MaxString()** individua i valori di stringa nell'espressione e restituisce l'ultimo valore di testo ordinato su un insieme di record, come definito dalla clausola **group by**.

```
MaxString (expression )
```

#### MinString

**MinString()** individua i valori di stringa nell'espressione e restituisce il primo valore di testo ordinato su un insieme di record, come definito dalla clausola **group by**.

```
MinString (expression )
```

### Funzioni di aggregazione delle stringhe nei grafici

Le seguenti funzioni grafiche sono disponibili per l'aggregazione delle stringhe nei grafici

#### Concat

**Concat()** viene utilizzata per unire le stringhe. La funzione restituisce la concatenazione di stringhe aggregata di tutti i valori dell'espressione valutata su ciascuna dimensione.

```
Concat - funzione per grafici {[SetExpression] [DISTINCT] [TOTAL [<fld{, fld}>]] string[, delimiter[, sort_weight]]}
```

### MaxString

**MaxString()** trova valori di stringa nell'espressione o nel campo e restituisce l'ultimo valore di testo nel criterio di ordinamento alfabetico.

```
MaxString - funzione per grafici {[SetExpression] [TOTAL [<fld{, fld}>]]}  
expr)
```

### MinString

**MinString()** trova valori di stringa nell'espressione o nel campo e restituisce il primo valore di testo nel criterio di ordinamento alfabetico.

```
MinString - funzione per grafici {[SetExpression] [TOTAL [<fld {, fld}>]]}  
expr)
```

## Concat

**Concat()** viene utilizzato per unire le stringhe. Questa funzione di script restituisce la concatenazione di stringhe aggregata di tutti i valori dell'espressione ripetuti su un insieme di record, come definito da una clausola **group by**.

### Sintassi:

```
Concat ([ distinct ] string [, delimiter [, sort-weight]])
```

**Tipo di dati restituiti:** stringa

### Argomenti:

L'espressione o il campo contenente la stringa da elaborare.

#### Argomenti

| Argomento   | Descrizione                                                                                                                                                                                                   |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| string      | L'espressione o il campo contenente la stringa da elaborare.                                                                                                                                                  |
| delimiter   | Ogni valore può essere separato dalla stringa trovata in delimiter.                                                                                                                                           |
| sort-weight | L'ordine di concatenazione può essere determinato dal valore della dimensione <b>sort-weight</b> , se presente, con la stringa corrispondente al valore più basso che compare per primo nella concatenazione. |
| distinct    | Se la parola <b>distinct</b> è riportata prima dell'espressione, tutti i duplicati vengono ignorati.                                                                                                          |

### Esempi e risultati:

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

## Esempi e risultati

| Esempio                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Risultato                                  | Risultati una volta aggiunti a un foglio                                |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|-------------------------------------------------------------------------|
| <b>TeamData:</b><br>LOAD * inline [<br>SalesGroup Team Date Amount<br>East Gamma 01/05/2013 20000<br>East Gamma 02/05/2013 20000<br>West Zeta 01/06/2013 19000<br>East Alpha 01/07/2013 25000<br>East Delta 01/08/2013 14000<br>West Epsilon 01/09/2013 17000<br>West Eta 01/10/2013 14000<br>East Beta 01/11/2013 20000<br>West Theta 01/12/2013 23000<br>] (delimiter is ' ');<br><br><b>Concat1:</b><br><br>LOAD SalesGroup,Concat(Team) as TeamConcat1<br>Resident TeamData Group By SalesGroup; | SalesGroup<br><br><br>East<br><br><br>West | TeamConcat1<br><br>AlphaBetaDeltaGammaGamma<br><br>EpsilonEtaThetaZeta  |
| Presupponendo che la tabella <b>TeamData</b> venga caricata come nell'esempio precedente:<br><br>LOAD SalesGroup,Concat(distinct Team,'-')<br>as TeamConcat2 Resident TeamData Group By SalesGroup;                                                                                                                                                                                                                                                                                                  | SalesGroup<br><br>East<br><br>West         | TeamConcat2<br><br>Alpha-Beta-Delta-Gamma<br><br>Epsilon-Eta-Theta-Zeta |
| Presupponendo che la tabella <b>TeamData</b> venga caricata come nell'esempio precedente. Poiché è stato aggiunto l'argomento per <b>sort-weight</b> , i risultati vengono ordinati in base al valore della dimensione Amount:<br><br>LOAD SalesGroup,Concat(distinct Team,'-',Amount) as TeamConcat2 Resident TeamData Group By SalesGroup;                                                                                                                                                         | SalesGroup<br><br>East<br><br>West         | TeamConcat2<br><br>Delta-Beta-Gamma-Alpha<br><br>Eta-Epsilon-Zeta-Theta |

## Concat - funzione per grafici

**Concat()** viene utilizzata per unire le stringhe. La funzione restituisce la concatenazione di stringhe aggregata di tutti i valori dell'espressione valutata su ciascuna dimensione.

## Sintassi:

```
Concat({[SetExpression] [DISTINCT] [TOTAL [<fld{, fld}>]]} string[, delimiter[, sort_weight]])
```

Tipo di dati restituiti: stringa

Argomenti:

Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| string        | L'espressione o il campo contenente la stringa da elaborare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| delimiter     | Ogni valore può essere separato dalla stringa trovata in delimiter.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| sort-weight   | L'ordine di concatenazione può essere determinato dal valore della dimensione <b>sort-weight</b> , se presente, con la stringa corrispondente al valore più basso che compare per primo nella concatenazione.                                                                                                                                                                                                                                                                                                                                                     |
| SetExpression | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                                                                                                                                                                                                                               |
| DISTINCT      | Se la parola <b>DISTINCT</b> è riportata prima degli argomenti della funzione, i duplicati risultanti dalla valutazione degli argomenti della funzione vengono ignorati.                                                                                                                                                                                                                                                                                                                                                                                          |
| TOTAL         | <p>Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.</p> <p>Utilizzando <b>TOTAL [&lt;fld {fld}&gt;]</b>, dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili.</p> |

Esempi e risultati:

Results table

| SalesGroup | Amount | Concat(Team)   | Concat(TOTAL <SalesGroup> Team) |
|------------|--------|----------------|---------------------------------|
| East       | 25000  | Alpha          | AlphaBetaDeltaGammaGamma        |
| East       | 20000  | BetaGammaGamma | AlphaBetaDeltaGammaGamma        |
| East       | 14000  | Delta          | AlphaBetaDeltaGammaGamma        |
| West       | 17000  | Epsilon        | EpsilonEtaThetaZeta             |
| West       | 14000  | Eta            | EpsilonEtaThetaZeta             |
| West       | 23000  | Theta          | EpsilonEtaThetaZeta             |
| West       | 19000  | Zeta           | EpsilonEtaThetaZeta             |

## Esempi di funzioni

| Esempio                                                           | Risultato                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>Concat(Team)</code>                                         | La tabella viene creata dalle dimensioni SalesGroup e Amount e dalle variazioni sulla misura Concat(Team). Ignorando il risultato di Totali, è necessario ricordare che sebbene esistano dati per otto valori di Team distribuiti tra due valori di SalesGroup, l'unico risultato della misura Concat(Team) che concatena più di un valore di stringa Team nella tabella è la riga contenente la dimensione Amount 20000, che restituisce il risultato BetaGammaGamma. Ciò si verifica perché esistono tre valori per Amount 20000 nei dati di input. Tutti gli altri risultati restano non concatenati quando la misura viene estesa tra le dimensioni perché esiste solo un valore di Team per ciascuna combinazione di SalesGroup e Amount. |
| <code>Concat (DISTINCT Team, ', ')</code>                         | Beta, Gamma perché il qualificatore DISTINCT significa che il risultato Gamma duplicato viene ignorato. Inoltre, l'argomento del delimitatore viene definito come virgola seguita da uno spazio.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <code>Concat (TOTAL &lt;SalesGroup&gt; Team)</code>               | Tutti i valori di stringa per tutti i valori di Team sono concatenati se viene utilizzato il qualificatore TOTAL. Con la selezione del campo <SalesGroup> specificata, vengono suddivisi i risultati nei due valori della dimensione SalesGroup. Per SalesGroupEast, i risultati sono AlphaBetaDeltaGammaGamma. Per SalesGroupWest, i risultati sono EpsilonEtaThetaZeta.                                                                                                                                                                                                                                                                                                                                                                      |
| <code>Concat (TOTAL &lt;SalesGroup&gt; Team, '; ', Amount)</code> | Aggiungendo l'argomento per <b>sort-weight</b> : Amount, i risultati vengono ordinati in base al valore della dimensione Amount. Il risultato diventa DeltaBetaGammaGammaAlpha e EtaEpsilonZetaTheta.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

Dati utilizzati nell'esempio:

```
TeamData:
LOAD * inline [
SalesGroup|Team|Date|Amount
East|Gamma|01/05/2013|20000
East|Gamma|02/05/2013|20000
West|Zeta|01/06/2013|19000
East|Alpha|01/07/2013|25000
East|Delta|01/08/2013|14000
West|Epsilon|01/09/2013|17000
West|Eta|01/10/2013|14000
East|Beta|01/11/2013|20000
West|Theta|01/12/2013|23000
] (delimiter is '|');
```

## FirstValue

**FirstValue()** restituisce il valore che è stato caricato per primo dai record definiti dall'espressione, ordinato in base a una clausola **group by**.



*Questa funzione è disponibile solo come funzione di script.*

**Sintassi:****FirstValue** ( expr )**Tipo di dati restituiti:** duale**Argomenti:**

## Argomenti

| Argomento | Descrizione                                             |
|-----------|---------------------------------------------------------|
| expr      | L'espressione o il campo contenente i dati da misurare. |

**Limiti:**

Se non viene trovato nessun valore di testo, viene restituito NULL.

**Esempi e risultati:**

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

## Dati risultanti

| Esempio                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Risultato                          | Risultati su un foglio                   |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|------------------------------------------|
| TeamData:<br>LOAD * inline [<br>SalesGroup Team Date Amount<br>East Gamma 01/05/2013 20000<br>East Gamma 02/05/2013 20000<br>West Zeta 01/06/2013 19000<br>East Alpha 01/07/2013 25000<br>East Delta 01/08/2013 14000<br>West Epsilon 01/09/2013 17000<br>West Eta 01/10/2013 14000<br>East Beta 01/11/2013 20000<br>West Theta 01/12/2013 23000<br>] (delimiter is ' ');<br><br>FirstValue1:<br>LOAD SalesGroup,FirstValue(Team) as FirstTeamLoaded<br>Resident TeamData Group By SalesGroup; | SalesGroup<br><br>East<br><br>West | FirstTeamLoaded<br><br>Gamma<br><br>Zeta |

**LastValue**

**LastValue()** restituisce il valore che è stato caricato per ultimo dai record definiti dall'espressione, ordinato in base a una clausola **group by**.



Questa funzione è disponibile solo come funzione di script.

**Sintassi:****LastValue** ( expr )**Tipo di dati restituiti:** duale**Argomenti:**

## Argomenti

| Argomento | Descrizione                                             |
|-----------|---------------------------------------------------------|
| expr      | L'espressione o il campo contenente i dati da misurare. |

**Limiti:**

Se non viene trovato nessun valore di testo, viene restituito NULL.

**Esempi e risultati:**

Aggiungere lo script di esempio all'app ed eseguirlo. Aggiungere quindi a un foglio nell'app almeno i campi elencati nella colonna dei risultati per visualizzare il risultato.

Per ottenere lo stesso aspetto della colonna dei risultati mostrata di seguito, nel pannello delle proprietà, in Ordinamento passare da Automatico a Personalizza, quindi deselezionare l'ordinamento numerico e alfabetico.

| Esempio                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Risultato                          | Risultato con ordinamento personalizzato |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|------------------------------------------|
| TeamData:<br>LOAD * inline [<br>SalesGroup Team Date Amount<br>East Gamma 01/05/2013 20000<br>East Gamma 02/05/2013 20000<br>West Zeta 01/06/2013 19000<br>East Alpha 01/07/2013 25000<br>East Delta 01/08/2013 14000<br>West Epsilon 01/09/2013 17000<br>West Eta 01/10/2013 14000<br>East Beta 01/11/2013 20000<br>West Theta 01/12/2013 23000<br>] (delimiter is ' ');<br><br>LastValue1:<br>LOAD SalesGroup,LastValue(Team) as LastTeamLoaded<br>Resident TeamData Group By SalesGroup; | SalesGroup<br><br>East<br><br>West | LastTeamLoaded<br><br>Beta<br><br>Theta  |

**MaxString**

**MaxString()** individua i valori di stringa nell'espressione e restituisce l'ultimo valore di testo ordinato su un insieme di record, come definito dalla clausola **group by**.

**Sintassi:****MaxString** ( expr )**Tipo di dati restituiti:** duale**Argomenti:**

| Argomento | Descrizione                                             |
|-----------|---------------------------------------------------------|
| expr      | L'espressione o il campo contenente i dati da misurare. |

**Limiti:**

Se non viene trovato nessun valore di testo, viene restituito NULL.

**Esempi e risultati:**

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

| Esempio                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Risultato  |            |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|------------|
| <b>TeamData:</b><br>LOAD * inline [<br>SalesGroup Team Date Amount<br>East Gamma 01/05/2013 20000<br>East Gamma 02/05/2013 20000<br>West Zeta 01/06/2013 19000<br>East Alpha 01/07/2013 25000<br>East Delta 01/08/2013 14000<br>West Epsilon 01/09/2013 17000<br>West Eta 01/10/2013 14000<br>East Beta 01/11/2013 20000<br>West Theta 01/12/2013 23000<br>] (delimiter is ' ');<br><br><b>Concat1:</b><br><br>LOAD SalesGroup,MaxString(Team) as MaxString1 Resident<br>TeamData Group By SalesGroup; | SalesGroup | MaxString1 |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | East       | Gamma      |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | West       | Zeta       |
| Presupponendo che la tabella <b>TeamData</b> venga caricata come nell'esempio precedente e che lo script di caricamento dei dati presenti l'istruzione SET:<br>SET DateFormat='DD/MM/YYYY';<br><br>LOAD SalesGroup,MaxString(Date) as MaxString2 Resident<br>TeamData Group By SalesGroup;                                                                                                                                                                                                             | SalesGroup | MaxString2 |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | East       | 01/11/2013 |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | West       | 01/12/2013 |

## MaxString - funzione per grafici

**MaxString()** trova valori di stringa nell'espressione o nel campo e restituisce l'ultimo valore di testo nel criterio di ordinamento alfabetico.

### Sintassi:

```
MaxString( {[SetExpression] [TOTAL [<fld{, fld}>]]} expr )
```

**Tipo di dati restituiti:** duale

### Argomenti:

#### Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| expr          | L'espressione o il campo contenente i dati da misurare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| SetExpression | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                                                                                                                                                                                                                                 |
| TOTAL         | <p>Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.</p> <p>Utilizzando <b>TOTAL [&lt;fld {, fld}&gt;]</b>, dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili.</p> |

### Limiti:

Se l'espressione non include alcun valore con una rappresentazione di stringa, verrà restituito NULL.

### Esempi e risultati:

Tabella dei risultati

| SalesGroup | Amount | MaxString(Team) | MaxString(Date) |
|------------|--------|-----------------|-----------------|
| East       | 14000  | Delta           | 2013/08/01      |
| East       | 20000  | Gamma           | 2013/11/01      |
| East       | 25000  | Alpha           | 2013/07/01      |
| West       | 14000  | Eta             | 2013/10/01      |
| West       | 17000  | Epsilon         | 2013/09/01      |
| West       | 19000  | Zeta            | 2013/06/01      |
| West       | 23000  | Theta           | 2013/12/01      |

## Esempi di funzioni

| Esempio             | Risultato                                                                                                                                                                                                                       |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MaxString<br>(Team) | Vi sono tre valori pari a 20000 per la dimensione Amount: due di Gamma (in date differenti) e uno di Beta. Il risultato della misura MaxString (Team) è pertanto Gamma, perché è il valore più elevato nelle stringhe ordinate. |
| MaxString<br>(Date) | 2013/11/01 è il valore Date più elevato dei tre associati alla dimensione Amount. Questo presuppone che lo script includa l'istruzione <code>SET SET DateFormat='YYYY-MM-DD';'</code>                                           |

Dati utilizzati nell'esempio:

```
TeamData:
LOAD * inline [
SalesGroup|Team|Date|Amount
East|Gamma|01/05/2013|20000
East|Gamma|02/05/2013|20000
West|Zeta|01/06/2013|19000
East|Alpha|01/07/2013|25000
East|Delta|01/08/2013|14000
West|Epsilon|01/09/2013|17000
West|Eta|01/10/2013|14000
East|Beta|01/11/2013|20000
West|Theta|01/12/2013|23000
] (delimiter is '|');
```

## MinString

**MinString()** individua i valori di stringa nell'espressione e restituisce il primo valore di testo ordinato su un insieme di record, come definito dalla clausola **group by**.

**Sintassi:**

```
MinString ( expr )
```

**Tipo di dati restituiti:** duale

**Argomenti:**

## Argomenti

| Argomento | Descrizione                                             |
|-----------|---------------------------------------------------------|
| expr      | L'espressione o il campo contenente i dati da misurare. |

**Limiti:**

Se non viene trovato nessun valore di testo, viene restituito NULL.

**Esempi e risultati:**

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

## Dati risultanti

| Esempio                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Risultato                          |                                                |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|------------------------------------------------|
| TeamData:<br>LOAD * inline [<br>SalesGroup Team Date Amount<br>East Gamma 01/05/2013 20000<br>East Gamma 02/05/2013 20000<br>West Zeta 01/06/2013 19000<br>East Alpha 01/07/2013 25000<br>East Delta 01/08/2013 14000<br>West Epsilon 01/09/2013 17000<br>West Eta 01/10/2013 14000<br>East Beta 01/11/2013 20000<br>West Theta 01/12/2013 23000<br>] (delimiter is ' ');<br><br>Concat1:<br><br>LOAD SalesGroup,MinString(Team) as MinString1 Resident<br>TeamData Group By SalesGroup; | SalesGroup<br><br>East<br><br>West | MinString1<br><br>Alpha<br><br>Epsilon         |
| Presupponendo che la tabella <b>TeamData</b> venga caricata come nell'esempio precedente e che lo script di caricamento dei dati presenti l'istruzione SET:<br>SET DateFormat='DD/MM/YYYY';<br><br>LOAD SalesGroup,MinString(Date) as MinString2 Resident<br>TeamData Group By SalesGroup;                                                                                                                                                                                               | SalesGroup<br><br>East<br><br>West | MinString2<br><br>01/05/2013<br><br>01/06/2013 |

## MinString - funzione per grafici

**MinString()** trova valori di stringa nell'espressione o nel campo e restituisce il primo valore di testo nel criterio di ordinamento alfabetico.

## Sintassi:

**MinString** ([SetExpression] [TOTAL [<fld {, fld}>]]) expr)

Tipo di dati restituiti: duale

## Argomenti:

## Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                         |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| expr          | L'espressione o il campo contenente i dati da misurare.                                                                                                                                                             |
| SetExpression | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis. |

| Argomento | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TOTAL     | <p>Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.</p> <p>Utilizzando <b>TOTAL [&lt;fld {fld}&gt;]</b>, dove il qualificatore <b>TOTAL</b> è seguito da un elenco di uno o più nomi di campo come sottogruppo delle variabili di dimensione del grafico, si crea un sottogruppo di tutti i valori possibili.</p> |

### Esempi e risultati:

Dati campione

| SalesGroup | Amount | MinString(Team) | MinString(Date) |
|------------|--------|-----------------|-----------------|
| East       | 14000  | Delta           | 2013/08/01      |
| East       | 20000  | Beta            | 2013/05/01      |
| East       | 25000  | Alpha           | 2013/07/01      |
| West       | 14000  | Eta             | 2013/10/01      |
| West       | 17000  | Epsilon         | 2013/09/01      |
| West       | 19000  | Zeta            | 2013/06/01      |
| West       | 23000  | Theta           | 2013/12/01      |

Esempi di funzioni

| Esempi              | Risultati                                                                                                                                                                                                                       |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MinString<br>(Team) | Vi sono tre valori pari a 20000 per la dimensione Amount: due di Gamma (in date differenti) e uno di Beta. Il risultato della misura MinString (Team) è pertanto Beta, perché questo è il primo valore nelle stringhe ordinate. |
| MinString<br>(Date) | 2013/11/01 è il primo valore Date dei tre valori associati alla dimensione Amount. Questo presuppone che lo script includa l'istruzione SET SET DateFormat='YYYY-MM-DD';                                                        |

### Dati utilizzati nell'esempio:

```
TeamData:
LOAD * inline [
SalesGroup|Team|Date|Amount
East|Gamma|01/05/2013|20000
East|Gamma|02/05/2013|20000
West|Zeta|01/06/2013|19000
East|Alpha|01/07/2013|25000
East|Delta|01/08/2013|14000
West|Epsilon|01/09/2013|17000
West|Eta|01/10/2013|14000
East|Beta|01/11/2013|20000
```

```
West|Theta|01/12/2013|23000  
] (delimiter is '|');
```

### Funzioni di dimensione sintetica

Una dimensione sintetica viene creata nell'app a partire dai valori generati dalle funzioni di dimensione sintetica e non direttamente dai campi nel modello dati. Quando i valori generati da una funzione di dimensione sintetica vengono utilizzati in un grafico come dimensione calcolata, viene creata una dimensione sintetica. Le dimensioni sintetiche consentono, ad esempio, di creare grafici con dimensioni con valori derivanti dai dati, vale a dire dimensioni dinamiche.



*Le dimensioni sintetiche non vengono influenzate dalle selezioni.*

Le seguenti funzioni di dimensione sintetica possono essere utilizzate nei grafici.

#### ValueList

**ValueList()** restituisce una serie di valori elencati che, se utilizzati in una dimensione calcolata, formano una dimensione sintetica.

**ValueList - funzione per grafici** (v1 {, Expression})

#### ValueLoop

**ValueLoop()** restituisce un set di valori ripetuti che, se utilizzati in una dimensione calcolata, formano una dimensione sintetica.

**ValueLoop - funzione per grafici** (from [, to [, step ]])

### ValueList - funzione per grafici

**ValueList()** restituisce una serie di valori elencati che, se utilizzati in una dimensione calcolata, formano una dimensione sintetica.



*Nei grafici con una dimensione sintetica creata con la funzione **ValueList**, è possibile fare riferimento al valore di dimensione corrispondente a una specifica cella di espressione dichiarando nuovamente la funzione **ValueList** con gli stessi parametri nell'espressione grafica. La funzione può essere ovviamente utilizzata ovunque nel layout, ma, tranne quando viene utilizzata per le dimensioni sintetiche, avrà significato solamente all'interno di una funzione di aggregazione.*



*Le dimensioni sintetiche non vengono influenzate dalle selezioni.*

#### Sintassi:

**ValueList** (v1 {, ...})

**Tipo di dati restituiti:** duale

**Argomenti:**

Argomenti

| Argomento | Descrizione                                                               |
|-----------|---------------------------------------------------------------------------|
| v1        | Valore statico (generalmente una stringa, anche se può essere un numero). |
| {,...}    | Elenco opzionale di valori statici.                                       |

**Esempi e risultati:**

Esempi di funzioni

| Esempio                                                                                                                                                                                                                                                                                                  | Risultato                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |             |                  |  |  |                   |  |      |                  |  |  |  |        |  |  |  |  |                  |  |      |      |                  |  |      |      |                    |  |      |       |                    |  |      |       |              |  |      |       |              |  |      |        |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|------------------|--|--|-------------------|--|------|------------------|--|--|--|--------|--|--|--|--|------------------|--|------|------|------------------|--|------|------|--------------------|--|------|-------|--------------------|--|------|-------|--------------|--|------|-------|--------------|--|------|--------|
| valueList<br>( 'Number of<br>orders',<br>'Average Order<br>Size', 'Total<br>Amount' )                                                                                                                                                                                                                    | Se utilizzato per creare una dimensione in una tabella, ad esempio, i tre valori della stringa verranno utilizzati come etichette della riga nella tabella, a cui è possibile fare riferimento in un'espressione.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |             |                  |  |  |                   |  |      |                  |  |  |  |        |  |  |  |  |                  |  |      |      |                  |  |      |      |                    |  |      |       |                    |  |      |       |              |  |      |       |              |  |      |        |
| =IF( valueList<br>( 'Number of<br>Orders',<br>'Average Order<br>Size', 'Total<br>Amount' ) =<br>'Number of<br>Orders', count<br>(SaleID), IF(<br>valueList<br>( 'Number of<br>Orders',<br>'Average Order<br>Size', 'Total<br>Amount' ) =<br>'Average Order<br>Size', avg<br>(Amount), sum<br>(Amount) )) | <p>Questa espressione prende i valori dalla dimensione creata e vi fa riferimento in un'istruzione IF nidificata come input per tre funzioni di aggregazione:</p> <table><tr><th colspan="4">ValueList()</th></tr><tr><th colspan="2">Created dimension</th><th>Year</th><th>Added expression</th></tr><tr><td></td><td></td><td></td><td>522.00</td></tr><tr><td colspan="4"></td></tr><tr><td colspan="2">Number of Orders</td><td>2012</td><td>5.00</td></tr><tr><td colspan="2">Number of Orders</td><td>2013</td><td>7.00</td></tr><tr><td colspan="2">Average Order Size</td><td>2012</td><td>13.20</td></tr><tr><td colspan="2">Average Order Size</td><td>2013</td><td>15.43</td></tr><tr><td colspan="2">Total Amount</td><td>2012</td><td>66.00</td></tr><tr><td colspan="2">Total Amount</td><td>2013</td><td>108.00</td></tr></table> | ValueList() |                  |  |  | Created dimension |  | Year | Added expression |  |  |  | 522.00 |  |  |  |  | Number of Orders |  | 2012 | 5.00 | Number of Orders |  | 2013 | 7.00 | Average Order Size |  | 2012 | 13.20 | Average Order Size |  | 2013 | 15.43 | Total Amount |  | 2012 | 66.00 | Total Amount |  | 2013 | 108.00 |
| ValueList()                                                                                                                                                                                                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |             |                  |  |  |                   |  |      |                  |  |  |  |        |  |  |  |  |                  |  |      |      |                  |  |      |      |                    |  |      |       |                    |  |      |       |              |  |      |       |              |  |      |        |
| Created dimension                                                                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Year        | Added expression |  |  |                   |  |      |                  |  |  |  |        |  |  |  |  |                  |  |      |      |                  |  |      |      |                    |  |      |       |                    |  |      |       |              |  |      |       |              |  |      |        |
|                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |             | 522.00           |  |  |                   |  |      |                  |  |  |  |        |  |  |  |  |                  |  |      |      |                  |  |      |      |                    |  |      |       |                    |  |      |       |              |  |      |       |              |  |      |        |
|                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |             |                  |  |  |                   |  |      |                  |  |  |  |        |  |  |  |  |                  |  |      |      |                  |  |      |      |                    |  |      |       |                    |  |      |       |              |  |      |       |              |  |      |        |
| Number of Orders                                                                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 2012        | 5.00             |  |  |                   |  |      |                  |  |  |  |        |  |  |  |  |                  |  |      |      |                  |  |      |      |                    |  |      |       |                    |  |      |       |              |  |      |       |              |  |      |        |
| Number of Orders                                                                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 2013        | 7.00             |  |  |                   |  |      |                  |  |  |  |        |  |  |  |  |                  |  |      |      |                  |  |      |      |                    |  |      |       |                    |  |      |       |              |  |      |       |              |  |      |        |
| Average Order Size                                                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 2012        | 13.20            |  |  |                   |  |      |                  |  |  |  |        |  |  |  |  |                  |  |      |      |                  |  |      |      |                    |  |      |       |                    |  |      |       |              |  |      |       |              |  |      |        |
| Average Order Size                                                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 2013        | 15.43            |  |  |                   |  |      |                  |  |  |  |        |  |  |  |  |                  |  |      |      |                  |  |      |      |                    |  |      |       |                    |  |      |       |              |  |      |       |              |  |      |        |
| Total Amount                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 2012        | 66.00            |  |  |                   |  |      |                  |  |  |  |        |  |  |  |  |                  |  |      |      |                  |  |      |      |                    |  |      |       |                    |  |      |       |              |  |      |       |              |  |      |        |
| Total Amount                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 2013        | 108.00           |  |  |                   |  |      |                  |  |  |  |        |  |  |  |  |                  |  |      |      |                  |  |      |      |                    |  |      |       |                    |  |      |       |              |  |      |       |              |  |      |        |

**Dati utilizzati negli esempi:**

```

SalesPeople:
LOAD * INLINE [
SalesID|SalesPerson|Amount|Year
1|1|12|2013
2|1|23|2013
3|1|17|2013
4|2|9|2013
5|2|14|2013
6|2|29|2013

```

```

7|2|4|2013
8|1|15|2012
9|1|16|2012
10|2|11|2012
11|2|17|2012
12|2|7|2012
] (delimiter is '|');

```

## ValueLoop - funzione per grafici

ValueLoop() restituisce un set di valori ripetuti che, se utilizzati in una dimensione calcolata, formano una dimensione sintetica.

I valori generati iniziano con il valore **from** e finiscono con il valore **to** includendo i valori intermedi con incrementi di step.



*Nei grafici con una dimensione sintetica creata con la funzione **ValueLoop**, è possibile fare riferimento al valore di dimensione corrispondente a una specifica cella di espressione dichiarando nuovamente la funzione **ValueLoop** con gli stessi parametri nell'espressione grafica. La funzione può essere ovviamente utilizzata ovunque nel layout, ma, tranne quando viene utilizzata per le dimensioni sintetiche, avrà significato solamente all'interno di una funzione di aggregazione.*



*Le dimensioni sintetiche non vengono influenzate dalle selezioni.*

### Sintassi:

```
ValueLoop (from [, to [, step ]])
```

Tipo di dati restituiti: duale

### Argomenti:

#### Argomenti

| Argomenti | Descrizione                                    |
|-----------|------------------------------------------------|
| from      | Valore iniziale nel set di valori da generare. |
| to        | Valore finale nel set di valori da generare.   |
| step      | Grandezza dell'incremento tra i valori.        |

### Esempi e risultati:

#### Esempi di funzioni

| Esempio              | Risultato                                                                                                                                                                                                                         |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ValueLoop<br>(1, 10) | Ciò crea una dimensione in una tabella che, ad esempio, può essere utilizzata per l'etichettatura numerata. Questo esempio restituisce valori numerati da 1 a 10. È possibile fare riferimento a questi valori in un'espressione. |

| Esempio                 | Risultato                                                                                                   |
|-------------------------|-------------------------------------------------------------------------------------------------------------|
| ValueLoop<br>(2, 10, 2) | Questo esempio restituisce valori numerati 2, 4, 6, 8 e 10 perché l'argomento step presenta un valore di 2. |

## Aggregazioni nidificate

Si potrebbero verificare situazioni in cui è necessario applicare un'aggregazione al risultato di un'altra aggregazione. Questa operazione è denominata nidificazione delle applicazioni.

Non è possibile nidificare le aggregazioni in gran parte delle espressioni del grafico. Tuttavia, è possibile nidificare le aggregazioni se si utilizza il qualificatore **TOTAL** nella funzione di aggregazione interna.



È consentito un massimo di 100 livelli di nidificazione.

## Aggregazioni nidificate con il qualificatore TOTAL

### Esempio:

Si desidera calcolare la somma del campo **Sales**, includendo solo le transazioni con **OrderDate** uguali all'anno scorso. L'anno precedente può essere ottenuto tramite la funzione di aggregazione **Max (TOTAL Year (OrderDate))**.

La seguente aggregazione restituirebbe il risultato desiderato:

```
Sum(If(Year(OrderDate)=Max(TOTAL Year(OrderDate)), Sales))
```

Qlik Sense richiede l'inclusione del qualificatore **TOTAL** in questo tipo di nidificazione. È necessario per il confronto desiderato. Questo tipo di nidificazione è abbastanza comune e dovrebbe essere utilizzata quando richiesto.

### Vedere anche:

p [Aggr - funzione per grafici \(page 605\)](#)

## 8.3 Aggr - funzione per grafici

**Aggr()** restituisce una matrice di valori per l'espressione calcolata in base alla dimensione o alle dimensioni dichiarate. Ad esempio, il valore massimo delle vendite, per cliente, per regione.

La funzione **Aggr** è utilizzata per le aggregazioni nidificate, in cui il relativo primo parametro (l'aggregazione interna) è calcolato una volta per valore dimensionale. Le dimensioni sono specificate nel secondo parametro (e nei parametri successivi).

Inoltre, la funzione **Aggr** deve essere racchiusa in una funzione di aggregazione esterna, utilizzando la gamma di risultati dalla funzione **Aggr** come input per l'aggregazione in cui risulta nidificata.

### Sintassi:

```
Aggr ({SetExpression} [DISTINCT] [NODISTINCT] expr, StructuredParameter{, StructuredParameter})
```

**Tipo di dati restituiti:** duale

### Argomenti:

#### Argomenti

| Argomento           | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| expr                | Un'espressione costituita da una funzione di aggregazione. Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| StructuredParameter | <p>StructuredParameter è composto da una dimensione e, facoltativamente, da criteri di ordinamento nel formato: (Dimension(Sort-type, Ordering))</p> <p>La dimensione è un campo singolo e non può essere un'espressione. Viene utilizzata per determinare la matrice di valori per cui viene calcolata l'espressione Aggr.</p> <p>Se sono inclusi criteri di ordinamento, la matrice di valori creata dalla funzione Aggr, calcolata per la dimensione, verrà ordinata. Questo è importante quando l'ordinamento influisce sul risultato dell'espressione in cui è racchiusa la funzione Aggr.</p> <p>Per informazioni dettagliate su come utilizzare i criteri di ordinamento, vedere <a href="#">Aggiunta di criteri di ordinamento alla dimensione nel parametro strutturato</a>.</p> |
| SetExpression       | Per impostazione predefinita, la funzione di aggregazione aggrega la serie di possibili record definiti dalla selezione. È possibile definire una serie di record alternativa mediante un'espressione Set Analysis.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| DISTINCT            | Se l'argomento dell'espressione è preceduto dal qualificatore <b>distinct</b> o se non viene utilizzato alcun qualificatore, ogni combinazione distinta di valori di dimensione restituirà un solo valore. Questa è la modalità normale in cui vengono create le aggregazioni; ciascuna combinazione distinta di valori di dimensione restituisce una linea nel grafico.                                                                                                                                                                                                                                                                                                                                                                                                                  |
| NODISTINCT          | Se l'argomento dell'espressione è preceduto dal qualificatore <b>nodistinct</b> , ogni combinazione di valori di dimensione può restituire più valori, a seconda della struttura dati sottostante. Se esiste una sola dimensione, la funzione <b>aggr</b> restituirà una matrice con un numero di elementi uguale al numero di righe presenti nei dati sorgente.                                                                                                                                                                                                                                                                                                                                                                                                                          |

Le funzioni di aggregazione di base, quali **Sum**, **Min** e **Avg**, restituiscono un singolo valore numerico, mentre la funzione **Aggr()** può essere paragonata alla creazione di un set di risultati temporaneo (una tabella virtuale) su cui è possibile effettuare un'altra aggregazione. Ad esempio, è possibile calcolare il valore medio delle vendite sommando le vendite per cliente in un'istruzione **Aggr()** e calcolando quindi la media dei risultati sommati: **Avg(TOTAL Aggr(Sum(Sales),Customer))**.



*È possibile utilizzare la funzione **Aggr()** nelle dimensioni calcolate se si desidera creare aggregazioni di grafici nidificate su più livelli.*

### Limiti:

Ogni dimensione presente in una funzione **Aggr()** deve corrispondere a un campo singolo e non può essere un'espressione (dimensione calcolata).

### Aggiunta di criteri di ordinamento alla dimensione nel parametro strutturato

Nella sua forma di base, l'argomento **StructuredParameter** nella sintassi della funzione **Aggr** è una dimensione singola. L'espressione **Aggr(Sum(Sales, Month))** trova il valore totale delle vendite per ogni mese. Tuttavia, se viene inclusa in un'altra funzione di aggregazione, potrebbero venire restituiti risultati imprevisti, a meno che non vengano utilizzati i criteri di ordinamento. Ciò è dovuto al fatto che alcune dimensioni possono essere ordinate con criterio numerico o con criterio alfabetico e così via.

Nell'argomento **StructuredParameter** all'interno della funzione **Aggr** è possibile specificare criteri di ordinamento per la dimensione nell'espressione. In questo modo si impone un ordinamento alla tabella virtuale prodotta dalla funzione **Aggr**.

La sintassi dell'argomento **StructuredParameter** è la seguente:

```
(FieldName, (Sort-type, Ordering))
```

I parametri strutturati possono essere nidificati:

```
(FieldName, (FieldName2, (Sort-type, Ordering)))
```

Sort-type può essere: **NUMERIC**, **TEXT**, **FREQUENCY** o **LOAD\_ORDER**.

I tipi di ordinamento associati a ciascun Sort-type sono i seguenti:

Tipi di ordinamento consentiti

| Sort-type  | Tipi di ordinamento consentiti            |
|------------|-------------------------------------------|
| NUMERIC    | ASCENDING, DESCENDING o REVERSE           |
| TEXT       | ASCENDING, A2Z, DESCENDING, REVERSE o Z2A |
| FREQUENCY  | DESCENDING, REVERSE o ASCENDING           |
| LOAD_ORDER | ASCENDING, ORIGINAL, DESCENDING o REVERSE |

I tipi di ordinamento **REVERSE** e **DESCENDING** sono equivalenti.

Per Sort-type uguale a TEXT, i tipi di ordinamento ASCENDING e A2Z sono equivalenti e DESCENDING, REVERSE e Z2A sono equivalenti.

Per Sort-type uguale a LOAD\_ORDER, i tipi di ordinamento ASCENDING e ORIGINAL sono equivalenti.

### Esempi: espressioni del grafico che utilizzano Aggr

Esempi: espressioni del grafico

#### Esempio 1 espressione del grafico

##### Script di caricamento

Caricare i dati seguenti come un caricamento inline nell'editor di caricamento dati per creare l'esempio di espressione del grafico in basso.

ProductData:

```
LOAD * inline [
Customer|Product|UnitsSales|UnitPrice
Astrida|AA|4|16
Astrida|AA|10|15
Astrida|BB|9|9
Betacab|BB|5|10
Betacab|CC|2|20
Betacab|DD|25|25
Canutility|AA|8|15
Canutility|CC|0|19
] (delimiter is '|');
```

##### Espressione del grafico

Creare una visualizzazione KPI in un foglio di Qlik Sense. Aggiungere l'espressione seguente al KPI, come misura:

```
Avg(Aggr(Sum(UnitsSales*UnitPrice), Customer))
```

##### Risultato

376.7

##### Spiegazione

L'espressione `Aggr(Sum(UnitsSales*UnitPrice), Customer)` trova il valore totale delle vendite per **Customer** e restituisce una matrice di valori: 295, 715 e 120 per i tre valori di **Customer**.

In modo efficiente, è stato creato un elenco temporaneo di valori, senza una tabella esplicita o una colonna contenente tali valori.

Questi valori vengono utilizzati come input nella funzione **Avg()** per trovare il valore medio delle vendite, 376.7.

## Esempio 2 espressione del grafico

### Script di caricamento

Caricare i dati seguenti come un caricamento inline nell'editor di caricamento dati per creare l'esempio di espressione del grafico in basso.

ProductData:

```
LOAD * inline [
Customer|Product|UnitSales|UnitPrice
Astrida|AA|4|16
Astrida|AA|10|15
Astrida|BB|10|15
Astrida|BB|9|9
Betacab|BB|5|10
Betacab|BB|7|12
Betacab|CC|2|22
Betacab|CC|4|20
Betacab|DD|25|25
Canutility|AA|8|15
Canutility|AA|5|11
Canutility|CC|0|19
] (delimiter is '|');
```

### Espressione del grafico

Creare una visualizzazione tabella in un foglio Qlik Sense con **Customer**, **Product**, **UnitPrice** e **UnitSales** come dimensioni. Aggiungere la seguente espressione alla tabella come misura:

```
Aggr(NODISTINCT Max(UnitPrice), Customer, Product)
```

### Risultato

| Customer | Product | UnitPrice | UnitSales | Aggr(NODISTINCT Max(UnitPrice), Customer, Product) |
|----------|---------|-----------|-----------|----------------------------------------------------|
| Astrida  | AA      | 15        | 10        | 16                                                 |
| Astrida  | AA      | 16        | 4         | 16                                                 |
| Astrida  | BB      | 9         | 9         | 15                                                 |
| Astrida  | BB      | 15        | 10        | 15                                                 |
| Betacab  | BB      | 10        | 5         | 12                                                 |
| Betacab  | BB      | 12        | 7         | 12                                                 |
| Betacab  | CC      | 20        | 4         | 22                                                 |
| Betacab  | CC      | 22        | 2         | 22                                                 |
| Betacab  | DD      | 25        | 25        | 25                                                 |

| Customer   | Product | UnitPrice | UnitSales | Aggr(NODISTINCT Max(UnitPrice), Customer, Product) |
|------------|---------|-----------|-----------|----------------------------------------------------|
| Canutility | AA      | 11        | 5         | 15                                                 |
| Canutility | AA      | 15        | 8         | 15                                                 |
| Canutility | CC      | 19        | 0         | 19                                                 |

### Spiegazione

Una matrice di valori: 16, 16, 15, 15, 12, 12, 22, 22, 25, 15, 15 e 19. Il qualificatore **nodistinct** indica che la matrice contiene un elemento per ogni riga dei dati sorgente: ognuno è il valore **UnitPrice** massimo per ogni valore **Customer** e **Product**.

### Esempio 3 espressione del grafico

#### Script di caricamento

Caricare i dati seguenti come un caricamento inline nell'editor caricamento dati per creare l'esempio di espressione del grafico in basso.

```
Set vNumberOfOrders = 1000;
```

```
OrderLines:
```

```
Load
```

```
    RowNo() as OrderLineID,
    OrderID,
    OrderDate,
    Round((Year(OrderDate)-2005)*1000*Rand()*Rand()*Rand1) as Sales
    while Rand() <= 0.5 or IterNo()=1;
```

```
Load * Where OrderDate <= Today();
```

```
Load
```

```
    Rand() as Rand1,
    Date(MakeDate(2013)+Floor((365*4+1)*Rand())) as OrderDate,
    RecNo() as OrderID
    Autogenerate vNumberOfOrders;
```

```
Calendar:
```

```
Load distinct
```

```
    Year(OrderDate) as Year,
    Month(OrderDate) as Month,
    OrderDate
    Resident OrderLines;
```

### Espressioni del grafico

Creare una visualizzazione tabella in un foglio Qlik Sense con **Anno** e **Mese** come dimensioni. Aggiungere alla tabella le espressioni seguenti come misure:

- Sum(Sales)
- Sum(Aggr( Rangesum(Above(Sum(Sales),0,12)), (Year, (Numeric, Ascending)), (Month, (Numeric, Ascending)) )) etichettato come Structured Aggr() nella tabella.

## Risultato

| Year | Month | Sum(Sales) | Structured Aggr() |
|------|-------|------------|-------------------|
| 2013 | Jan   | 53495      | 53495             |
| 2013 | Feb   | 48580      | 102075            |
| 2013 | Mar   | 25651      | 127726            |
| 2013 | Apr   | 36585      | 164311            |
| 2013 | May   | 61211      | 225522            |
| 2013 | Jun   | 23689      | 249211            |
| 2013 | Jul   | 42311      | 291522            |
| 2013 | Aug   | 41913      | 333435            |
| 2013 | Sep   | 28886      | 362361            |
| 2013 | Oct   | 25977      | 388298            |
| 2013 | Nov   | 44455      | 432753            |
| 2013 | Dec   | 64144      | 496897            |
| 2014 | Jan   | 67775      | 67775             |

## Spiegazione

Questo esempio visualizza i valori aggregati su un periodo di dodici mesi per ogni anno in ordine cronologico crescente, da cui i parametri strutturati (Numeric, Ascending) parte dell'espressione **Aggr()**. Sono richieste due dimensioni specifiche come parametri strutturati: **Year** e **Month**, con ordinamento (1) **Year** (numerico) e (2) **Month** (numerico). Queste due dimensioni devono essere utilizzate nella visualizzazione tabella o grafico. Ciò risulta necessario affinché l'elenco dimensioni della funzione **Aggr()** corrisponda alle dimensioni dell'oggetto utilizzato nella visualizzazione.

È possibile confrontare le differenze tra queste misure in una tabella o in grafici lineari separati:

- `Sum(Aggr( Rangesum(Above(Sum(Sales),0,12)), (Year), (Month) ))`
- `Sum(Aggr( Rangesum(Above(Sum(Sales),0,12)), (Year, (Numeric, Ascending)), (Month, (Numeric, Ascending)) ))`

Dovrebbe risultare evidente che solo l'ultima espressione esegue l'accumulo desiderato dei valori aggregati.

## Vedere anche:

p [Funzioni di aggregazione di base \(page 357\)](#)

## 8.4 Funzioni colore

Queste funzioni possono essere utilizzate nelle espressioni associate con l'impostazione e la valutazione delle proprietà del colore degli oggetti dei grafici così come negli script di caricamento dei dati.



*Qlik Sense supporta le funzioni colore **Color()**, **qliktechblue** e **qliktechgray** per compatibilità con le versioni precedenti, sebbene il loro utilizzo non sia consigliato.*

### ARGB

**ARGB()** viene utilizzata nelle espressioni per impostare o valutare le proprietà del colore di un oggetto del grafico, in cui il colore è definito da un componente rosso **r**, un componente verde **g** e un componente blu **b**, con un fattore alfa (opacità) di **alpha**.

```
ARGB (alpha, r, g, b)
```

### HSL

**HSL()** viene utilizzata nelle espressioni per impostare o valutare le proprietà del colore di un oggetto del grafico, dove il colore è definito dai valori **hue**, **saturation** e **luminosity** tra 0 e 1.

```
HSL (hue, saturation, luminosity)
```

### RGB

**RGB()** restituisce un intero corrispondente al codice colore del colore definito dai tre parametri: la componente rossa **r**, la componente verde **g** e la componente blu **b**. Queste componenti devono avere valori interi compresi tra 0 e 255. La funzione può essere usata nelle espressioni per impostare o valutare le proprietà di colore di un oggetto grafico.

```
RGB (r, g, b)
```

### Colormix1

La funzione **Colormix1()** viene utilizzata nelle espressioni per restituire una rappresentazione cromatica ARGB da un gradiente di due colori, basato su un valore compreso tra 0 e 1.

```
Colormix1 (Value , ColorZero , ColorOne)
```

Value è un numero reale compreso tra 0 e 1.

- Se Value = 0, viene restituito ColorZero .
- Se Value = 1, viene restituito ColorOne .
- Se 0 < Value < 1 verrà restituita la sfumatura intermedia appropriata.

ColorZero è una rappresentazione di colore RGB valida per il colore da associare con il limite minimo dell'intervallo.

ColorOne è una rappresentazione cromatica RGB valida per il colore da associare con il livello finale massimo dell'intervallo.

### Esempio:

`colormix1(0.5, red(), blue())`

restituisce:

`ARGB(255,64,0,64)` (purple)



### Colormix2

La funzione **Colormix2()** viene utilizzata nelle espressioni per restituire una rappresentazione cromatica ARGB da un gradiente di due colori, basato su un valore compreso tra -1 e 1 con la possibilità di specificare un colore intermedio per la posizione centrale (0).

**Colormix2** (Value ,ColorMinusOne , ColorOne[ , ColorZero])

Value è un numero reale compreso tra -1 e 1.

- Se Value = -1 verrà restituito il primo colore.
- Se Value = 1 verrà restituito il secondo colore.
- Se  $-1 < \text{Value} < 1$ , verrà restituita la combinazione di colori appropriata.

ColorMinusOne è una rappresentazione di colore RGB valida per il colore da associare con il limite minimo dell'intervallo.

ColorOne è una rappresentazione cromatica RGB valida per il colore da associare con il livello finale massimo dell'intervallo.

ColorZero è una rappresentazione cromatica RGB opzionale valida per il colore da associare con il centro dell'intervallo.

**SysColor**

**SysColor()** restituisce la rappresentazione cromatica ARGB per il colore di sistema di Windows nr, dove nr corrisponde al parametro della funzione **GetSysColor(nr)** dell'API di Windows.

**SysColor** (nr)

**ColorMapHue**

**ColorMapHue()** restituisce il valore ARGB di un colore da una mappa dei colori che varia il componente di tonalità del modello cromatico HSV. La mappa dei colori inizia con il rosso, quindi passa al giallo, verde, ciano, blu, magenta per poi tornare al rosso. x deve essere specificato come un valore compreso tra 0 e 1.

**ColorMapHue** (x)

**ColorMapJet**

**ColorMapJet()** restituisce un valore ARGB di un colore da una mappa dei colori che inizia con il blu, passando dal ciano, giallo e arancione per poi tornare al rosso. x deve essere specificato come un valore compreso tra 0 e 1.

**ColorMapJet** (x)

## Funzioni colori predefiniti

Per i colori predefiniti nelle espressioni è possibile utilizzare le funzioni descritte di seguito. Ciascuna funzione restituisce una rappresentazione cromatica RGB.

In alternativa, è possibile specificare un parametro per il fattore alfa, nel qual caso verrà restituita la rappresentazione cromatica ARGB. Un fattore alfa pari a 0 corrisponde alla trasparenza completa, mentre un fattore alfa pari a 255 corrisponde all'opacità completa. Se non viene immesso un valore per alfa, verrà utilizzato 255.

Funzioni colori predefiniti

| Funzione colore       | RGB Valore    |
|-----------------------|---------------|
| black([alpha])        | (0,0,0)       |
| blue([alpha])         | (0,0,128)     |
| brown([alpha])        | (128,128,0)   |
| cyan([alpha])         | (0,128,128)   |
| darkgray([alpha])     | (128,128,128) |
| green([alpha])        | (0,128,0)     |
| lightblue([alpha])    | (0,0,255)     |
| lightcyan([alpha])    | (0,255,255)   |
| lightgray([alpha])    | (192,192,192) |
| lightgreen([alpha])   | (0,255,0)     |
| lightmagenta([alpha]) | (255,0,255)   |

|                   |               |
|-------------------|---------------|
| lightred([alpha]) | (255,0,0)     |
| magenta([alpha])  | (128,0,128)   |
| red([alpha])      | (128,0,0)     |
| white([alpha])    | (255,255,255) |
| yellow([alpha])   | (255,255,0)   |

**Esempi e risultati:**

## Esempi e risultati

| Esempi    | Risultati         |
|-----------|-------------------|
| Blue()    | RGB(0,0,128)      |
| Blue(128) | ARGB(128,0,0,128) |

**ARGB**

**ARGB()** viene utilizzata nelle espressioni per impostare o valutare le proprietà del colore di un oggetto del grafico, in cui il colore è definito da un componente rosso **r**, un componente verde **g** e un componente blu **b**, con un fattore alfa (opacità) di **alpha**.

**Sintassi:**

**ARGB** (alpha, r, g, b)

**Tipo di dati restituiti:** duale

## Argomenti

| Argomento      | Descrizione                                                                                                                                        |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>alpha</b>   | Valore della trasparenza nella scala 0 - 255. Il valore 0 corrisponde alla trasparenza completa e 255 corrisponde all'opacità completa.            |
| <b>r, g, b</b> | Valori del componente rosso, verde e blu. Un componente a colori pari a 0 corrisponde a nessun contributo e uno pari a 255 al contributo completo. |



*Tutti gli argomenti devono essere espressioni che si risolvono in numeri interi nella scala da 0 a 255.*

Se si interpreta il componente numerico e lo si formatta in notazione esadecimale, i valori dei componenti a colori saranno più facilmente visibili. Ad esempio, il verde chiaro ha il numero 4 278 255 360, che in notazione esadecimale è FF00FF00. Le prime due posizioni 'FF' (255) indicano il canale **alpha**. Le due posizioni successive '00' indicano la quantità di **red**, le due posizioni successive 'FF' indicano la quantità di **green** e le sue posizioni finali '00' indicano la quantità di **blue**.

| Esempio              | Esempio: espressione del grafico                              |
|----------------------|---------------------------------------------------------------|
| ARGB(128, 0, 128, 0) | Risultati<br>Restituisce il colore verde con semitrasparenza. |

### Esempio: principi fondamentali della funzione ARGB

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Product`
  - `Sales`

#### Script di caricamento

Example:

```
LOAD * Inline [  
Product, Sales  
Product A, 200  
Product B, 500  
Product C, 100  
Product D, 300  
Product E, 700  
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `Product`

Creare la seguente misura:

- `=Sum(Sales)`, per calcolare la somma delle vendite.

Nel pannello **Proprietà** della misura `sum(Sales)`, inserire l'espressione seguente come **Espressione colore sfondo**. Questa azione applicherà un colore per la cella in base al calcolo delle vendite per il prodotto.

- `=If(Sales > 400, ARGB(128, 0, 128, 0), If(Sales >= 200 and Sales <= 400, ARGB(128, 255, 255, 0), ARGB(128, 255, 0, 0)) )`

Results table

| Product   | Sum(Sales) |
|-----------|------------|
| Product A | 200        |
| Product B | 500        |
| Product C | 100        |
| Product D | 300        |
| Product E | 700        |

In questo esempio, fare quanto segue:

- I valori di vendita superiori a 400 sono evidenziati con uno sfondo verde.
- I valori di vendita tra 200 e 400 sono evidenziati con uno sfondo giallo.
- Tutti gli altri valori di vendita sono evidenziati con uno sfondo rosso.

### Esempio: evidenziazione dei dati sugli utili in un grafico

Espressione del grafico

#### Panoramica

Un set di dati contiene i dati sulle vendite suddivisi per cliente. Bisogna analizzare i dati ed evidenziare i margini di profitto a seconda che siano bassi, moderati o alti.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Customer`
  - `SalesValue`
  - `Cost`
  - `ProfitMargin`

#### Script di caricamento

Example:

```
LOAD * Inline [  
Customer, SalesValue, Cost, ProfitMargin  
Customer A, 500, 300, 40%  
Customer B, 800, 750, 6.25%  
Customer C, 400, 300, 25%  
Customer D, 600, 550, 8.33%  
Customer E, 900, 700, 22.22%  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- Customer

Creare le seguenti misure:

- =Sum(SalesValue), per calcolare la somma delle vendite.
- =Sum(Cost), per calcolare il valore totale del costo delle vendite.
- =Sum(ProfitMargin), per sommare il margine di profitto. Per mostrare questo valore come una percentuale, sotto la voce **Formattazione numero**, selezionare **Numero > Formattazione semplice > 12,34%**.

Nel pannello **Proprietà** della misura Sum(ProfitMargin), inserire l'espressione seguente come **Espressione colore sfondo**.

```
If(ProfitMargin > 0.3, ARGB(128, 0, 128, 0), // Green for high-profit customers
  If(ProfitMargin >= 0.1 and ProfitMargin <= 0.3, ARGB(128, 255, 165, 0), // Orange for
moderate-profit customers
  ARGB(128, 255, 0, 0) // Red for low-profit customers
)
```

Nel pannello **Aspetto**, alla voce **Presentazione**, disattivare **Totali**.

Results table

| Customer   | Sum(SalesValue) | Sum(Cost) | Sum(Profit Margin) |
|------------|-----------------|-----------|--------------------|
| Customer A | 500             | 300       | 40.00%             |
| Customer B | 800             | 750       | 6.25%              |
| Customer C | 400             | 300       | 25.00%             |
| Customer D | 600             | 550       | 8.33%              |
| Customer E | 900             | 700       | 22.22%             |

L'output della funzione ARGB rende i seguenti colori in base al margine di profitto del cliente:

- Verde, ARGB(128, 0, 128, 0), per i clienti con margini di profitto superiori al 30%.
- Arancione, ARGB(128, 255, 165, 0), per i clienti con margini di profitto moderati tra il 10% e il 30%.
- Rosso, ARGB(128, 255, 0, 0), per i clienti con un margine di profitto inferiore al 10%.

### Esempio: scenario per il grafico ARGB

Espressione del grafico

#### Panoramica

Un set di dati contiene i dati di vendita effettivi e previsti per Paese. Un direttore vendite vuole visualizzare i dati in un grafico a barre ed evidenziare i dati di vendita che sono superiori al valore previsto per il periodo.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Country`
  - `ActualSales`
  - `ProjectedSales`

#### Script di caricamento

Example:

```
Load * Inline
[Country, ActualSales, ProjectedSales
Sweden, 100000, 50000
Germany, 125000, 175000
Norway, 74850, 68500
Ireland, 45000, 48000
Sweden, 98000, 50000
Germany, 115000, 175000
Norway, 71850, 68500
Ireland, 31000, 48000
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare un nuovo grafico a barre e aggiungere questo campo come dimensione:

- `Country`

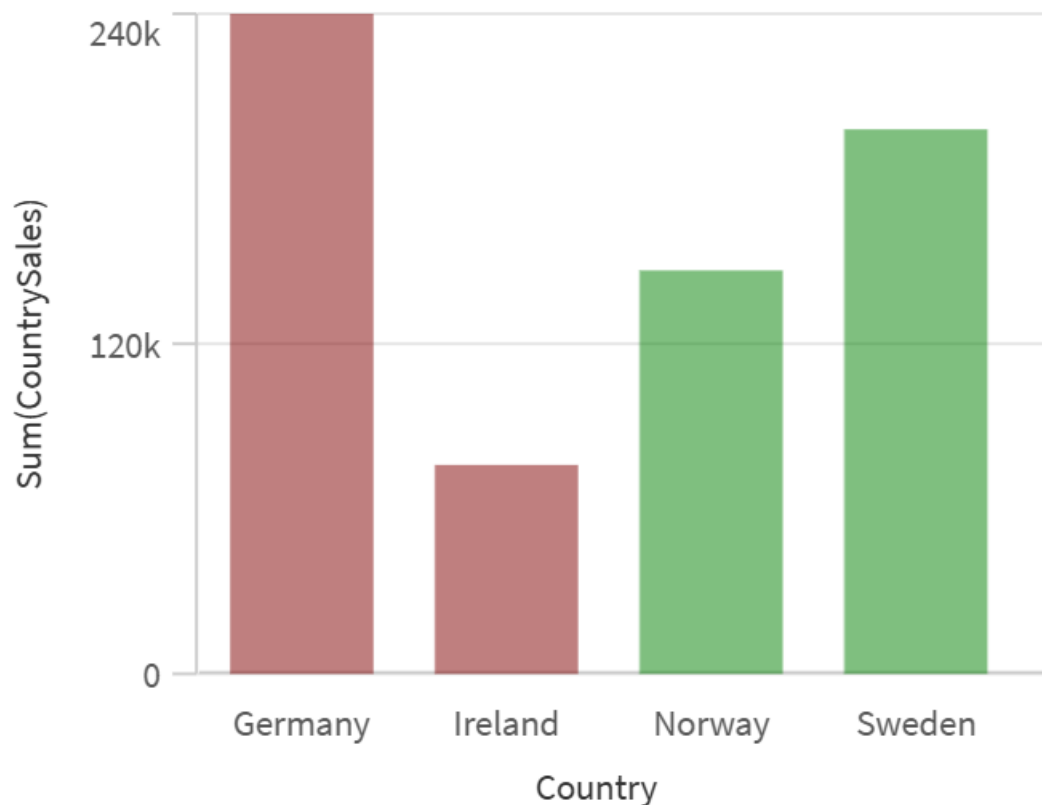
Creare la seguente misura:

- `=Sum(ActualSales)`, per calcolare il valore totale delle vendite per il Paese.

Aggiungere un'espressione cromatica per confrontare le vendite effettive con quelle previste. Nel pannello delle proprietà, in **Aspetto > Colori e legenda**, disattivare **Colori: Automatico** per passare a **Personalizzato**, quindi selezionare **Per espressione** dall'elenco. Inserire la seguente espressione:

- `=If(Sum(ActualSales)>Sum(ProjectedSales),ARGB(128,0,128,0),ARGB(128,128,0,0))`, per evidenziare in verde le occorrenze in cui le vendite effettive superano le vendite previste e in rosso quelle in cui le vendite previste superano le vendite effettive.

Risultati: grafico a barre delle vendite per Paese



È possibile vedere che il colore della barra è verde per i Paesi con vendite superiori al valore previsto, mentre quelli con vendite inferiori al valore previsto appaiono in rosso. Per esempio, sweden è verde perché il valore di vendita di 98,000 è maggiore del valore previsto di 50,000.

### RGB

**RGB()** restituisce un intero corrispondente al codice colore del colore definito dai tre parametri: la componente rossa *r*, la componente verde *g* e la componente blu *b*. Queste componenti devono avere valori interi compresi tra 0 e 255. La funzione può essere usata nelle espressioni per impostare o valutare le proprietà di colore di un oggetto grafico.

#### Sintassi:

**RGB** (*r*, *g*, *b*)

Tipo di dati restituiti: duale

Argomenti

| Argomento      | Descrizione                                                                                                                                        |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>r, g, b</b> | Valori del componente rosso, verde e blu. Un componente a colori pari a 0 corrisponde a nessun contributo e uno pari a 255 al contributo completo. |



*Tutti gli argomenti devono essere espressioni che si risolvono in numeri interi nella scala da 0 a 255.*

Se si interpreta il componente numerico e lo si formatta in notazione esadecimale, i valori dei componenti a colori saranno più facilmente visibili. Ad esempio, il verde chiaro ha il numero 4 278 255 360, che in notazione esadecimale è FF00FF00. Le prime due posizioni 'FF' (255) indicano il canale **alpha**. Nelle funzioni **RGB** e **HSL**, è sempre 'FF' (opaco). Le due posizioni successive '00' indicano la quantità di **red**, le due posizioni successive 'FF' indicano la quantità di **green** e le sue posizioni finali '00' indicano la quantità di **blue**.

Esempio: espressione del grafico

### Esempio

```
If(Sum(Sales)>Sum(Budget), RGB
(255,0,0), RGB(100,80,120))
```

### Risultati

Restituisce il colore viola o rosso a seconda del valore del confronto tra Sales e Budget.

## Esempio: principi fondamentali della funzione RGB

Espressione del grafico

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata Example.
- I seguenti campi nella tabella dati:
  - Product
  - Sales

### Script di caricamento

Example:

```
LOAD * Inline [
Product, Sales
Product A, 200
Product B, 500
Product C, 100
```

```
Product D, 300  
Product E, 700  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- Product

Creare la seguente misura:

- =Sum(Sales), per calcolare la somma delle vendite.

Nel pannello **Proprietà** della misura sum(Sales), inserire l'espressione seguente come **Espressione colore sfondo**. Questa azione applicherà un colore per la cella in base al calcolo delle vendite per il prodotto.

- =If(Sales > 400, RGB(0, 128, 0), If(Sales >= 200 and Sales <= 400, RGB( 255, 255, 0), RGB(255, 0, 0)) )

Results table

| Product   | Sum(Sales) |
|-----------|------------|
| Product A | 200        |
| Product B | 500        |
| Product C | 100        |
| Product D | 300        |
| Product E | 700        |

In questo esempio, fare quanto segue:

- I valori di vendita superiori a 400 sono evidenziati con uno sfondo verde.
- I valori di vendita tra 200 e 400 sono evidenziati con uno sfondo giallo.
- Tutti gli altri valori di vendita sono evidenziati con uno sfondo rosso.

### Esempio: evidenziazione dei dati sugli utili in un grafico

Espressione del grafico

#### Panoramica

Un set di dati contiene i dati sulle vendite suddivisi per cliente. Bisogna analizzare i dati ed evidenziare i margini di profitto a seconda che siano bassi, moderati o alti.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Customer`
  - `SalesValue`
  - `Cost`
  - `ProfitMargin`

### Script di caricamento

Example:

```
LOAD * Inline [  
Customer, SalesValue, Cost, ProfitMargin  
Customer A, 500, 300, 40%  
Customer B, 800, 750, 6.25%  
Customer C, 400, 300, 25%  
Customer D, 600, 550, 8.33%  
Customer E, 900, 700, 22.22%  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `Customer`

Creare le seguenti misure:

- `=Sum(SalesValue)`, per calcolare la somma delle vendite.
- `=Sum(Cost)`, per calcolare il valore totale del costo delle vendite.
- `=Sum(ProfitMargin)`, per sommare il margine di profitto. Per mostrare questo valore come una percentuale, sotto la voce **Formattazione numero**, selezionare **Numero > Formattazione semplice > 12,34%**.

Nel pannello **Proprietà** della misura `Sum(ProfitMargin)`, inserire l'espressione seguente come **Espressione colore sfondo**.

```
If(ProfitMargin > 0.3, RGB(0, 128, 0), // Green for high-profit customers  
If(ProfitMargin >= 0.1 and ProfitMargin <= 0.3, RGB(255, 165, 0), // Orange for moderate-  
profit customers  
RGB(255, 0, 0) // Red for low-profit customers  
)  
)
```

Nel pannello **Aspetto**, alla voce **Presentazione**, disattivare **Totali**.

Results table

| Customer   | Sum(SalesValue) | Sum(Cost) | Sum(Profit Margin) |
|------------|-----------------|-----------|--------------------|
| Customer A | 500             | 300       | 40.00%             |

| Customer   | Sum(SalesValue) | Sum(Cost) | Sum(Profit Margin) |
|------------|-----------------|-----------|--------------------|
| Customer B | 800             | 750       | 6.25%              |
| Customer C | 400             | 300       | 25.00%             |
| Customer D | 600             | 550       | 8.33%              |
| Customer E | 900             | 700       | 22.22%             |

L'output della funzione RGB rende i seguenti colori in base al margine di profitto del cliente:

- Verde, RGB(0, 128, 0), per i clienti con margini di profitto superiori al 30%.
- Arancione, RGB(255, 165, 0), per i clienti con margini di profitto moderati tra il 10% e il 30%.
- Rosso, RGB(255, 0, 0), per i clienti con un margine di profitto inferiore al 10%.

### Esempio: scenario per il grafico RGB

Espressione del grafico

#### Panoramica

Un set di dati contiene i dati di vendita effettivi e previsti per Paese. Un direttore vendite vuole visualizzare i dati in un grafico a barre ed evidenziare i dati di vendita che sono superiori al valore previsto per quel periodo.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata Example.
- I seguenti campi nella tabella dati:
  - Country
  - ActualSales
  - ProjectedSales

#### Script di caricamento

Example:

```
Load * Inline
[Country, ActualSales, ProjectedSales
Sweden, 100000, 50000
Germany, 125000, 175000
Norway, 74850, 68500
Ireland, 45000, 48000
Sweden, 98000, 50000
Germany, 115000, 175000
Norway, 71850, 68500
Ireland, 31000, 48000
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare un nuovo grafico a barre e aggiungere questo campo come dimensione:

- Country

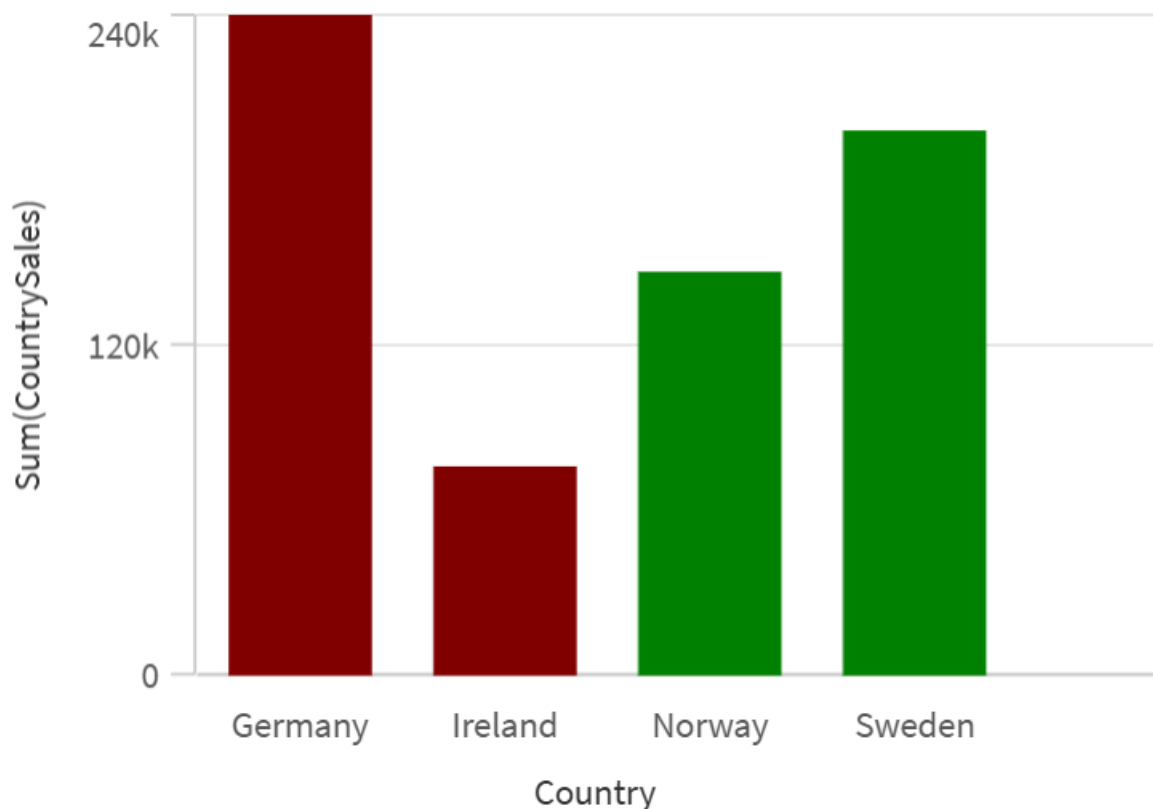
Creare la seguente misura:

- `=Sum(ActualSales)`, per calcolare il valore totale delle vendite per il Paese.

Aggiungere un'espressione cromatica personalizzata per confrontare le vendite effettive con quelle previste. Nel pannello delle proprietà, in **Aspetto > Colori e legenda**, disattivare **Colori: Automatico** per passare a **Personalizzato**, quindi selezionare **Per espressione** dall'elenco. Inserire la seguente espressione:

- `=If(Sum(ActualSales)>Sum(ProjectedSales),RGB(0,128,0),RGB(128,0,0))`, per evidenziare in verde le occorrenze in cui le vendite effettive superano le vendite previste e in rosso quelle in cui le vendite previste superano le vendite effettive.

*Risultati: grafico a barre delle vendite per Paese*



È possibile vedere che il colore della barra è verde per i Paesi con vendite superiori al valore previsto, mentre quelli con vendite inferiori al valore previsto appaiono in rosso. Per esempio, sweden è verde perché il valore di vendita di 98,000 è maggiore del valore previsto di 50,000.

### HSL

**HSL()** viene utilizzata nelle espressioni per impostare o valutare le proprietà del colore di un oggetto del grafico, dove il colore è definito dai valori **hue**, **saturation** e **luminosity** tra 0 e 1.

#### Sintassi:

```
HSL (hue, saturation, luminosity)
```

Tipo di dati restituiti: duale

#### Argomenti

| Argomento                   | Descrizione                                                               |
|-----------------------------|---------------------------------------------------------------------------|
| hue, saturation, luminosity | I valori dei componenti Hue, saturation, e luminosity compresi tra 0 e 1. |



*Tutti gli argomenti devono essere espressioni che si risolvono in numeri interi nella scala da 0 a 1.*

Se si interpreta il componente numerico e lo si formatta in notazione esadecimale, i valori RGB dei componenti di colore saranno più facilmente visibili. Ad esempio, il verde chiaro ha il numero 4 278 255 360, che in notazione esadecimale è FF00FF00 e RGB (0,255,0). Questo equivale a HSL (80/240, 240/240, 120/240) - un valore HSL di (0.33, 1, 0.5).

#### Esempio: espressione del grafico

##### Esempio

```
If(Sum(Sales)>Sum(Budget),HSL( 0.33, 1, 0.25),HSL(0, 1, 0.5))
```

##### Risultati

Restituisce il colore viola o rosso a seconda del valore del confronto tra sales e Budget.

### Esempio: principi fondamentali della funzione HSL

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:

- Product
- Sales

### Script di caricamento

Example:

```
LOAD * Inline [  
Product, Sales  
Product A, 200  
Product B, 500  
Product C, 100  
Product D, 300  
Product E, 700  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- Product

Creare la seguente misura:

- =Sum(Sales), per calcolare la somma delle vendite.

Nel pannello **Proprietà** della misura sum(Sales), inserire l'espressione seguente come **Espressione colore sfondo**. Questa azione applicherà un colore per la cella in base al calcolo delle vendite per il prodotto.

- =If(Sales > 400, HSL(.33, 1, 0.25), If(Sales >= 200 and Sales <= 400, HSL(.167, 1, 0.5), HSL(0, 1, 0.5)))

Results table

| Product   | Sum(Sales) |
|-----------|------------|
| Product A | 200        |
| Product B | 500        |
| Product C | 100        |
| Product D | 300        |
| Product E | 700        |

In questo esempio, fare quanto segue:

- I valori di vendita superiori a 400, come Product B e E, sono evidenziati con uno sfondo verde.
- I valori di vendita compresi tra 200 e 400, come Product A e D, sono evidenziati con uno sfondo giallo.
- Tutti gli altri valori di vendita sono evidenziati con uno sfondo rosso.

### Esempio: evidenziazione dei dati sugli utili in un grafico

Espressione del grafico

#### Panoramica

Un set di dati contiene i dati delle vendite. Bisogna analizzare i dati ed evidenziare i margini di profitto a seconda che siano bassi, moderati o alti.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Customer`
  - `SalesValue`
  - `Cost`
  - `ProfitMargin`

#### Script di caricamento

Example:

```
LOAD * Inline [
Customer, SalesValue, Cost, ProfitMargin
Customer A, 500, 300, 40%
Customer B, 800, 750, 6.25%
Customer C, 400, 300, 25%
Customer D, 600, 550, 8.33%
Customer E, 900, 700, 22.22%
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `Customer`

Creare le seguenti misure:

- `=Sum(SalesValue)`, per calcolare la somma delle vendite.
- `=Sum(Cost)`, per calcolare il valore totale del costo delle vendite.
- `=Sum(ProfitMargin)`, per sommare il margine di profitto. Per mostrare questo valore come una percentuale, sotto la voce **Formattazione numero**, selezionare **Numero > Formattazione semplice > 12,34%**.

Nel pannello **Proprietà** della misura `Sum(ProfitMargin)`, inserire l'espressione seguente come **Espressione colore sfondo**.

```
If(ProfitMargin > 0.3, HSL( 0.33, 1, 0.25), // Green for high-profit customers
  If(ProfitMargin >= 0.1 and ProfitMargin <= 0.3, HSL( 0.11, 1, 0.5), // Orange for
moderate-profit customers
    HSL(0, 1, 0.5) // Red for low-profit customers
  )
)
```

Nel pannello **Aspetto**, alla voce **Presentazione**, disattivare **Totali**.

Results table

| Customer   | Sum(SalesValue) | Sum(Cost) | Sum([Profit Margin]) |
|------------|-----------------|-----------|----------------------|
| Customer A | 500             | 300       | 40.00%               |
| Customer B | 800             | 750       | 6.25%                |
| Customer C | 400             | 300       | 25.00%               |
| Customer D | 600             | 550       | 8.33%                |
| Customer E | 900             | 700       | 22.22%               |

L'output della funzione HSL rende i seguenti colori in base al margine di profitto del cliente:

- Verde, HSL( 0.33, 1, 0.25), per i clienti con margini di profitto superiori al 30%.
- Arancione, HSL( 0.11, 1, 0.5), per i clienti con margini di profitto moderati tra il 10% e il 30%.
- Rosso, HSL(0, 1, 0.5), per i clienti con un margine di profitto inferiore al 10%.

### Esempio: scenario per il grafico HSL

Espressione del grafico

#### Panoramica

Un set di dati contiene i dati di vendita effettivi e previsti per Paese. Un responsabile vendite vuole visualizzare i dati in un grafico a barre ed evidenziare i dati di vendita che sono superiori al valore previsto per quel periodo.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Country`
  - `ActualSales`
  - `ProjectedSales`

#### Script di caricamento

Example:

Load \* Inline

[Country, ActualSales, ProjectedSales

```
Sweden, 100000, 50000  
Germany, 125000, 175000  
Norway, 74850, 68500  
Ireland, 45000, 48000  
Sweden, 98000, 50000  
Germany, 115000, 175000  
Norway, 71850, 68500  
Ireland, 31000, 48000  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare un nuovo grafico a barre e aggiungere questo campo come dimensione:

- Country

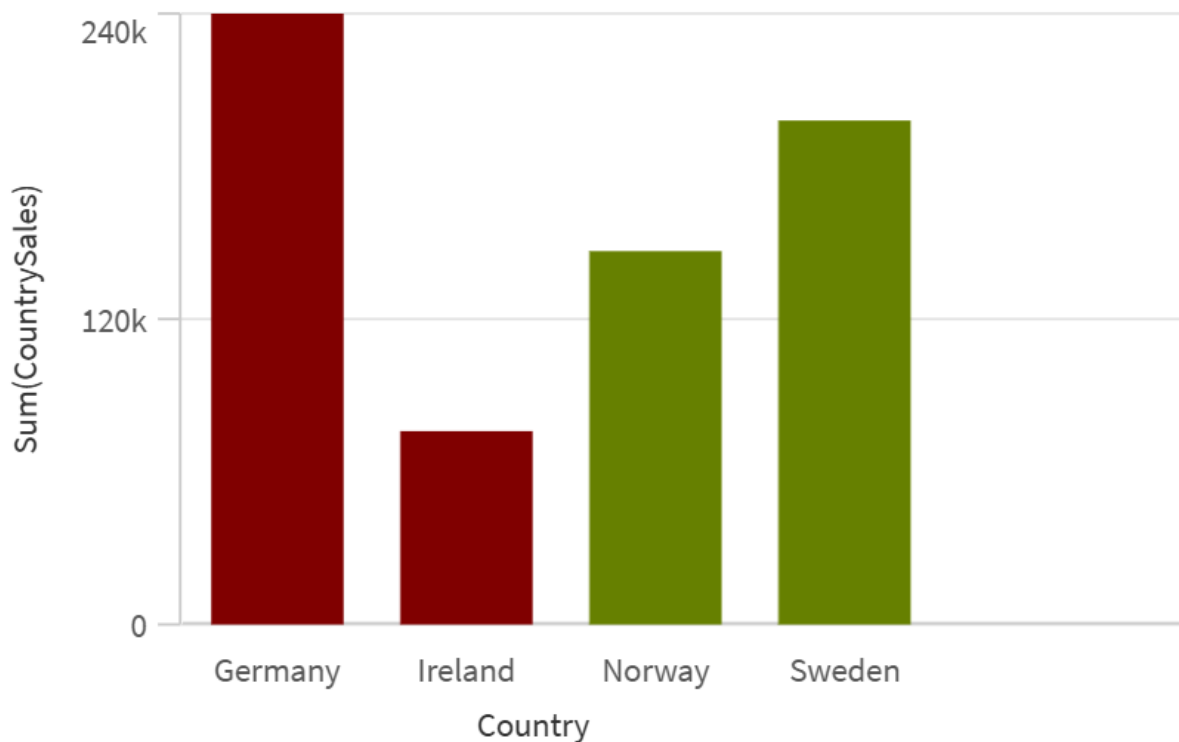
Creare la seguente misura:

- `=Sum(ActualSales)`, per calcolare il valore totale delle vendite per il Paese.

Aggiungere un'espressione cromatica per confrontare le vendite effettive con quelle previste. Nel pannello delle proprietà, in **Aspetto > Colori e legenda**, disattivare **Colori: Automatico** per passare a **Personalizzato**, quindi selezionare **Per espressione** dall'elenco. Inserire la seguente espressione:

- `=If(Sum(ActualSales)>Sum(ProjectedSales),HSL(1.2,1.0,0.25),HSL(0, 1.0, 0.25))`, per evidenziare in verde le occorrenze in cui le vendite effettive superano le vendite previste e in rosso quelle in cui le vendite previste superano le vendite effettive.

Risultati: grafico a barre delle vendite per Paese



È possibile vedere che il colore della barra è verde per i Paesi con vendite superiori al valore previsto, mentre quelli con vendite inferiori al valore previsto appaiono in rosso. Per esempio, sweden è verde perché il valore di vendita di 98,000 è maggiore del valore previsto di 50,000.

### 8.5 Funzioni condizionali

Tutte le funzioni condizionali valutano una condizione e restituiscono differenti risposte in base al valore della condizione. Le funzioni possono essere utilizzate nello script di caricamento dei dati e nelle espressioni grafiche.

#### Panoramica sulle funzioni condizionali

Ciascuna funzione viene descritta ulteriormente dopo la panoramica. È inoltre possibile fare clic sul nome della funzione nella sintassi per accedere immediatamente ai dettagli per tale funzione specifica.

##### **alt**

La funzione **alt** restituisce il primo dei parametri che presenta una rappresentazione numerica valida. Se nessuna corrispondenza viene trovata, verrà restituito l'ultimo parametro. Può essere utilizzato un numero qualsiasi di parametri.

```
alt(expr1[ , expr2 , expr3 , ...] , else)
```

### class

La funzione **class** assegna il primo parametro a un intervallo di classe. Viene restituito un valore duale in cui  $a \leq x < b$  rappresenta il valore testuale dove  $a$  e  $b$  rappresentano i limiti superiore e inferiore del contenitore e il limite inferiore è un valore numerico.

```
class (expression, interval [ , label [ , offset ]])
```

### coalesce

La funzione **coalesce** restituisce il primo dei parametri che presenta una rappresentazione non-NULL valida. Può essere utilizzato un numero qualsiasi di parametri.

```
coalesce(expr1 [ , expr2 , expr3 , ...])
```

### if

La funzione **if** restituisce un valore a seconda che la condizione fornita con la funzione esegua la valutazione come True o come False.

```
if (condition , then , else)
```

### match

La funzione **match** confronta il primo parametro con tutti i parametri seguenti e restituisce la posizione numerica delle espressioni corrispondenti. Il confronto rispetta la distinzione maiuscole/minuscole.

```
match ( str, expr1 [ , expr2,...exprN ])
```

### mixmatch

La funzione **mixmatch** confronta il primo parametro con tutti i parametri seguenti e restituisce la posizione numerica delle espressioni corrispondenti. Il confronto non distingue tra caratteri maiuscoli o minuscoli né i sistemi di caratteri del giapponese hiragana e katakana.

```
mixmatch ( str, expr1 [ , expr2,...exprN ])
```

### pick

La funzione **pick** restituisce l'espressione numero  $n$  nell'elenco.

```
pick (n, expr1 [ , expr2,...exprN])
```

### wildmatch

La funzione **wildmatch** confronta il primo parametro con quelli seguenti e restituisce il numero di espressioni corrispondenti. Consente l'utilizzo di caratteri jolly (  $*$  e  $?$  ) per il confronto delle stringhe.  $*$  corrisponde a qualsiasi sequenza di caratteri.  $?$  corrisponde a qualsiasi carattere singolo. Il confronto non distingue tra caratteri maiuscoli o minuscoli né i sistemi di caratteri del giapponese hiragana e katakana.

```
wildmatch ( str, expr1 [ , expr2,...exprN ])
```

### alt

La funzione **alt** restituisce il primo dei parametri che presenta una rappresentazione numerica valida. Se nessuna corrispondenza viene trovata, verrà restituito l'ultimo parametro. Può essere utilizzato un numero qualsiasi di parametri.

### Sintassi:

```
alt(expr1[ , expr2 , expr3 , ...] , else)
```

**Tipo di dati restituiti:** numero intero

### Argomenti

| Argomento    | Descrizione                                                                                    |
|--------------|------------------------------------------------------------------------------------------------|
| <b>expr1</b> | La prima espressione da controllare per una valida rappresentazione numerica.                  |
| <b>expr2</b> | La seconda espressione da controllare per una valida rappresentazione numerica.                |
| <b>expr3</b> | La terza espressione da controllare per una valida rappresentazione numerica.                  |
| <b>else</b>  | Valore restituito se nessuno dei parametri precedenti ha una valida rappresentazione numerica. |

La funzione alt viene spesso utilizzata con le funzioni di interpretazione numerica o della data. In questo modo Qlik Sense può provare differenti formati di data in un ordine con priorità. Può anche essere utilizzata per gestire valori NULL in espressioni numeriche.

### Esempio: espressioni dei grafici

| Esempio                                                                                                                            | Risultato                                                                                                                                                                                                                                                                                                                                                                                        |
|------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>alt( date#( dat , 'YYYY/MM/DD' ),<br/>date#( dat , 'MM/DD/YYYY' ),<br/>date#( dat , 'MM/DD/YY' ),<br/>'No valid date' )</pre> | Questa espressione verificherà se la data del campo contiene una data conforme a uno dei tre formati specificati. In tal caso, restituirà un valore duale contenente la stringa originale e una rappresentazione numerica valida di una data. Se non viene trovata nessuna corrispondenza, verrà restituito il testo <code>No valid date</code> (senza alcuna rappresentazione numerica valida). |
| <pre>alt(Sales,0) + alt(Margin,0)</pre>                                                                                            | L'espressione aggiunge i campi <code>Sales</code> e <code>Margin</code> , sostituendo qualsiasi valore mancante (NULL) con il valore 0.                                                                                                                                                                                                                                                          |

### Esempio: principi fondamentali della funzione alt

#### Espressione del grafico

#### Panoramica

Un set di dati contiene dati incompleti sulle vendite e sulle previsioni. Si desidera sostituire i valori mancanti con valori alternativi validi per colmare le lacune.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `ID`
  - `Sales`
  - `Forecast`

### Script di caricamento

Example:

```
LOAD * INLINE [  
ID, Sales, Forecast  
1, 500, 450  
2, , 300  
3, 200, 250  
4, , 400  
5, 350,  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `ID`
- `Sales`
- `Forecast`

Creare la seguente misura:

- `=Sum(alt(Sales, Forecast))` per restituire una rappresentazione numerica valida, sostituendo i valori mancanti con valori alternativi validi.

| Tabella dei risultati |       |          |                           |
|-----------------------|-------|----------|---------------------------|
| ID                    | Sales | Forecast | Sum(alt(Sales, Forecast)) |
| Totals                |       |          | 1750                      |
| 1                     | 500   | 450      | 500                       |
| 2                     |       | 300      | 300                       |
| 3                     | 200   | 250      | 200                       |
| 4                     |       | 400      | 400                       |
| 5                     | 350   |          | 350                       |

La funzione `alt` restituisce il valore `Forecast` nel calcolo della misura per le righe `ID` con i valori `Sales` mancanti. Sostituendo i valori mancanti con valori alternativi validi, è possibile generare un set di dati completamente popolato.

### Esempio: utilizzo di alt per sostituire i valori mancanti

Espressione del grafico

#### Panoramica

Un'azienda raccoglie i dati relativi alla temperatura da più sensori. Si desidera sostituire i dati mancanti di un sensore con i dati dell'altro sensore.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Time`
  - `Temp_Sensor1`
  - `Temp_Sensor2`

#### Script di caricamento

Example:

```
LOAD * INLINE [
Time, Temp_Sensor1, Temp_Sensor2
08:00, 23.5, 24.0
09:00, , 23.8
10:00, 24.1,
11:00, , 24.5
12:00, 23.9, 24.1
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `Time`
- `Temp_Sensor1`
- `Temp_Sensor2`

Creare la seguente misura:

- `=Sum(alt(Temp_Sensor1, Temp_Sensor2))` per restituire una rappresentazione numerica valida.

Tabella dei risultati

| Time   | Temp_Sensor1 | Temp_Sensor2 | Sum(alt(Temp_Sensor1, Temp_Sensor2)) |
|--------|--------------|--------------|--------------------------------------|
| Totals |              |              | 119.8                                |
| 08:00  | 23.5         | 24.0         | 23.5                                 |

| Time   | Temp_Sensor1 | Temp_Sensor2 | Sum(alt(Temp_Sensor1, Temp_Sensor2)) |
|--------|--------------|--------------|--------------------------------------|
| Totals |              |              | 119.8                                |
| 09:00  |              | 23.8         | 23.8                                 |
| 10:00  | 24.1         |              | 24.1                                 |
| 11:00  |              | 24.5         | 24.5                                 |
| 12:00  | 23.9         | 24.1         | 23.9                                 |

Osservando i risultati, la funzione `alt` ha sostituito i dati mancanti del sensore con risultati alternativi validi dell'altro sensore. Questo consente di completare il set di dati riempiendo le lacune.

### Esempio: scenario per l'applicazione della funzione `alt`

Espressione del grafico

#### Panoramica

Questo scenario mostra diversi modi in cui è possibile utilizzare la funzione `alt` per sostituire i valori mancanti o null.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `ProductID`
  - `ProductName`
  - `SalesAmount`
  - `ReturnAmount`
  - `Profit`
  - `Discount`
  - `Category`
  - `Year`
  - `Region`
  - `CustomerRating`

#### Script di caricamento

```
Set NullInterpret = '';
```

Example:

```
LOAD * INLINE [  
ProductID, ProductName, SalesAmount, ReturnAmount, Profit, Discount, Category, Year, Region,  
CustomerRating  
101, "Laptop", 1200, 200, 500, , "Electronics", 2023, "North",  
102, "Tablet", 800, , 300, 50, "Electronics", 2023, "South", 4  
103, "Smartphone", 1500, , 600, 100, , 2022, "East", 5
```

```
104, "Printer", 400, , , 20, "Accessories", 2021, , 3
105, , 1000, 100, , 30, "Electronics", 2023, "West", 4
106, "Monitor", , 150, 250, , "Accessories", 2022, "North", 5
107, "Keyboard", 200, , 80, 5, , 2022, "South", 2
108, "Mouse", , 50, , , "Accessories", , "East",
109, , 600, , , , "Electronics", 2023, , 4
110, "Headphones", 300, 30, 100, , "Accessories", 2021, "West", 3
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- ProductID

Creare la seguente dimensione calcolata:

- =Coalesce( ProductName, 'Unknown Product'), per fornire la descrizione Unknown Product per tutti i prodotti che non hanno un valore per ProductName.

Creare le seguenti misure:

- =Sum(alt(SalesAmount, ReturnAmount, Profit)), per restituire un numero valido se non viene trovato nessun valore per SalesAmount.
- =avg(alt(CustomerRating, 3)), per restituire una valutazione standard valida del cliente, se non è stata fornita inizialmente.
- =Sum(alt(Discount, SalesAmount \* 0.05)), per restituire uno sconto valido.
- =Sum(alt(Profit, SalesAmount \* 0.1)), per restituire un valore valido per gli utili.
- =Sum(alt(SalesAmount, avg(TOTAL SalesAmount))), per restituire un valore valido per le vendite.
- =Sum(alt(Discount, avg(TOTAL Discount))), per restituire un valore valido per lo sconto.

Tabella dei risultati

| ProductID     | Coalesce(ProductName, 'Unknown Product') | Sum(alt(SalesAmount, ReturnAmount, Profit)) | avg(alt(CustomerRating, 3)) | Sum(alt(Discount, SalesAmount * 0.05)) | Sum(alt(Profit, SalesAmount * 0.1)) | Sum(alt(SalesAmount, avg(TOTAL SalesAmount))) | Sum(alt(Discount, avg(TOTAL Discount))) |
|---------------|------------------------------------------|---------------------------------------------|-----------------------------|----------------------------------------|-------------------------------------|-----------------------------------------------|-----------------------------------------|
| <b>Totali</b> |                                          | <b>6200</b>                                 | <b>3.6</b>                  | <b>310</b>                             | <b>2030</b>                         | <b>7500</b>                                   | <b>410</b>                              |
| 101           | Computer portatile                       | 1200                                        | 3                           | 60                                     | 500                                 | 1200                                          | 41                                      |
| 102           | Tablet                                   | 800                                         | 4                           | 50                                     | 300                                 | 500                                           | 50                                      |
| 103           | Smartphone                               | 1500                                        | 5                           | 100                                    | 600                                 | 800                                           | 100                                     |
| 104           | Stampante                                | 400                                         | 3                           | 20                                     | 40                                  | 400                                           | 20                                      |

| ProductID     | Coalesce(ProductName, 'Unknown Product') | Sum(alt (SalesAmount, ReturnAmount, Profit)) | avg(alt (CustomerRating, 3)) | Sum(alt (Discount, SalesAmount * 0.05)) | Sum(alt (Profit, SalesAmount * 0.1)) | Sum(alt (SalesAmount, avg (TOTAL SalesAmount))) | Sum(alt (Discount, avg (TOTAL Discount))) |
|---------------|------------------------------------------|----------------------------------------------|------------------------------|-----------------------------------------|--------------------------------------|-------------------------------------------------|-------------------------------------------|
| <b>Totali</b> |                                          | <b>6200</b>                                  | <b>3.6</b>                   | <b>310</b>                              | <b>2030</b>                          | <b>7500</b>                                     | <b>410</b>                                |
| 105           | Prodotto sconosciuto                     | 1000                                         | 4                            | 30                                      | 100                                  | 1000                                            | 30                                        |
| 106           | Monitoraggio                             | 150                                          | 5                            | 0                                       | 250                                  | 750                                             | 41                                        |
| 107           | Tastiera                                 | 200                                          | 2                            | 5                                       | 80                                   | 200                                             | 5                                         |
| 108           | Mouse                                    | 50                                           | 3                            | 0                                       | 0                                    | 750                                             | 41                                        |
| 109           | Prodotto sconosciuto                     | 600                                          | 4                            | 30                                      | 60                                   | 600                                             | 41                                        |
| 110           | Cuffie                                   | 300                                          | 3                            | 15                                      | 100                                  | 300                                             | 41                                        |

La funzione `alt` assicura che quando mancano dei valori, è possibile calcolare e restituire dei valori alternativi. Osservare i seguenti risultati:

- La colonna 2 utilizza la funzione `Coalesce` per restituire il valore `unknown Product` per tutti i valori null. La funzione `alt` non può essere utilizzata perché si applica solo ai valori numerici.
- La colonna 3 restituisce il valore di `SalesAmount`. Se non viene trovato alcun valore, viene restituito il valore `ReturnAmount`. Se non viene trovato il valore `ReturnAmount`, viene restituito il valore `Profit`. Questo aiuta a garantire che la colonna venga popolata con un valore se non è presente alcun valore per `SalesAmount`.
- La colonna 4 restituisce il valore per `CustomerRating`. Se non viene trovato alcun valore, il valore predefinito è 3. Ad esempio, il prodotto `Laptop` non ha una valutazione dei clienti, quindi viene restituito il valore 3. La misura calcola quindi la media della colonna.
- La colonna 5 restituisce il valore dello sconto. Se non è disponibile alcun valore, viene calcolato come il 5% del valore `SalesAmount`.
- La colonna 6 restituisce il valore `Profit`. Se non è disponibile alcun valore, viene restituito il 10% per il valore `SalesAmount`.
- La colonna 7 restituisce il valore per `SalesAmount`. Se non viene fornito alcun valore, viene restituita la media di `SalesAmount` per l'intero set di dati. Ad esempio, i prodotti `Monitor` e `Mouse` non hanno alcun valore per `SalesAmount`, quindi entrambi restituiscono 750, che è la media di tutti i valori di `SalesAmount`.
- La colonna 8 restituisce il valore `Discount`. Se non viene trovato alcun valore, il sistema calcola e restituisce lo sconto medio per l'intero set di dati.

## class

La funzione **class** assegna il primo parametro a un intervallo di classe. Viene restituito un valore duale in cui  $a \leq x < b$  rappresenta il valore testuale dove a e b rappresentano i limiti superiore e inferiore del contenitore e il limite inferiore è un valore numerico.

### Sintassi:

```
class(expression, interval [ , label [ , offset ]])
```

Tipo di dati restituiti: duale

### Argomenti

| Argomento       | Descrizione                                                                                                                                              |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>interval</b> | Un numero che specifica la larghezza del contenitore.                                                                                                    |
| <b>label</b>    | Una stringa arbitraria che può sostituire la 'x' nel testo del risultato.                                                                                |
| <b>offset</b>   | Un numero che può essere utilizzato come offset dal punto di partenza predefinito della classificazione. Il punto di partenza predefinito è in genere 0. |

### Esempio: espressioni dei grafici

| Esempio                                          | Risultato                               |
|--------------------------------------------------|-----------------------------------------|
| <code>class( var,10 ) con var = 23</code>        | Restituisce $20 \leq x < 30$            |
| <code>class( var,5,'value' ) con var = 23</code> | Restituisce $20 \leq \text{value} < 25$ |
| <code>class( var,10,'x',5 ) con var = 23</code>  | Restituisce $15 \leq x < 25$            |

## Esempio: principi fondamentali della funzione class

Espressione del grafico

### Panoramica

Viene caricata una tabella dati che contiene nomi ed età. Si desidera classificare le persone in base a gruppi di età (con un intervallo di dieci anni) e fornire il conteggio del numero di persone in ciascun gruppo.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - Age
  - Name

### Script di caricamento

Example:

```
LOAD * INLINE [  
Age, Name  
25, John  
42, Karen  
53, Yoshi  
43, Bob  
27, Charles  
];
```

### Risultati

1. Caricare i dati e aprire un foglio. Creare due nuove tabelle.
2. Nella prima tabella, fare quanto segue:
  - a. Aggiungere le seguenti dimensioni:
    - Age
    - Name
  - b. Creare la seguente misura:
    - =class(Age, 10, 'age'), per classificare i dati in categorie di età.
3. Nella seconda tabella, fare quanto segue:
  - a. Aggiungere questa espressione come dimensione.
    - =class(Age, 10, 'age')
  - b. Creare la seguente misura:
    - =Count(Age), per calcolare il numero di persone classificate in base all'età.

La prima tabella mostra i dati non elaborati per Name e Age e la categoria di età in cui sono classificati.

Tabella dei risultati 1

| Name    | Age | class(Age, 10, 'age') |
|---------|-----|-----------------------|
| Bob     | 43  | 40 <= age < 50        |
| Charles | 27  | 20 <= age < 30        |
| John    | 25  | 20 <= age < 30        |
| Karen   | 42  | 40 <= age < 50        |
| Yoshi   | 53  | 50 <= age < 60        |

Nella seconda tabella, è possibile vedere come i dati sono ora classificati e riassunti. I risultati della funzione class hanno creato 3 categorie di età nella prima misura. La seconda misura calcola il numero di persone in ciascuna categoria di età.

Tabella dei risultati 2

| <code>class(Age, 10, 'age')</code> | <code>Count(Age)</code> |
|------------------------------------|-------------------------|
| Totals                             | 5                       |
| 20 <= age < 30                     | 2                       |
| 40 <= age < 50                     | 2                       |
| 50 <= age < 60                     | 1                       |

### Script di caricamento

#### Panoramica

Viene caricata una tabella dati che contiene nomi ed età. Si desidera aggiungere un campo che classifichi ciascun utente in base a un gruppo di età con un intervallo temporale di dieci anni. La tabella di origine originale ha un aspetto simile al seguente.

Tabella di origine originale

| Name  | Age |
|-------|-----|
| John  | 25  |
| Karen | 42  |
| Yoshi | 53  |

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- L'istruzione di un caricamento precedente che usa la funzione **class** per creare un nuovo campo denominato Agegroup.

### Script di caricamento

```
LOAD *,
class(Age, 10, 'age') As Agegroup;
```

```
LOAD * INLINE
[ Age, Name
25, John
42, Karen
53, Yoshi];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- Name
- Age
- Agegroup

Tabella dei risultati

| Name  | Age | Agegroup       |
|-------|-----|----------------|
| John  | 25  | 20 <= age < 30 |
| Karen | 42  | 40 <= age < 50 |
| Yoshi | 53  | 50 <= age < 60 |

Il nuovo campo Agegroup classifica ogni persona per categoria di età.

### Esempio: scenario per l'applicazione della funzione class

Espressione del grafico

#### Panoramica

Viene caricata una tabella dati che contiene i dati dei clienti. Si desidera classificare i clienti in base al numero di punti valore ottenuti e fornire un conteggio del numero di clienti in ciascuna categoria.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata Example.
- I seguenti campi nella tabella dati:
  - CustomerID
  - CustomerName
  - LifetimeValue

#### Script di caricamento

Example:

```
Load * Inline [  
CustomerID, CustomerName, LifetimeValue  
1, John Doe, 12500  
2, Jane Smith, 36000  
3, Sam Brown, 15000  
4, Lisa White, 50000  
5, Tom Harris, 22000  
];
```

### Risultati

1. Caricare i dati e aprire un foglio. Creare due nuove tabelle.
2. Nella prima tabella, fare quanto segue:
  - a. Aggiungere le seguenti dimensioni:
    - CustomerID
    - CustomerName
    - LifetimeValue
  - b. Creare la seguente dimensione calcolata:
    - =Class(LifetimeValue, 10000, 'Lifetime Value Group'), per classificare i dati in base al campo LifetimeValue.
3. Nella seconda tabella, fare quanto segue:
  - a. Aggiungere questa espressione come dimensione.
    - =Class(LifetimeValue, 10000, 'Lifetime Value Group')
  - b. Creare la seguente misura:
    - =Count(CustomerID), per calcolare il numero di persone classificate in base all'età.

La prima tabella mostra i dati non elaborati che sono stati inseriti e i risultati dell'utilizzo della funzione di classe.

Tabella dei risultati 1

| CustomerID | CustomerName | LifetimeValue | Class(LifetimeValue, 10000, 'Lifetime Value Group') |
|------------|--------------|---------------|-----------------------------------------------------|
| 1          | John Doe     | 12500         | 10000 <= Lifetime Value Group < 20000               |
| 2          | Jane Smith   | 36000         | 30000 <= Lifetime Value Group < 40000               |
| 3          | Sam Brown    | 15000         | 10000 <= Lifetime Value Group < 20000               |
| 4          | Lisa White   | 50000         | 50000 <= Lifetime Value Group < 60000               |
| 5          | Tom Harris   | 22000         | 20000 <= Lifetime Value Group < 30000               |

La seconda tabella mostra come i dati ora possono essere classificati e riassunti. I dati vengono categorizzati in quattro gruppi utilizzando la funzione class, quindi vengono conteggiati per mostrare il numero di clienti in ciascun gruppo.

Tabella dei risultati 2

| Class(LifetimeValue, 10000, 'Lifetime Value Group') | Count(CustomerID) |
|-----------------------------------------------------|-------------------|
| Totals                                              | 5                 |
| 10000 <= Lifetime Value Group < 20000               | 2                 |
| 20000 <= Lifetime Value Group < 30000               | 1                 |
| 30000 <= Lifetime Value Group < 40000               | 1                 |
| 50000 <= Lifetime Value Group < 60000               | 1                 |

## coalesce

La funzione **coalesce** restituisce il primo dei parametri che presenta una rappresentazione non-NULL valida. Può essere utilizzato un numero qualsiasi di parametri.

### Sintassi:

```
coalesce(expr1[ , expr2 , expr3 , ...])
```

Tipo di dati restituiti: duale

### Argomenti

| Argomento    | Descrizione                                                                     |
|--------------|---------------------------------------------------------------------------------|
| <b>expr1</b> | La prima espressione da controllare con una rappresentazione non-NULL valida.   |
| <b>expr2</b> | La seconda espressione da controllare con una rappresentazione non-NULL valida. |
| <b>expr3</b> | La terza espressione da controllare con una rappresentazione non-NULL valida.   |

### Esempio: espressioni del grafico

| Esempio                                                                            | Risultato                                                                                                                                                                                                                                                                                                                                          |
|------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Coalesce(ProductDescription, ProductName, ProductCode, 'no description available') | Questa espressione selezionerà tra tre diversi campi con la descrizione dei prodotti, utili quando alcuni campi non dispongono di valori per il prodotto. Seguendo l'ordine elencato nella funzione, viene restituito il primo campo con un valore non nullo. Se nessuno dei campi contiene un valore, il risultato sarà no description available. |

| Esempio                                                        | Risultato                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|----------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>Coalesce(TextBetween(FileName, '', ''), FileName)</code> | Questa espressione rimuoverà potenziali virgolette di chiusura dal campo <code>FileName</code> . Se il valore <code>FileName</code> fornito appare tra virgolette, queste verranno rimosse e verrà restituito il valore <code>FileName</code> senza virgolette. Se la funzione <code>TextBetween</code> non trova i delimitatori restituisce null, che <b>Coalesce</b> rifiuta, restituendo invece il valore <code>FileName</code> non elaborato. |

### Esempio: principi fondamentali della funzione Coalesce

Espressione del grafico

#### Panoramica

Viene caricata una tabella dati che contiene i dati su prodotti e prezzi. Bisogna assicurarsi che qualsiasi valore null venga sostituito con un valore alternativo.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `ProductID`
  - `ProductName`
  - `Price`
- Un'istruzione di variabile `NullInterpret` consente di garantire che qualsiasi valore mancante sia trattato come `NULL`.

#### Script di caricamento

```
Set NullInterpret = '';
Example:
LOAD * INLINE [
ProductID, ProductName, Price
1, Laptop, 1000
2, , 800
3, Desktop, 1200
4, , 500
```

```
5, Tablet, 600  
];
```

### Risultati

1. Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:
  - ProductID
  - ProductName
2. Creare la seguente dimensione calcolata:
  - =Coalesce(ProductName, 'Unknown Product')
3. Creare la seguente misura:
  - =Sum(Price), per restituire una rappresentazione numerica valida.
4. Creare una nuova tabella e aggiungere la seguente espressione come dimensione calcolata:
  - =Coalesce(ProductName, 'Unknown Product')
5. Creare la seguente misura:
  - =Sum(Price), per restituire una rappresentazione numerica valida.

I risultati nelle tabelle seguenti mostrano come la funzione `coalesce` può sostituire i valori null nel set di dati originale con il valore alternativo `Unknown Product`.

Tabella dei risultati 1

| ProductID | ProductName | Coalesce(ProductName, 'Unknown Product') | Sum(Price) |
|-----------|-------------|------------------------------------------|------------|
| Totals    |             |                                          | 4100       |
| 1         | Laptop      | Laptop                                   | 1000       |
| 2         | -           | Unknown Product                          | 800        |
| 3         | Desktop     | Desktop                                  | 1200       |
| 4         | -           | Unknown Product                          | 500        |
| 5         | Tablet      | Tablet                                   | 600        |

Tabella dei risultati 2

| Coalesce(ProductName, 'Unknown Product') | Sum(Price) |
|------------------------------------------|------------|
| Totals                                   | 5          |
| Desktop                                  | 600        |
| Laptop                                   | 1000       |
| Tablet                                   | 600        |
| Unknown Product                          | 1300       |

### Esempio: sostituzione dei valori null con un valore alternativo prioritario

Espressione del grafico

#### Panoramica

Questo esempio utilizza la funzione `coalesce` per sostituire i dati di posizione mancanti con un valore alternativo in base alla logica di priorità specificata.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `LocationID`
  - `Country`
  - `State`
  - `City`
- Un'istruzione di variabile `NullInterpret` consente di garantire che qualsiasi valore mancante sia trattato come `NULL`.

#### Script di caricamento

```
Set NullInterpret = '';
Example:
LOAD * INLINE [
LocationID, Country, State, City
L001, USA, California, Los Angeles
L002, USA, , 
L003, Canada, Ontario, 
L004, , New South Wales, Sydney
L005, UK, , London
L006, , , London
];
```

#### Risultati

1. Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:
  - `LocationID`
  - `Country`
  - `State`
  - `City`
2. Creare la seguente dimensione calcolata:
  - `=Coalesce(Country, State, City)`, per riempire i valori null con valori alternativi.

Tabella dei risultati

| LocationID | Country | State           | City        | Coalesce(Country, State, City) |
|------------|---------|-----------------|-------------|--------------------------------|
| L001       | USA     | California      | Los Angeles | USA                            |
| L002       | USA     | -               | -           | USA                            |
| L003       | Canada  | Ontario         | -           | Canada                         |
| L004       | -       | New South Wales | Sydney      | New South Wales                |
| L005       | UK      | -               | London      | UK                             |
| L006       | -       | -               | London      | London                         |

I risultati mostrano che, sebbene vi siano dei valori mancanti, è possibile utilizzare la funzione `coalesce` per fornire una logica su come popolare queste istanze con un valore alternativo per completare il set di dati. Ad esempio, in L004 manca un valore per `country`. La funzione `coalesce` restituisce `New South Wales` perché `state` è il secondo campo elencato nella logica di priorità da utilizzare quando il valore di `country` è mancante.

## if

La funzione `if` restituisce un valore a seconda che la condizione fornita con la funzione esegua la valutazione come `True` o come `False`.

### Sintassi:

```
if(condition , then [, else])
```

**Tipo di dati restituiti:** duale

Argomenti

| Argomento        | Descrizione                                                                                                                                                                                                                                                                                                                             |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>condition</b> | Espressione interpretata in modo logico.                                                                                                                                                                                                                                                                                                |
| <b>then</b>      | Espressione che può essere di qualsiasi tipo. Se <i>condition</i> è <code>True</code> , quindi la funzione <code>if</code> restituisce il valore dell'espressione <i>then</i> .                                                                                                                                                         |
| <b>else</b>      | Espressione che può essere di qualsiasi tipo. Se <i>condition</i> è <code>False</code> , quindi la funzione <code>if</code> restituisce il valore dell'espressione <i>else</i> .<br>Questo parametro è facoltativo. Se <i>condition</i> è <code>False</code> , verrà restituito <code>NULL</code> se non si è specificato <i>else</i> . |

Esempio: espressioni dei grafico

| Esempio                                          | Risultato                                                                                                                                                                                |
|--------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>if( Amount &gt;= 0, 'OK', 'Alarm' )</code> | Questa espressione verifica se l'importo è un numero positivo (0 o superiore) e restituisce <code>OK</code> se lo è. Se l'importo è inferiore a 0, viene restituito <code>Alarm</code> . |

### Esempio - Script di caricamento usando if

#### Script di caricamento

#### Panoramica

L'istruzione If può essere utilizzata in uno script di caricamento con altri metodi e oggetti, comprese le variabili. Questo esempio utilizza la variabile `threshold` e include un nuovo campo nel modello dati basato sulla soglia.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in due tabelle dati denominate:
  - `Transactions`: include i campi delle transazioni
  - `Transaction_Buckets`: include un campo `transaction_id` e crea un nuovo campo, `[Compared to Threshold]`, utilizzando la funzione `if` per confrontare il campo `transaction_amount` con l'impostazione della variabile `threshold`.
- Un'istruzione di variabile `threshold` impostata su 100.

#### Script di caricamento

```
Transactions:
Load * Inline [
transaction_id, transaction_date, transaction_amount, transaction_quantity, customer_id, size,
color_code
3750, 20180830, 23.56, 2, 2038593, L, Red
3751, 20180907, 556.31, 6, 203521, m, orange
3752, 20180916, 5.75, 1, 5646471, S, blue
3753, 20180922, 125.00, 7, 3036491, l, Black
3754, 20180922, 484.21, 13, 049681, xs, Red
3756, 20180922, 59.18, 2, 2038593, M, Blue
3757, 20180923, 177.42, 21, 203521, XL, Black
];

set threshold = 100;

/* Create new table called Transaction_Buckets
Compare transaction_amount field from Transaction table to threshold of 100.
Output results into a new field called Compared to Threshold
*/

Transaction_Buckets:
Load
    transaction_id,
    If(transaction_amount > $(threshold), 'Greater than $(threshold)', 'Less than $(threshold)')
as [Compared to Threshold]
Resident Transactions;
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- transaction\_id
- [Compared to Threshold]

La tabella dei risultati che mostra l'output derivante dall'utilizzo della funzione *if* nello script di caricamento.

| transaction_id | In confronto alla soglia |
|----------------|--------------------------|
| 3750           | Inferiore a 100          |
| 3751           | Superiore a 100          |
| 3752           | Inferiore a 100          |
| 3753           | Superiore a 100          |
| 3754           | Superiore a 100          |
| 3756           | Inferiore a 100          |
| 3757           | Superiore a 100          |

### Esempio: valutazione dei dati dei prodotti utilizzando la funzione if

Espressione del grafico

#### Panoramica

Un set di dati contiene i dati dei prodotti. Si desidera identificare e classificare i prodotti classificandoli come di prodotti di consumo o meno.

#### Script di caricamento

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - ProductID
  - ProductName
  - ProductCategory

Example:

```
LOAD * INLINE [  
ProductID, ProductName, ProductCategory
```

```
1, "Laptop", "Electronics"  
2, "Banana", "Fruits"  
3, "Shampoo", "Toiletries"  
4, "TV", "Electronics"  
5, "Apple", "Fruits"  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- ProductID
- ProductName
- ProductCategory

Creare la seguente misura:

- =If(ProductCategory = 'Toiletries' or ProductCategory = 'Fruits', 'Consumable', 'other'), per restituire i valori richiesti.

Tabella dei risultati

| ProductID | ProductName | ProductCategory | If(ProductCategory = 'Toiletries' or ProductCategory = 'Fruits', 'Consumable', 'Other') |
|-----------|-------------|-----------------|-----------------------------------------------------------------------------------------|
| 1         | Laptop      | Electronics     | Other                                                                                   |
| 2         | Banana      | Fruits          | Consumable                                                                              |
| 3         | Shampoo     | Toiletries      | Consumable                                                                              |
| 4         | TV          | Electronics     | Other                                                                                   |
| 5         | Apple       | Fruits          | Consumable                                                                              |

Dai risultati si può vedere che la funzione `if` valuta l'espressione e restituisce i valori richiesti. Per esempio, i prodotti della categoria `Fruits` o `Toiletries` sono classificati come `Consumable`.

### Esempio: valutazione e classificazione dei dati dei clienti utilizzando la funzione `if`

Espressione del grafico

#### Panoramica

Un set di dati contiene i dati dei clienti. Si desidera identificare e classificare i clienti la cui regione e categoria aziendale soddisfano i requisiti di classificazione della propria organizzazione.

### Script di caricamento

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `ClientID`
  - `ClientName`
  - `Region`
  - `Industry`

Example:

```
LOAD * INLINE [
ClientID, ClientName, Region, Industry
1, "Acme Corp", "North", "Manufacturing"
2, "Beta Ltd", "East", "Technology"
3, "Gamma Inc", "West", "Retail"
4, "Delta LLC", "North", "Technology"
5, "Epsilon GmbH", "South", "Healthcare"
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `ClientID`
- `ClientName`
- `Region`
- `Industry`

Creare la seguente misura:

- `=If((Region = 'North' or Region = 'East') AND (Industry= 'Technology' or Industry = 'Manufacturing'), 'Target Segment', 'Secondary Segment')`, per restituire il valore richiesto.

Tabella dei risultati

| ClientID | ClientName   | Region | Industry      | If((Region = 'North' or Region = 'East') AND (Industry= 'Technology' or Industry = 'Manufacturing'), 'Target Segment', 'Secondary Segment') |
|----------|--------------|--------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| 1        | Acme Corp    | North  | Manufacturing | Target Segment                                                                                                                              |
| 2        | Beta Ltd     | East   | Technology    | Target Segment                                                                                                                              |
| 3        | Gamma Inc    | West   | Retail        | Secondary Segment                                                                                                                           |
| 4        | Delta LLC    | North  | Technology    | Target Segment                                                                                                                              |
| 5        | Epsilon GmbH | South  | Healthcare    | Secondary Segment                                                                                                                           |

Dai risultati si può vedere che la funzione `if` valuta l'espressione e restituisce i valori richiesti. Ad esempio, Acme Corp è classificato come Target Segment perché soddisfa i requisiti per far parte della regione North o East e del settore Manufacturing o Technology.

### Esempio: valutazione dei dati degli incidenti utilizzando la funzione `if`

Espressione del grafico

#### Panoramica

Utilizzare la funzione `if` per categorizzare un set di dati con i report sugli incidenti.

#### Script di caricamento

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Date`
  - `Location`
  - `Incidents`

Example:

```
LOAD * inline [
Date, Location, Incidents
1/3/2016, Beijing, 0
1/3/2016, Boston, 12
1/3/2016, Stockholm, 3
1/3/2016, Toronto, 0
1/4/2016, Beijing, 0
1/4/2016, Boston, 8];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- Data
- Località
- Incidenti

Creare le seguenti dimensioni calcolate:

- =if(Incidents>=10, 'Critical', 'Ok' )
- =if(Incidents>=10, 'Critical', If( Incidents>=1 and Incidents<10, 'Warning', 'Ok'))

Tabella dei risultati

| Date     | Location  | Incidents | if(Incidents>=10, 'Critical', 'Ok' ) | if(Incidents>=10, 'Critical', If( Incidents>=1 and Incidents<10, 'Warning', 'Ok')) |
|----------|-----------|-----------|--------------------------------------|------------------------------------------------------------------------------------|
| 1/3/2016 | Beijing   | 0         | Ok                                   | Ok                                                                                 |
| 1/3/2016 | Boston    | 12        | Critical                             | Critical                                                                           |
| 1/3/2016 | Stockholm | 3         | Ok                                   | Warning                                                                            |
| 1/3/2016 | Toronto   | 0         | Ok                                   | Ok                                                                                 |
| 1/4/2016 | Beijing   | 0         | Ok                                   | Ok                                                                                 |
| 1/4/2016 | Boston    | 8         | Ok                                   | Warning                                                                            |

### Esempio: categorizzazione delle date utilizzando la funzione if

Espressione del grafico

#### Panoramica

Utilizzare la funzione if per classificare le date.

#### Script di caricamento

Aprire Editor caricamento dati, quindi aggiungere lo script di caricamento in basso in una nuova sezione.

```
SET FirstWeekDay=0;  
Load  
Date(MakeDate(2022)+RecNo()-1) as Date  
Autogenerate 14;
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- Date

Creare le seguenti dimensioni calcolate:

- WeekDay(Date)
- =If(WeekDay(Date)>=5,'WeekEnd','Normal Day')

Tabella dei risultati

| Date      | WeekDay(Date) | If(WeekDay<br>(Date)>=5,'WeekEnd','Normal<br>Day') |
|-----------|---------------|----------------------------------------------------|
| 1/1/2022  | Sat           | WeekEnd                                            |
| 1/2/2022  | Sun           | WeekEnd                                            |
| 1/3/2022  | Mon           | Normal Day                                         |
| 1/4/2022  | Tue           | Normal Day                                         |
| 1/5/2022  | Wed           | Normal Day                                         |
| 1/6/2022  | Thu           | Normal Day                                         |
| 1/7/2022  | Fri           | Normal Day                                         |
| 1/8/2022  | Sat           | WeekEnd                                            |
| 1/9/2022  | Sun           | WeekEnd                                            |
| 1/10/2022 | Mon           | Normal Day                                         |
| 1/11/2022 | Tue           | Normal Day                                         |
| 1/12/2022 | Wed           | Normal Day                                         |
| 1/13/2022 | Thu           | Normal Day                                         |
| 1/14/2022 | Fri           | Normal Day                                         |

### match

La funzione **match** confronta il primo parametro con tutti i parametri seguenti e restituisce la posizione numerica delle espressioni corrispondenti. Il confronto rispetta la distinzione maiuscole/minuscole.

**Sintassi:**

```
match( str, expr1 [ , expr2,...exprN ] )
```

**Tipo di dati restituiti:** numero intero



*Se si desidera utilizzare il confronto senza distinzione tra lettere minuscole e maiuscole, servirsi della funzione **mixmatch**. Se si desidera utilizzare il confronto senza distinzione tra lettere minuscole e maiuscole e i caratteri speciali, servirsi della funzione **wildmatch**.*

**Argomenti**

| Argomento    | Descrizione                                                                     |
|--------------|---------------------------------------------------------------------------------|
| <b>str</b>   | La stringa da valutare.                                                         |
| <b>expr1</b> | La prima espressione da controllare per una valida rappresentazione numerica.   |
| <b>expr2</b> | La seconda espressione da controllare per una valida rappresentazione numerica. |
| <b>expr3</b> | La terza espressione da controllare per una valida rappresentazione numerica.   |

**Esempio: espressioni dei grafico**

| Esempio                                          | Risultato                                                                                                                                                                                                                                                                                                                                                           |
|--------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>match( colorCode, 'Blue', 'Black' )</code> | Questa espressione restituisce l'indice della corrispondenza per il campo <code>colorCode</code> . Se il valore di <code>colorCode</code> è <code>Blue</code> , viene restituito il valore 1. Se il valore di <code>colorCode</code> è <code>Black</code> , viene restituito il valore 2. Se non viene trovata alcuna corrispondenza, viene restituito il valore 0. |
| <code>match( Region, 'North', 'East' )</code>    | Questa espressione restituisce l'indice della corrispondenza per il campo <code>Region</code> . Se il valore di <code>Region</code> è <code>North</code> , viene restituito il valore 1. Se il valore di <code>Region</code> è <code>East</code> , viene restituito il valore 2. Se non viene trovata alcuna corrispondenza, viene restituito il valore 0.          |

**Esempio: categorizzazione dei prodotti utilizzando la funzione match**

Espressione del grafico

**Panoramica**

Un set di dati contiene i dati dei prodotti. Si desidera identificare e classificare i prodotti di consumo.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:

- ProductID
- ProductName
- ProductCategory

### Script di caricamento

Example:

```
LOAD * INLINE [  
ProductID, ProductName, ProductCategory  
1, "Laptop", "Electronics"  
2, "Banana", "Fruits"  
3, "Shampoo", "Toiletries"  
4, "TV", "Electronics"  
5, "Apple", "Fruits"  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- ProductID
- ProductName
- ProductCategory

Creare le seguenti misure:

- =match(ProductCategory, 'Toiletries', 'Fruits'), per restituire l'indice in cui ProductCategory ha il valore Toiletries o Fruits.
- =If(match(ProductCategory, 'Toiletries', 'Fruits'), 'Consumable', 'Other'), per restituire l'indice in cui ProductCategory ha il valore Toiletries o Fruits. Se viene trovato uno dei due, viene restituito il valore consumable. Altrimenti, viene restituito il valore other.

Tabella dei risultati

| ProductID | ProductName | ProductCategory | match<br>(ProductCategory, 'Toiletries', 'Fruits') | If(match<br>(ProductCategory, 'Toiletries', 'Fruits'), 'Consumable', 'Other') |
|-----------|-------------|-----------------|----------------------------------------------------|-------------------------------------------------------------------------------|
| 1         | Laptop      | Electronics     | 0                                                  | Other                                                                         |
| 2         | Banana      | Fruits          | 2                                                  | Consumable                                                                    |
| 3         | Shampoo     | Toiletries      | 1                                                  | Consumable                                                                    |
| 4         | TV          | Electronics     | 0                                                  | Other                                                                         |
| 5         | Apple       | Fruits          | 2                                                  | Consumable                                                                    |

La funzione `match` restituisce l'indice della stringa di ricerca nella prima misura. Nella seconda misura, questo risultato viene poi utilizzato in una funzione `if` per classificare i prodotti come `Consumable` o `Other`.

### Esempio: valutazione e classificazione dei dati dei clienti utilizzando la funzione `match`

Espressione del grafico

#### Panoramica

Un set di dati contiene i dati dei clienti. Si desidera identificare e classificare i clienti la cui regione e categoria aziendale soddisfano i requisiti di classificazione della propria organizzazione.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `ClientID`
  - `ClientName`
  - `Region`
  - `Industry`

#### Script di caricamento

Example:

```
LOAD * INLINE [
ClientID, ClientName, Region, Industry
1, "Acme Corp", "North", "Manufacturing"
2, "Beta Ltd", "East", "Technology"
3, "Gamma Inc", "West", "Retail"
4, "Delta LLC", "North", "Technology"
5, "Epsilon GmbH", "South", "Healthcare"
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `ClientID`
- `ClientName`
- `Region`
- `Industry`

Creare la seguente misura:

=IF(match(Region, 'North', 'East') AND match(Industry, 'Technology', 'Manufacturing'), 'Target Segment', 'Secondary Segment'), per restituire l'indice della stringa di ricerca. Se viene trovato, viene restituito il valore Target Segment. Se non viene trovato, viene restituito il valore Secondary Segment.

Tabella dei risultati

| ClientID | ClientName   | Region | Industry      | IF(match(Region, 'North', 'East') AND match(Industry, 'Technology', 'Manufacturing'), 'Target Segment', 'Secondary Segment') |
|----------|--------------|--------|---------------|------------------------------------------------------------------------------------------------------------------------------|
| 1        | Acme Corp    | North  | Manufacturing | Target Segment                                                                                                               |
| 2        | Beta Ltd     | East   | Technology    | Target Segment                                                                                                               |
| 3        | Gamma Inc    | West   | Retail        | Secondary Segment                                                                                                            |
| 4        | Delta LLC    | North  | Technology    | Target Segment                                                                                                               |
| 5        | Epsilon GmbH | South  | Healthcare    | Secondary Segment                                                                                                            |

La funzione match cerca il testo e restituisce un indice che può essere utilizzato per eseguire ulteriori analisi e classificazioni dei dati. Per esempio, il campo Acme Corp è classificato come Target Segment perché soddisfa i requisiti di ricerca della funzione match per entrambi i campi Region e Industry.

### Esempio: script di caricamento che usa match

#### Script di caricamento

#### Panoramica

È possibile utilizzare match per caricare un sottogruppo dei dati. Ad esempio, è possibile restituire un valore numerico per un'espressione nella funzione. È quindi possibile limitare i dati caricati in base al valore numerico. La funzione Match restituisce 0 se non vi sono corrispondenze. Tutte le espressioni senza corrispondenze in questo esempio restituiranno quindi 0 e saranno escluse dal caricamento dei dati dall'istruzione WHERE.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in due tabelle dati denominate:
  - Transactions: include i campi delle transazioni
  - Transaction\_Buckets: crea due nuovi campi, Customer e Color Code Blue and Black.L'istruzione where carica i risultati della funzione match.

### Script di caricamento

```
Transactions:
Load * Inline [
transaction_id, transaction_date, transaction_amount, transaction_quantity, customer_id, size,
color_code
3750, 20180830, 23.56, 2, 2038593, L, Red
3751, 20180907, 556.31, 6, 203521, m, orange
3752, 20180916, 5.75, 1, 5646471, S, blue
3753, 20180922, 125.00, 7, 3036491, l, Black
3754, 20180922, 484.21, 13, 049681, xs, Red
3756, 20180922, 59.18, 2, 2038593, M, Blue
3757, 20180923, 177.42, 21, 203521, XL, Black
];

/*
Create new table called Transaction_Buckets
Create new fields called Customer, and Color code Blue and Black
Load Transactions table.
Match returns 1 for 'Blue', 2 for 'Black'.
Does not return a value for 'blue' because match is case sensitive.
Only values that returned numeric value greater than 0
are loaded by WHERE statement into Transactions_Buckets table.
*/

Transaction_Buckets:
Load
customer_id,
customer_id as [Customer],
color_code as [Color Code Blue and Black]
Resident Transactions
where match(color_code,'Blue','Black') > 0;
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- [Color Code Blue and Black]
- Customer

La tabella dei risultati che mostra l'output  
derivante dall'utilizzo della funzione match  
nello script di caricamento

| Color Code Blue and Black | Customer |
|---------------------------|----------|
| Black                     | 203521   |
| Black                     | 3036491  |
| Blue                      | 2038593  |

### Esempio: espressione del grafico che usa match

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata Example.
- I seguenti campi nella tabella dati:
  - Cities
  - Count

#### Script di caricamento

Example:

```
Load * inline [  
Cities, Count  
Toronto, 123  
Toronto, 234  
Toronto, 231  
Boston, 32  
Boston, 23  
Boston, 1341  
Beijing, 234  
Beijing, 45  
Beijing, 235  
Stockholm, 938  
Stockholm, 39  
Stockholm, 189  
zurich, 2342  
zurich, 9033  
zurich, 0039];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- Cities

Creare le seguenti misure:

- `=match( Cities, 'Toronto', 'Boston', 'Beijing', 'zurich'),` per restituire l'indice della stringa di ricerca.
- `=match( Cities, 'Toronto', 'Boston', 'Beijing', 'Stockholm', 'zurich'),` per restituire l'indice della stringa di ricerca.
- `=Cities & ' - ' & match ( Cities, 'Toronto', 'Boston', 'Beijing', 'Stockholm', 'zurich'),` per restituire il campo cities e l'indice della stringa di ricerca.

La prima misura nella tabella sottostante restituisce 0 per stockholm perché il valore stockholm non è incluso nell'elenco di espressioni nella funzione match. Restituisce anche il valore 0 per zurich perché il confronto con match distingue tra maiuscole e minuscole.

Tabella dei risultati

| Cities    | match(Cities,'Toronto','Boston','Beijing','Zurich') | match(Cities,'Toronto','Boston','Beijing','Stockholm','zurich') | Cities & ' - ' & match ( Cities, 'Toronto','Boston','Beijing','Stockholm','zurich') |
|-----------|-----------------------------------------------------|-----------------------------------------------------------------|-------------------------------------------------------------------------------------|
| Beijing   | 3                                                   | 3                                                               | Beijing - 3                                                                         |
| Boston    | 2                                                   | 2                                                               | Boston - 2                                                                          |
| Stockholm | 0                                                   | 4                                                               | Stockholm - 4                                                                       |
| Toronto   | 1                                                   | 1                                                               | Toronto - 1                                                                         |
| zurich    | 0                                                   | 5                                                               | zurich - 5                                                                          |

### Esempio: espressione di ordinamento personalizzata che usa match

Espressione del grafico

#### Panoramica

È possibile utilizzare match per eseguire un ordinamento personalizzato per un'espressione. Questo esempio utilizza lo stesso set di dati dell'esempio precedente.

Per impostazione predefinita, le colonne vengono ordinate numericamente o alfabeticamente a seconda dei dati.

Tabella dei risultati che mostra l'ordinamento predefinito

| Cities    |
|-----------|
| Beijing   |
| Boston    |
| Stockholm |
| Toronto   |
| zurich    |

Per cambiare l'ordinamento, fare quanto segue:

1. Aprire la sezione **Ordinamento** del grafico nel pannello **Proprietà**.
2. Disattivare l'ordinamento automatico per la colonna su cui si desidera eseguire un ordinamento personalizzato.

3. Deselezionare le caselle di controllo **Ordina per numero** e **Ordina per lettera**.
4. Selezionare **Ordina per espressione** e quindi inserire un'espressione simile alla seguente:  
`=match( Cities, 'Toronto','Boston','Beijing','Stockholm','zurich')`  
 L'ordinamento della colonna Cities verrà modificato.

Tabella dei risultati che mostra l'ordinamento usando la funzione *match*

| Cities    |
|-----------|
| Toronto   |
| Boston    |
| Beijing   |
| Stockholm |
| zurich    |

## mixmatch

La funzione **mixmatch** confronta il primo parametro con tutti i parametri seguenti e restituisce la posizione numerica delle espressioni corrispondenti. Il confronto non distingue tra caratteri maiuscoli o minuscoli né i sistemi di caratteri del giapponese hiragana e katakana.

**Sintassi:**

```
mixmatch( str, expr1 [ , expr2,...exprN ] )
```

**Tipo di dati restituiti:** numero intero

Se si desidera utilizzare il confronto con distinzione tra lettere minuscole e maiuscole, servirsi della funzione **match**. Se si desidera utilizzare il confronto senza distinzione tra lettere minuscole e maiuscole e i caratteri speciali, servirsi della funzione **wildmatch**.

Argomenti

| Argomento    | Descrizione                                                                     |
|--------------|---------------------------------------------------------------------------------|
| <b>str</b>   | La stringa da valutare.                                                         |
| <b>expr1</b> | La prima espressione da controllare per una valida rappresentazione numerica.   |
| <b>expr2</b> | La seconda espressione da controllare per una valida rappresentazione numerica. |
| <b>expr3</b> | La terza espressione da controllare per una valida rappresentazione numerica.   |

Esempio: espressioni del grafico

| Esempio                                             | Risultato                                                                                                                                                            |
|-----------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>mixmatch( ColorCode, 'black', 'Blue' )</code> | Questa espressione restituisce 1 se il valore di <code>ColorCode</code> è <code>black</code> , oppure 2 se il valore di <code>ColorCode</code> è <code>Blue</code> . |

| Esempio                                                                  | Risultato                                                                                                       |
|--------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| <code>mixmatch(Cities, 'toronto', 'Boston', 'Beijing', 'Zurich' )</code> | Questa espressione restituisce l'indice dell'elemento che corrisponde al valore nel campo <code>Cities</code> . |

### Esempio: categorizzazione dei prodotti utilizzando la funzione `mixmatch`

Espressione del grafico

#### Panoramica

Un set di dati contiene i dati dei prodotti. Si desidera identificare e classificare i prodotti di consumo.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `ProductID`
  - `ProductName`
  - `ProductCategory`

#### Script di caricamento

Example:

```
LOAD * INLINE [
ProductID, ProductName, ProductCategory
1, "Laptop", "Electronics"
2, "Banana", "Fruits"
3, "Shampoo", "toiletries"
4, "TV", "Electronics"
5, "Apple", "fruits"
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `ProductID`
- `ProductName`
- `ProductCategory`

Creare le seguenti misure:

- `=mixmatch(ProductCategory, 'Toiletries', 'Fruits')`, per restituire l'indice in cui `ProductCategory` ha il valore `Toiletries` o `Fruits`.

- `=If(mismatch(ProductCategory, 'Toiletries', 'Fruits'), 'Consumable', 'Other')`, per tradurre una corrispondenza `ProductCategory` con `Toiletries` o `Fruits` come `Consumable`. Se non viene trovata alcuna corrispondenza, viene restituito il valore `Other`.



La funzione `mismatch` restituisce un valore positivo non nullo se viene trovata una corrispondenza, altrimenti restituisce zero se non viene trovata alcuna corrispondenza. L'istruzione `if` interpreta a sua volta i risultati non nulli come `true` e i risultati nulli come `false`. Combinando un'istruzione `if` con le funzioni `mismatch`, `wildmatch` o `match`, è possibile decodificare facilmente le corrispondenze e le mancate corrispondenze.

Tabella dei risultati

| ProductID | ProductName | ProductCategory | mismatch<br>(ProductCategory, 'Toiletries', 'Fruits') | If(mismatch<br>(ProductCategory, 'Toiletries', 'Fruits'), 'Consumable', 'Other') |
|-----------|-------------|-----------------|-------------------------------------------------------|----------------------------------------------------------------------------------|
| 1         | Laptop      | Electronics     | 0                                                     | Other                                                                            |
| 2         | Banana      | Fruits          | 2                                                     | Consumable                                                                       |
| 3         | Shampoo     | toiletries      | 1                                                     | Consumable                                                                       |
| 4         | TV          | Electronics     | 0                                                     | Other                                                                            |
| 5         | Apple       | fruits          | 2                                                     | Consumable                                                                       |

La funzione `mismatch` restituisce l'indice della stringa di ricerca nella prima misura. Nella seconda misura, questo risultato viene poi utilizzato in una funzione `if` per fornire risultati personalizzati. Notare che la funzione `mismatch` non fa distinzione tra maiuscole e minuscole.

### Esempio: valutazione e classificazione dei dati dei clienti utilizzando la funzione `mismatch`

Espressione del grafico

#### Panoramica

Un set di dati contiene i dati dei clienti. Si desidera identificare e classificare i clienti la cui regione e categoria aziendale soddisfano i requisiti di classificazione della propria organizzazione.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:

- ClientID
- ClientName
- Region
- Industry

### Script di caricamento

Example:

```
LOAD * INLINE [  
ClientID, ClientName, Region, Industry  
1, "Acme Corp", "north", "manufacturing"  
2, "Beta Ltd", "East", "Technology"  
3, "Gamma Inc", "west", "Retail"  
4, "Delta LLC", "North", "Technology"  
5, "Epsilon GmbH", "South", "healthcare"  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- ClientID
- ClientName
- Region
- Industry

Creare la seguente misura:

=If(mixmatch(Region, 'North', 'East') AND mixmatch(Industry, 'Technology', 'Manufacturing'), 'Target segment', 'Secondary segment'), per restituire l'indice della stringa di ricerca. Se viene trovato, viene restituito il valore Target segment. Se non viene trovato, viene restituito il valore secondary segment.

Tabella dei risultati

| ClientID | ClientName | Region | Industry      | If(mixmatch<br>(Region, 'North',<br>'East') AND<br>mixmatch<br>(Industry,<br>'Technology',<br>'Manufacturing'),<br>'Target Segment',<br>'Secondary<br>Segment') |
|----------|------------|--------|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1        | Acme Corp  | north  | manufacturing | Target Segment                                                                                                                                                  |
| 2        | Beta Ltd   | East   | Technology    | Target Segment                                                                                                                                                  |

| ClientID | ClientName   | Region | Industry   | If(mismatch<br>(Region, 'North',<br>'East') AND<br>mismatch<br>(Industry,<br>'Technology',<br>'Manufacturing'),<br>'Target Segment',<br>'Secondary<br>Segment') |
|----------|--------------|--------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3        | Gamma Inc    | west   | Retail     | Secondary<br>Segment                                                                                                                                            |
| 4        | Delta LLC    | North  | Technology | Target Segment                                                                                                                                                  |
| 5        | Epsilon GmbH | South  | healthcare | Secondary<br>Segment                                                                                                                                            |

La funzione `mismatch` cerca il testo e restituisce un indice che può essere utilizzato per eseguire ulteriori analisi e classificazioni dei dati. Per esempio, il campo `Acme Corp` è classificato come `Target Segment` perché soddisfa i requisiti di ricerca della funzione `mismatch` per entrambi i campi `Region` e `Industry`.  
 Notare che la funzione `mismatch` non fa distinzione tra maiuscole e minuscole.

## Esempio - Script di caricamento usando mismatch

Script di caricamento

### Panoramica

È possibile utilizzare `mismatch` per caricare un sottogruppo dei dati. Ad esempio, è possibile restituire un valore numerico per un'espressione nella funzione. È quindi possibile limitare i dati caricati in base al valore numerico. `Mismatch` restituisce 0 se non vi sono corrispondenze. Tutte le espressioni senza corrispondenze in questo esempio restituiranno quindi 0 e saranno escluse dal caricamento dei dati dall'istruzione `WHERE`.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in due tabelle dati denominate:
  - `Transactions`: include i campi delle transazioni
  - `Transaction_Buckets`: crea due nuovi campi, `Customer` e `Color Code` - `Black`, `Blue`, `blue`.  
L'istruzione `where` carica i risultati della funzione `mismatch`.

### Script di caricamento

```
Transactions:
Load * Inline [
transaction_id, transaction_date, transaction_amount, transaction_quantity, customer_id, size,
```

```
color_code
3750, 20180830, 23.56, 2, 2038593, L, Red
3751, 20180907, 556.31, 6, 203521, m, orange
3752, 20180916, 5.75, 1, 5646471, S, blue
3753, 20180922, 125.00, 7, 3036491, l, Black
3754, 20180922, 484.21, 13, 049681, xs, Red
3756, 20180922, 59.18, 2, 2038593, M, Blue
3757, 20180923, 177.42, 21, 203521, xL, Black
];

/*
Create new table called Transaction_Buckets
Create new fields called Customer, and Color code - Black, Blue, blue
Load Transactions table.
Mixmatch returns 1 for 'Black', 2 for 'Blue'.
Also returns 3 for 'blue' because mixmatch is not case sensitive.
Only values that returned numeric value greater than 0
are loaded by WHERE statement into Transactions_Buckets table.
*/

Transaction_Buckets:
Load
    customer_id,
    customer_id as [Customer],
    color_code as [Color Code - Black, Blue, blue]
Resident Transactions
Where mixmatch(color_code,'Black','Blue') > 0;
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- [Color Code - Black, Blue, blue]
- Customer

La tabella dei risultati che mostra l'output  
derivante dall'utilizzo della funzione mixmatch  
nello script di caricamento.

| Color Code - Black, Blue, blue | Customer |
|--------------------------------|----------|
| Black                          | 203521   |
| Black                          | 3036491  |
| Blue                           | 2038593  |
| blue                           | 5646471  |

### Esempio: espressione del grafico che usa mixmatch

#### Espressione del grafico

##### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Cities`
  - `Count`

##### Script di caricamento

Example:

```
Load * inline [Cities, Count
Toronto, 123
Toronto, 234
Toronto, 231
Boston, 32
Boston, 23
Boston, 1341
Beijing, 234
Beijing, 45
Beijing, 235
Stockholm, 938
Stockholm, 39
Stockholm, 189
zurich, 2342
zurich, 9033
zurich, 0039];
```

##### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `Cities`

Creare le seguenti misure:

- `=mixmatch( Cities, 'Toronto', 'Boston', 'Beijing', 'Zurich')`, per restituire l'indice della stringa di ricerca.
- `=mixmatch( Cities, 'Toronto', 'Boston', 'Beijing', 'Stockholm', 'Zurich')`, per restituire l'indice della stringa di ricerca.
- `=Cities & ' - ' & mixmatch ( Cities, 'Toronto', 'Boston', 'Beijing', 'Stockholm', 'Zurich')`, per restituire il campo `Cities` e l'indice della stringa di ricerca.

La prima misura nella tabella sottostante restituisce 0 per stockholm perché il valore stockholm non è incluso nell'elenco di espressioni nella funzione `mixmatch`. Restituisce anche il valore 4 per zurich perché il confronto con `mixmatch` non fa distinzione tra maiuscole e minuscole.

Tabella dei risultati

| Cities    | <code>mixmatch(Cities, 'Toronto', 'Boston', 'Beijing', 'Zurich')</code> | <code>mixmatch(Cities, 'Toronto', 'Boston', 'Beijing', 'Stockholm', 'Zurich')</code> | Cities & ' - ' & <code>mixmatch(Cities, 'Toronto', 'Boston', 'Beijing', 'Stockholm', 'Zurich')</code> |
|-----------|-------------------------------------------------------------------------|--------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|
| Beijing   | 3                                                                       | 3                                                                                    | Beijing - 3                                                                                           |
| Boston    | 2                                                                       | 2                                                                                    | Boston - 2                                                                                            |
| Stockholm | 0                                                                       | 4                                                                                    | Stockholm - 4                                                                                         |
| Toronto   | 1                                                                       | 1                                                                                    | Toronto - 1                                                                                           |
| zurich    | 4                                                                       | 5                                                                                    | zurich - 5                                                                                            |

### Esempio: espressione di ordinamento personalizzata che usa `mixmatch`

Espressione del grafico

#### Panoramica

È possibile utilizzare `mixmatch` per eseguire un ordinamento personalizzato per un'espressione. Questo esempio utilizza lo stesso set di dati dell'esempio precedente.

Per impostazione predefinita, le colonne vengono ordinate numericamente o alfabeticamente a seconda dei dati.

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- Cities

Tabella dei risultati che mostra l'ordinamento predefinito

| Cities    |
|-----------|
| Beijing   |
| Boston    |
| Stockholm |
| Toronto   |
| zurich    |

Per cambiare l'ordinamento, fare quanto segue:

1. Aprire la sezione **Ordinamento** del grafico nel pannello **Proprietà**.
2. Disattivare l'ordinamento automatico per la colonna su cui si desidera eseguire un ordinamento personalizzato.
3. Deselezionare le caselle di controllo **Ordina per numero** e **Ordina per lettera**.
4. Selezionare **Ordina per espressione** e quindi inserire un'espressione simile alla seguente:  
`=mixmatch( Cities, 'Toronto','Boston','Beijing','Stockholm','Zurich')`  
L'ordinamento della colonna Cities verrà modificato.

Tabella dei risultati che mostra l'ordinamento usando la funzione *mixmatch*

| Cities    |
|-----------|
| Toronto   |
| Boston    |
| Beijing   |
| Stockholm |
| zurich    |

### pick

La funzione pick restituisce l'espressione numero *n* nell'elenco.

**Sintassi:**

```
pick(n, expr1[ , expr2,...exprN])
```

**Tipo di dati restituiti:** duale

Argomenti

| Argomento | Descrizione                                     |
|-----------|-------------------------------------------------|
| <b>n</b>  | <i>n</i> è un numero intero compreso tra 1 e N. |

Esempio: espressione del grafico

| Esempio                              | Risultato                                        |
|--------------------------------------|--------------------------------------------------|
| <code>pick( N, 'A','B',4, 6 )</code> | restituisce B se N = 2<br>restituisce 4 se N = 3 |

### Esempio: assegnazione di etichette ai prodotti utilizzando pick

Espressione del grafico

#### Panoramica

Un set di dati contiene i dati dei prodotti. Si desidera assegnare le etichette di categoria ai prodotti in base al codice di categoria.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `ProductID`
  - `ProductName`
  - `CategoryCode`

#### Script di caricamento

Example:

```
LOAD * INLINE [  
ProductID, ProductName, CategoryCode  
1, "Laptop", 1  
2, "Headphones", 2  
3, "Printer", 3  
4, "Smartphone", 1  
5, "Tablet", 1  
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `ProductID`
- `ProductName`
- `CategoryCode`

Creare la seguente misura:

- `=Pick(CategoryCode, 'Electronics', 'Accessories', 'Office Equipment')`, per restituire un'espressione valida dall'elenco.

Tabella dei risultati

| ProductID | ProductName | CategoryCode | Pick<br>(CategoryCode,<br>'Electronics',<br>'Accessories',<br>'Office<br>Equipment') |
|-----------|-------------|--------------|--------------------------------------------------------------------------------------|
| 1         | Laptop      | 1            | Electronics                                                                          |
| 2         | Headphone   | 2            | Accessories                                                                          |
| 3         | Printer     | 3            | Office Equipment                                                                     |
| 4         | Smartphone  | 1            | Electronics                                                                          |
| 5         | Tablet      | 1            | Electronics                                                                          |

La funzione `pick` restituisce un elenco di valori associati all'indice corrispondente dal campo numerico `categoryCode`. Ad esempio, l'etichetta `Electronics` è il primo parametro della funzione e viene restituita per tutti i prodotti con `categoryCode` uguale a 1.

### Esempio: assegnazione delle etichette delle regioni utilizzando `pick`

Espressione del grafico

#### Panoramica

Un set di dati contiene codici per Paesi e regioni. Si desidera assegnare le etichette delle regioni ai Paesi in base al codice regionale.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Country`
  - `RegionCode`

#### Script di caricamento

Example:

```
LOAD * INLINE [  
Country, RegionCode  
"USA", 1  
"Canada", 1  
"Germany", 2  
"Japan", 3  
"Australia", 4  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- Country
- RegionCode

Creare la seguente misura:

- `=Pick(RegionCode, 'North America', 'Europe', 'Asia', 'Australia')`, per restituire un elemento valido dall'elenco.

Tabella dei risultati

| Country   | RegionCode |   | Pick<br>(RegionCode,<br>'North America',<br>'Europe', 'Asia',<br>'Australia') |
|-----------|------------|---|-------------------------------------------------------------------------------|
| Australia |            | 4 | Australia                                                                     |
| Canada    |            | 1 | North America                                                                 |
| Germany   |            | 2 | Europe                                                                        |
| Japan     |            | 3 | Asia                                                                          |
| USA       |            | 1 | North America                                                                 |

La funzione `pick` restituisce un elenco di valori associati all'indice corrispondente dal campo numerico `RegionCode`. Ad esempio, l'etichetta `North America` è il primo parametro della funzione e viene restituita per tutte le regioni con `RegionCode` uguale a 1.

### Esempio: combinazione di `pick` e `match` per tradurre elenchi di corrispondenze di testo alternative

Espressione del grafico

#### Panoramica

Questo scenario avanzato descrive come utilizzare efficacemente una combinazione delle funzioni `pick` e `match` per mappare le corrispondenze di testo in linea. In termini di programmazione, ha uno scopo simile a quello di un'istruzione `case`. Lo scenario esplora questa funzionalità mappando i nomi dei Paesi in nomi equivalenti alle regioni, fornendo anche un valore predefinito `Region` per qualsiasi nome precedentemente non mappato `Country`.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Countries`.
  - La tabella dati ha un campo denominato `country`.

### Script di caricamento

```
Countries:
LOAD * INLINE [
Country
"USA"
"Canada"
"Germany"
"Japan"
"Australia"
"South Africa"
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `Country`
- `RegionCode`

Creare la seguente misura:

- `=Pick(1+match(Country,'USA','Canada','Germany','Japan','Australia'), 'Unmapped', 'North America', 'North America', 'Europe', 'Asia', 'Australia')`

La funzione `match` restituirà la corrispondenza dell'indice delle alternative fornite separate da virgole, a partire da 1 = 'USA', 2 = 'Canada' e così via. Se non viene trovata alcuna corrispondenza, viene restituito il valore zero. Questo risultato viene poi incrementato di 1 e inserito nella funzione `pick`, in modo che 1 = default value se non viene trovata alcuna corrispondenza, 2 = 'USA', 3 = 'Canada' e così via.

Questa combinazione di chiamate di funzione `match` e `Pick` decodifica i nomi `country` corrispondenti in un nome `region` equivalente. Qualsiasi valore non mappato per `country` restituirà il valore predefinito per `Region`, `Unmapped`.

Tabella dei risultati

| Country      | Pick(1+match(Country,'USA','Canada','Germany','Japan','Australia'), 'Unmapped', 'North America', 'North America', 'Europe', 'Asia', 'Australia') |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| Australia    | Australia                                                                                                                                        |
| Canada       | North America                                                                                                                                    |
| Germany      | Europe                                                                                                                                           |
| Japan        | Asia                                                                                                                                             |
| South Africa | Unmapped                                                                                                                                         |
| USA          | North America                                                                                                                                    |

## wildmatch

La funzione **wildmatch** confronta il primo parametro con quelli seguenti e restituisce il numero di espressioni corrispondenti. Consente l'utilizzo di caratteri jolly ( \* e ? ) per il confronto delle stringhe. \* corrisponde a qualsiasi sequenza di caratteri. ? corrisponde a qualsiasi carattere singolo. Il confronto non distingue tra caratteri maiuscoli o minuscoli né i sistemi di caratteri del giapponese hiragana e katakana.

### Sintassi:

```
wildmatch( str, expr1 [ , expr2,...exprN ] )
```

**Tipo di dati restituiti:** numero intero

Se si desidera utilizzare il confronto senza caratteri speciali, usare le funzioni **match** o **mixmatch**.

### Argomenti

| Argomento    | Descrizione                                                                     |
|--------------|---------------------------------------------------------------------------------|
| <b>str</b>   | La stringa da valutare.                                                         |
| <b>expr1</b> | La prima espressione da controllare per una valida rappresentazione numerica.   |
| <b>expr2</b> | La seconda espressione da controllare per una valida rappresentazione numerica. |
| <b>expr3</b> | La terza espressione da controllare per una valida rappresentazione numerica.   |

### Esempio: espressioni dei grafico

| Esempio                                                       | Risultato                                                                                                                          |
|---------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| wildmatch( colorCode, 'black', '*Blue*' )                     | Questa espressione restituisce 1 se il valore di colorCode è black, oppure 2 se il valore di colorCode è Blue o Light Blue o blue. |
| wildmatch( Cities, 'toronto', 'Boston', 'Beijing', 'Zurich' ) | Questa espressione restituisce l'indice dell'elemento che corrisponde al valore nel campo Cities.                                  |

## Esempio: categorizzazione dei prodotti utilizzando la funzione wildmatch

Espressione del grafico

### Panoramica

Un set di dati contiene i dati dei prodotti. Si desidera identificare e classificare i prodotti di consumo.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `ProductID`
  - `ProductName`
  - `ProductCategory`

### Script di caricamento

Example:

```
LOAD * INLINE [
ProductID, ProductName, ProductCategory
1, "Laptop", "Electronics"
2, "Banana", "fruit"
3, "Shampoo", "toiletries"
4, "TV", "Electronics"
5, "Apple", "Fruits and vegetables"
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `ProductID`
- `ProductName`
- `ProductCategory`

Creare le seguenti misure:

- `=wildmatch(ProductCategory, 'Toiletries*', 'Fruit*')`, per restituire l'indice in cui `ProductCategory` ha il valore che inizia con `Toiletries` o `Fruit`.
- `=If(wildmatch(ProductCategory, 'Toiletries*', 'Fruit*'), 'Consumable', 'other')`, per restituire l'indice in cui `ProductCategory` ha il valore che inizia con `Toiletries` o `Fruits`. Se viene trovato uno dei due, viene restituito il valore `consumable`. Altrimenti, viene restituito il valore `other`.



*La funzione `wildmatch` restituisce un valore positivo non nullo se viene trovata una corrispondenza, altrimenti restituisce zero se non viene trovata alcuna corrispondenza. L'istruzione `if` interpreta a sua volta i risultati diversi da zero come `true` e zero come `false`. Combinando un'istruzione `if` con le funzioni `mismatch`, `wildmatch` o `match`, è possibile decodificare facilmente le corrispondenze e le mancate corrispondenze.*

Tabella dei risultati

| ProductID | ProductName | ProductCategory | wildmatch<br>(ProductCategory,<br>'Toiletries*', 'Fruit*') | If(wildmatch<br>(ProductCategory,<br>'Toiletries*', 'Fruit*') |
|-----------|-------------|-----------------|------------------------------------------------------------|---------------------------------------------------------------|
| 1         | Laptop      | Electronics     | 0                                                          | Other                                                         |

| ProductID | ProductName | ProductCategory          | wildmatch<br>(ProductCategory,<br>'Toiletries*', 'Fruit*') | If(wildmatch<br>(ProductCategory,<br>'Toiletries*', 'Fruit*') |
|-----------|-------------|--------------------------|------------------------------------------------------------|---------------------------------------------------------------|
| 2         | Banana      | fruit                    | 2                                                          | Consumable                                                    |
| 3         | Shampoo     | toiletries               | 1                                                          | Consumable                                                    |
| 4         | TV          | Electronics              | 0                                                          | Other                                                         |
| 5         | Apple       | Fruits and<br>vegetables | 2                                                          | Consumable                                                    |

La funzione `wildmatch` restituisce l'indice della stringa di ricerca nella prima misura. Nella seconda misura, questo risultato viene poi utilizzato in una funzione `if` per fornire risultati personalizzati. Notare che la funzione `wildmatch` non fa distinzione tra maiuscole e minuscole e permette i caratteri speciali (\* e ?).

## Esempio: valutazione e classificazione dei dati dei clienti utilizzando la funzione `wildmatch`

Espressione del grafico

### Panoramica

Un set di dati contiene i dati dei clienti. Si desidera identificare e classificare i clienti la cui regione e categoria aziendale soddisfano i requisiti di classificazione della propria organizzazione.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `ClientID`
  - `ClientName`
  - `Region`
  - `Industry`

### Script di caricamento

Example:

```
LOAD * INLINE [
ClientID, ClientName, Region, Industry
1, "Acme Corp", "north territory", "Vehicle Manufacturing"
2, "Beta Ltd", "East location", "Science & Technology"
3, "Gamma Inc", "west", "Retail Sales"
4, "Delta LLC", "North", "Technology"
5, "Epsilon GmbH", "South", "healthcare"
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- ClientID
- ClientName
- Region
- Industry

Creare la seguente misura:

=If(wildmatch(Region, 'North\*', 'East\*') AND wildmatch(Industry, '\*Technology\*', '\*Manufacturing\*'), 'Target Segment', 'Secondary Segment'), per restituire l'indice della stringa di ricerca. Se viene trovato, viene restituito il valore Target Segment. Se non viene trovato, viene restituito il valore Secondary Segment.

Tabella dei risultati

| ClientID | ClientName   | Region          | Industry              | If(wildmatch(Region, 'North*', 'East*') AND wildmatch(Industry, '*Technology*', '*Manufacturing*'), 'Target Segment', 'Secondary Segment') |
|----------|--------------|-----------------|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| 1        | Acme Corp    | north territory | Vehicle Manufacturing | Target Segment                                                                                                                             |
| 2        | Beta Ltd     | East location   | Science & Technology  | Target Segment                                                                                                                             |
| 3        | Gamma Inc    | west            | Retail Sales          | Secondary Segment                                                                                                                          |
| 4        | Delta LLC    | North           | Technology            | Target Segment                                                                                                                             |
| 5        | Epsilon GmbH | South           | healthcare            | Secondary Segment                                                                                                                          |

La funzione `wildmatch` cerca il testo e restituisce un indice che può essere utilizzato per eseguire ulteriori analisi e classificazioni dei dati. Per esempio, il campo `Acme Corp` è classificato come `Target Segment` perché soddisfa i requisiti di ricerca della funzione `wildmatch` per entrambi i campi `Region` e `Industry`. Notare che la funzione `wildmatch` non fa distinzione tra maiuscole e minuscole e permette i caratteri speciali (\* e ?).

### Esempio: script di caricamento che usa wildmatch

#### Script di caricamento

##### Panoramica

È possibile utilizzare wildmatch per caricare un sottogruppo dei dati. Ad esempio, è possibile restituire un valore numerico per un'espressione nella funzione. È quindi possibile limitare i dati caricati in base al valore numerico. Wildmatch restituisce 0 se non vi sono corrispondenze. Tutte le espressioni senza corrispondenze in questo esempio restituiranno quindi 0 e saranno escluse dal caricamento dei dati dall'istruzione WHERE.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in due tabelle dati denominate:
  - Transactions: include i campi delle transazioni
  - Transaction\_Buckets: crea due nuovi campi, Customer e Color code - Black, Blue, blue, red. L'istruzione where carica i risultati della funzione wildmatch.

##### Script di caricamento

```
Transactions:
Load * Inline [
transaction_id, transaction_date, transaction_amount, transaction_quantity, customer_id, size,
color_code
3750, 20180830, 23.56, 2, 2038593, L, Red
3751, 20180907, 556.31, 6, 203521, m, orange
3752, 20180916, 5.75, 1, 5646471, s, blue
3753, 20180922, 125.00, 7, 3036491, l, Black
3754, 20180922, 484.21, 13, 049681, xs, Red
3756, 20180922, 59.18, 2, 2038593, M, Blue
3757, 20180923, 177.42, 21, 203521, XL, Black
];

/*
Create new table called Transaction_Buckets
Create new fields called Customer, and Color code - Black, Blue, blue, red
Load Transactions table.
wildmatch returns 1 for 'Black', 'Blue', and 'blue', and 2 for 'Red'.
Only values that returned numeric value greater than 0
are loaded by WHERE statement into Transactions_Buckets table.
*/

Transaction_Buckets:
Load
    customer_id,
    customer_id as [Customer],
    color_code as [Color Code Black, Blue, blue, Red]
Resident Transactions
where wildmatch(color_code, 'Bl*', 'R??') > 0;
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- [Color code - Black, Blue, blue, red]
- Customer

La tabella dei risultati che mostra l'output derivante dall'utilizzo della funzione wildmatch nello script di caricamento.

| Color code - Black, Blue, blue, red | Customer |
|-------------------------------------|----------|
| Black                               | 203521   |
| Black                               | 3036491  |
| Blue                                | 2038593  |
| blue                                | 5646471  |
| Red                                 | 049681   |
| Red                                 | 2038593  |

### Esempio: espressione del grafico che usa wildmatch

Espressione del grafico

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata Example.
- I seguenti campi nella tabella dati:
  - Cities
  - Count

### Script di caricamento

Example:

```
Load * inline [Cities, Count
Toronto, 123
Toronto, 234
Toronto, 231
Boston, 32
Boston, 23
Boston, 1341
Beijing, 234
Beijing, 45
```

```
Beijing, 235
Stockholm, 938
Stockholm, 39
Stockholm, 189
zurich, 2342
zurich, 9033
zurich, 0039];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- Cities

Creare le seguenti misure:

- =wildmatch( Cities,'Tor\*','?ton','Beijing','\*urich') , per restituire l'indice della stringa di ricerca.
- =wildmatch( Cities,'Tor\*','???ton','Beijing','Stockholm','\*urich'), per restituire l'indice della stringa di ricerca.
- =Cities & ' - ' & wildmatch ( Cities, 'Tor\*','???ton','Beijing','Stockholm','\*urich'), per restituire il campo cities e l'indice della stringa di ricerca.

La prima misura nella tabella sottostante restituisce 0 per stockholm perché il valore stockholm non è incluso nell'elenco di espressioni nella funzione wildmatch. Restituisce anche 0 per Boston, perché ? corrisponde solo a un singolo carattere.

Tabella dei risultati

| Cities    | wildmatch(<br>Cities,'Tor*','?ton','Beijing','*urich') | wildmatch(<br>Cities,'Tor*','???ton','Beijing','Stockholm','*urich') | Cities & ' - ' & wildmatch ( Cities,<br>'Tor*','???ton','Beijing','Stockholm','*urich') |
|-----------|--------------------------------------------------------|----------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
| Beijing   | 3                                                      | 3                                                                    | Beijing - 3                                                                             |
| Boston    | 0                                                      | 2                                                                    | Boston - 2                                                                              |
| Stockholm | 0                                                      | 4                                                                    | Stockholm - 4                                                                           |
| Toronto   | 1                                                      | 1                                                                    | Toronto - 1                                                                             |
| zurich    | 4                                                      | 5                                                                    | zurich - 5                                                                              |

## Esempio: espressione di ordinamento personalizzata che usa wildmatch

Espressione del grafico

### Panoramica

È possibile utilizzare `wildmatch` per eseguire un ordinamento personalizzato per un'espressione. Questo esempio utilizza lo stesso set di dati dell'esempio precedente.

Per impostazione predefinita, le colonne vengono ordinate numericamente o alfabeticamente, a seconda dei dati.

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `Cities`

Tabella dei risultati che mostra l'ordinamento predefinito

| Cities    |
|-----------|
| Beijing   |
| Boston    |
| Stockholm |
| Toronto   |
| zurich    |

Per cambiare l'ordinamento, fare quanto segue:

1. Aprire la sezione **Ordinamento** del grafico nel pannello **Proprietà**.
2. Disattivare l'ordinamento automatico per la colonna su cui si desidera eseguire un ordinamento personalizzato.
3. Deselezionare le caselle di controllo **Ordina per numero** e **Ordina per lettera**.
4. Selezionare **Ordina per espressione** e quindi inserire un'espressione simile alla seguente:  
`=wildmatch( Cities, 'Tor*', '???ton', 'Beijing', 'Stockholm', '*urich')`  
 L'ordinamento della colonna Cities verrà modificato.

Tabella dei risultati che mostra l'ordinamento usando la funzione *wildmatch*

| Cities    |
|-----------|
| Toronto   |
| Boston    |
| Beijing   |
| Stockholm |
| zurich    |

## 8.6 Funzioni di conteggio

In questa sezione vengono descritte le funzioni correlate ai contatori di record durante la valutazione dell'istruzione **LOAD** nello script di caricamento dei dati. L'unica funzione che è possibile utilizzare nelle espressioni grafiche è **RowNo()**.

Ad alcune funzioni di conteggio non sono associati parametri, ma sono comunque richieste le parentesi finali.

### Prospetto delle funzioni di conteggio

Ciascuna funzione viene descritta ulteriormente dopo la panoramica. È inoltre possibile fare clic sul nome della funzione nella sintassi per accedere immediatamente ai dettagli per tale funzione specifica.

#### **autonumber**

Questa funzione dello script restituisce un valore intero univoco per ciascun valore calcolato distinto di *expression* rilevato durante l'esecuzione dello script. Questa funzione può essere utilizzata per creare una rappresentazione compatta di memoria che rappresenta una chiave complessa.

```
autonumber (expression[ , AutoID])
```

#### **autonumberhash128**

Questa funzione di script calcola un hash a 128 bit dei valori di espressione di input combinati e restituisce un valore intero univoco per ciascun valore hash distinto rilevato durante l'esecuzione dello script. Questa funzione può essere, ad esempio, utilizzata per creare una rappresentazione compatta di memoria che rappresenta una chiave complessa.

```
autonumberhash128 (expression {, expression})
```

#### **autonumberhash256**

Questa funzione di script calcola un hash a 256 bit dei valori di espressione di input combinati e restituisce un valore intero univoco per ciascun valore hash distinto rilevato durante l'esecuzione dello script. Questa funzione può essere utilizzata per creare una rappresentazione compatta di memoria che rappresenta una chiave complessa.

```
autonumberhash256 (expression {, expression})
```

#### **IterNo**

Questa funzione dello script restituisce un numero intero che indica il numero di volte in cui verrà valutato un singolo record in un'istruzione **LOAD** con una clausola **while**. La prima ripetizione ha valore 1. La funzione **IterNo** è significativa solo se utilizzata in combinazione con una clausola **while**.

```
IterNo ( )
```

#### **RecNo**

Questa funzione di script restituisce un valore intero relativo al numero della riga della tabella corrente attualmente in corso di lettura. Il primo record è il numero 1.

```
RecNo ( )
```

### RowNo - script function

Questa funzione dello script restituisce un numero intero relativo alla posizione della riga attuale nella tabella interna risultante in Qlik Sense. La prima riga è il numero 1.

**RowNo** ( )

### RowNo - chart function

**RowNo()** restituisce il numero della riga attuale nel segmento di colonna attuale in una tabella. Per i grafici bitmap, **RowNo()** restituisce il numero della riga attuale nell'equivalente di tabella lineare del grafico.

**RowNo - funzione per grafici** ([TOTAL])

## autonumber

Questa funzione dello script restituisce un valore intero univoco per ciascun valore calcolato distinto di *expression* rilevato durante l'esecuzione dello script. Questa funzione può essere utilizzata per creare una rappresentazione compatta di memoria che rappresenta una chiave complessa.



È possibile connettere solo chiavi **autonumber** che sono state generate nello stesso caricamento di dati, poiché il numero intero viene generato in base all'ordine di lettura della tabella. Se si necessita di utilizzare chiavi permanenti tra i caricamenti di dati indipendentemente dall'ordinamento dei dati sorgente è necessario utilizzare le funzioni **hash128**, **hash160** o **hash256**.

### Sintassi:

**autonumber** (expression[ , AutoID])

### Argomenti:

| Argomento | Descrizione                                                                                                                                                                                                                        |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AutoID    | Per creare più istanze di conteggio qualora si utilizzi la funzione <b>autonumber</b> su chiavi differenti all'interno dello script, è possibile utilizzare un parametro opzionale <i>AutoID</i> per il nome di ciascun contatore. |

### Esempio: Creazione di una chiave composta

Nell'esempio viene creata una chiave composta tramite la funzione **autonumber** per conservare la memoria. L'esempio è breve a scopo dimostrativo, ma risulterebbe significativo con una tabella contenente un elevato numero di righe.

Dati di esempio

| Region | Year | Month | Sales |
|--------|------|-------|-------|
| North  | 2014 | May   | 245   |

| Region | Year | Month | Sales |
|--------|------|-------|-------|
| North  | 2014 | May   | 347   |
| North  | 2014 | June  | 127   |
| South  | 2014 | June  | 645   |
| South  | 2013 | May   | 367   |
| South  | 2013 | May   | 221   |

I dati sorgente vengono caricati tramite dati inline. Quindi, è necessario aggiungere un caricamento precedente, che crea una chiave composita dai campi Region, Year e Month.

```
RegionSales:
LOAD *,
AutoNumber(Region&Year&Month) as RYMkey;
```

```
LOAD * INLINE
[ Region, Year, Month, Sales
North, 2014, May, 245
North, 2014, May, 347
North, 2014, June, 127
South, 2014, June, 645
South, 2013, May, 367
South, 2013, May, 221
];
```

La tabella risultante avrà l'aspetto seguente:

Tabella dei risultati

| Region | Year | Month | Sales | RYMkey |
|--------|------|-------|-------|--------|
| North  | 2014 | May   | 245   | 1      |
| North  | 2014 | May   | 347   | 1      |
| North  | 2014 | June  | 127   | 2      |
| South  | 2014 | June  | 645   | 3      |
| South  | 2013 | May   | 367   | 4      |
| South  | 2013 | May   | 221   | 4      |

In questo esempio è possibile fare riferimento a RYMkey, per l'esempio 1, anziché alla stringa 'North2014May' se si desidera effettuare il collegamento a un'altra tabella.

Ora viene caricata una tabella sorgente dei costi in modo simile. I campi Region, Year e Month vengono esclusi nel caricamento precedente per evitare di creare una chiave sintetica. È stata già creata una chiave composita con la funzione **autonumber** tramite il collegamento delle tabelle.

```
RegionCosts:
LOAD Costs,
AutoNumber(Region&Year&Month) as RYMkey;
```

```
LOAD * INLINE
[ Region, Year, Month, Costs
South, 2013, May, 167
North, 2014, May, 56
North, 2014, June, 199
South, 2014, June, 64
South, 2013, May, 172
South, 2013, May, 126
];
```

Ora è possibile aggiungere la visualizzazione di una tabella a un foglio e aggiungere i campi Region, Year e Month così come le misure Sum per le vendite e i costi. La tabella avrà il seguente aspetto:

Tabella dei risultati

| Region | Year | Month | Sum([Sales]) | Sum([Costs]) |
|--------|------|-------|--------------|--------------|
| Totals | -    | -     | 1952         | 784          |
| North  | 2014 | June  | 127          | 199          |
| North  | 2014 | May   | 592          | 56           |
| South  | 2014 | June  | 645          | 64           |
| South  | 2013 | May   | 588          | 465          |

## autonumberhash128

Questa funzione di script calcola un hash a 128 bit dei valori di espressione di input combinati e restituisce un valore intero univoco per ciascun valore hash distinto rilevato durante l'esecuzione dello script. Questa funzione può essere, ad esempio, utilizzata per creare una rappresentazione compatta di memoria che rappresenta una chiave complessa.



*È possibile connettere solo chiavi **autonumberhash128** che sono state generate nello stesso caricamento di dati, poiché il numero intero viene generato in base all'ordine di lettura della tabella. Se si necessita di utilizzare chiavi permanenti tra i caricamenti di dati indipendentemente dall'ordinamento dei dati sorgente è necessario utilizzare le funzioni **hash128**, **hash160** o **hash256**.*

### Sintassi:

```
autonumberhash128(expression {, expression})
```

### Esempio: Creazione di una chiave composita

Nell'esempio viene creata una chiave composita tramite la funzione **autonumberhash128** per conservare la memoria. L'esempio è breve a scopo dimostrativo, ma risulterebbe significativo con una tabella contenente un elevato numero di righe.

Dati di esempio

| Region | Year | Month | Sales |
|--------|------|-------|-------|
| North  | 2014 | May   | 245   |
| North  | 2014 | May   | 347   |
| North  | 2014 | June  | 127   |
| South  | 2014 | June  | 645   |
| South  | 2013 | May   | 367   |
| South  | 2013 | May   | 221   |

I dati sorgente vengono caricati tramite dati inline. Quindi, è necessario aggiungere un caricamento precedente, che crea una chiave composita dai campi Region, Year e Month.

```
RegionSales:
LOAD *,
AutoNumberHash128(Region, Year, Month) as RYMkey;
```

```
LOAD * INLINE
[ Region, Year, Month, Sales
North, 2014, May, 245
North, 2014, May, 347
North, 2014, June, 127
South, 2014, June, 645
South, 2013, May, 367
South, 2013, May, 221
];
```

La tabella risultante avrà l'aspetto seguente:

Tabella dei risultati

| Region | Year | Month | Sales | RYMkey |
|--------|------|-------|-------|--------|
| North  | 2014 | May   | 245   | 1      |
| North  | 2014 | May   | 347   | 1      |
| North  | 2014 | June  | 127   | 2      |
| South  | 2014 | June  | 645   | 3      |
| South  | 2013 | May   | 367   | 4      |
| South  | 2013 | May   | 221   | 4      |

In questo esempio è possibile fare riferimento a RYMkey, per l'esempio 1, anziché alla stringa 'North2014May' se si desidera effettuare il collegamento a un'altra tabella.

Ora viene caricata una tabella sorgente dei costi in modo simile. I campi Region, Year e Month vengono esclusi nel caricamento precedente per evitare di creare una chiave sintetica. È stata già creata una chiave composita con la funzione **autonumberhash128** tramite il collegamento delle tabelle.

```
RegionCosts:
LOAD Costs,
AutoNumberHash128(Region, Year, Month) as RYMkey;
```

```
LOAD * INLINE
[ Region, Year, Month, Costs
South, 2013, May, 167
North, 2014, May, 56
North, 2014, June, 199
South, 2014, June, 64
South, 2013, May, 172
South, 2013, May, 126
];
```

Ora è possibile aggiungere la visualizzazione di una tabella a un foglio e aggiungere i campi Region, Year e Month così come le misure Sum per le vendite e i costi. La tabella avrà il seguente aspetto:

Tabella dei risultati

| Region | Year | Month | Sum([Sales]) | Sum([Costs]) |
|--------|------|-------|--------------|--------------|
| Totals | -    | -     | 1952         | 784          |
| North  | 2014 | June  | 127          | 199          |
| North  | 2014 | May   | 592          | 56           |
| South  | 2014 | June  | 645          | 64           |
| South  | 2013 | May   | 588          | 465          |

## autonumberhash256

Questa funzione di script calcola un hash a 256 bit dei valori di espressione di input combinati e restituisce un valore intero univoco per ciascun valore hash distinto rilevato durante l'esecuzione dello script. Questa funzione può essere utilizzata per creare una rappresentazione compatta di memoria che rappresenta una chiave complessa.



*È possibile connettere solo chiavi **autonumberhash256** che sono state generate nello stesso caricamento di dati, poiché il numero intero viene generato in base all'ordine di lettura della tabella. Se si necessita di utilizzare chiavi permanenti tra i caricamenti di dati indipendentemente dall'ordinamento dei dati sorgente è necessario utilizzare le funzioni **hash128**, **hash160** o **hash256**.*

### Sintassi:

```
autonumberhash256 (expression {, expression})
```

### Esempio: Creazione di una chiave composita

Nell'esempio viene creata una chiave composita tramite la funzione **autonumberhash256** per conservare la memoria. L'esempio è breve a scopo dimostrativo, ma risulterebbe significativo con una tabella contenente un elevato numero di righe.

Tabella di esempio

| Region | Year | Month | Sales |
|--------|------|-------|-------|
| North  | 2014 | May   | 245   |
| North  | 2014 | May   | 347   |
| North  | 2014 | June  | 127   |
| South  | 2014 | June  | 645   |
| South  | 2013 | May   | 367   |
| South  | 2013 | May   | 221   |

I dati sorgente vengono caricati tramite dati inline. Quindi, è necessario aggiungere un caricamento precedente, che crea una chiave composita dai campi Region, Year e Month.

```
RegionSales:
LOAD *,
AutoNumberHash256(Region, Year, Month) as RYMkey;
```

```
LOAD * INLINE
[ Region, Year, Month, Sales
North, 2014, May, 245
North, 2014, May, 347
North, 2014, June, 127
South, 2014, June, 645
South, 2013, May, 367
South, 2013, May, 221
];
```

La tabella risultante avrà l'aspetto seguente:

Tabella dei risultati

| Region | Year | Month | Sales | RYMkey |
|--------|------|-------|-------|--------|
| North  | 2014 | May   | 245   | 1      |
| North  | 2014 | May   | 347   | 1      |
| North  | 2014 | June  | 127   | 2      |
| South  | 2014 | June  | 645   | 3      |
| South  | 2013 | May   | 367   | 4      |
| South  | 2013 | May   | 221   | 4      |

In questo esempio è possibile fare riferimento a RYMkey, per l'esempio 1, anziché alla stringa 'North2014May' se si desidera effettuare il collegamento a un'altra tabella.

Ora viene caricata una tabella sorgente dei costi in modo simile. I campi Region, Year e Month vengono esclusi nel caricamento precedente per evitare di creare una chiave sintetica. È stata già creata una chiave composita con la funzione **autonumberhash256** tramite il collegamento delle tabelle.

```
RegionCosts:
LOAD Costs,
AutoNumberHash256(Region, Year, Month) as RYMkey;
```

```
LOAD * INLINE
[ Region, Year, Month, Costs
South, 2013, May, 167
North, 2014, May, 56
North, 2014, June, 199
South, 2014, June, 64
South, 2013, May, 172
South, 2013, May, 126
];
```

Ora è possibile aggiungere la visualizzazione di una tabella a un foglio e aggiungere i campi Region, Year e Month così come le misure Sum per le vendite e i costi. La tabella avrà il seguente aspetto:

Tabella dei risultati

| Region | Year | Month | Sum([Sales]) | Sum([Costs]) |
|--------|------|-------|--------------|--------------|
| Totals | -    | -     | 1952         | 784          |
| North  | 2014 | June  | 127          | 199          |
| North  | 2014 | May   | 592          | 56           |
| South  | 2014 | June  | 645          | 64           |
| South  | 2013 | May   | 588          | 465          |

### IterNo

Questa funzione dello script restituisce un numero intero che indica il numero di volte in cui verrà valutato un singolo record in un'istruzione **LOAD** con una clausola **while**. La prima ripetizione ha valore 1. La funzione **IterNo** è significativa solo se utilizzata in combinazione con una clausola **while**.

**Sintassi:**

```
IterNo ( )
```

Esempi e risultati:

**Esempio:**

```
LOAD
    IterNo() as Day,
    Date( StartDate + IterNo() - 1 ) as Date
    while StartDate + IterNo() - 1 <= EndDate;

LOAD * INLINE
[StartDate, EndDate
2014-01-22, 2014-01-26
];
```

L'istruzione **LOAD** genererà un record per la data compresa nella scala definita da **StartDate** e **EndDate**.

In questo esempio la tabella risultante sarà simile alla seguente:

| Tabella dei risultati |            |
|-----------------------|------------|
| Day                   | Date       |
| 1                     | 2014-01-22 |
| 2                     | 2014-01-23 |
| 3                     | 2014-01-24 |
| 4                     | 2014-01-25 |
| 5                     | 2014-01-26 |

### RecNo

Questa funzione di script restituisce un valore intero relativo al numero della riga della tabella corrente attualmente in corso di lettura. Il primo record è il numero 1.

**Sintassi:**

```
RecNo ( )
```

Diversamente da **RowNo( )**, che conta le righe nella tabella di Qlik Sense risultante, **RecNo( )** conteggia i record nella tabella dati non elaborati e viene reimpostata quando una tabella di dati non elaborati viene concatenata con un'altra.

**Esempio: Script di caricamento dei dati**

Caricamento di tabelle di dati non elaborati:

```
Tab1:
LOAD * INLINE
[A, B
1, aa
2, cc
3, ee];
```

```
Tab2:  
LOAD * INLINE  
[C, D  
5, xx  
4, yy  
6, zz];
```

Caricamento dei numeri di record e riga per le righe selezionate:

QTab:

```
LOAD *,  
  
RecNo( ),  
  
RowNo( )  
  
resident Tab1 where A<>2;
```

```
LOAD  
  
C as A,  
  
D as B,  
  
RecNo( ),  
  
RowNo( )  
  
resident Tab2 where A<>5;
```

```
//We don't need the source tables anymore, so we drop them
```

```
Drop tables Tab1, Tab2;
```

Tabella interna di Qlik Sense risultante:

Tabella dei risultati

| A | B  | RecNo( ) | RowNo( ) |
|---|----|----------|----------|
| 1 | aa | 1        | 1        |
| 3 | ee | 3        | 2        |
| 4 | yy | 2        | 3        |
| 6 | zz | 3        | 4        |

## RowNo

Questa funzione dello script restituisce un numero intero relativo alla posizione della riga attuale nella tabella interna risultante in Qlik Sense. La prima riga è il numero 1.

### Sintassi:

```
RowNo ( [TOTAL] )
```

Diversamente da **RecNo( )**, che conta i record nella tabella di dati non elaborati, la funzione **RowNo( )** non conta i record che sono esclusi da clausole **where** e non viene reimpostata quando una tabella di dati non elaborati risulta concatenata con un'altra.



*Se si utilizza un caricamento precedente, ossia più istruzioni **LOAD** in pila, lette dalla stessa tabella, è possibile utilizzare solo **RowNo( )** nell'istruzione **LOAD** di livello superiore. Se si utilizza **RowNo( )** nelle istruzioni **LOAD** successive, viene restituito 0.*

### Esempio: Script di caricamento dei dati

Caricamento di tabelle di dati non elaborati:

```
Tab1:  
LOAD * INLINE  
[A, B  
1, aa  
2, cc  
3, ee];
```

```
Tab2:  
LOAD * INLINE  
[C, D  
5, xx  
4, yy  
6, zz];
```

Caricamento dei numeri di record e riga per le righe selezionate:

```
QTab:  
  
LOAD *,  
  
RecNo( ),  
  
RowNo( )  
  
resident Tab1 where A<>2;  
  
LOAD
```

```
C as A,
```

```
D as B,
```

```
RecNo( ),
```

```
RowNo( )
```

```
resident Tab2 where A<>5;
```

```
//We don't need the source tables anymore, so we drop them
```

```
Drop tables Tab1, Tab2;
```

Tabella interna di Qlik Sense risultante:

Tabella dei risultati

| A | B  | RecNo( ) | RowNo( ) |
|---|----|----------|----------|
| 1 | aa | 1        | 1        |
| 3 | ee | 3        | 2        |
| 4 | yy | 2        | 3        |
| 6 | zz | 3        | 4        |

## RowNo - funzione per grafici

**RowNo()** restituisce il numero della riga attuale nel segmento di colonna attuale in una tabella. Per i grafici bitmap, **RowNo()** restituisce il numero della riga attuale nell'equivalente di tabella lineare del grafico.

Se la tabella o l'equivalente di tabella include più dimensioni verticali, il segmento colonna attuale includerà solo righe contenenti gli stessi valori della riga attuale in tutte le colonne di dimensione, eccetto la colonna che mostra l'ultima dimensione nell'ordinamento tra campi.

### Segmenti delle colonne

|                   | Region   | Country                  | Population  | Rank(Population) |
|-------------------|----------|--------------------------|-------------|------------------|
| Column segment #1 | Americas | Mexico                   | 128,932,753 | 2                |
|                   | Americas | Canada                   | 37,742,154  | 3                |
|                   | Americas | United States of America | 331,000,651 | 1                |
| Column segment #2 | Europe   | Sweden                   | 10,099,265  | 4                |
|                   | Europe   | United Kingdom           | 67,886,011  | 2                |
|                   | Europe   | France                   | 65,273,511  | 3                |
|                   | Europe   | Germany                  | 83,783,942  | 1                |



*L'ordinamento sui valori y nei grafici o l'ordinamento per colonne di espressione nelle tabelle non è consentito quando si utilizza questa funzione di grafico in una qualsiasi delle espressioni del grafico. Queste alternative di ordinamento verranno perciò disattivate automaticamente. Quando si utilizza questa funzione di grafico in una visualizzazione o in una tabella, l'ordinamento della visualizzazione torna all'ordinamento inserito in questa funzione.*

### Sintassi:

```
RowNo ( [ TOTAL ] )
```

**Tipo di dati restituiti:** numero intero

**Argomenti:**

| Argomento | Descrizione                                                                                                                                                        |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TOTAL     | Se la tabella è unidimensionale o se è utilizzato il qualificatore <b>TOTAL</b> come argomento, il segmento colonna attuale sarà sempre uguale all'intera colonna. |

## Esempio: espressione del grafico che utilizza RowNo

Esempio: espressione del grafico

### Script di caricamento

Caricare i dati seguenti come un caricamento inline nell'editor caricamento dati per creare gli esempi di espressione del grafico in basso.

```
Temp:
LOAD * inline [
Customer|Product|OrderNumber|UnitSales|UnitPrice
Astrida|AA|1|4|16
Astrida|AA|7|10|15
Astrida|BB|4|9|9
Betacab|CC|6|5|10
Betacab|AA|5|2|20
Betacab|BB|1|25| 25
Canutility|AA|3|8|15
Canutility|CC|5|4|19
Divadip|CC|2|4|16
Divadip|DD|3|1|25
] (delimiter is '|');
```

### Espressione del grafico

Creare la visualizzazione di una tabella in un foglio di Qlik Sense con **Customer** e **UnitSales** come dimensioni. Aggiungere rispettivamente **RowNo( )** e **RowNo(TOTAL)** come misure etichettate **Riga nel Segmento** e **Row Number**. Aggiungere la seguente espressione alla tabella come misura:

```
If( RowNo( )=1, 0, UnitSales / Above( UnitSales ))
```

### Risultato

| Customer | UnitSales | Row in Segment | Row Number | If( RowNo( )=1, 0, UnitSales / Above( UnitSales )) |
|----------|-----------|----------------|------------|----------------------------------------------------|
| Astrida  | 4         | 1              | 1          | 0                                                  |
| Astrida  | 9         | 2              | 2          | 2.25                                               |
| Astrida  | 10        | 3              | 3          | 1.11111111111111                                   |

| Customer   | UnitSales | Row in Segment | Row Number | If( RowNo( )=1, 0, UnitSales / Above( UnitSales )) |
|------------|-----------|----------------|------------|----------------------------------------------------|
| Betacab    | 2         | 1              | 4          | 0                                                  |
| Betacab    | 5         | 2              | 5          | 2.5                                                |
| Betacab    | 25        | 3              | 6          | 5                                                  |
| Canutility | 4         | 1              | 7          | 0                                                  |
| Canutility | 8         | 2              | 8          | 2                                                  |
| Divadip    | 1         | 1              | 9          | 0                                                  |
| Divadip    | 4         | 2              | 10         | 4                                                  |

### Spiegazione

La colonna **Row in Segment** restituisce i risultati 1,2,3 per il segmento di colonna contenente i valori di UnitSales per il cliente Astrida. La numerazione delle righe riparte da 1 per il segmento di colonna successivo, vale a dire Betacab.

La colonna **Row Number** ignora le dimensioni a causa dell'argomento TOTAL per RowNo() e conteggia le righe nella tabella.

Questa espressione restituisce 0 per la prima riga di ogni segmento di colonna, quindi la colonna mostrerà: 0, 2.25, 1.1111111, 0, 2.5, 5, 0, 2, 0 e 4.

### Vedere anche:

p [Above - funzione per grafici \(page 1464\)](#)

## 8.7 Funzioni data e ora

Le funzioni data e ora di Qlik Sense consentono di trasformare e convertire i valori di data e ora. Tutte le funzioni possono essere utilizzate sia nello script di caricamento dei dati che nelle espressioni grafiche.

Le funzioni sono basate su un numero seriale di data e ora che equivale al numero di giorni trascorsi dal 30 dicembre 1899. Il valore di numero intero rappresenta il giorno e il valore frazionale rappresenta l'ora del giorno.

Qlik Sense utilizza il valore numerico del parametro, pertanto un numero è valido come parametro anche quando non è formattato come data oppure come ora. Se il parametro non corrisponde a un valore numerico, ad esempio una stringa, Qlik Sense tenta di interpretare la stringa in base alle variabili ambientali di data e ora.

Se il formato dell'ora utilizzato nel parametro non corrisponde a quello impostato nelle variabili di ambiente, Qlik Sense non sarà in grado di eseguire un'interpretazione corretta. Per risolvere questo problema, modificare le impostazioni o utilizzare una funzione di interpretazione.

Negli esempi per ciascuna funzione vengono utilizzati i formati predefiniti di ora e data hh:mm:ss e YYYY-MM-DD (ISO 8601).



*Quando elabora un valore di data e ora con una funzione data o ora, Qlik Sense ignora eventuali parametri relativi all'ora legale, a meno che la funzione data o ora non comprenda una posizione geografica.*

*Ad esempio, la funzione `convertToLocalTime( filetime('Time.qvd'), 'Paris')` utilizzerà i parametri relativi all'ora legale, mentre la funzione `convertToLocalTime(filetime('Time.qvd'), 'GMT-01:00')` non li utilizzerà.*

### Prospetto delle funzioni data e ora

Ciascuna funzione viene descritta ulteriormente dopo la panoramica. È inoltre possibile fare clic sul nome della funzione nella sintassi per accedere immediatamente ai dettagli per tale funzione specifica.

#### Espressioni di numero intero dell'ora

##### **second**

Questa funzione restituisce un numero intero che rappresenta il secondo in cui la frazione di **expression** viene interpretata come ora in base all'interpretazione numerica standard.

**second** (expression)

##### **minute**

Questa funzione restituisce un numero intero che rappresenta il minuto in cui la frazione di **expression** viene interpretata come ora in base all'interpretazione numerica standard.

**minute** (expression)

##### **hour**

Questa funzione restituisce un numero intero che rappresenta l'ora in cui la frazione di **expression** viene interpretata come ora in base all'interpretazione numerica standard.

**hour** (expression)

##### **day**

Questa funzione restituisce un numero intero che rappresenta il giorno in cui la frazione di **expression** viene interpretata come data in base all'interpretazione numerica standard.

**day** (expression)

##### **week**

Questa funzione restituisce un numero intero che rappresenta il numero della settimana in base allo standard ISO 8601. Il numero della settimana viene calcolato a partire dall'interpretazione della data dell'espressione in base all'interpretazione numerica standard.

**week** (expression)

### month

Questa funzione restituisce un valore duale: il nome del mese come definito nella variabile di ambiente **MonthNames** e un numero intero compreso tra 1 e 12. Il numero del mese viene calcolato a partire dall'interpretazione della data dell'espressione in base all'interpretazione numerica standard.

```
month (expression)
```

### year

Questa funzione restituisce un numero intero che rappresenta l'anno in cui **expression** viene interpretato come data in base all'interpretazione numerica standard.

```
year (expression)
```

### weekyear

Questa funzione restituisce l'anno a cui appartiene il numero della settimana in base alle variabili di ambiente. I numeri della settimana rientrano in un intervallo approssimativo compreso tra 1 e 52.

```
weekyear (expression)
```

### weekday

Questa funzione restituisce un valore duale con:

- Il nome di un giorno come definito nella variabile di ambiente **DayNames**.
- Un numero intero compreso tra 0 e 6 che corrisponde al giorno nominale della settimana (0-6).

```
weekday (date)
```

## Funzioni timestamp

### now

Questa funzione restituisce un indicatore temporale dell'ora attuale. La funzione restituisce i valori nel formato della variabile di sistema **TimeStamp**. Il valore predefinito di **timer\_mode** è 1.

```
now ([ timer_mode])
```

### today

Questa funzione restituisce la data attuale. La funzione restituisce i valori nel formato della variabile di sistema **DateFormat**.

```
today ([timer_mode])
```

### LocalTime

Questa funzione restituisce un indicatore temporale dell'ora attuale fornita per il fuso orario specificato.

```
localtime ([timezone [, ignoreDST ]])
```

## Funzioni make

### makedate

Questa funzione restituisce una data calcolata dall'anno **YYYY**, dal mese **MM** e dal giorno **DD**.

```
makedate (YYYY [ , MM [ , DD ] ])
```

### **makeweekdate**

Questa funzione restituisce una data calcolata dall'anno, dal numero di settimana e dal giorno della settimana.

```
makeweekdate (YYYY [ , WW [ , D ] ])
```

### **maketime**

Questa funzione restituisce una data calcolata dall'ora **hh**, dal minuto **mm** e dal secondo **ss**.

```
maketime (hh [ , mm [ , ss [ .fff ] ] ])
```

## Altre funzioni date

### **AddMonths**

Questa funzione restituisce la data che ricorre **n** mesi dopo **startdate** oppure, se **n** è negativo, la data che ricorre **n** mesi prima di **startdate**.

```
addmonths (startdate, n , [ , mode])
```

### **AddYears**

Questa funzione restituisce la data che ricorre **n** anni dopo **startdate** oppure, se **n** è negativo, la data che ricorre **n** anni prima di **startdate**.

```
addyears (startdate, n)
```

### **yeartodate**

Questa funzione stabilisce se l'indicatore temporale di input ricade all'interno dell'anno in cui lo script è stato caricato per l'ultima volta e restituisce True in caso affermativo e False in caso negativo.

```
yeartodate (date [ , yearoffset [ , firstmonth [ , todaydate] ] ])
```

## Funzioni timezone

### **timezone**

Questa funzione restituisce il fuso orario impostato sul computer dove è in esecuzione il motore Qlik.

```
timezone ( )
```

### **GMT**

Questa funzione restituisce il valore Greenwich Mean Time attuale ricavato dalle impostazioni locali.

```
GMT ( )
```

### **UTC**

Restituisce il Coordinated Universal Time attuale.

```
UTC ( )
```

### **daylightsaving**

Restituisce il valore di regolazione attuale per l'ora legale, come definito in Windows.

```
daylightsaving ( )
```

### **converttolocaltime**

Converte un indicatore temporale UTC o GMT in ora locale come valore duale. Il luogo può essere qualsiasi città, località o fuso orario nel mondo.

```
converttolocaltime (timestamp [, place [, ignore_dst=false]])
```

### Funzioni set time

#### **setdateyear**

Questa funzione utilizza come input un **timestamp** e un **year** e aggiorna il **timestamp** con l'**year** specificato nell'input.

```
setdateyear (timestamp, year)
```

#### **setdateyearmonth**

Questa funzione utilizza come input un **timestamp**, un **month** e un **year** e aggiorna il **timestamp** con l'**year** e il **month** specificati nell'input.

```
setdateyearmonth (timestamp, year, month)
```

### Funzioni in...

#### **inyear**

Questa funzione restituisce True se **timestamp** ricade all'interno dell'anno contenente **base\_date**.

```
inyear (date, basedate , shift [, first_month_of_year = 1])
```

#### **inyeartodate**

Questa funzione restituisce True se **timestamp** ricade all'interno della parte dell'anno contenente **base\_date** fino a includere l'ultimo millisecondo di **base\_date**.

```
inyeartodate (date, basedate , shift [, first_month_of_year = 1])
```

#### **inquarter**

Questa funzione restituisce True se **timestamp** ricade all'interno del trimestre contenente **base\_date**.

```
inquarter (date, basedate , shift [, first_month_of_year = 1])
```

#### **inquartertodate**

Questa funzione restituisce True se **timestamp** ricade all'interno della parte del trimestre contenente **base\_date** fino a includere l'ultimo millisecondo di **base\_date**.

```
inquartertodate (date, basedate , shift [, first_month_of_year = 1])
```

#### **inmonth**

Questa funzione restituisce True se **timestamp** ricade all'interno del mese contenente **base\_date**.

```
inmonth (date, basedate , shift)
```

#### **inmonthtodate**

Restituisce True se **date** ricade nella parte di mese contenente **basedate** fino a includere l'ultimo millisecondo di **basedate**.

```
inmonthtodate (date, basedate , shift)
```

### inmonths

Questa funzione consente di verificare se un indicatore temporale cade nello stesso mese, bimestre, trimestre, quadrimestre o semestre di una data base. È inoltre possibile stabilire se l'indicatore temporale ricade all'interno di un periodo di tempo precedente o successivo.

```
inmonths (n, date, basedate , shift [, first_month_of_year = 1])
```

### inmonthstodate

Questa funzione stabilisce se un indicatore temporale ricade all'interno della parte di un periodo di un mese, bimestre, trimestre, quadrimestre o semestre fino a includere l'ultimo millisecondo di **base\_date**. È inoltre possibile stabilire se l'indicatore temporale ricade all'interno di un periodo di tempo precedente o successivo.

```
inmonthstodate (n, date, basedate , shift [, first_month_of_year = 1])
```

### inweek

Questa funzione restituisce True se **timestamp** ricade all'interno della settimana contenente **base\_date**.

```
inweek (date, basedate , shift [, weekstart])
```

### inweektodate

Questa funzione restituisce True se **timestamp** ricade all'interno della parte della settimana contenente **base\_date** fino a includere l'ultimo millisecondo di **base\_date**.

```
inweektodate (date, basedate , shift [, weekstart])
```

### inlunarweek

Questa funzione determina se **timestamp** ricade all'interno della settimana lunare contenente **base\_date**. Le settimane lunari in Qlik Sense sono definite contando il 1° gennaio come primo giorno della settimana, a parte l'ultima settimana dell'anno, ogni settimana conterrà esattamente sette giorni.

```
inlunarweek (date, basedate , shift [, weekstart])
```

### inlunarweektodate

Questa funzione stabilisce se **timestamp** ricade all'interno della parte della settimana lunare fino a includere l'ultimo millisecondo di **base\_date**. Le settimane lunari in Qlik Sense sono definite contando il 1° gennaio come primo giorno della settimana e, a parte l'ultima settimana dell'anno, conterranno esattamente sette giorni.

```
inlunarweektodate (date, basedate , shift [, weekstart])
```

### inday

Questa funzione restituisce True se **timestamp** ricade all'interno del giorno contenente **base\_timestamp**.

```
inday (timestamp, basetimestamp , shift [, daystart])
```

### indaytotime

Questa funzione restituisce True se **timestamp** ricade nella parte del giorno contenente **base\_timestamp** fino a includere il millisecondo esatto di **base\_timestamp**.

```
indaytotime (timestamp, basetimestamp , shift [, daystart])
```

### Funzioni start ... end

#### yearstart

Questa funzione restituisce un indicatore temporale corrispondente all'inizio del primo giorno dell'anno contenente **date**. Il formato di output predefinito sarà il formato **DateFormat** impostato nello script.

```
yearstart ( date [, shift = 0 [, first_month_of_year = 1]])
```

#### yearend

Questa funzione restituisce un valore corrispondente a un indicatore temporale recante l'ultimo millisecondo dell'ultimo giorno dell'anno contenente **date**. Il formato di output predefinito sarà il formato **DateFormat** impostato nello script.

```
yearend ( date [, shift = 0 [, first_month_of_year = 1]])
```

#### yearname

Questa funzione restituisce un anno di quattro cifre come valore di visualizzazione con un valore numerico sottostante corrispondente a un indicatore temporale recante il primo millisecondo del primo giorno dell'anno contenente **date**.

```
yearname (date [, shift = 0 [, first_month_of_year = 1]] )
```

#### quarterstart

Questa funzione restituisce un valore corrispondente a un indicatore temporale recante il primo millisecondo del trimestre contenente **date**. Il formato di output predefinito sarà il formato **DateFormat** impostato nello script.

```
quarterstart (date [, shift = 0 [, first_month_of_year = 1]])
```

#### quarterend

Questa funzione restituisce un valore corrispondente a un indicatore temporale recante l'ultimo millisecondo del trimestre contenente **date**. Il formato di output predefinito sarà il formato **DateFormat** impostato nello script.

```
quarterend (date [, shift = 0 [, first_month_of_year = 1]])
```

#### quartername

Questa funzione restituisce un valore di visualizzazione che mostra i mesi del trimestre (formattati in base alla variabile di script **MonthNames**) e l'anno con valore numerico sottostante corrispondente a un indicatore temporale recante il primo millisecondo del primo giorno del trimestre.

```
quartername (date [, shift = 0 [, first_month_of_year = 1]])
```

### monthstart

Questa funzione restituisce un valore corrispondente a un indicatore temporale recante il primo millisecondo del primo giorno del mese contenente **date**. Il formato di output predefinito sarà il formato **DateFormat** impostato nello script.

```
monthstart (date [, shift = 0])
```

### monthend

Questa funzione restituisce un valore corrispondente a un indicatore temporale recante l'ultimo millisecondo dell'ultimo giorno del mese contenente **date**. Il formato di output predefinito sarà il formato **DateFormat** impostato nello script.

```
monthend (date [, shift = 0])
```

### monthname

Questa funzione restituisce un valore di visualizzazione che mostra il mese (formattato in base alla variabile di script **MonthNames**) e l'anno il cui valore numerico sottostante corrisponde a un indicatore temporale recante il primo millisecondo del primo giorno del mese.

```
monthname (date [, shift = 0])
```

### monthsstart

Questa funzione restituisce un valore corrispondente a un indicatore temporale recante il primo millisecondo del mese, del bimestre, del trimestre, del quadrimestre o del semestre contenente una data di base. È inoltre possibile individuare il timestamp per un periodo di tempo precedente o successivo. Il formato di output predefinito è **DateFormat**, impostato nello script.

```
monthsstart (n, date [, shift = 0 [, first_month_of_year = 1]])
```

### monthsend

Questa funzione restituisce un valore corrispondente a un indicatore temporale recante l'ultimo millisecondo del mese, del bimestre, del trimestre, del quadrimestre o del semestre contenente una data di base. È inoltre possibile individuare l'indicatore temporale per un periodo di tempo precedente o successivo.

```
monthsend (n, date [, shift = 0 [, first_month_of_year = 1]])
```

### monthsname

Questa funzione restituisce un valore di visualizzazione che rappresenta l'intervallo dei mesi del periodo (formattati in base alla variabile di script **MonthNames**) e l'anno. Il valore numerico sottostante corrisponde a un indicatore temporale recante il primo millisecondo del mese, del bimestre, del trimestre, del quadrimestre o del semestre contenente una data di base.

```
monthsname (n, date [, shift = 0 [, first_month_of_year = 1]])
```

### weekstart

Questa funzione restituisce un valore corrispondente a un indicatore temporale recante il primo millisecondo del primo giorno della settimana di calendario contenente **date**. Il formato di output predefinito è il formato **DateFormat** impostato nello script.

```
weekstart (date [, shift = 0 [,weekoffset = 0]])
```

### weekend

Questa funzione restituisce un valore corrispondente a un timestamp recante l'ultimo millisecondo dell'ultimo giorno della settimana di calendario contenente **date**. Il formato di output predefinito sarà il formato **DateFormat** impostato nello script.

```
weekend (date [, shift = 0 [,weekoffset = 0]])
```

### weekname

Questa funzione restituisce un valore che mostra l'anno e il numero della settimana con un valore numerico sottostante corrispondente a un indicatore temporale recante il primo millisecondo del primo giorno della settimana contenente **date**.

```
weekname (date [, shift = 0 [,weekoffset = 0]])
```

### lunarweekstart

Questa funzione restituisce un valore corrispondente a un indicatore temporale del primo millisecondo del primo giorno della settimana lunare contenente **date**. Le settimane lunari in Qlik Sense sono definite contando il 1° gennaio come primo giorno della settimana e, a parte l'ultima settimana dell'anno, conterranno esattamente sette giorni.

```
lunarweekstart (date [, shift = 0 [,weekoffset = 0]])
```

### lunarweekend

Questa funzione restituisce un valore corrispondente a un indicatore temporale recante l'ultimo millisecondo dell'ultimo giorno della settimana lunare contenente **date**. Le settimane lunari in Qlik Sense sono definite contando il 1° gennaio come primo giorno della settimana e, a parte l'ultima settimana dell'anno, conterranno esattamente sette giorni.

```
lunarweekend (date [, shift = 0 [,weekoffset = 0]])
```

### lunarweekname

Questa funzione restituisce un valore di visualizzazione che mostra l'anno e il numero della settimana lunare corrispondente a un indicatore temporale del primo millisecondo del primo giorno della settimana lunare contenente **date**. Le settimane lunari in Qlik Sense sono definite contando il 1° gennaio come primo giorno della settimana e, a parte l'ultima settimana dell'anno, conterranno esattamente sette giorni.

```
lunarweekname (date [, shift = 0 [,weekoffset = 0]])
```

### daystart

Questa funzione restituisce un valore corrispondente a un indicatore temporale con il primo millisecondo del giorno contenuto nell'argomento **time**. Il formato di output predefinito sarà il formato **TimestampFormat** impostato nello script.

```
daystart (timestamp [, shift = 0 [, dayoffset = 0]])
```

### dayend

Questa funzione restituisce un valore corrispondente a un indicatore temporale dell'ultimo millisecondo del giorno contenuto in **time**. Il formato di output predefinito sarà il formato **TimestampFormat** impostato nello script.

```
dayend (timestamp [, shift = 0 [, dayoffset = 0]])
```

### dayname

Questa funzione restituisce un valore che mostra la data con un valore numerico sottostante corrispondente a un indicatore temporale recante il primo millisecondo del giorno contenente **time**.

```
dayname (timestamp [, shift = 0 [, dayoffset = 0]])
```

## Funzioni di numerazione del giorno

### age

La funzione **age** restituisce l'età al momento dell'indicazione della data e dell'ora **timestamp** (in anni completi) di un soggetto nato nella data **date\_of\_birth**.

```
age (timestamp, date_of_birth)
```

### networkdays

La funzione **networkdays** restituisce il numero di giorni lavorativi (dal lunedì al venerdì) compresi tra e inclusi in **start\_date** e **end\_date**, tenendo in considerazione qualsiasi eventuale valore di festività **holiday** nel calendario.

```
networkdays (start:date, end_date {, holiday})
```

### firstworkdate

La funzione **firstworkdate** restituisce la data di inizio più recente per fare in modo che il valore **no\_of\_workdays** (dal lunedì al venerdì) non termini oltre la data **end\_date**, tenendo in considerazione tutte le festività eventualmente in calendario. **end\_date** e **holiday** devono essere date o indicatori temporali validi.

```
firstworkdate (end_date, no_of_workdays {, holiday} )
```

### lastworkdate

La funzione **lastworkdate** restituisce la data di fine più prossima per ottenere **no\_of\_workdays** (dal lunedì al venerdì) se si inizia dalla data **start\_date** tenendo in considerazione tutte le festività **holiday** eventualmente in calendario. **start\_date** e **holiday** devono essere date o indicatori temporali validi.

```
lastworkdate (start_date, no_of_workdays {, holiday})
```

### daynumberofyear

Questa funzione calcola il numero del giorno dell'anno a cui è stato assegnato un indicatore temporale. Il calcolo viene eseguito partendo dal primo millisecondo del primo giorno dell'anno, tuttavia il primo mese può essere differito.

```
daynumberofyear (date[, firstmonth])
```

**daynumberofquarter**

Questa funzione calcola il numero del giorno del trimestre a cui è stato assegnato un indicatore temporale. Questa funzione viene utilizzata al momento di creare un Calendario principale.

```
daynumberofquarter (date[,firstmonth])
```

**addmonths**

Questa funzione restituisce la data che ricorre **n** mesi dopo **startdate** oppure, se **n** è negativo, la data che ricorre **n** mesi prima di **startdate**.

**Sintassi:**

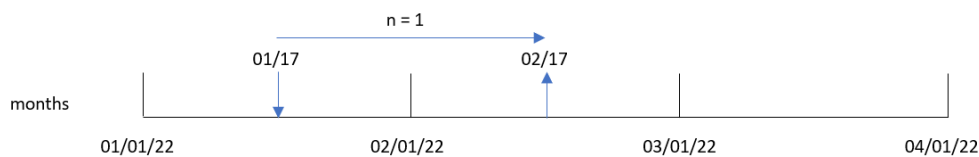
```
AddMonths (startdate, n , [ , mode])
```

**Tipo di dati restituiti:** duale

La funzione `addmonths()` aggiunge o sottrae un numero definito di mesi, **n**, da un **startdate** e restituisce la data risultante.

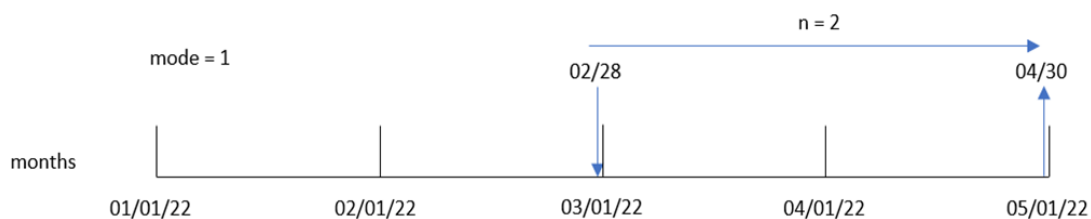
L'argomento **mode** avrà un impatto sui valori **startdate** non prima del 28 del mese. Impostando l'argomento **mode** a 1, la funzione `addmonths()` restituisce una data che equivale nella distanza relativa alla fine del mese come **startdate**.

*Schema esemplificativo della funzione `addmonths()`*



Ad esempio, il 28 febbraio è l'ultimo giorno del mese. Se nella funzione `addmonths()`, con un **mode** di 1, è utilizzato per restituire la data due mesi dopo, la funzione restituirà l'ultima data di aprile, il 30 aprile.

*Schema di esempio della funzione `addmonths()`, con **mode**=1.*

**Argomenti**

| Argomento | Descrizione                                                                |
|-----------|----------------------------------------------------------------------------|
| startdate | La data di inizio come indicazione di data e ora, ad esempio '2012-10-12'. |

| Argomento | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| n         | Numero di mesi come numero intero positivo o negativo.                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| mode      | Specifica se il mese viene aggiunto relativamente all'inizio o alla fine del mese. La modalità predefinita è 0 per le aggiunte relative all'inizio del mese. Impostare la modalità su 1 per le aggiunte relative alla fine del mese. Quando la modalità è impostata su 1 e la data di input è il 28 o successiva, la funzione controlla quanti giorni mancano per raggiungere la fine del mese nella data di inizio. Sulla data restituita viene impostato lo stesso numero di giorni per raggiungere la fine del mese. |

## Casi di utilizzo

La funzione `addmonths()` verrà utilizzata comunemente in un'espressione per trovare una data un certo numero di mesi prima o dopo un periodo di tempo.

Ad esempio, la funzione `addmonths()` può essere utilizzata per identificare la data di fine dei contratti per la telefonia mobile.

### Esempi di funzioni

| Esempio                                      | Risultato                 |
|----------------------------------------------|---------------------------|
| <code>addmonths ('01/29/2003' , 3)</code>    | Restituisce '04/29/2003'. |
| <code>addmonths ('01/29/2003' , 3, 0)</code> | Restituisce '04/29/2003'. |
| <code>addmonths ('01/29/2003' , 3, 1)</code> | Restituisce '04/28/2003'. |
| <code>addmonths ('01/29/2003' , 1, 0)</code> | Restituisce '02/28/2003'. |
| <code>addmonths ('01/29/2003' , 1, 1)</code> | Restituisce '02/26/2003'. |
| <code>addmonths ('02/28/2003' , 1, 0)</code> | Restituisce '03/28/2003'. |
| <code>addmonths ('02/28/2003' , 1, 1)</code> | Restituisce '03/31/2003'. |
| <code>addmonths ('01/29/2003' , -3)</code>   | Restituisce '10/29/2002'. |

## Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - Nessun argomento aggiuntivo

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni tra il 2020 e il 2022, caricato in una tabella denominata `Transactions`.
- Il campo della data fornito nel formato della variabile di sistema `DateFormat` (MM/GG/AAAA).
- La creazione di un campo, `two_months_later`, che restituisce la data per due mesi dopo l'avvenuta transazione.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        addmonths(date,2) as two_months_later
    ;
```

```
Load
```

```
*
```

```
InLine
```

```
[
```

```
id,date,amount
```

```
8188,'01/10/2020',37.23
```

```
8189,'02/28/2020',17.17
```

```
8190,'04/09/2020',88.27
```

```
8191,'04/16/2020',57.42
```

```
8192,'05/21/2020',53.80
```

```
8193,'08/14/2020',82.06
```

```
8194,'10/07/2020',40.39
```

```
8195,'12/05/2020',87.21
```

```
8196,'01/22/2021',95.93
```

```
8197,'02/03/2021',45.89
```

```
8198,'03/17/2021',36.23
```

```
8199,'04/23/2021',25.66
```

```
8200,'05/04/2021',82.77
```

```
8201,'06/30/2021',69.98
```

```
8202,'07/26/2021',76.11
```

```
8203,'12/27/2021',25.12
```

```
8204,'02/02/2022',46.23
```

```
8205,'02/26/2022',84.21
```

```
8206,'03/07/2022',96.24
```

```
8207,'03/11/2022',67.67
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

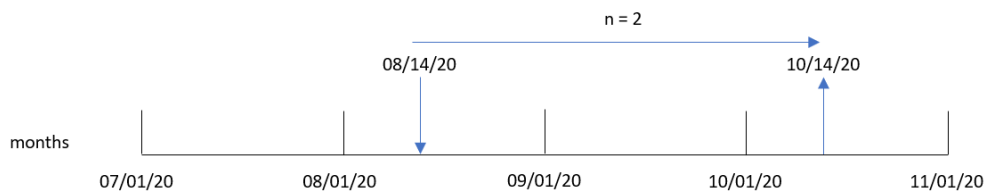
- date
- two\_months\_later

Tabella dei risultati

| date       | two_months_later |
|------------|------------------|
| 01/10/2020 | 03/10/2020       |
| 02/28/2020 | 04/28/2020       |
| 04/09/2020 | 06/09/2020       |
| 04/16/2020 | 06/16/2020       |
| 05/21/2020 | 07/21/2020       |
| 08/14/2020 | 10/14/2020       |
| 10/07/2020 | 12/07/2020       |
| 12/05/2020 | 02/05/2021       |
| 01/22/2021 | 03/22/2021       |
| 02/03/2021 | 04/03/2021       |
| 03/17/2021 | 05/17/2021       |
| 04/23/2021 | 06/23/2021       |
| 05/04/2021 | 07/04/2021       |
| 06/30/2021 | 08/30/2021       |
| 07/26/2021 | 09/26/2021       |
| 12/27/2021 | 02/27/2022       |
| 02/02/2022 | 04/02/2022       |
| 02/26/2022 | 04/26/2022       |
| 03/07/2022 | 05/07/2022       |
| 03/11/2022 | 05/11/2022       |

Il campo `two_months_later` viene creato nell'istruzione caricamento precedente utilizzando la funzione `addmonths()`. Il primo argomento fornito identifica la data da valutare. Il secondo argomento è il numero di mesi da aggiungere o sottrarre all'intervallo `startdate`. In questo caso, viene fornito il valore 2.

Schema della funzione `addmonths()`, esempio senza argomenti aggiuntivi



La transazione 8193 è avvenuta il 14 agosto. Pertanto, la funzione `addmonths()` restituisce il 14 ottobre 2020 per il campo `two_months_later`.

### Esempio 2 - Fine mese relativo

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente una serie di transazioni di fine mese nel 2022, che viene caricato in una tabella chiamata `Transactions`.
- Il campo della data fornito nel formato della variabile di sistema `DateFormat` (MM/GG/AAAA).
- La creazione di un campo, `relative_two_months_prior`, che restituisce la data di fine mese relativa ai due mesi precedenti la transazione.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
    *,
    addmonths(date,-2,1) as relative_two_months_prior
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
id,date,amount
8188,'01/28/2022',37.23
8189,'01/31/2022',57.54
8190,'02/28/2022',17.17
8191,'04/29/2022',88.27
8192,'04/30/2022',57.42
8193,'05/31/2022',53.80
8194,'08/14/2022',82.06
8195,'10/07/2022',40.39
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

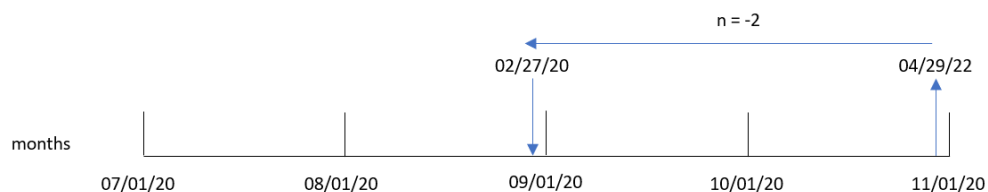
- date
- relative\_two\_months\_prior

Tabella dei risultati

| date       | relative_two_months_prior |
|------------|---------------------------|
| 01/28/2022 | 11/27/2021                |
| 01/31/2022 | 11/30/2021                |
| 02/28/2022 | 12/31/2021                |
| 04/29/2022 | 02/27/2022                |
| 04/30/2022 | 02/28/2022                |
| 05/31/2022 | 03/31/2022                |
| 08/14/2022 | 06/14/2022                |
| 10/07/2022 | 08/07/2022                |

Il campo `relative_two_months_prior` viene creato nell'istruzione `LOAD` precedente utilizzando la funzione `addmonths()`. Il primo argomento fornito identifica la data da valutare. Il secondo argomento è il numero di mesi da aggiungere o sottrarre all'intervallo `startdate`. In questo caso, viene fornito il valore `-2`. L'ultimo argomento è la modalità, con un valore di `1`, che costringe la funzione a calcolare la data di fine mese relativa per tutte le date maggiori o uguali a 28.

*Schema della funzione `addmonths()`, esempio con  $n=-2$ .*



La transazione 8191 avrà luogo il 29 aprile 2022. Inizialmente, due mesi prima si sarebbe impostato il mese a febbraio. Quindi, dato che il terzo argomento della funzione imposta la modalità a 1 e il valore del giorno è successivo al 27, la funzione calcola il valore relativo alla fine del mese. La funzione identifica che il 29 è il penultimo giorno di aprile e pertanto restituisce il penultimo giorno di febbraio, il 27.

### Esempio 3 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor di caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento contiene lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, in questo esempio, il set di dati invariato viene caricato nell'applicazione. Il calcolo che restituisce la data di due mesi dopo la transazione viene creato come misura in un oggetto grafico.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'01/10/2020',37.23
```

```
8189,'02/28/2020',17.17
```

```
8190,'04/09/2020',88.27
```

```
8191,'04/16/2020',57.42
```

```
8192,'05/21/2020',53.80
```

```
8193,'08/14/2020',82.06
```

```
8194,'10/07/2020',40.39
```

```
8195,'12/05/2020',87.21
```

```
8196,'01/22/2021',95.93
```

```
8197,'02/03/2021',45.89
```

```
8198,'03/17/2021',36.23
```

```
8199,'04/23/2021',25.66
```

```
8200,'05/04/2021',82.77
```

```
8201,'06/30/2021',69.98
```

```
8202,'07/26/2021',76.11
```

```
8203,'12/27/2021',25.12
```

```
8204,'02/02/2022',46.23
```

```
8205,'02/26/2022',84.21
```

```
8206,'03/07/2022',96.24
```

```
8207,'03/11/2022',67.67
```

```
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: date.

Creare la seguente misura:

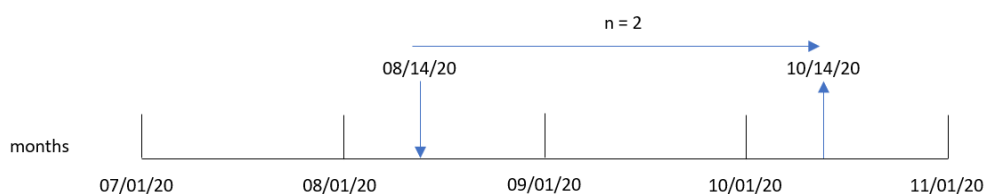
```
=addmonths(date,2)
```

Tabella dei risultati

| date       | =addmonths(date,2) |
|------------|--------------------|
| 01/10/2020 | 03/10/2020         |
| 02/28/2020 | 04/28/2020         |
| 04/09/2020 | 06/09/2020         |
| 04/16/2020 | 06/16/2020         |
| 05/21/2020 | 07/21/2020         |
| 08/14/2020 | 10/14/2020         |
| 10/07/2020 | 12/07/2020         |
| 12/05/2020 | 02/05/2021         |
| 01/22/2021 | 03/22/2021         |
| 02/03/2021 | 04/03/2021         |
| 03/17/2021 | 05/17/2021         |
| 04/23/2021 | 06/23/2021         |
| 05/04/2021 | 07/04/2021         |
| 06/30/2021 | 08/30/2021         |
| 07/26/2021 | 09/26/2021         |
| 12/27/2021 | 02/27/2022         |
| 02/02/2022 | 04/02/2022         |
| 02/26/2022 | 04/26/2022         |
| 03/07/2022 | 05/07/2022         |
| 03/11/2022 | 05/11/2022         |

La misura `two_months_later` viene creata nell'oggetto grafico utilizzando la funzione `addmonths()`. Il primo argomento fornito identifica la data da valutare. Il secondo argomento è il numero di mesi da aggiungere o sottrarre all'intervallo `startdate`. In questo caso, viene fornito il valore 2.

*Schema della funzione `addmonths()`, esempio di oggetto grafico*



La transazione 8193 è avvenuta il 14 agosto. Pertanto, la funzione `addmonths()` restituisce il 14 ottobre 2020 per il campo `two_months_later`.

### Esempio 4 - Scenario

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella chiamata `Mobile_Plans`.
- Informazioni con l'ID del contratto, la data di inizio, la durata del contratto e il canone mensile.

L'utente finale vorrebbe un oggetto grafico che visualizzi, per ID contratto, la data di scadenza di ogni contratto telefonico.

#### Script di caricamento

```
Mobile_Plans:
Load
*
Inline
[
contract_id,start_date,contract_length,monthly_fee
8188,'01/13/2020',18,37.23
8189,'02/26/2020',24,17.17
8190,'03/27/2020',36,88.27
8191,'04/16/2020',24,57.42
8192,'05/21/2020',24,53.80
8193,'08/14/2020',12,82.06
8194,'10/07/2020',18,40.39
8195,'12/05/2020',12,87.21
8196,'01/22/2021',12,95.93
8197,'02/03/2021',18,45.89
8198,'03/17/2021',24,36.23
8199,'04/23/2021',24,25.66
8200,'05/04/2021',12,82.77
8201,'06/30/2021',12,69.98
8202,'07/26/2021',12,76.11
8203,'12/27/2021',36,25.12
8204,'06/06/2022',24,46.23
8205,'07/18/2022',12,84.21
8206,'11/14/2022',12,96.24
8207,'12/12/2022',18,67.67
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- contract\_id
- start\_date
- contract\_length

Crea la seguente misura per calcolare la data di scadenza di ogni contratto:

```
=addmonths(start_date,contract_length, 0)
```

Tabella dei risultati

| contract_id | start_date | contract_length | =addmonths(start_date,contract_length,0) |
|-------------|------------|-----------------|------------------------------------------|
| 8188        | 01/13/2020 | 18              | 07/13/2021                               |
| 8189        | 02/26/2020 | 24              | 02/26/2022                               |
| 8190        | 03/27/2020 | 36              | 03/27/2023                               |
| 8191        | 04/16/2020 | 24              | 04/16/2022                               |
| 8192        | 05/21/2020 | 24              | 05/21/2022                               |
| 8193        | 08/14/2020 | 12              | 08/14/2021                               |
| 8194        | 10/07/2020 | 18              | 04/07/2022                               |
| 8195        | 12/05/2020 | 12              | 12/05/2021                               |
| 8196        | 01/22/2021 | 12              | 01/22/2022                               |
| 8197        | 02/03/2021 | 18              | 08/03/2022                               |
| 8198        | 03/17/2021 | 24              | 03/17/2023                               |
| 8199        | 04/23/2021 | 24              | 04/23/2023                               |
| 8200        | 05/04/2021 | 12              | 05/04/2022                               |
| 8201        | 06/30/2021 | 12              | 06/30/2022                               |
| 8202        | 07/26/2021 | 12              | 07/26/2022                               |
| 8203        | 12/27/2021 | 36              | 12/27/2024                               |
| 8204        | 06/06/2022 | 24              | 06/06/2024                               |
| 8205        | 07/18/2022 | 12              | 07/18/2023                               |
| 8206        | 11/14/2022 | 12              | 11/14/2023                               |
| 8207        | 12/12/2022 | 18              | 06/12/2024                               |

### addyears

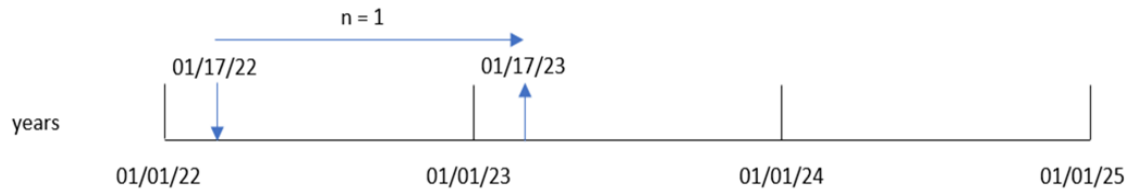
Questa funzione restituisce la data che ricorre **n** anni dopo **startdate** oppure, se **n** è negativo, la data che ricorre **n** anni prima di **startdate**.

Sintassi:

```
AddYears (startdate, n)
```

**Tipo di dati restituiti:** duale

*Schema esemplificativo della funzione `addyears()`*



La funzione `addyears()` aggiunge o sottrae un numero definito di anni, *n*, da un valore di `startdate`. Quindi restituisce la data risultante.

#### Argomenti

| Argomento              | Descrizione                                                                |
|------------------------|----------------------------------------------------------------------------|
| <code>startdate</code> | La data di inizio come indicazione di data e ora, ad esempio '2012-10-12'. |
| <i>n</i>               | Numero di anni come numero intero positivo o negativo.                     |

#### Esempi di funzioni

| Esempio                                  | Risultato                 |
|------------------------------------------|---------------------------|
| <code>addyears ('01/29/2010', 3)</code>  | Restituisce '01/29/2013'. |
| <code>addyears ('01/29/2010', -1)</code> | Restituisce '01/29/2009'. |

## Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - Esempio semplice

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni tra il 2020 e il 2022, caricato in una tabella denominata `Transactions`.
- Il campo della data fornito nel formato della variabile di sistema `DateFormat` (MM/GG/AAAA).
- La creazione di un campo, `two_years_later`, che restituisce la data per due anni dopo l'avvenuta transazione.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        addyears(date,2) as two_years_later
    ;
```

```
Load
```

```
*
```

```
InLine
```

```
[
```

```
id,date,amount
```

```
8188,'01/10/2020',37.23
```

```
8189,'02/28/2020',17.17
```

```
8190,'04/09/2020',88.27
```

```
8191,'04/16/2020',57.42
```

```
8192,'05/21/2020',53.80
```

```
8193,'08/14/2020',82.06
```

```
8194,'10/07/2020',40.39
```

```
8195,'12/05/2020',87.21
```

```
8196,'01/22/2021',95.93
```

```
8197,'02/03/2021',45.89
```

```
8198,'03/17/2021',36.23
```

```
8199,'04/23/2021',25.66
```

```
8200,'05/04/2021',82.77
```

```
8201,'06/30/2021',69.98
```

```
8202,'07/26/2021',76.11
```

```
8203,'12/27/2021',25.12
```

```
8204,'02/02/2022',46.23
```

```
8205,'02/26/2022',84.21
```

```
8206,'03/07/2022',96.24
```

```
8207,'03/11/2022',67.67
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

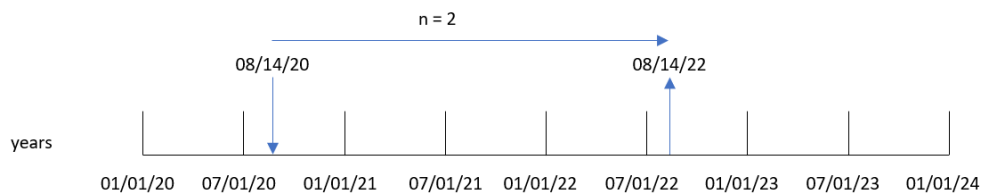
- date
- two\_years\_later

Tabella dei risultati

| date       | due_anni_dopo |
|------------|---------------|
| 01/10/2020 | 01/10/2022    |
| 02/28/2020 | 02/28/2022    |
| 04/09/2020 | 04/09/2022    |
| 04/16/2020 | 04/16/2022    |
| 05/21/2020 | 05/21/2022    |
| 08/14/2020 | 08/14/2022    |
| 10/07/2020 | 10/07/2022    |
| 12/05/2020 | 12/05/2022    |
| 01/22/2021 | 01/22/2023    |
| 02/03/2021 | 02/03/2023    |
| 03/17/2021 | 03/17/2023    |
| 04/23/2021 | 04/23/2023    |
| 05/04/2021 | 05/04/2023    |
| 06/30/2021 | 06/30/2023    |
| 07/26/2021 | 07/26/2023    |
| 12/27/2021 | 12/27/2023    |
| 02/02/2022 | 02/02/2024    |
| 02/26/2022 | 02/26/2024    |
| 03/07/2022 | 03/07/2024    |
| 03/11/2022 | 03/11/2024    |

Il campo `two_years_later` viene creato nell'istruzione caricamento precedente utilizzando la funzione `addyears()`. Il primo argomento fornito identifica la data da valutare. Il secondo argomento è il numero di anni da aggiungere o sottrarre dalla data di inizio. In questo caso, viene fornito il valore 2.

*Schema della funzione addyears(), esempio di base*



La transazione 8193 è avvenuta il 14 agosto 2020. Pertanto, la funzione addyears() restituisce il 14 agosto 2022 per il campo two\_years\_later.

### Esempio 2 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni tra il 2020 e il 2022, caricato in una tabella denominata Transactions.
- Il campo della data fornito nel formato della variabile di sistema DateFormat (MM/GG/AAAA).

In un oggetto grafico, creare una misura, prior\_year\_date, che restituisce la data di un anno precedente a quella in cui avviene la transazione.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

Transactions:

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'01/10/2020',37.23
```

```
8189,'02/28/2020',17.17
```

```
8190,'04/09/2020',88.27
```

```
8191,'04/16/2020',57.42
```

```
8192,'05/21/2020',53.80
```

```
8193,'08/14/2020',82.06
```

```
8194,'10/07/2020',40.39
```

```
8195,'12/05/2020',87.21
```

```
8196,'01/22/2021',95.93
```

```
8197,'02/03/2021',45.89
```

```
8198,'03/17/2021',36.23
```

```
8199,'04/23/2021',25.66
```

```
8200, '05/04/2021', 82.77
8201, '06/30/2021', 69.98
8202, '07/26/2021', 76.11
8203, '12/27/2021', 25.12
8204, '02/02/2022', 46.23
8205, '02/26/2022', 84.21
8206, '03/07/2022', 96.24
8207, '03/11/2022', 67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: date.

Crea la seguente misura per calcolare la data di un anno prima di ogni transazione:

```
=addyears(date, -1)
```

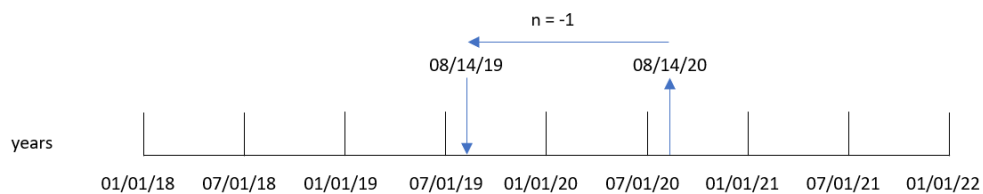
Tabella dei risultati

| date       | =addyears(data,-1) |
|------------|--------------------|
| 01/10/2020 | 01/10/2019         |
| 02/28/2020 | 02/28/2019         |
| 04/09/2020 | 04/09/2019         |
| 04/16/2020 | 04/16/2019         |
| 05/21/2020 | 05/21/2019         |
| 08/14/2020 | 08/14/2019         |
| 10/07/2020 | 10/07/2019         |
| 12/05/2020 | 12/05/2019         |
| 01/22/2021 | 01/22/2020         |
| 02/03/2021 | 02/03/2020         |
| 03/17/2021 | 03/17/2020         |
| 04/23/2021 | 04/23/2020         |
| 05/04/2021 | 05/04/2020         |
| 06/30/2021 | 06/30/2020         |
| 07/26/2021 | 07/26/2020         |
| 12/27/2021 | 12/27/2020         |
| 02/02/2022 | 02/02/2021         |
| 02/26/2022 | 02/26/2021         |

| date       | =addyears(data,-1) |
|------------|--------------------|
| 03/07/2022 | 03/07/2021         |
| 03/11/2022 | 03/11/2021         |

La misura `one_year_prior` viene creata nell'oggetto grafico utilizzando la funzione `addyears()`. Il primo argomento fornito identifica la data da valutare. Il secondo argomento è il numero di anni da aggiungere o sottrarre da `startdate`. In questo caso, viene fornito il valore `-1`.

*Schema della funzione `addyears()`, esempio di oggetto grafico*



La transazione 8193 è avvenuta il 14 agosto. Pertanto, la funzione `addyears()` restituisce il 14 agosto 2019 per il campo `one_year_prior`.

### Esempio 3 - Scenario

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella chiamata `warranties`.
- Informazioni con l'ID del prodotto, la data di acquisto, la durata della garanzia e il prezzo di acquisto.

L'utente finale vorrebbe un oggetto grafico che visualizzi, per ID prodotto, la data di cessazione della garanzia di ogni prodotto.

#### Script di caricamento

Warranties:

Load

\*

Inline

[

`product_id,purchase_date,warranty_length,purchase_price`

`8188,'01/13/2020',4,32000`

`8189,'02/26/2020',2,28000`

`8190,'03/27/2020',3,41000`

`8191,'04/16/2020',4,17000`

`8192,'05/21/2020',2,25000`

```
8193, '08/14/2020', 1, 59000
8194, '10/07/2020', 2, 12000
8195, '12/05/2020', 3, 12000
8196, '01/22/2021', 4, 24000
8197, '02/03/2021', 1, 50000
8198, '03/17/2021', 2, 80000
8199, '04/23/2021', 3, 10000
8200, '05/04/2021', 4, 30000
8201, '06/30/2021', 3, 30000
8202, '07/26/2021', 4, 20000
8203, '12/27/2021', 4, 10000
8204, '06/06/2022', 2, 25000
8205, '07/18/2022', 1, 32000
8206, '11/14/2022', 1, 30000
8207, '12/12/2022', 4, 22000
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- product\_id
- purchase\_date
- warranty\_length

Crea la seguente misura per calcolare la data di scadenza della garanzia di ogni prodotto:

```
=addyears(purchase_date,warranty_length)
```

Tabella dei risultati

| id_prodotto | data_acquisto | durata_garanzia | =aggiungianni(data_acquisto,durata_garanzia) |
|-------------|---------------|-----------------|----------------------------------------------|
| 8188        | 01/13/2020    | 4               | 01/13/2024                                   |
| 8189        | 02/26/2020    | 2               | 02/26/2022                                   |
| 8190        | 03/27/2020    | 3               | 03/27/2023                                   |
| 8191        | 04/16/2020    | 4               | 04/16/2024                                   |
| 8192        | 05/21/2020    | 2               | 21/05/2022                                   |
| 8193        | 08/14/2020    | 1               | 08/14/2021                                   |
| 8194        | 10/07/2020    | 2               | 10/07/2022                                   |
| 8195        | 12/05/2020    | 3               | 12/05/2023                                   |
| 8196        | 01/22/2021    | 4               | 01/22/2025                                   |
| 8197        | 02/03/2021    | 1               | 02/03/2022                                   |
| 8198        | 03/17/2021    | 2               | 17/03/2023                                   |
| 8199        | 04/23/2021    | 3               | 04/23/2024                                   |

| id_prodotto | data_acquisto | durata_garanzia | =aggiungianni(data_acquisto,durata_garanzia) |
|-------------|---------------|-----------------|----------------------------------------------|
| 8200        | 05/04/2021    | 4               | 05/04/2025                                   |
| 8201        | 06/30/2021    | 3               | 06/30/2024                                   |
| 8202        | 07/26/2021    | 4               | 07/26/2025                                   |
| 8203        | 12/27/2021    | 4               | 12/27/2025                                   |
| 8204        | 06/06/2022    | 2               | 06/06/2024                                   |
| 8205        | 07/18/2022    | 1               | 07/18/2023                                   |
| 8206        | 11/14/2022    | 1               | 11/14/2023                                   |
| 8207        | 12/12/2022    | 4               | 12/12/2026                                   |

## age

La funzione **age** restituisce l'età al momento dell'indicazione della data e dell'ora **timestamp** (in anni completi) di un soggetto nato nella data **date\_of\_birth**.

### Sintassi:

```
age (timestamp, date_of_birth)
```

Può essere un'espressione.

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento            | Descrizione                                                                                                                       |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| <b>timestamp</b>     | L'indicatore temporale, o l'espressione che risolve l'indicatore temporale, fino al quale calcolare il numero di anni completati. |
| <b>date_of_birth</b> | Data di nascita della persona di cui si sta calcolando l'età. Può essere un'espressione.                                          |

### Esempi e risultati:

In questi esempi viene utilizzato il formato della data **DD/MM/YYYY**. Il formato della data viene specificato nell'istruzione **SET DateFormat** nella parte superiore dello script di caricamento dei dati. Modificare il formato negli esempi in base alle proprie necessità.

#### Esempi di script

| Esempio                                      | Risultato      |
|----------------------------------------------|----------------|
| <code>age('25/01/2014', '29/10/2012')</code> | Restituisce 1. |
| <code>age('29/10/2014', '29/10/2012')</code> | Restituisce 2. |

### Esempio:

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

```
Employees :
LOAD * INLINE [
Member|DateOfBirth
John|28/03/1989
Linda|10/12/1990
Steve|5/2/1992
Birg|31/3/1993
Raj|19/5/1994
Prita|15/9/1994
Su|11/12/1994
Goran|2/3/1995
Sunny|14/5/1996
Ajoa|13/6/1996
Daphne|7/7/1998
Biffy|4/8/2000
] (delimiter is |);
AgeTable:
Load *,
age('20/08/2015', DateOfBirth) As Age
Resident Employees;
Drop table Employees;
```

La tabella risultante mostra i valori restituiti in age per ciascun record della tabella.

Tabella dei risultati

| Member | DateOfBirth | Age |
|--------|-------------|-----|
| John   | 28/03/1989  | 26  |
| Linda  | 10/12/1990  | 24  |
| Steve  | 5/2/1992    | 23  |
| Birg   | 31/3/1993   | 22  |
| Raj    | 19/5/1994   | 21  |
| Prita  | 15/9/1994   | 20  |
| Su     | 11/12/1994  | 20  |
| Goran  | 2/3/1995    | 20  |
| Sunny  | 14/5/1996   | 19  |
| Ajoa   | 13/6/1996   | 19  |
| Daphne | 7/7/1998    | 17  |
| Biffy  | 4/8/2000    | 15  |

## converttolocaltime

Converte un indicatore temporale UTC o GMT in ora locale come valore duale. Il luogo può essere qualsiasi città, località o fuso orario nel mondo.

Sintassi:

```
ConvertToLocalTime(timestamp [, place [, ignore_dst=false]])
```

Tipo di dati restituiti: duale

### Argomenti

| Argomento        | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>timestamp</b> | L'indicatore temporale, o l'espressione che restituisce un indicatore temporale, da convertire.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>place</b>     | <p>Una località o un fuso orario contenuti nella tabella seguente relativa alle località e ai fusi orari validi. In alternativa, è possibile utilizzare GMT o UTC per definire l'ora locale. I valori e gli intervalli di differimento temporale seguenti sono validi:</p> <ul style="list-style-type: none"> <li>• GMT</li> <li>• GMT-12:00 - GMT-01:00</li> <li>• GMT+01:00 - GMT+14:00</li> <li>• UTC</li> <li>• UTC-12:00 - UTC-01:00</li> <li>• UTC+01:00 - UTC+14:00</li> </ul> <div> <p> Se si utilizza un differimento dalla DST (ovvero, si specifica un valore argomento <b>ignore_dst</b> che restituisce False), è necessario specificare una località, piuttosto che un differimento dall'ora GMT, nell'argomento <b>place</b>. Infatti, un adattamento in funzione dell'ora legale richiede informazioni latitudinali oltre a quelle longitudinali offerte da un differimento dall'ora GMT. Per informazioni, vedere <a href="#">Utilizzo dei differimenti dall'ora GMT in combinazione con la DST (page 730)</a>.</p> <p> È possibile utilizzare differimenti temporali standard. Non è possibile utilizzare un differimento temporale arbitrario, ad esempio GMT-04:27.</p> </div> |

| Argomento         | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ignore_dst</b> | <p>Se l'argomento restituisce True, la DST (ora legale) viene ignorata. I valori argomento validi che restituiscono True includono -1 e <code>True()</code>.</p> <p>Se questo argomento restituisce False, l'indicatore temporale viene adattato in funzione dell'ora legale. I valori argomento validi che restituiscono False includono 0 e <code>False()</code>.</p> <p>Se il valore argomento <b>ignore_dst</b> non è valido, la funzione valuta l'espressione come se il valore <b>ignore_dst</b> restituisse True. Se il valore argomento <b>ignore_dst</b> non è specificato, la funzione valuta l'espressione come se il valore <b>ignore_dst</b> restituisse False.</p> |

Località e fusi orari validi

| A-C                    | D-K                          | L-R                         | S-Z                 |
|------------------------|------------------------------|-----------------------------|---------------------|
| Abu Dhabi              | Darwin                       | La Paz                      | Samoa               |
| Adelaide               | Dhaka                        | Lima                        | Santiago            |
| Alaska                 | Eastern Time (US & Canada)   | Lisbon                      | Sapporo             |
| Amsterdam              | Edinburgh                    | Ljubljana                   | Sarajevo            |
| Arizona                | Ekaterinburg                 | London                      | Saskatchewan        |
| Astana                 | Fiji                         | Madrid                      | Seoul               |
| Athens                 | Georgetown                   | Magadan                     | Singapore           |
| Atlantic Time (Canada) | Greenland                    | Mazatlan                    | Skopje              |
| Auckland               | Greenwich Mean Time : Dublin | Melbourne                   | Sofia               |
| Azores                 | Guadalajara                  | Mexico City                 | Solomon Is.         |
| Baghdad                | Guam                         | Mid-Atlantic                | Sri Jayawardenepura |
| Baku                   | Hanoi                        | Minsk                       | St. Petersburg      |
| Bangkok                | Harare                       | Monrovia                    | Stockholm           |
| Beijing                | Hawaii                       | Monterrey                   | Sydney              |
| Belgrade               | Helsinki                     | Moscow                      | Taipei              |
| Berlin                 | Hobart                       | Mountain Time (US & Canada) | Tallinn             |
| Bern                   | Hong Kong                    | Mumbai                      | Tashkent            |
| Bogota                 | Indiana (East)               | Muscat                      | Tbilisi             |

| A-C                        | D-K                          | L-R                        | S-Z                 |
|----------------------------|------------------------------|----------------------------|---------------------|
| Brasilia                   | International Date Line West | Nairobi                    | Tehran              |
| Bratislava                 | Irkutsk                      | New Caledonia              | Tokyo               |
| Brisbane                   | Islamabad                    | New Delhi                  | Urumqi              |
| Brussels                   | Istanbul                     | Newfoundland               | Warsaw              |
| Bucharest                  | Jakarta                      | Novosibirsk                | Wellington          |
| Budapest                   | Jerusalem                    | Nuku'alofa                 | West Central Africa |
| Buenos Aires               | Kabul                        | Osaka                      | Vienna              |
| Cairo                      | Kamchatka                    | Pacific Time (US & Canada) | Vilnius             |
| Canberra                   | Karachi                      | Paris                      | Vladivostok         |
| Cape Verde Is.             | Kathmandu                    | Perth                      | Volgograd           |
| Caracas                    | Kolkata                      | Port Moresby               | Yakutsk             |
| Casablanca                 | Krasnoyarsk                  | Prague                     | Yerevan             |
| Central America            | Kuala Lumpur                 | Pretoria                   | Zagreb              |
| Central Time (US & Canada) | Kuwait                       | Quito                      | -                   |
| Chennai                    | Kyiv                         | Riga                       | -                   |
| Chihuahua                  | -                            | Riyadh                     | -                   |
| Chongqing                  | -                            | Rome                       | -                   |
| Copenhagen                 | -                            | -                          | -                   |

Esempi e risultati:

Esempi di script

| Esempio                                                        | Risultato                                                                                     |
|----------------------------------------------------------------|-----------------------------------------------------------------------------------------------|
| <code>ConvertToLocalTime('2023-08-14 08:39:47','Paris')</code> | Restituisce '2023-08-14 10:39:47' e la corrispondente rappresentazione interna di data e ora. |
| <code>ConvertToLocalTime(UTC(), 'Stockholm')</code>            | Restituisce l'ora di Stoccolma, adattandola in funzione dell'ora legale.                      |
| <code>ConvertToLocalTime(UTC(), 'Stockholm', -1)</code>        | Restituisce l'ora di Stoccolma, senza adattamento dell'ora legale.                            |

| Esempio                                                     | Risultato                                                                                                                                                                                                                           |
|-------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>ConvertToLocalTime(UTC(), 'GMT-05:00')</code>         | Restituisce l'ora della costa Orientale degli Stati Uniti, ad esempio l'ora di New York. Non viene effettuato nessun adattamento in funzione dell'ora legale perché è specificato il differimento dall'ora GMT, invece di un luogo. |
| <code>ConvertToLocalTime(UTC(), 'New York', -1)</code>      | Restituisce l'ora della costa Orientale degli Stati Uniti (New York), senza adattamento dell'ora legale.                                                                                                                            |
| <code>ConvertToLocalTime(UTC(), 'New York', True())</code>  | Restituisce l'ora della costa Orientale degli Stati Uniti (New York), senza adattamento dell'ora legale.                                                                                                                            |
| <code>ConvertToLocalTime(UTC(), 'New York', 0)</code>       | Restituisce l'ora della costa Orientale degli Stati Uniti (New York), adattandola in funzione dell'ora legale.                                                                                                                      |
| <code>ConvertToLocalTime(UTC(), 'New York', False())</code> | Restituisce l'ora della costa Orientale degli Stati Uniti (New York), adattandola in funzione dell'ora legale.                                                                                                                      |

### Utilizzo dei differimenti dall'ora GMT in combinazione con la DST

In seguito all'implementazione delle librerie ICU (International Components for Unicode) in Qlik Sense, l'utilizzo di differimenti dall'ora GMT (ora media di Greenwich) in combinazione con la DST (ora legale) richiede informazioni latitudinali aggiuntive.

L'ora GMT è un differimento longitudinale (est-ovest), mentre la DST è un differimento latitudinale (nord-sud). Per esempio, Helsinki (Finlandia) e Johannesburg (Sudafrica) condividono lo stesso differimento GMT+02:00, ma non condividono lo stesso differimento dalla DST. Ciò significa che, ulteriormente al differimento dall'ora GMT, qualsiasi differimento dalla DST richiede informazioni sulla posizione latitudinale del fuso orario locale (input del fuso orario geografico) per avere informazioni complete sulle condizioni relative alla DST locale.

## day

Questa funzione restituisce un numero intero che rappresenta il giorno in cui la frazione di **expression** viene interpretata come data in base all'interpretazione numerica standard.

La funzione restituisce il giorno del mese per una data particolare. In genere viene utilizzata per derivare un campo giorno come parte di una dimensione di calendario.

#### Sintassi:

**day** (expression)

**Tipo di dati restituiti:** numero intero

Esempi di funzioni

| Esempio           | Risultato                               |
|-------------------|-----------------------------------------|
| day( 1971-10-12 ) | restituisce 12                          |
| day( 35648 )      | restituisce 6 poiché 35648 = 1997-08-06 |

### Esempio 1 - set di dati DateFormat (script)

Script di caricamento e risultati

#### Panoramica

Aprire Editor caricamento dati, quindi aggiungere lo script di caricamento in basso in una nuova scheda.

Lo script di caricamento contiene:

- Un set di dati per le date denominato `master_calendar`. La variabile di sistema `DateFormat` è impostata su `GG/MM/AAAA`.
- Un caricamento precedente che crea un campo aggiuntivo, denominato `day_of_month`, mediante la funzione `day()`.
- Un campo aggiuntivo, denominato `long_date`, che utilizza la funzione `date()` per esprimere il nome mese completo.

#### Script di caricamento

```
SET DateFormat='DD/MM/YYYY';
```

```
Master_Calendar:
```

```
Load
```

```
    date,  
    date(date,'dd-MMMM-YYYY') as long_date,  
    day(date) as day_of_month
```

```
Inline
```

```
[
```

```
date
```

```
03/11/2022
```

```
03/12/2022
```

```
03/13/2022
```

```
03/14/2022
```

```
03/15/2022
```

```
03/16/2022
```

```
03/17/2022
```

```
03/18/2022
```

```
03/19/2022
```

```
03/20/2022
```

```
03/21/2022
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- long\_date
- day\_of\_month

Tabella dei risultati

| data       | long_date      | day_of_month |
|------------|----------------|--------------|
| 03/11/2022 | 11-March- 2022 | 11           |
| 03/12/2022 | 12-March- 2022 | 12           |
| 03/13/2022 | 13-March- 2022 | 13           |
| 03/14/2022 | 14-March- 2022 | 14           |
| 03/15/2022 | 15-March- 2022 | 15           |
| 03/16/2022 | 16-March- 2022 | 16           |
| 03/17/2022 | 17-March- 2022 | 17           |
| 03/18/2022 | 18-March- 2022 | 18           |
| 03/19/2022 | 19-March- 2022 | 19           |
| 03/20/2022 | 20-March- 2022 | 20           |
| 03/21/2022 | 21-March- 2022 | 21           |

Il giorno del mese viene valutato correttamente dalla funzione `day()` nello script.

### Esempio 2 - Date ANSI (script)

Script di caricamento e risultati

#### Panoramica

Aprire Editor caricamento dati, quindi aggiungere lo script di caricamento in basso in una nuova scheda.

Lo script di caricamento contiene:

- Un set di dati di date denominato `master_calendar`. Viene utilizzata la variabile di sistema `DateFormat GG/MM/AAAA`. Tuttavia, le date incluse nel set di dati sono nel formato data standard ANSI.
- Un caricamento precedente che crea un campo aggiuntivo, denominato `day_of_month`, mediante la funzione `date()`.
- Un campo aggiuntivo, denominato `long_date`, che utilizza la funzione `date()` per esprimere la data con il nome mese completo.

### Script di caricamento

```
SET DateFormat='DD/MM/YYYY';
Master_Calendar:
Load
    date,
    date(date,'dd-MMMM-YYYY') as long_date,
    day(date) as day_of_month

Inline
[
date
2022-03-11
2022-03-12
2022-03-13
2022-03-14
2022-03-15
2022-03-16
2022-03-17
2022-03-18
2022-03-19
2022-03-20
2022-03-21
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- long\_date
- day\_of\_month

Tabella dei risultati

| data       | long_date      | day_of_month |
|------------|----------------|--------------|
| 03/11/2022 | 11-March- 2022 | 11           |
| 03/12/2022 | 12-March- 2022 | 12           |
| 03/13/2022 | 13-March- 2022 | 13           |
| 03/14/2022 | 14-March- 2022 | 14           |
| 03/15/2022 | 15-March- 2022 | 15           |
| 03/16/2022 | 16-March- 2022 | 16           |
| 03/17/2022 | 17-March- 2022 | 17           |
| 03/18/2022 | 18-March- 2022 | 18           |
| 03/19/2022 | 19-March- 2022 | 19           |

| data       | long_date      | day_of_month |
|------------|----------------|--------------|
| 03/20/2022 | 20-March- 2022 | 20           |
| 03/21/2022 | 21-March- 2022 | 21           |

Il giorno del mese viene valutato correttamente dalla funzione `day()` nello script.

### Esempio 3 - Date non formattate (script)

Script di caricamento e risultati

#### Panoramica

Aprire Editor caricamento dati, quindi aggiungere lo script di caricamento in basso in una nuova scheda.

Lo script di caricamento contiene:

- Un set di dati di date denominato `Master_Calendar`. Viene utilizzata la variabile di sistema `DateFormat GG/MM/AAAA`.
- Un caricamento precedente che crea un campo aggiuntivo, denominato `day_of_month`, mediante la funzione `day()`.
- La data non formattata originale, denominata `unformatted_date`.
- Un campo aggiuntivo, denominato `long_date`, che utilizza `date()` viene impiegato per convertire la data numerica in un campo data formattato.

#### Script di caricamento

```
SET DateFormat='DD/MM/YYYY';
```

```
Master_Calendar:
```

```
Load
    unformatted_date,
    date(unformatted_date,'dd-MMMM-YYYY') as long_date,
    day(date) as day_of_month
```

```
Inline
```

```
[
unformatted_date
44868
44898
44928
44958
44988
45018
45048
45078
45008
45038
45068
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- unformatted\_date
- long\_date
- day\_of\_month

Tabella dei risultati

| unformatted_date | long_date         | day_of_month |
|------------------|-------------------|--------------|
| 44868            | 03-November- 2022 | 3            |
| 44898            | 03-December- 2022 | 3            |
| 44928            | 02-January- 2023  | 2            |
| 44958            | 01-February- 2023 | 1            |
| 44988            | 03-March- 2023    | 3            |
| 45008            | 23-March- 2023    | 23           |
| 45018            | 02-April- 2023    | 2            |
| 45038            | 22-April- 2023    | 22           |
| 45048            | 2-May- 2023       | 2            |
| 45068            | 22-May- 2023      | 22           |
| 45078            | 01-June- 2023     | 1            |

Il giorno del mese viene valutato correttamente dalla funzione day() nello script.

### Esempio 4 - Calcolo del mese di scadenza (grafico)

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire Editor caricamento dati, quindi aggiungere lo script di caricamento in basso in una nuova scheda.

Lo script di caricamento contiene:

- Un set di dati degli ordini effettuati a marzo denominato orders. La tabella contiene tre campi:
  - id
  - order\_date
  - importo

### Script di caricamento

```
Orders:
Load
    id,
    order_date,
    amount
Inline
[
id,order_date,amount
1,03/01/2022,231.24
2,03/02/2022,567.28
3,03/03/2022,364.28
4,03/04/2022,575.76
5,03/05/2022,638.68
6,03/06/2022,785.38
7,03/07/2022,967.46
8,03/08/2022,287.67
9,03/09/2022,764.45
10,03/10/2022,875.43
11,03/11/2022,957.35
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:order\_date.

Per calcolare la data di consegna, creare questa misura: =day(order\_date+5).

Tabella dei risultati

| order_date | =day(order_date+5) |
|------------|--------------------|
| 03/11/2022 | 16                 |
| 03/12/2022 | 17                 |
| 03/13/2022 | 18                 |
| 03/14/2022 | 19                 |
| 03/15/2022 | 20                 |
| 03/16/2022 | 21                 |
| 03/17/2022 | 22                 |
| 03/18/2022 | 23                 |
| 03/19/2022 | 24                 |
| 03/20/2022 | 25                 |
| 03/21/2022 | 26                 |

La funzione `day()` attualmente determina che un ordine effettuato l'11 marzo verrà consegnato il 16 sulla base di una tempistica di consegna di 5 giorni.

### dayend

Questa funzione restituisce un valore corrispondente a un indicatore temporale dell'ultimo millisecondo del giorno contenuto in **time**. Il formato di output predefinito sarà il formato **TimestampFormat** impostato nello script.

#### Sintassi:

```
DayEnd (time[, [period_no[, day_start]])
```

#### Casi di utilizzo

La funzione `dayend()` viene comunemente utilizzata come parte di un'espressione quando l'utente desidera che il calcolo utilizzi la frazione di giorno non ancora trascorsa. Ad esempio, per calcolare il totale delle spese ancora da sostenere durante la giornata.

**Tipo di dati restituiti:** duale

#### Argomenti

| Argomento        | Descrizione                                                                                                                                                                                                                                                                                                                                  |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>time</b>      | L'indicatore temporale da valutare.                                                                                                                                                                                                                                                                                                          |
| <b>period_no</b> | <b>period_no</b> è un numero intero o un'espressione la cui risoluzione è un numero intero, in cui il valore 0 indica il giorno che contiene il valore <b>time</b> . I valori negativi di <b>period_no</b> indicano i giorni precedenti, mentre i valori positivi indicano i giorni successivi.                                              |
| <b>day_start</b> | Per specificare che i giorni non iniziano a mezzanotte, indicare un differimento come frazione di un giorno in <b>day_start</b> . Ad esempio, 0.125 per denotare 3:00 AM. In altre parole, per creare l'offset dividere l'ora di inizio per 24 ore. Ad esempio, per un giorno che inizia alle 7:00 del mattino, utilizzare la frazione 7/24. |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempi di funzioni

| Esempio                               | Risultato                            |
|---------------------------------------|--------------------------------------|
| dayend('01/25/2013 16:45:00')         | Restituisce: 01/24/2013 23:59:59. PM |
| dayend('01/25/2013 16:45:00', -1)     | Restituisce 01/24/2013 23:59:59. PM  |
| dayend('01/25/2013 16:45:00', 0, 0.5) | Restituisce: 01/26/2013 11:59:59. PM |

### Esempio 1 - Script base

#### Script di caricamento e risultati

#### Panoramica

Aprire Editor caricamento dati, quindi aggiungere lo script di caricamento in basso in una nuova scheda.

Lo script di caricamento contiene:

- Un set di dati contenente un elenco di date viene caricato in una tabella denominata "Calendario".
- La variabile di sistema DateFormat predefinita (MM/DD/YYYY).
- Un caricamento precedente per creare un campo aggiuntivo, denominato 'EOD\_timestamp', mediante la funzione dayend().

#### Script di caricamento

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

```
Calendar:
```

```
    Load
        date,
        dayend(date) as EOD_timestamp
    ;
```

```
Load
```

```
date
```

```
Inline
```

```
[
```

```
date
```

```
03/11/2022 1:47:15 AM
```

```
03/12/2022 4:34:58 AM
```

```
03/13/2022 5:15:55 AM
```

```
03/14/2022 9:25:14 AM
```

```
03/15/2022 10:06:54 AM
```

```
03/16/2022 10:44:42 AM
```

```
03/17/2022 11:33:30 AM
```

```
03/18/2022 12:58:14 PM
```

```
03/19/2022 4:23:12 PM
```

```
03/20/2022 6:42:15 PM
```

```
03/21/2022 7:41:16 PM
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- EOD\_timestamp

Tabella dei risultati

| date                   | EOD_timestamp         |
|------------------------|-----------------------|
| 03/11/2022 1:47:15 AM  | 3/11/2022 11:59:59 PM |
| 03/12/2022 4:34:58 AM  | 3/12/2022 11:59:59 PM |
| 03/13/2022 5:15:55 AM  | 3/13/2022 11:59:59 PM |
| 03/14/2022 9:25:14 AM  | 3/14/2022 11:59:59 PM |
| 03/15/2022 10:06:54 AM | 3/15/2022 11:59:59 PM |
| 03/16/2022 10:44:42 AM | 3/16/2022 11:59:59 PM |
| 03/17/2022 11:33:30 AM | 3/17/2022 11:59:59 PM |
| 03/18/2022 12:58:14 PM | 3/18/2022 11:59:59 PM |
| 03/19/2022 4:23:12 PM  | 3/19/2022 11:59:59 PM |
| 03/20/2022 6:42:15 PM  | 3/20/2022 11:59:59 PM |
| 03/21/2022 7:41:16 PM  | 3/21/2022 11:59:59 PM |

Come indicato nella tabella sopra, la fine del timestamp giorno viene generata per ciascuna data nel nostro set di dati. Il timestamp è nel formato della variabile di sistema `TimestampFormat M/D/YYYY h:mm:ss[.fff] TT`.

### Esempio 2 - period\_no

#### Script di caricamento e risultati

##### Panoramica

Aprire Editor caricamento dati, quindi aggiungere lo script di caricamento in basso in una nuova scheda.

Si caricherà un set di dati contenente le prenotazioni di servizio in una tabella denominata 'Servizi'.

Il set di dati include i seguenti campi:

- service\_id
- service\_date
- amount

Si creeranno due nuovi campi nella tabella:

- `deposit_due_date`: la data in cui si dovrebbe ricevere il deposito. Questa è la fine del giorno tre giorni prima del `service_date`.
- `final_payment_due_date`: la data in cui si dovrebbe ricevere il pagamento finale. Questa è la fine del giorno sette giorni dopo il `service_date`.

I due campi sopra sono creati in un caricamento precedente mediante la funzione `dayend()` e possono fornire i primi due parametri, `time` e `period_no`.

### Script di caricamento

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

```
Services:
```

```
Load
*,
dayend(service_date,-3) as deposit_due_date,
dayend(service_date,7) as final_payment_due_date
;
```

```
Load
service_id,
service_date,
amount
Inline
[
service_id, service_date, amount
1,03/11/2022 9:25:14 AM,231.24
2,03/12/2022 10:06:54 AM,567.28
3,03/13/2022 10:44:42 AM,364.28
4,03/14/2022 11:33:30 AM,575.76
5,03/15/2022 12:58:14 PM,638.68
6,03/16/2022 4:23:12 PM,785.38
7,03/17/2022 6:42:15 PM,967.46
8,03/18/2022 7:41:16 PM,287.67
9,03/19/2022 8:14:15 PM,764.45
10,03/20/2022 9:23:51 PM,875.43
11,03/21/2022 10:04:41 PM,957.35
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `service_date`
- `deposit_due_date`
- `final_payment_due_date`

Tabella dei risultati

| <code>service_date</code> | <code>deposit_due_date</code> | <code>final_payment_due_date</code> |
|---------------------------|-------------------------------|-------------------------------------|
| 03/11/2022 9:25:14 AM     | 3/8/2022 11:59:59 PM          | 3/18/2022 11:59:59 PM               |

| service_date           | deposit_due_date      | final_payment_due_date |
|------------------------|-----------------------|------------------------|
| 03/12/2022 10:06:54 AM | 3/9/2022 11:59:59 PM  | 3/19/2022 11:59:59 PM  |
| 03/13/2022 10:44:42 AM | 3/10/2022 11:59:59 PM | 3/20/2022 11:59:59 PM  |
| 03/14/2022 11:33:30 AM | 3/11/2022 11:59:59 PM | 3/21/2022 11:59:59 PM  |
| 03/15/2022 12:58:14 PM | 3/12/2022 11:59:59 PM | 3/22/2022 11:59:59 PM  |
| 03/16/2022 4:23:12 PM  | 3/13/2022 11:59:59 PM | 3/23/2022 11:59:59 PM  |
| 03/17/2022 6:42:15 PM  | 3/14/2022 11:59:59 PM | 3/24/2022 11:59:59 PM  |
| 03/18/2022 7:41:16 PM  | 3/15/2022 11:59:59 PM | 3/25/2022 11:59:59 PM  |
| 03/19/2022 8:14:15 PM  | 3/16/2022 11:59:59 PM | 3/26/2022 11:59:59 PM  |
| 03/20/2022 9:23:51 PM  | 3/17/2022 11:59:59 PM | 3/27/2022 11:59:59 PM  |
| 03/21/2022 10:04:41 PM | 3/18/2022 11:59:59 PM | 3/28/2022 11:59:59 PM  |

I valori dei nuovi campi sono in `TimestampFormat M/D/YYYY h:mm:ss[.fff] TT`. Essendo stata usata la funzione `dayend()`, i valori data e ora sono tutti all'ultimo millisecondo del giorno.

I valori della data di scadenza del deposito sono tre giorni prima della data del servizio perché il secondo argomento passato nella funzione `dayend()` è negativo.

I valori della data di scadenza del pagamento finale sono sette giorni dopo la data del servizio perché il secondo argomento passato nella funzione `dayend()` è positivo.

### Esempio 3 - day\_start script

#### Script di caricamento e risultati

##### Panoramica

Aprire Editor caricamento dati, quindi aggiungere lo script di caricamento in basso in una nuova scheda.

Il set di dati e lo scenario utilizzati in questo esempio sono gli stessi dell'esempio precedente.

Come nell'esempio precedente, si creeranno due nuovi campi:

- `deposit_due_date`: la data in cui si dovrebbe ricevere il deposito. Questa è la fine del giorno tre giorni prima del `service_date`.
- `final_payment_due_date`: la data in cui si dovrebbe ricevere il pagamento finale. Questa è la fine del giorno sette giorni dopo il `service_date`.

Tuttavia, la propria azienda desidera operare in base a una policy in cui il giorno lavorativo inizia alle 17:00 e termina alle 17:00 del giorno successivo. La propria azienda può quindi monitorare le transazioni che si verificano in tali ore di lavoro.

Per ottenere questi requisiti, i due campi sopra vengono creati in un caricamento precedente usando la funzione `dayend()` e impiegando tutti e tre gli argomenti, `time`, `period_no` e `day_start`.

### Script di caricamento

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

Services:

```
Load
    *,
    dayend(service_date,-3,17/24) as deposit_due_date,
    dayend(service_date,7,17/24) as final_payment_due_date
;
Load
service_id,
service_date,
amount
Inline
[
service_id, service_date,amount
1,03/11/2022 9:25:14 AM,231.24
2,03/12/2022 10:06:54 AM,567.28
3,03/13/2022 10:44:42 AM,364.28
4,03/14/2022 11:33:30 AM,575.76
5,03/15/2022 12:58:14 PM,638.68
6,03/16/2022 4:23:12 PM,785.38
7,03/17/2022 6:42:15 PM,967.46
8,03/18/2022 7:41:16 PM,287.67
9,03/19/2022 8:14:15 PM,764.45
10,03/20/2022 9:23:51 PM,875.43
11,03/21/2022 10:04:41 PM,957.35
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- service\_date
- deposit\_due\_date
- final\_payment\_due\_date

Tabella dei risultati

| service_date           | deposit_due_date     | final_payment_due_date |
|------------------------|----------------------|------------------------|
| 03/11/2022 9:25:14 AM  | 3/8/2022 4:59:59 PM  | 3/18/2022 4:59:59 PM   |
| 03/12/2022 10:06:54 AM | 3/9/2022 4:59:59 PM  | 3/19/2022 4:59:59 PM   |
| 03/13/2022 10:44:42 AM | 3/10/2022 4:59:59 PM | 3/20/2022 4:59:59 PM   |
| 03/14/2022 11:33:30 AM | 3/11/2022 4:59:59 PM | 3/21/2022 4:59:59 PM   |
| 03/15/2022 12:58:14 PM | 3/12/2022 4:59:59 PM | 3/22/2022 4:59:59 PM   |
| 03/16/2022 4:23:12 PM  | 3/13/2022 4:59:59 PM | 3/23/2022 4:59:59 PM   |

| service_date           | deposit_due_date     | final_payment_due_date |
|------------------------|----------------------|------------------------|
| 03/17/2022 6:42:15 PM  | 3/14/2022 4:59:59 PM | 3/24/2022 4:59:59 PM   |
| 03/18/2022 7:41:16 PM  | 3/15/2022 4:59:59 PM | 3/25/2022 4:59:59 PM   |
| 03/19/2022 8:14:15 PM  | 3/16/2022 4:59:59 PM | 3/26/2022 4:59:59 PM   |
| 03/20/2022 9:23:51 PM  | 3/17/2022 4:59:59 PM | 3/27/2022 4:59:59 PM   |
| 03/21/2022 10:04:41 PM | 3/18/2022 4:59:59 PM | 3/28/2022 4:59:59 PM   |

Mentre le date rimangono le stesse dell'Esempio 2, le date ora hanno un timestamp dell'ultimo millisecondo prima delle 17:00 a causa del fatto che il valore del terzo argomento, `day_start`, trasferito nella funzione `dayend()`, è 17/24.

### Esempio 4 - Esempio grafico

#### Script di caricamento ed espressione del grafico

##### Panoramica

Aprire Editor caricamento dati, quindi aggiungere lo script di caricamento in basso in una nuova scheda.

Il set di dati e lo scenario utilizzati in questo esempio corrispondono a quelli dei due esempi precedenti. La propria azienda desidera operare in base a una policy in cui il giorno lavorativo inizia alle 17:00 e termina alle 17:00 del giorno successivo.

Come nell'esempio precedente, si creeranno due nuovi campi:

- `deposit_due_date`: la data in cui si dovrebbe ricevere il deposito. Questa è la fine del giorno tre giorni prima del `service_date`.
- `final_payment_due_date`: la data in cui si dovrebbe ricevere il pagamento finale. Questa è la fine del giorno sette giorni dopo il `service_date`.

#### Script di caricamento

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

```
Services:
```

```
Load
```

```
service_id,
```

```
service_date,
```

```
amount
```

```
Inline
```

```
[
```

```
service_id, service_date, amount
```

```
1,03/11/2022 9:25:14 AM,231.24
```

```
2,03/12/2022 10:06:54 AM,567.28
```

```
3,03/13/2022 10:44:42 AM,364.28
```

```
4,03/14/2022 11:33:30 AM,575.76
```

```
5,03/15/2022 12:58:14 PM,638.68
```

```
6,03/16/2022 4:23:12 PM,785.38
```

```
7,03/17/2022 6:42:15 PM,967.46
8,03/18/2022 7:41:16 PM,287.67
9,03/19/2022 8:14:15 PM,764.45
10,03/20/2022 9:23:51 PM,875.43
11,03/21/2022 10:04:41 PM,957.35
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

service\_date.

Per creare il campo deposit\_due\_date, creare questa misura:

```
=dayend(service_date,-3,17/24).
```

Quindi, per creare il campo final\_payment\_due\_date, creare questa misura:

```
=dayend(service_date,7,17/24).
```

Tabella dei risultati

| service_date | =dayend(service_date,-3,17/24) | =dayend(service_date,7,17/24) |
|--------------|--------------------------------|-------------------------------|
| 03/11/2022   | 3/8/2022 16:59:59 PM           | 3/18/2022 16:59:59 PM         |
| 03/12/2022   | 3/9/2022 16:59:59 PM           | 3/19/2022 16:59:59 PM         |
| 03/13/2022   | 3/10/2022 16:59:59 PM          | 3/20/2022 16:59:59 PM         |
| 03/14/2022   | 3/11/2022 16:59:59 PM          | 3/21/2022 16:59:59 PM         |
| 03/15/2022   | 3/12/2022 16:59:59 PM          | 3/22/2022 16:59:59 PM         |
| 03/16/2022   | 3/13/2022 16:59:59 PM          | 3/23/2022 16:59:59 PM         |
| 03/17/2022   | 3/14/2022 16:59:59 PM          | 3/24/2022 16:59:59 PM         |
| 03/18/2022   | 3/15/2022 16:59:59 PM          | 3/25/2022 16:59:59 PM         |
| 03/19/2022   | 3/16/2022 16:59:59 PM          | 3/26/2022 16:59:59 PM         |
| 03/20/2022   | 3/17/2022 16:59:59 PM          | 3/27/2022 16:59:59 PM         |
| 03/21/2022   | 3/18/2022 16:59:59 PM          | 3/28/2022 16:59:59 PM         |

I valori dei nuovi campi sono in TimestampFormat M/D/YYYY h:mm:ss[.fff] TT. Essendo stata usata la funzione dayend(), i valori data e ora sono tutti all'ultimo millisecondo del giorno.

I valori della data di pagamento del deposito sono tre giorni prima della data del servizio perché il secondo argomento passato nella funzione dayend() è negativo.

I valori della data di scadenza del pagamento finale sono sette giorni dopo la data del servizio perché il secondo argomento passato nella funzione dayend() è positivo.

Le date hanno un timestamp dell'ultimo millisecondo prima delle 17:00 a causa del fatto che il valore del terzo argomento, day\_start, trasferito nella funzione dayend(), è 17/24.

## Argomenti

| Argomento        | Descrizione                                                                                                                                                                                                                                                                                     |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>time</b>      | L'indicatore temporale da valutare.                                                                                                                                                                                                                                                             |
| <b>period_no</b> | <b>period_no</b> è un numero intero o un'espressione la cui risoluzione è un numero intero, in cui il valore 0 indica il giorno che contiene il valore <b>time</b> . I valori negativi di <b>period_no</b> indicano i giorni precedenti, mentre i valori positivi indicano i giorni successivi. |
| <b>day_start</b> | Per specificare che i giorni non iniziano a mezzanotte, indicare un differimento come frazione di un giorno in <b>day_start</b> . Ad esempio, 0.125 per denotare 3:00 AM.                                                                                                                       |

## daylightsaving

Restituisce il valore di regolazione attuale per l'ora legale, come definito in Windows.

Sintassi:

```
DaylightSaving( )
```

Tipo di dati restituiti: duale

Esempio:

```
daylightsaving( )
```

## dayname

Questa funzione restituisce un valore che mostra la data con un valore numerico sottostante corrispondente a un indicatore temporale recante il primo millisecondo del giorno contenente **time**.

Sintassi:

```
DayName (time[, period_no [, day_start]])
```

Tipo di dati restituiti: duale

Argomenti:

## Argomenti

| Argomento        | Descrizione                                                                                                                                                                                                                                                                                     |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>time</b>      | L'indicatore temporale da valutare.                                                                                                                                                                                                                                                             |
| <b>period_no</b> | <b>period_no</b> è un numero intero o un'espressione la cui risoluzione è un numero intero, in cui il valore 0 indica il giorno che contiene il valore <b>time</b> . I valori negativi di <b>period_no</b> indicano i giorni precedenti, mentre i valori positivi indicano i giorni successivi. |
| <b>day_start</b> | Per specificare che i giorni non iniziano a mezzanotte, indicare un differimento come frazione di un giorno in <b>day_start</b> . Ad esempio, 0.125 per denotare 3:00 AM.                                                                                                                       |

### Esempi e risultati:

In questi esempi viene utilizzato il formato della data **DD/MM/YYYY**. Il formato della data viene specificato nell'istruzione **SET DateFormat** nella parte superiore dello script di caricamento dei dati. Modificare il formato negli esempi in base alle proprie necessità.

Esempi di script

| Esempio                                              | Risultato                                                                                                                                                              |
|------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>dayname('25/01/2013 16:45:00')</code>          | Restituisce 25/01/2013.                                                                                                                                                |
| <code>dayname('25/01/2013 16:45:00', -1)</code>      | Restituisce 24/01/2013.                                                                                                                                                |
| <code>dayname('25/01/2013 16:45:00', 0, 0.5 )</code> | Restituisce 25/01/2013.<br><br>La visualizzazione dell'indicatore temporale completo mostra che il valore numerico sottostante corrisponde a '25/01/2013 12:00:00.000. |

### Esempio:

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

In questo esempio il nome del giorno viene creato a partire dall'indicatore temporale che contrassegna l'inizio del giorno dopo ciascuna data della fattura nella tabella.

TempTable:

```
LOAD RecNo() as InvID, * Inline [
```

```
InvDate
```

```
28/03/2012
```

```
10/12/2012
```

```
5/2/2013
```

```
31/3/2013
```

```
19/5/2013
```

```
15/9/2013
```

```
11/12/2013
```

```
2/3/2014
```

```
14/5/2014
```

```
13/6/2014
```

7/7/2014

4/8/2014

];

InvoiceData:

LOAD \*,

DayName(InvDate, 1) AS DName

Resident TempTable;

Drop table TempTable;

La tabella risultante contiene le date originali e una colonna con il valore restituito della funzione dayname (). È possibile visualizzare l'indicatore temporale completo specificando la formattazione nel pannello delle proprietà.

Tabella dei risultati

| InvDate    | DName               |
|------------|---------------------|
| 28/03/2012 | 29/03/2012 00:00:00 |
| 10/12/2012 | 11/12/2012 00:00:00 |
| 5/2/2013   | 07/02/2013 00:00:00 |
| 31/3/2013  | 01/04/2013 00:00:00 |
| 19/5/2013  | 20/05/2013 00:00:00 |
| 15/9/2013  | 16/09/2013 00:00:00 |
| 11/12/2013 | 12/12/2013 00:00:00 |
| 2/3/2014   | 03/03/2014 00:00:00 |
| 14/5/2014  | 15/05/2014 00:00:00 |
| 13/6/2014  | 14/06/2014 00:00:00 |
| 7/7/2014   | 08/07/2014 00:00:00 |
| 4/8/2014   | 05/08/2014 00:00:00 |

### daynumberofquarter

Questa funzione calcola il numero del giorno del trimestre a cui è stato assegnato un indicatore temporale. Questa funzione viene utilizzata al momento di creare un Calendario principale.

**Sintassi:**

```
DayNumberOfQuarter(timestamp[,start_month])
```

**Tipo di dati restituiti:** numero intero

#### Argomenti

| Argomento          | Descrizione                                                                                                                                                                                                                                                                               |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>timestamp</b>   | La data o la data e ora da valutare.                                                                                                                                                                                                                                                      |
| <b>start_month</b> | Se si specifica un valore <b>start_month</b> compreso tra 2 e 12 (1 se omissso), l'inizio dell'anno potrà essere spostato in avanti sul primo giorno di qualsiasi mese. Se, ad esempio, si intende utilizzare un anno fiscale che inizi il 1° marzo, specificare <b>start_month = 3</b> . |

In questi esempi viene utilizzato il formato della data **DD/MM/YYYY**. Il formato della data viene specificato nell'istruzione **SET DateFormat** nella parte superiore dello script di caricamento dei dati. Modificare il formato negli esempi in base alle proprie necessità.

#### Esempi di funzioni

| Esempio                                          | Risultato                                                                                                                                                                                                                                                |
|--------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>DayNumberOfQuarter('12/09/2014')</code>    | Restituisce 74, il numero del giorno del trimestre attuale.                                                                                                                                                                                              |
| <code>DayNumberOfQuarter('12/09/2014', 3)</code> | Restituisce 12, il numero del giorno del trimestre attuale. In questo caso, il primo trimestre inizia a marzo (in quanto <b>start_month</b> è specificato come 3). Ciò significa che il trimestre attuale è il terzo trimestre, iniziato il 1 settembre. |

## Esempio 1 - Gennaio inizio dell'anno (script)

Script di caricamento e risultati

### Panoramica

Aprire Editor caricamento dati, quindi aggiungere lo script di caricamento in basso in una nuova scheda.

Lo script di caricamento contiene:

- Un set di dati semplice contenente un elenco di date, caricato in una tabella denominata `calendar`. È utilizzata la variabile di sistema predefinita `DateFormat MM/GG/AAAA`.
- Un caricamento precedente che crea un campo aggiuntivo, denominato `DayNrQtr`, mediante la funzione `DayNumberOfQuarter()`.

Oltre alla data, non viene fornito alla funzione alcun parametro aggiuntivo.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
calendar:
Load
```

```
        date,
        DayNumberOfQuarter(date) as DayNrQtr
    ;
Load
date
Inline
[
date
01/01/2022
01/10/2022
01/31/2022
02/01/2022
02/10/2022
02/28/2022
03/01/2022
03/31/2022
04/01/2022
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- daynrqtr

Tabella dei risultati

| data       | daynrqtr |
|------------|----------|
| 01/01/2022 | 1        |
| 01/10/2022 | 10       |
| 01/31/2022 | 31       |
| 02/01/2022 | 32       |
| 02/10/2022 | 41       |
| 02/28/2022 | 59       |
| 03/01/2022 | 61       |
| 03/31/2022 | 91       |
| 04/01/2022 | 1        |

Il primo giorno dell'anno era il 1° di gennaio dato che nessun secondo argomento era stato trasferito nella funzione DayNumberOfQuarter().

Il 1° gennaio è il primo giorno del trimestre, mentre il 1° febbraio è il 32esimo giorno del trimestre. Il 31 marzo è il 91esimo e ultimo giorno del trimestre, mentre il 1° aprile è il primo giorno del 2° trimestre.

### Esempio 2 - Febbraio inizio dell'anno (script)

Script di caricamento e risultati

#### Panoramica

Aprire Editor caricamento dati, quindi aggiungere lo script di caricamento in basso in una nuova scheda.

Lo script di caricamento contiene:

- Lo stesso set di dati dal primo esempio.
- È utilizzata la variabile di sistema predefinita `DateFormat` `MM/GG/AAAA`.
- Un argomento `start_month` che inizia il 1° febbraio. Ciò imposta l'anno finanziario al 1° febbraio.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Calendar:
```

```
Load
```

```
    date,  
    DayNumberOfQuarter(date,2) as DayNrQtr  
    ;
```

```
Load
```

```
date
```

```
Inline
```

```
[
```

```
date
```

```
01/01/2022
```

```
01/10/2022
```

```
01/31/2022
```

```
02/01/2022
```

```
02/10/2022
```

```
02/28/2022
```

```
03/01/2022
```

```
03/31/2022
```

```
04/01/2022
```

```
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `date`
- `daynrqtr`

Tabella dei risultati

| data       | daynrqtr |
|------------|----------|
| 01/01/2022 | 62       |

| data       | daynrqtr |
|------------|----------|
| 01/10/2022 | 71       |
| 01/31/2022 | 92       |
| 02/01/2022 | 1        |
| 02/10/2022 | 10       |
| 02/28/2022 | 28       |
| 03/01/2022 | 30       |
| 03/31/2022 | 60       |
| 04/01/2022 | 61       |

Il primo giorno dell'anno è il 1° di febbraio, dato che il secondo argomento trasferito nella funzione `DayNumberOfQuarter()` era 2.

Il primo trimestre dell'anno va da febbraio ad aprile, mentre il quarto trimestre va da novembre a gennaio. Questo viene mostrato nella tabella dei risultati, dove il 1° febbraio è il primo giorno del trimestre mentre il 31 gennaio è il 92esimo e ultimo giorno del trimestre.

### Esempio 3 - Gennaio inizio dell'anno (grafico)

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire Editor caricamento dati, quindi aggiungere lo script di caricamento in basso in una nuova scheda.

Lo script di caricamento contiene:

- Lo stesso set di dati dal primo esempio.
- È utilizzata la variabile di sistema predefinita `DateFormat MM/GG/AAAA`.

Tuttavia, in questo esempio, il set di dati invariato viene caricato nell'applicazione. Il valore del giorno del trimestre viene calcolato mediante una misura in un oggetto del grafico.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Calendar:
```

```
Load
```

```
date
```

```
Inline
```

```
[
```

```
date
```

```
01/01/2022
```

```
01/10/2022
```

```
01/31/2022
```

```
02/01/2022  
02/10/2022  
02/28/2022  
03/01/2022  
03/31/2022  
04/01/2022  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: date.

Creare la seguente misura:

```
=daynumberofquarter(date)
```

Tabella dei risultati

| data       | =daynumberofquarter(date) |
|------------|---------------------------|
| 01/01/2022 | 1                         |
| 01/10/2022 | 10                        |
| 01/31/2022 | 31                        |
| 02/01/2022 | 32                        |
| 02/10/2022 | 41                        |
| 02/28/2022 | 59                        |
| 03/01/2022 | 61                        |
| 03/31/2022 | 91                        |
| 04/01/2022 | 1                         |

Il primo giorno dell'anno era il 1° di gennaio dato che nessun secondo argomento era stato trasferito nella funzione `DayNumberOfQuarter()`.

Il 1° gennaio è il primo giorno del trimestre, mentre il 1° febbraio è il 32esimo giorno del trimestre. Il 31 marzo è il 91esimo e ultimo giorno del trimestre, mentre il 1° aprile è il primo giorno del 2° trimestre.

### Esempio 4 - Febbraio inizio dell'anno (grafico)

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire Editor caricamento dati, quindi aggiungere lo script di caricamento in basso in una nuova scheda.

Lo script di caricamento contiene:

- Lo stesso set di dati dal primo esempio.
- È utilizzata la variabile di sistema predefinita `DateFormat` `MM/GG/AAAA`.
- L'anno finanziario va dal 1° febbraio al 31 gennaio.

Tuttavia, in questo esempio, il set di dati invariato viene caricato nell'applicazione. Il valore del giorno del trimestre viene calcolato mediante una misura in un oggetto del grafico.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Calendar:
Load
date
inline
[
date
01/01/2022
01/10/2022
01/31/2022
02/01/2022
02/10/2022
02/28/2022
03/01/2022
03/31/2022
04/01/2022
];
```

### Oggetto grafico

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: `date`.

Creare la seguente misura:

```
=daynumberofquarter(date,2)
```

### Risultati

Tabella dei risultati

| data       | =daynumberofquarter(date,2) |
|------------|-----------------------------|
| 01/01/2022 | 62                          |
| 01/10/2022 | 71                          |
| 01/31/2022 | 92                          |
| 02/01/2022 | 1                           |
| 02/10/2022 | 10                          |

| data       | =daynumberofquarter(date,2) |
|------------|-----------------------------|
| 02/28/2022 | 28                          |
| 03/01/2022 | 30                          |
| 03/31/2022 | 60                          |
| 04/01/2022 | 61                          |

Il primo giorno dell'anno è il 1° di gennaio, dato che il secondo argomento trasferito nella funzione `DayNumberOfQuarter()` era 2.

Il primo trimestre dell'anno va da febbraio ad aprile, mentre il quarto trimestre va da novembre a gennaio. Questo viene evidenziato nella tabella dei risultati, dove il 1° febbraio è il primo giorno del trimestre mentre il 31 gennaio è il 92esimo e ultimo giorno del trimestre.

## daynumberofyear

Questa funzione calcola il numero del giorno dell'anno a cui è stato assegnato un indicatore temporale. Il calcolo viene eseguito partendo dal primo millisecondo del primo giorno dell'anno, tuttavia il primo mese può essere differito.

### Sintassi:

```
DayNumberOfYear (timestamp[, start_month])
```

**Tipo di dati restituiti:** numero intero

### Argomenti

| Argomento          | Descrizione                                                                                                                                                                                                                                                                              |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>timestamp</b>   | La data o la data e ora da valutare.                                                                                                                                                                                                                                                     |
| <b>start_month</b> | Se si specifica un valore <b>start_month</b> compreso tra 2 e 12 (1 se omissso), l'inizio dell'anno potrà essere spostato in avanti sul primo giorno di qualsiasi mese. Se, ad esempio, si intende utilizzare un anno fiscale che inizi il 1° marzo, specificare <b>start_month</b> = 3. |

In questi esempi viene utilizzato il formato della data **DD/MM/YYYY**. Il formato della data viene specificato nell'istruzione **SET DateFormat** nella parte superiore dello script di caricamento dei dati. Modificare il formato negli esempi in base alle proprie necessità.

### Esempi di funzioni

| Esempio                                         | Risultato                                                                     |
|-------------------------------------------------|-------------------------------------------------------------------------------|
| <code>DayNumberOfYear( '12/09/2014' )</code>    | Restituisce 256, il numero del giorno conteggiato dal primo giorno dell'anno. |
| <code>DayNumberOfYear( '12/09/2014', 3 )</code> | Restituisce 196, il numero del giorno conteggiato a partire dal 1° marzo.     |

### Esempio 1 - Gennaio inizio dell'anno (script)

Script di caricamento e risultati

#### Panoramica

Aprire Editor caricamento dati, quindi aggiungere lo script di caricamento in basso in una nuova scheda.

Lo script di caricamento contiene:

- Un set di dati semplice contenente un elenco di date, caricato in una tabella denominata `calendar`. È utilizzata la variabile di sistema predefinita `DateFormat MM/GG/AAAA`.
- Un caricamento precedente che crea un campo aggiuntivo, denominato `daynryear`, mediante la funzione `DayNumberOfYear()`.

Oltre alla data, non viene fornito alla funzione alcun parametro aggiuntivo.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
calendar:
```

```
Load
```

```
    date,  
    DayNumberOfYear(date) as daynryear  
;
```

```
Load
```

```
date
```

```
Inline
```

```
[
```

```
date
```

```
01/01/2022
```

```
01/10/2022
```

```
01/31/2022
```

```
02/01/2022
```

```
02/10/2022
```

```
06/30/2022
```

```
07/26/2022
```

```
10/31/2022
```

```
11/01/2022
```

```
12/31/2022
```

```
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `date`
- `daynryear`

Tabella dei risultati

| data       | daynryear |
|------------|-----------|
| 01/01/2022 | 1         |
| 01/10/2022 | 10        |
| 01/31/2022 | 31        |
| 02/01/2022 | 32        |
| 02/10/2022 | 41        |
| 06/30/2022 | 182       |
| 07/26/2022 | 208       |
| 10/31/2022 | 305       |
| 11/01/2022 | 306       |
| 12/31/2022 | 366       |

Il primo giorno dell'anno era il 1° di gennaio dato che nessun secondo argomento era stato trasferito nella funzione `DayNumberOfYear()`.

Il 1° gennaio è il primo giorno del trimestre, mentre il 1° febbraio è il 32esimo giorno dell'anno. Il 30 giugno è il 182esimo giorno dell'anno mentre il 31 dicembre è il 366esimo e ultimo giorno dell'anno.

### Esempio 2 - Novembre inizio dell'anno (script)

Script di caricamento e risultati

#### Panoramica

Aprire Editor caricamento dati, quindi aggiungere lo script di caricamento in basso in una nuova scheda.

Lo script di caricamento contiene:

- Lo stesso set di dati dal primo esempio.
- È utilizzata la variabile di sistema predefinita `DateFormat MM/GG/AAAA`
- Un argomento `start_month` che inizia il 1° novembre. Ciò imposta l'anno finanziario al 1° novembre.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Calendar:
```

```
Load
```

```
    date,  
    DayNumberOfYear(date,11) as daynryear  
    ;
```

```
Load
```

```
date
```

```
inline  
[  
date  
01/01/2022  
01/10/2022  
01/31/2022  
02/01/2022  
02/10/2022  
06/30/2022  
07/26/2022  
10/31/2022  
11/01/2022  
12/31/2022  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- daynryear

Tabella dei risultati

| data       | daynryear |
|------------|-----------|
| 01/01/2022 | 62        |
| 01/10/2022 | 71        |
| 01/31/2022 | 92        |
| 02/01/2022 | 93        |
| 02/10/2022 | 102       |
| 06/30/2022 | 243       |
| 07/26/2022 | 269       |
| 10/31/2022 | 366       |
| 11/01/2022 | 1         |
| 12/31/2022 | 61        |

Il primo giorno dell'anno è il 1° di novembre, dato che il secondo argomento trasferito nella funzione `DayNumberOfYear()` era 11.

Il 1° gennaio è il primo giorno del trimestre, mentre il 1° febbraio è il 32esimo giorno dell'anno. Il 30 giugno è il 182esimo giorno dell'anno mentre il 31 dicembre è il 366esimo e ultimo giorno dell'anno.

### Esempio 3 - Gennaio inizio dell'anno (grafico)

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire Editor caricamento dati, quindi aggiungere lo script di caricamento in basso in una nuova scheda.

Lo script di caricamento contiene:

- Lo stesso set di dati dal primo esempio.
- È utilizzata la variabile di sistema predefinita `DateFormat` `MM/GG/AAAA`.

Tuttavia, in questo esempio, il set di dati invariato viene caricato nell'applicazione. Il valore del giorno del trimestre viene calcolato mediante una misura in un oggetto del grafico.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Calendar:
```

```
Load
```

```
date
```

```
Inline
```

```
[
```

```
date
```

```
01/01/2022
```

```
01/10/2022
```

```
01/31/2022
```

```
02/01/2022
```

```
02/10/2022
```

```
06/30/2022
```

```
07/26/2022
```

```
10/31/2022
```

```
11/01/2022
```

```
12/31/2022
```

```
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: `date`.

Creare la seguente misura:

```
=daynumberofyear(date)
```

Tabella dei risultati

| data       | =daynumberofyear(date) |
|------------|------------------------|
| 01/01/2022 | 1                      |

| data       | =daynumberofyear(date) |
|------------|------------------------|
| 01/10/2022 | 10                     |
| 01/31/2022 | 31                     |
| 02/01/2022 | 32                     |
| 02/10/2022 | 41                     |
| 06/30/2022 | 182                    |
| 07/26/2022 | 208                    |
| 10/31/2022 | 305                    |
| 11/01/2022 | 306                    |
| 12/31/2022 | 366                    |

Il primo giorno dell'anno era il 1° di gennaio dato che nessun secondo argomento era stato trasferito nella funzione `DayNumberOfYear()`.

Il 1° gennaio è il primo giorno dell'anno, mentre il 1° febbraio è il 32esimo giorno dell'anno. Il 30 giugno è il 182esimo giorno dell'anno mentre il 31 dicembre è il 366esimo e ultimo giorno dell'anno.

### Esempio 4 - Novembre inizio dell'anno (grafico)

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire Editor caricamento dati, quindi aggiungere lo script di caricamento in basso in una nuova scheda.

Lo script di caricamento contiene:

- Lo stesso set di dati dal primo esempio.
- È utilizzata la variabile di sistema predefinita `DateFormat MM/GG/AAAA`.
- L'anno finanziario va dal 1° novembre al 31 ottobre.

Tuttavia, in questo esempio, il set di dati invariato viene caricato nell'applicazione. Il valore del giorno dell'anno viene calcolato mediante una misura in un oggetto del grafico.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
Calendar:
Load
date
Inline
[
date
01/01/2022
01/10/2022
```

```
01/31/2022  
02/01/2022  
02/10/2022  
06/30/2022  
07/26/2022  
10/31/2022  
11/01/2022  
12/31/2022  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: date.

Creare la seguente misura:

```
=daynumberofyear(date)
```

Tabella dei risultati

| data       | =daynumberofyear(date,11) |
|------------|---------------------------|
| 01/01/2022 | 62                        |
| 01/10/2022 | 71                        |
| 01/31/2022 | 92                        |
| 02/01/2022 | 93                        |
| 02/10/2022 | 102                       |
| 06/30/2022 | 243                       |
| 07/26/2022 | 269                       |
| 10/31/2022 | 366                       |
| 11/01/2022 | 1                         |
| 12/31/2022 | 61                        |

Il primo giorno dell'anno è il 1° di novembre, dato che il secondo argomento trasferito nella funzione DayNumberOfYear() era 11.

L'anno finanziario va da novembre a ottobre. Questo viene mostrato nella tabella dei risultati, dove il 1° novembre è il primo giorno dell'anno mentre il 31 ottobre è il 366esimo e ultimo giorno dell'anno.

### daystart

Questa funzione restituisce un valore corrispondente a un indicatore temporale con il primo millisecondo del giorno contenuto nell'argomento **time**. Il formato di output predefinito sarà il formato **TimestampFormat** impostato nello script.

#### Sintassi:

```
DayStart(time[, [period_no[, day_start]])
```

Tipo di dati restituiti: duale

#### Argomenti

| Argomento        | Descrizione                                                                                                                                                                                                                                                                                                                                  |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>time</b>      | L'indicatore temporale da valutare.                                                                                                                                                                                                                                                                                                          |
| <b>period_no</b> | <b>period_no</b> è un numero intero o un'espressione la cui risoluzione è un numero intero, in cui il valore 0 indica il giorno che contiene il valore <b>time</b> . I valori negativi di <b>period_no</b> indicano i giorni precedenti, mentre i valori positivi indicano i giorni successivi.                                              |
| <b>day_start</b> | Per specificare che i giorni non iniziano a mezzanotte, indicare un differimento come frazione di un giorno in <b>day_start</b> . Ad esempio, 0.125 per denotare 3:00 AM. In altre parole, per creare l'offset dividere l'ora di inizio per 24 ore. Ad esempio, per un giorno che inizia alle 7:00 del mattino, utilizzare la frazione 7/24. |

### Casi di utilizzo

La funzione `daystart()` viene comunemente utilizzata come parte di un'espressione quando l'utente desidera che il calcolo utilizzi la frazione del giorno trascorsa finora. Ad esempio, può essere utilizzata per calcolare il totale dei salari guadagnati dai dipendenti nella giornata fino a quel momento.

Questi esempi utilizzano il formato timestamp 'M/D/YYYY h:mm:ss[.fff] TT'. Il formato timestamp viene specificato nell'istruzione `SET Timestamp` sopra lo script di caricamento dei dati. Modificare il formato negli esempi in base alle proprie necessità.

#### Esempi di funzioni

| Esempio                                              | Risultato                          |
|------------------------------------------------------|------------------------------------|
| <code>daystart('01/25/2013 4:45:00 PM')</code>       | Restituisce 1/25/2013 12:00:00 AM. |
| <code>daystart('1/25/2013 4:45:00 PM', -1)</code>    | Restituisce 1/24/2013 12:00:00 AM. |
| <code>daystart('1/25/2013 16:45:00', 0, 0.5 )</code> | Restituisce 1/25/2013 12:00:00 PM. |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - Esempio semplice

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati semplice contenente un elenco di date, caricato in una tabella denominata `calendar`.
- Viene utilizzata la variabile di sistema `TimestampFormat` predefinita (`(M/D/YYYY h:mm:ss[.fff] TT)`).
- Un caricamento precedente che crea un campo aggiuntivo, denominato `SOD_timestamp`, mediante la funzione `daystart()`.

Oltre alla data, non viene fornito alla funzione alcun parametro aggiuntivo.

#### Script di caricamento

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

```
calendar:
    Load
        date,
        daystart(date) as SOD_timestamp
    ;
Load
date
Inline
[
date
03/11/2022 1:47:15 AM
03/12/2022 4:34:58 AM
03/13/2022 5:15:55 AM
03/14/2022 9:25:14 AM
03/15/2022 10:06:54 AM
03/16/2022 10:44:42 AM
03/17/2022 11:33:30 AM
03/18/2022 12:58:14 PM
03/19/2022 4:23:12 PM
03/20/2022 6:42:15 PM
03/21/2022 7:41:16 PM
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `date`
- `SOD_timestamp`

Tabella dei risultati

| date                   | SOD_timestamp         |
|------------------------|-----------------------|
| 03/11/2022 1:47:15 AM  | 3/11/2022 12:00:00 AM |
| 03/12/2022 4:34:58 AM  | 12/3/2022 00:00:00    |
| 03/13/2022 5:15:55 AM  | 13/03/2022 00:00:00   |
| 03/14/2022 9:25:14 AM  | 3/14/2022 12:00:00 AM |
| 03/15/2022 10:06:54 AM | 15/03/2022 00:00:00   |
| 03/16/2022 10:44:42 AM | 16/03/2022 00:00:00   |
| 03/17/2022 11:33:30 AM | 3/17/2022 12:00:00 AM |
| 03/18/2022 12:58:14 PM | 3/18/2022 12:00:00 AM |
| 03/19/2022 4:23:12 PM  | 19/3/2022 00:00:00    |
| 03/20/2022 6:42:15 PM  | 20/03/2022 00:00:00   |
| 03/21/2022 7:41:16 PM  | 3/21/2022 12:00:00 AM |

Come appare nella tabella sopra, la fine del timestamp giorno viene generata per ciascuna data nel nostro set di dati. Il timestamp è nel formato della variabile di sistema `TimestampFormat M/D/YYYY h:mm:ss[.fff] TT`.

### Esempio 2 - period\_no

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente le multe per divieto di sosta, che viene caricato in una tabella denominata `Fines`. Il set di dati include i seguenti campi:
  - `id`
  - `due_date`
  - `number_plate`
  - `amount`
- Un caricamento precedente che utilizza la funzione `daystart()` e fornisce tutti e tre i parametri: `time`, `period_no` e `day_start`. Questo caricamento crea i seguenti due nuovi campi data:
  - Un campo di data `early_repayment_period`, che inizia sette giorni prima della scadenza del pagamento.
  - Un campo di data `late_penalty_period`, che inizia 14 giorni dopo la scadenza del pagamento.

### Script di caricamento

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';

Fines:
    Load
        *,
        daystart(due_date,-7) as early_repayment_period,
        daystart(due_date,14) as late_penalty_period
    ;

Load
*
Inline
[
id, due_date, number_plate, amount
1,02/11/2022, 573RJG,50.00
2,03/25/2022, SC41854,50.00
3,04/14/2022, 8EHZ378,50.00
4,06/28/2022, 8HSS198,50.00
5,08/15/2022, 1221665,50.00
6,11/16/2022, EAK473,50.00
7,01/17/2023, KD6822,50.00
8,03/22/2023, 1GGLB,50.00
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- due\_date
- early\_repayment\_period
- late\_penalty\_period

Tabella dei risultati

| due_date               | early_repayment_period | late_penalty_period    |
|------------------------|------------------------|------------------------|
| 02/11/2022 9:25:14 AM  | 2/4/2022 12:00:00 AM   | 2/25/2022 12:00:00 AM  |
| 03/25/2022 10:06:54 AM | 3/18/2022 12:00:00 AM  | 4/8/2022 12:00:00 AM   |
| 04/14/2022 10:44:42 AM | 4/7/2022 12:00:00 AM   | 4/28/2022 12:00:00 AM  |
| 06/28/2022 11:33:30 AM | 21/06/2022 00:00:00    | 7/12/2022 12:00:00 AM  |
| 08/15/2022 12:58:14 PM | 8/8/2022 12:00:00 AM   | 8/29/2022 12:00:00 AM  |
| 11/16/2022 4:23:12 PM  | 11/9/2022 12:00:00 AM  | 11/30/2022 12:00:00 AM |
| 01/17/2023 6:42:15 PM  | 1/10/2023 12:00:00 AM  | 1/31/2023 12:00:00 AM  |
| 03/22/2023 7:41:16 PM  | 3/15/2023 12:00:00 AM  | 4/5/2023 12:00:00 AM   |

I valori dei nuovi campi sono in TimestampFormat M/DD/YYYY tt. Essendo stata usata la funzione daystart (), i valori timestamp sono tutti al primo millisecondo del giorno.

I valori del periodo di rimborso anticipato sono sette giorni prima della data di scadenza, in quanto il secondo argomento passato nella funzione `daystart()` è negativo.

I valori del periodo di rimborso tardivo sono 14 giorni dopo la data di scadenza, in quanto il secondo argomento passato nella funzione `daystart()` è positivo.

### Esempio 3 - day\_start

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati e lo stesso scenario del precedente esempio.
- Lo stesso caricamento precedente dell'esempio precedente.

In questo esempio, abbiamo impostato la giornata lavorativa in modo che inizi e termini alle 7:00 del mattino di ogni giorno.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Fines:
```

```
    Load
```

```
        *,
```

```
        daystart(due_date,-7,7/24) as early_repayment_period,
```

```
        daystart(due_date,14, 7/24) as late_penalty_period
```

```
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id, due_date, number_plate,amount
```

```
1,02/11/2022, 573RJG,50.00
```

```
2,03/25/2022, SC41854,50.00
```

```
3,04/14/2022, 8EHZ378,50.00
```

```
4,06/28/2022, 8HSS198,50.00
```

```
5,08/15/2022, 1221665,50.00
```

```
6,11/16/2022, EAK473,50.00
```

```
7,01/17/2023, KD6822,50.00
```

```
8,03/22/2023, 1GGLB,50.00
```

```
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- due\_date
- early\_repayment\_period
- late\_penalty\_period

Tabella dei risultati

| due_date   | early_repayment_period | late_penalty_period   |
|------------|------------------------|-----------------------|
| 02/11/2022 | 2/3/2022 7:00:00 AM    | 2/24/2022 7:00:00 AM  |
| 03/25/2022 | 3/17/2022 7:00:00 AM   | 4/7/2022 7:00:00 AM   |
| 04/14/2022 | 4/6/2022 7:00:00 AM    | 4/27/2022 7:00:00 AM  |
| 06/28/2022 | 6/20/2022 7:00:00 AM   | 7/11/2022 7:00:00 AM  |
| 08/15/2022 | 8/7/2022 7:00:00 AM    | 8/28/2022 7:00:00 AM  |
| 11/16/2022 | 11/8/2022 7:00:00 AM   | 11/29/2022 7:00:00 AM |
| 01/17/2023 | 1/9/2023 7:00:00 AM    | 1/30/2023 7:00:00 AM  |
| 03/22/2023 | 3/14/2023 7:00:00 AM   | 4/4/2023 7:00:00 AM   |

Le date ora hanno un timestamp di 7:00 AM perché il valore dell'argomento day\_start passato alla funzione daystart() è 7/24. In questo modo si imposta l'inizio del giorno alle 7:00 AM.

Poiché il campo due\_date non ha un timestamp, viene trattato come le 12:00 AM, che quindi fa ancora parte del giorno precedente, dato che i giorni iniziano e finiscono alle 7:00 del mattino. Pertanto, il periodo di rimborso anticipato per una multa in scadenza l'11 febbraio inizia il 3 febbraio alle ore 7:00.

### Esempio 4 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor di caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Questo esempio utilizza lo stesso set di dati e lo stesso scenario dell'esempio precedente.

Tuttavia, solo la tabella Fines originale viene caricata nell'applicazione, mentre i due valori aggiuntivi delle scadenze vengono calcolati in un oggetto grafico.

#### Script di caricamento

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

```
Fines:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id, due_date, numer_plate, amount
```

```
1,02/11/2022 9:25:14 AM, 573RJG,50.00
2,03/25/2022 10:06:54 AM, SC41854,50.00
3,04/14/2022 10:44:42 AM, 8EHZ378,50.00
4,06/28/2022 11:33:30 AM, 8HSS198,50.00
5,08/15/2022 12:58:14 PM, 1221665,50.00
6,11/16/2022 4:23:12 PM, EAK473,50.00
7,01/17/2023 6:42:15 PM, KD6822,50.00
8,03/22/2023 7:41:16 PM, 1GGLB,50.00
];
```

### Risultati

Procedere come indicato di seguito:

1. Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: `due_date`.
2. Per creare il campo `early_repayment_period`, creare la seguente misura:  
`=daystart(due_date,-7,7/24)`
3. Per creare il campo `late_penalty_period`, creare la seguente misura:  
`=daystart(due_date,14,7/24)`

Tabella dei risultati

| <code>due_date</code>  | <code>=daystart(due_date,-7,7/24)</code> | <code>=daystart(due_date,14,7/24)</code> |
|------------------------|------------------------------------------|------------------------------------------|
| 02/11/2022 9:25:14 AM  | 2/4/2022 7:00:00 AM                      | 2/25/2022 7:00:00 AM                     |
| 03/25/2022 10:06:54 AM | 3/18/2022 7:00:00 AM                     | 4/8/2022 7:00:00 AM                      |
| 04/14/2022 10:44:42 AM | 4/7/2022 7:00:00 AM                      | 4/28/2022 7:00:00 AM                     |
| 06/28/2022 11:33:30 AM | 6/21/2022 7:00:00 AM                     | 7/12/2022 7:00:00 AM                     |
| 08/15/2022 12:58:14 PM | 8/8/2022 7:00:00 AM                      | 8/29/2022 7:00:00 AM                     |
| 11/16/2022 4:23:12 PM  | 11/9/2022 7:00:00 AM                     | 11/30/2022 7:00:00 AM                    |
| 01/17/2023 6:42:15 PM  | 1/10/2023 7:00:00 AM                     | 1/31/2023 7:00:00 AM                     |
| 03/22/2023 7:41:16 PM  | 3/15/2023 7:00:00 AM                     | 4/5/2023 7:00:00 AM                      |

I valori dei nuovi campi sono in `TimestampFormat M/D/YYYY h:mm:ss[.fff] TT`. Essendo stata usata la funzione `daystart()`, i valori timestamp corrispondono tutti al primo millisecondo del giorno.

I valori del periodo di rimborso anticipato sono sette giorni prima della data di scadenza, poiché il secondo argomento passato nella funzione `daystart()` era negativo.

I valori del periodo di rimborso tardivo sono 14 giorni dopo la data di scadenza, poiché il secondo argomento passato nella funzione `daystart()` era positivo.

Le date hanno un timestamp corrispondente alle 7:00 AM perché il valore del terzo argomento passato alla funzione `daystart()`, `day_start`, era `7/24`.

## firstworkdate

La funzione **firstworkdate** restituisce la data di inizio più recente per fare in modo che il valore **no\_of\_workdays** (dal lunedì al venerdì) non termini oltre la data **end\_date**, tenendo in considerazione tutte le festività eventualmente in calendario. **end\_date** e **holiday** devono essere date o indicatori temporali validi.

### Sintassi:

```
firstworkdate(end_date, no_of_workdays {, holiday} )
```

**Tipo di dati restituiti:** numero intero

### Argomenti:

#### Argomenti

| Argomento             | Descrizione                                                                                                                                                                                                                                                         |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>end_date</b>       | L'indicatore temporale della data di fine da valutare.                                                                                                                                                                                                              |
| <b>no_of_workdays</b> | Il numero di giorni lavorativi da raggiungere.                                                                                                                                                                                                                      |
| <b>holiday</b>        | Periodi di vacanza da escludere dai giorni lavorativi. Una vacanza è definita come una stringa di data costante. È possibile specificare più periodi di vacanza, separati da virgole.<br><br><b>Esempio:</b> '12/25/2013', '12/26/2013', '12/31/2013', '01/01/2014' |

### Esempi e risultati:

In questi esempi viene utilizzato il formato della data **DD/MM/YYYY**. Il formato della data viene specificato nell'istruzione **SET DateFormat** nella parte superiore dello script di caricamento dei dati. Modificare il formato negli esempi in base alle proprie necessità.

#### Esempi di script

| Esempio                                                                  | Risultato                                                                            |
|--------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| <code>firstworkdate ('29/12/2014', 9)</code>                             | Restituisce '17/12/2014'.                                                            |
| <code>firstworkdate ('29/12/2014', 9, '25/12/2014', '26/12/2014')</code> | Restituisce 15/12/2014 perché viene considerato un periodo di vacanza di due giorni. |

### Esempio:

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

ProjectTable:

```
LOAD *, recno() as InVID, INLINE [
EndDate
```

```

28/03/2015
10/12/2015
5/2/2016
31/3/2016
19/5/2016
15/9/2016
] ;
NrDays:
Load *,
FirstWorkDate(EndDate,120) As StartDate
Resident ProjectTable;
Drop table ProjectTable;

```

La tabella risultante mostra i valori restituiti in FirstWorkDate per ciascun record della tabella.

Tabella dei risultati

| InvID | EndDate    | StartDate  |
|-------|------------|------------|
| 1     | 28/03/2015 | 13/10/2014 |
| 2     | 10/12/2015 | 26/06/2015 |
| 3     | 5/2/2016   | 24/08/2015 |
| 4     | 31/3/2016  | 16/10/2015 |
| 5     | 19/5/2016  | 04/12/2015 |
| 6     | 15/9/2016  | 01/04/2016 |

## GMT

Questa funzione restituisce il valore Greenwich Mean Time attuale ricavato dalle impostazioni locali. La funzione restituisce i valori nel formato della variabile di sistema `TimestampFormat`.

Ogni volta che l'app viene ricaricata, qualsiasi tabella, variabile o oggetto grafico dello script di caricamento che utilizza la funzione GMT verrà adattato all'ultimo orario medio di Greenwich derivato dall'orologio di sistema.

### Sintassi:

```
GMT ( )
```

**Tipo di dati restituiti:** duale

Questi esempi utilizzano il formato timestamp `M/D/YYYY h:mm:ss[.fff] TT`. Il formato della data viene specificato nell'istruzione `SET TimestampFormat` nella parte superiore dello script di caricamento dei dati. Modificare il formato negli esempi in base alle proprie necessità.

Esempi di funzioni

| Esempio | Risultato            |
|---------|----------------------|
| GMT()   | 3/28/2022 2:47:36 PM |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - Variabile (script)

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor di caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda. Questo esempio imposta l'attuale ora di Greenwich come variabile nello script di caricamento utilizzando la funzione `GMT`.

#### Script di caricamento

```
LET vGMT = GMT();
```

#### Risultati

Caricare i dati e creare un foglio. Creare una casella di testo utilizzando l'oggetto grafico **Testo e immagine**.

Aggiungere questa misura alla casella di testo:

```
=vGMT
```

La casella di testo deve contenere una riga di testo con data e ora, come quella mostrata di seguito:

```
3/28/2022 2:47:36 PM
```

### Esempio 2 - Inizio dell'anno a novembre (script)

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente i libri scaduti della biblioteca, caricati in una tabella denominata overdue. È utilizzata la variabile di sistema predefinita `DateFormat MM/GG/AAAA`.
- La creazione di un nuovo campo chiamato `days_overdue`, che calcola il numero di giorni di ritardo di ogni libro.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Overdue:
```

```
Load
```

```
*,
```

```
Floor(GMT()-due_date) as days_overdue
```

```
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
cust_id,book_id,due_date
```

```
1,4,01/01/2021,
```

```
2,24,01/10/2021,
```

```
6,173,01/31/2021,
```

```
31,281,02/01/2021,
```

```
86,265,02/10/2021,
```

```
52,465,06/30/2021,
```

```
26,537,07/26/2021,
```

```
92,275,10/31/2021,
```

```
27,455,11/01/2021,
```

```
27,46,12/31/2021
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `due_date`
- `book_id`
- `days_overdue`

Tabella dei risultati

| due_date   | book_id | days_overdue |
|------------|---------|--------------|
| 01/01/2021 | 4       | 455          |
| 01/10/2021 | 24      | 446          |
| 01/31/2021 | 173     | 425          |
| 02/01/2021 | 281     | 424          |

| due_date   | book_id | days_overdue |
|------------|---------|--------------|
| 02/10/2021 | 265     | 415          |
| 06/30/2021 | 465     | 275          |
| 07/26/2021 | 537     | 249          |
| 10/31/2021 | 275     | 152          |
| 11/01/2021 | 455     | 151          |
| 12/31/2021 | 46      | 91           |

I valori del campo `days_overdue` sono calcolati trovando la differenza tra l'ora di Greenwich attuale, utilizzando la funzione `GMT()`, e la data di scadenza originale. Per calcolare solo i giorni, i risultati vengono arrotondati al numero intero più vicino utilizzando la funzione `Floor()`.

### Esempio 3 - oggetto grafico (grafico)

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda. Lo script di caricamento contiene lo stesso set di dati dell'esempio precedente. È utilizzata la variabile di sistema predefinita `DateFormat MM/GG/AAAA`.

Tuttavia, in questo esempio, il set di dati invariato viene caricato nell'applicazione. Il valore del numero di giorni di ritardo viene calcolato tramite una misura in un oggetto grafico.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Overdue:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
  cust_id,book_id,due_date
```

```
  1,4,01/01/2021,
```

```
  2,24,01/10/2021,
```

```
  6,173,01/31/2021,
```

```
  31,281,02/01/2021,
```

```
  86,265,02/10/2021,
```

```
  52,465,06/30/2021,
```

```
  26,537,07/26/2021,
```

```
  92,275,10/31/2021,
```

```
  27,455,11/01/2021,
```

```
  27,46,12/31/2021
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- due\_date
- book\_id

Creare la seguente misura:

`=Floor(GMT() - due_date)`

Tabella dei risultati

| due_date   | book_id | =Floor(GMT()-due_date) |
|------------|---------|------------------------|
| 01/01/2021 | 4       | 455                    |
| 01/10/2021 | 24      | 446                    |
| 01/31/2021 | 173     | 425                    |
| 02/01/2021 | 281     | 424                    |
| 02/10/2021 | 265     | 415                    |
| 06/30/2021 | 465     | 275                    |
| 07/26/2021 | 537     | 249                    |
| 10/31/2021 | 275     | 152                    |
| 11/01/2021 | 455     | 151                    |
| 12/31/2021 | 46      | 91                     |

I valori del campo `days_overdue` sono calcolati trovando la differenza tra l'ora di Greenwich attuale, utilizzando la funzione `GMT()`, e la data di scadenza originale. Per calcolare solo i giorni, i risultati vengono arrotondati al numero intero più vicino utilizzando la funzione `Floor()`.

### hour

Questa funzione restituisce un numero intero che rappresenta l'ora in cui la frazione di **expression** viene interpretata come ora in base all'interpretazione numerica standard.

#### Sintassi:

**hour** (expression)

**Tipo di dati restituiti:** numero intero

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema,

a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

Esempi di funzioni

| Esempio                         | Risultato                                                                                                                                                                                                    |
|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>hour( '09:14:36' )</code> | La stringa di testo fornita è convertita in modo implicito in un data e ora in quanto corrisponde al formato data e ora definito nella variabile <code>TimestampFormat</code> . L'espressione restituisce 9. |
| <code>hour( '0.5555' )</code>   | L'espressione restituisce 13 (poiché $0.5555 = 13:19:55$ ).                                                                                                                                                  |

### Esempio 1 - Variabile (script)

Script di caricamento e risultati

#### Panoramica

Aprire Editor caricamento dati, quindi aggiungere lo script di caricamento in basso in una nuova scheda.

Lo script di caricamento contiene:

- Un set di dati contenente transazioni per data e ora
- La variabile di sistema `Timestamp` predefinita (`M/D/YYYY h:mm:ss[.fff] TT`)

Creare un campo, 'hour', calcolando quando sono avvenuti gli acquisti.

#### Script di caricamento

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

```
Transactions:
```

```
Load
    *,
    hour(date) as hour
;
Load
*
Inline
[
id,date,amount
9497,'2022-01-05 19:04:57',47.25,
9498,'2022-01-03 14:21:53',51.75,
```

```
9499, '2022-01-03 05:40:49', 73.53,  
9500, '2022-01-04 18:49:38', 15.35,  
9501, '2022-01-01 22:10:22', 31.43,  
9502, '2022-01-05 19:34:46', 13.24,  
9503, '2022-01-04 22:58:34', 74.34,  
9504, '2022-01-06 11:29:38', 50.00,  
9505, '2022-01-02 08:35:54', 36.34,  
9506, '2022-01-06 08:49:09', 74.23  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- hour

Tabella dei risultati

| data                | ora |
|---------------------|-----|
| 2022-01-01 22:10:22 | 22  |
| 2022-01-02 08:35:54 | 8   |
| 2022-01-03 05:40:49 | 5   |
| 2022-01-03 14:21:53 | 14  |
| 2022-01-04 18:49:38 | 18  |
| 2022-01-04 22:58:34 | 22  |
| 2022-01-05 19:04:57 | 19  |
| 2022-01-05 19:34:46 | 19  |
| 2022-01-06 08:49:09 | 8   |
| 2022-01-06 11:29:38 | 11  |

I valori nel campo ora sono creati usando la funzione `hour()` e trasferendo la data come espressione nell'istruzione di caricamento precedente.

### Esempio 2 - Oggetto grafico (grafico)

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire Editor caricamento dati, quindi aggiungere lo script di caricamento in basso in una nuova scheda.

Lo script di caricamento contiene:

- Lo stesso set di dati dal primo esempio.
- La variabile di sistema `Timestamp` predefinita (M/D/YYYY h:mm:ss[.fff] TT).

Tuttavia, in questo esempio, il set di dati, invariato, viene caricato nell'applicazione. I valori 'hour' sono calcolati mediante una misura in un oggetto grafico.

### Script di caricamento

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
9497,'2022-01-05 19:04:57',47.25,
```

```
9498,'2022-01-03 14:21:53',51.75,
```

```
9499,'2022-01-03 05:40:49',73.53,
```

```
9500,'2022-01-04 18:49:38',15.35,
```

```
9501,'2022-01-01 22:10:22',31.43,
```

```
9502,'2022-01-05 19:34:46',13.24,
```

```
9503,'2022-01-04 22:58:34',74.34,
```

```
9504,'2022-01-06 11:29:38',50.00,
```

```
9505,'2022-01-02 08:35:54',36.34,
```

```
9506,'2022-01-06 08:49:09',74.23
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: date.

Per calcolare il 'hour', creare la seguente misura:

```
=hour(date)
```

Tabella dei risultati

| due_date            | =hour(date) |
|---------------------|-------------|
| 2022-01-01 22:10:22 | 22          |
| 2022-01-02 08:35:54 | 8           |
| 2022-01-03 05:40:49 | 5           |
| 2022-01-03 14:21:53 | 14          |
| 2022-01-04 18:49:38 | 18          |
| 2022-01-04 22:58:34 | 22          |

| due_date            | =hour(date) |
|---------------------|-------------|
| 2022-01-05 19:04:57 | 19          |
| 2022-01-05 19:34:46 | 19          |
| 2022-01-06 08:49:09 | 8           |
| 2022-01-06 11:29:38 | 11          |

I valori per 'hour' sono creati usando la funzione `hour()` e trasferendo la data come espressione in una misura per l'oggetto grafico.

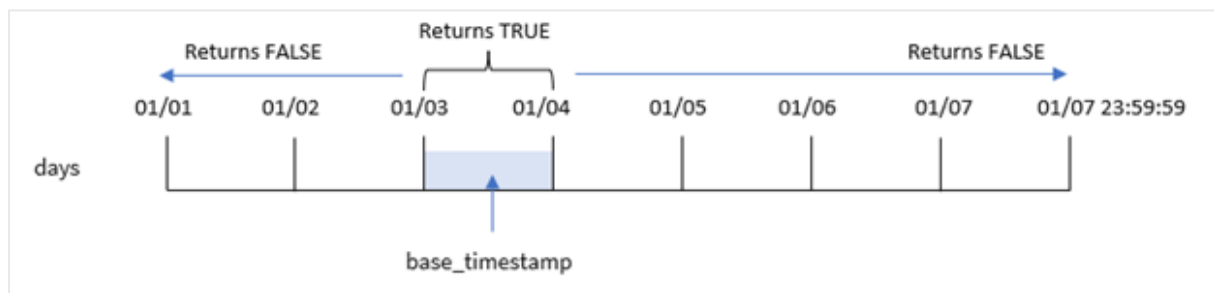
## inday

Questa funzione restituisce True se **timestamp** ricade all'interno del giorno contenente **base\_timestamp**.

Sintassi:

**InDay** (timestamp, base\_timestamp, period\_no[, day\_start])

Schema della funzione *inday*.



La funzione `inday()` utilizza l'argomento `base_timestamp` per identificare il giorno in cui cade il timestamp. L'ora di inizio del giorno è, per impostazione predefinita, la mezzanotte; ma è possibile modificare l'ora di inizio del giorno utilizzando l'argomento `day_start` della funzione `inday()`. Una volta definito il giorno, la funzione restituisce risultati booleani quando si confrontano i valori dei timestamp prescritti con quel giorno.

### Casi di utilizzo

La funzione `inday()` restituisce un risultato booleano. In genere, questo tipo di funzione viene utilizzato come condizione in un file `if expression`. Restituisce un'aggregazione o un calcolo che dipende dal fatto che una data valutata si sia verificata nel giorno del timestamp in questione.

Ad esempio, la funzione `inday()` può essere utilizzata per identificare tutte le apparecchiature prodotte in un determinato giorno.

**Tipo di dati restituiti:** Boolean

In Qlik Sense, il valore booleano vero è rappresentato da -1 e il valore falso è rappresentato da 0.

## Argomenti

| Argomento      | Descrizione                                                                                                                                                                                                                                                           |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| timestamp      | La data e l'ora che si desidera confrontare con base_timestamp.                                                                                                                                                                                                       |
| base_timestamp | La data e l'ora utilizzate per valutare l'indicatore temporale.                                                                                                                                                                                                       |
| period_no      | Il giorno può essere differito mediante period_no. period_no è un numero intero, in cui il valore 0 indica il giorno che contiene base_timestamp. I valori negativi di period_no indicano i giorni precedenti, mentre i valori positivi indicano i giorni successivi. |
| day_start      | Se si intende utilizzare giorni che non inizino alla mezzanotte, indicare un differimento come frazione di un giorno in day_start, ad esempio 0,125 per indicare le 3 del mattino.                                                                                    |

## Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

## Esempi di funzioni

| Esempio                                                                         | Risultato         |
|---------------------------------------------------------------------------------|-------------------|
| <code>inday ('01/12/2006 12:23:00 PM', '01/12/2006 12:00:00 AM', 0)</code>      | Restituisce True  |
| <code>inday ('01/12/2006 12:23:00 PM', '01/13/2006 12:00:00 AM', 0)</code>      | Restituisce False |
| <code>inday ('01/12/2006 12:23:00 PM', '01/12/2006 12:00:00 AM', -1)</code>     | Restituisce False |
| <code>inday ('01/11/2006 12:23:00 PM', '01/12/2006 12:00:00 AM', -1)</code>     | Restituisce True  |
| <code>inday ('01/12/2006 12:23:00 PM', '01/12/2006 12:00:00 AM', 0, 0.5)</code> | Restituisce False |
| <code>inday ('01/12/2006 11:23:00 AM', '01/12/2006 12:00:00 AM', 0, 0.5)</code> | Restituisce True  |

## Esempio 1 - Istruzione LOAD (script)

Script di caricamento e risultati

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente transazioni per timestamp, caricato in una tabella chiamata `Transactions`.
- Un campo data fornito nel formato `Timestamp` della variabile di sistema (`M/D/YYYY h:mm:ss[.fff] TT`).
- Un caricamento precedente che contiene la funzione `inday()` impostata come campo `in_day`.

### Script di caricamento

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

```
Transactions:
```

```
Load
    *,
    inday(date,'01/05/2022 12:00:00 AM', 0) as in_day
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
9497,'01/01/2022 7:34:46 PM',13.24
```

```
9498,'01/01/2022 10:10:22 PM',31.43
```

```
9499,'01/02/2022 8:35:54 AM',36.34
```

```
9500,'01/03/2022 2:21:53 PM',51.75
```

```
9501,'01/04/2022 6:49:38 PM',15.35
```

```
9502,'01/04/2022 10:58:34 PM',74.34
```

```
9503,'01/05/2022 5:40:49 AM',73.53
```

```
9504,'01/05/2022 11:29:38 AM',50.00
```

```
9505,'01/05/2022 7:04:57 PM',47.25
```

```
9506,'01/06/2022 8:49:09 AM',74.23
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `date`
- `in_day`

Tabella dei risultati

| <b>date</b>            | <b>in_day</b> |
|------------------------|---------------|
| 01/01/2022 7:34:46 PM  | 0             |
| 01/01/2022 10:10:22 PM | 0             |

| date                   | in_day |
|------------------------|--------|
| 01/02/2022 8:35:54 AM  | 0      |
| 01/03/2022 2:21:53 PM  | 0      |
| 01/04/2022 6:49:38 PM  | 0      |
| 01/04/2022 10:58:34 PM | 0      |
| 01/05/2022 5:40:49 AM  | -1     |
| 01/05/2022 11:29:38 AM | -1     |
| 01/05/2022 7:04:57 PM  | -1     |
| 01/06/2022 8:49:09 AM  | 0      |

Il campo `in_day` viene creato nell'istruzione di caricamento precedente mediante l'uso della funzione `inday()` e trasferendo il campo `date`, un timestamp con hard coding per il 5 gennaio e un `period_no` di 0 come argomenti della funzione.

### Esempio 2 - `period_no`

Script di caricamento e risultati

#### Panoramica

Lo script di caricamento utilizza lo stesso set di dati e lo stesso scenario utilizzati nel primo esempio.

Tuttavia, in questo esempio, l'attività viene calcolata se la data di transazione è occorsa due giorni prima del 5 gennaio.

#### Script di caricamento

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';

Transactions:
    Load
        *,
        inday(date, '01/05/2022 12:00:00 AM', -2) as in_day
    ;

Load
*
Inline
[
id,date,amount
9497,'01/01/2022 7:34:46 PM',13.24
9498,'01/01/2022 10:10:22 PM',31.43
9499,'01/02/2022 8:35:54 AM',36.34
9500,'01/03/2022 2:21:53 PM',51.75
9501,'01/04/2022 6:49:38 PM',15.35
9502,'01/04/2022 10:58:34 PM',74.34
9503,'01/05/2022 5:40:49 AM',73.53
9504,'01/05/2022 11:29:38 AM',50.00
9505,'01/05/2022 7:04:57 PM',47.25
```

```
9506, '01/06/2022 8:49:09 AM', 74.23  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- in\_day

Tabella dei risultati

| date                   | in_day |
|------------------------|--------|
| 01/01/2022 7:34:46 PM  | 0      |
| 01/01/2022 10:10:22 PM | 0      |
| 01/02/2022 8:35:54 AM  | 0      |
| 01/03/2022 2:21:53 PM  | -1     |
| 01/04/2022 6:49:38 PM  | 0      |
| 01/04/2022 10:58:34 PM | 0      |
| 01/05/2022 5:40:49 AM  | 0      |
| 01/05/2022 11:29:38 AM | 0      |
| 01/05/2022 7:04:57 PM  | 0      |
| 01/06/2022 8:49:09 AM  | 0      |

In questa istanza, dato che un `period_no` di -2 è stato utilizzato come argomento `offset` nella funzione `inday()`, la funzione determina se ciascuna data di transazione è avvenuta il 3 gennaio. Ciò può essere verificato nella tabella di output in cui una transazione restituisce un risultato booleano di `TRUE`.

### Esempio 3 - day\_start

Script di caricamento e risultati

#### Panoramica

Lo script di caricamento utilizza lo stesso set di dati e lo stesso scenario utilizzati negli esempi precedenti.

Tuttavia, in questo esempio, la policy aziendale indica che la giornata lavorativa inizia e termina alle 7:00.

#### Script di caricamento

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

```
Transactions:
```

```
Load
```

```
*,
```

```
inday(date, '01/05/2022 12:00:00 AM', 0, 7/24) as in_day
```

```
;
Load
*
Inline
[
id,date,amount
9497,'01/01/2022 7:34:46 PM',13.24
9498,'01/01/2022 10:10:22 PM',31.43
9499,'01/02/2022 8:35:54 AM',36.34
9500,'01/03/2022 2:21:53 PM',51.75
9501,'01/04/2022 6:49:38 PM',15.35
9502,'01/04/2022 10:58:34 PM',74.34
9503,'01/05/2022 5:40:49 AM',73.53
9504,'01/05/2022 11:29:38 AM',50.00
9505,'01/05/2022 7:04:57 PM',47.25
9506,'01/06/2022 8:49:09 AM',74.23
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- in\_day

Tabella dei risultati

| date                   | in_day |
|------------------------|--------|
| 01/01/2022 7:34:46 PM  | 0      |
| 01/01/2022 10:10:22 PM | 0      |
| 01/02/2022 8:35:54 AM  | 0      |
| 01/03/2022 2:21:53 PM  | 0      |
| 01/04/2022 6:49:38 PM  | -1     |
| 01/04/2022 10:58:34 PM | -1     |
| 01/05/2022 5:40:49 AM  | -1     |
| 01/05/2022 11:29:38 AM | 0      |
| 01/05/2022 7:04:57 PM  | 0      |
| 01/06/2022 8:49:09 AM  | 0      |

Dato che l'argomento `start_day` di `7/24`, che corrisponde alle 7:00, è usato nella funzione `inday()`, la funzione determina se la data di ciascuna transazione è avvenuta il 4 gennaio dalle 7:00 e il 5 gennaio prima delle 7:00.

Ciò può essere verificato nella tabella di output dove le transazioni che hanno avuto luogo dopo le 7:00 il 4 gennaio restituiscono un risultato booleano di `TRUE` mentre le transazioni che hanno avuto luogo dopo le 7:00 il 5 gennaio restituiscono un risultato booleano di `FALSE`.

### Esempio 4 - Oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Lo script di caricamento utilizza lo stesso set di dati e lo stesso scenario utilizzati negli esempi precedenti.

Tuttavia, in questo esempio, il set di dati è invariato e viene caricato nell'applicazione. Si calcolerà per determinare se una transazione è avvenuta il 5 gennaio creando una misura in un oggetto grafico.

#### Script di caricamento

Transactions:

Load

\*

Inline

[

id,date,amount

9497,'01/01/2022 7:34:46 PM',13.24

9498,'01/01/2022 10:10:22 PM',31.43

9499,'01/02/2022 8:35:54 AM',36.34

9500,'01/03/2022 2:21:53 PM',51.75

9501,'01/04/2022 6:49:38 PM',15.35

9502,'01/04/2022 10:58:34 PM',74.34

9503,'01/05/2022 5:40:49 AM',73.53

9504,'01/05/2022 11:29:38 AM',50.00

9505,'01/05/2022 7:04:57 PM',47.25

9506,'01/06/2022 8:49:09 AM',74.23

];

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- date

Per calcolare se una transazione è avvenuta il 5 gennaio, creare la seguente misura:

=inday(date,'01/05/2022 12:00:00 AM',0)

Tabella dei risultati

| date                   | inday(date,'01/05/2022 12:00:00 AM',0) |
|------------------------|----------------------------------------|
| 01/01/2022 7:34:46 PM  | 0                                      |
| 01/01/2022 10:10:22 PM | 0                                      |
| 01/02/2022 8:35:54 AM  | 0                                      |
| 01/03/2022 2:21:53 PM  | 0                                      |

| date                   | inday(date,'01/05/2022 12:00:00 AM',0) |
|------------------------|----------------------------------------|
| 01/04/2022 6:49:38 PM  | 0                                      |
| 01/04/2022 10:58:34 PM | 0                                      |
| 01/05/2022 5:40:49 AM  | -1                                     |
| 01/05/2022 11:29:38 AM | -1                                     |
| 01/05/2022 7:04:57 PM  | -1                                     |
| 01/06/2022 8:49:09 AM  | 0                                      |

## Esempio 5 - Scenario

Script di caricamento e risultati

### Panoramica

In questo esempio, è stato identificato che, a causa di un errore delle apparecchiature, i prodotti fabbricati il 5 gennaio erano difettosi. L'utente finale desidera un oggetto grafico che visualizzi, per data, lo stato dei prodotti fabbricati "difettosi" o "senza difetti" e il costo dei prodotti fabbricati il 5 gennaio.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella chiamata "Prodotti".
- La tabella contiene i seguenti campi:
  - ID prodotto
  - ora di produzione
  - prezzo di costo

### Script di caricamento

Products:

Load

\*

Inline

```
[
product_id,manufacture_date,cost_price
9497,'01/01/2022 7:34:46 PM',13.24
9498,'01/01/2022 10:10:22 PM',31.43
9499,'01/02/2022 8:35:54 AM',36.34
9500,'01/03/2022 2:21:53 PM',51.75
9501,'01/04/2022 6:49:38 PM',15.35
9502,'01/04/2022 10:58:34 PM',74.34
9503,'01/05/2022 5:40:49 AM',73.53
9504,'01/05/2022 11:29:38 AM',50.00
9505,'01/05/2022 7:04:57 PM',47.25
9506,'01/06/2022 8:49:09 AM',74.23
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

=dayname(manufacture\_date)

Creare le seguenti misure:

- =if(only(InDay(manufacture\_date,makedate(2022,01,05),0)), 'Defective', 'Faultless')
- =sum(cost\_price)

Impostare la misura **Formattazione numero** su **Denaro**.

In **Aspetto**, disattivare **Totals**.

Tabella dei risultati

| dayname<br>(manufacture_date) | =if(only(InDay(manufacture_date,makedate<br>(2022,01,05),0)), 'Defective', 'Faultless') | =sum(cost_<br>price) |
|-------------------------------|-----------------------------------------------------------------------------------------|----------------------|
| 01/01/2022                    | Non difettoso                                                                           | 44.67                |
| 01/02/2022                    | Non difettoso                                                                           | 36.34                |
| 01/03/2022                    | Non difettoso                                                                           | 51.75                |
| 01/04/2022                    | Non difettoso                                                                           | 89.69                |
| 01/05/2022                    | Difettoso                                                                               | 170.78               |
| 01/06/2022                    | Non difettoso                                                                           | 74.23                |

La funzione `inday()` restituisce un valore booleano quando valuta le date di produzione di ciascun prodotto. Per qualsiasi prodotto fabbricato il 5 gennaio, la funzione `inday()` restituisce un valore booleano TRUE e contrassegna i prodotti come "difettosi". Per tutti i prodotti che restituiscono un valore FALSE, e che quindi non sono stati fabbricati in quel giorno, vengono contrassegnati come "non difettosi".

### indaytotime

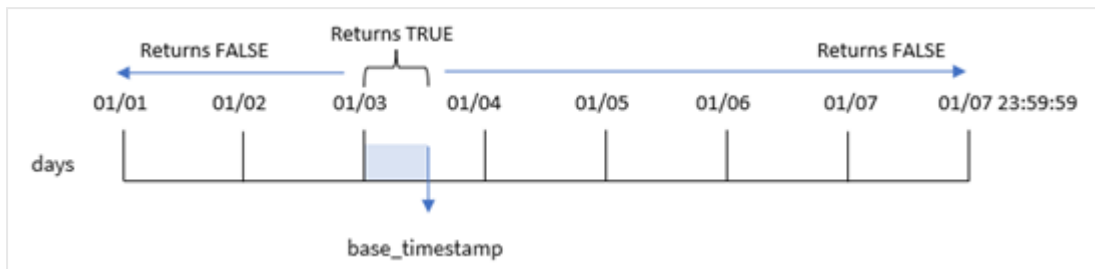
Questa funzione restituisce True se **timestamp** ricade nella parte del giorno contenente **base\_timestamp** fino a includere il millisecondo esatto di **base\_timestamp**.

Sintassi:

**InDayToTime** (timestamp, base\_timestamp, period\_no[, day\_start])

La funzione `indaytotime()` restituisce un risultato booleano a seconda di quando si verifica un valore di timestamp durante il segmento del giorno. Il limite iniziale di questo segmento è l'inizio del giorno, che per impostazione predefinita è la mezzanotte; l'inizio del giorno può essere modificato dall'argomento `day_start` della funzione `indaytotime()`. Il limite finale del segmento giornaliero è determinato da un argomento `base_timestamp` della funzione.

Schema della funzione *indaytotime*.



### Casi di utilizzo

La funzione `indaytotime()` restituisce un risultato booleano. In genere, questo tipo di funzione viene utilizzato come condizione in un file `if` expression. La funzione `indaytotime()` restituisce un'aggregazione o un calcolo a seconda che un timestamp si sia verificato nel segmento del giorno fino al momento del timestamp di base incluso.

Ad esempio, la funzione `indaytotime()` può essere utilizzata per mostrare la somma delle vendite di biglietti per gli spettacoli che hanno avuto luogo fino ad oggi.

**Tipo di dati restituiti:** Boolean

In Qlik Sense, il valore booleano vero è rappresentato da -1 e il valore falso è rappresentato da 0.

### Argomenti

| Argomento                   | Descrizione                                                                                                                                                                                                                                                                                                                 |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>timestamp</code>      | La data e l'ora che si desidera confrontare con <code>base_timestamp</code> .                                                                                                                                                                                                                                               |
| <code>base_timestamp</code> | La data e l'ora utilizzate per valutare l'indicatore temporale.                                                                                                                                                                                                                                                             |
| <code>period_no</code>      | Il giorno può essere differito mediante <code>period_no</code> . <code>period_no</code> è un numero intero, in cui il valore 0 indica il giorno che contiene <code>base_timestamp</code> . I valori negativi di <code>period_no</code> indicano i giorni precedenti, mentre i valori positivi indicano i giorni successivi. |
| <code>day_start</code>      | (Facoltativo) Se si intende utilizzare giorni che non inizino a mezzanotte, indicare un differimento come frazione di un giorno in <code>day_start</code> , ad esempio 0,125 per indicare le 3 del mattino.                                                                                                                 |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempi di funzioni

| Esempio                                                                          | Risultato         |
|----------------------------------------------------------------------------------|-------------------|
| <code>indaytotime ('01/12/2006 12:23:00 PM', '01/12/2006 11:59:00 PM', 0)</code> | Restituisce True  |
| <code>indaytotime ('01/12/2006 12:23:00 PM', '01/12/2006 12:00:00 AM', 0)</code> | Restituisce False |
| <code>indaytotime '01/11/2006 12:23:00 PM', '01/12/2006 11:59:00 PM', -1)</code> | Restituisce True  |

### Esempio 1 - nessun argomento aggiuntivo

Script di caricamento e risultati

#### Panoramica

Aprire Editor caricamento dati, quindi aggiungere lo script di caricamento in basso in una nuova scheda.

Lo script di caricamento contiene:

- Un set di dati contenente una serie di transazioni per il periodo tra il 4 e il 5 gennaio viene caricato in una tabella chiamata "Transazioni".
- Un campo data fornito nel formato `Timestamp` della variabile di sistema `(M/D/YYYY h:mm:ss[.fff] TT)`.
- Un caricamento precedente che contiene la funzione `indaytotime()` impostata come `'in_day_to_time'`, che determina se ciascuna delle transazioni avviene prima delle 9:00 del mattino.

#### Script di caricamento

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

```
Transactions:
```

```
Load
    *,
    indaytotime(date, '01/05/2022 9:00:00 AM', 0) as in_day_to_time
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188, '01/04/2022 3:41:54 AM', 25.66
```

```
8189, '01/04/2022 4:19:43 AM', 87.21
```

```
8190, '01/04/2022 4:53:47 AM', 53.80
```

```
8191, '01/04/2022 8:38:53 AM', 69.98
```

```
8192, '01/04/2022 10:37:52 AM', 57.42
```

```
8193, '01/04/2022 1:54:10 PM', 45.89
8194, '01/04/2022 5:53:23 PM', 82.77
8195, '01/04/2022 8:13:26 PM', 36.23
8196, '01/04/2022 10:00:49 PM', 76.11
8197, '01/05/2022 7:45:37 AM', 82.06
8198, '01/05/2022 8:44:36 AM', 17.17
8199, '01/05/2022 11:26:08 AM', 40.39
8200, '01/05/2022 6:43:08 PM', 37.23
8201, '01/05/2022 10:54:10 PM', 88.27
8202, '01/05/2022 11:09:09 PM', 95.93
];
```

### Risultati

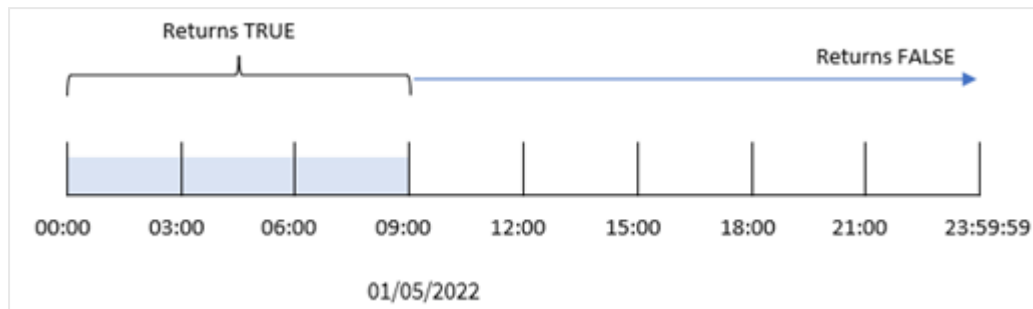
Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- in\_day\_to\_time

Tabella dei risultati

| date                   | in_day_to_time |
|------------------------|----------------|
| 01/04/2022 3:41:54 AM  | 0              |
| 01/04/2022 4:19:43 AM  | 0              |
| 01/04/2022 04:53:47 AM | 0              |
| 01/04/2022 8:38:53 AM  | 0              |
| 01/04/2022 10:37:52 AM | 0              |
| 01/04/2022 1:54:10 PM  | 0              |
| 01/04/2022 5:53:23 PM  | 0              |
| 01/04/2022 8:13:26 PM  | 0              |
| 01/04/2022 10:00:49 PM | 0              |
| 01/05/2022 7:45:37 AM  | -1             |
| 01/05/2022 8:44:36 AM  | -1             |
| 01/05/2022 11:26:08 AM | 0              |
| 01/05/2022 6:43:08 PM  | 0              |
| 01/05/2022 10:54:10 PM | 0              |
| 01/05/2022 11:09:09 PM | 0              |

Schema 1 di esempio della funzione `indaytotime` con il limite delle 9:00 AM.



`in_day_to_time` field creato nell'istruzione di caricamento precedente mediante l'uso della funzione `indaytotime()` e trasferendo il campo `data`, un timestamp con hard coding per il 5 gennaio alle 9:00 e un offset di 0 come argomenti della funzione. Qualsiasi transazione che si verifichi tra la mezzanotte e le 9:00 del 5 gennaio restituisce TRUE.

### Esempio 2 - `period_no`

Script di caricamento e risultati

#### Panoramica

Lo script di caricamento utilizza lo stesso set di dati e lo stesso scenario utilizzati nel primo esempio.

Tuttavia, in questo esempio, verrà calcolato se la data di transazione è occorsa un giorno prima delle 9:00 del 5 gennaio.

#### Script di caricamento

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

Transactions:

```
Load
    *,
    indaytotime(date,'01/05/2022 9:00:00 AM', -1) as in_day_to_time
;
```

Load

\*

Inline

[

id,date,amount

8188,'01/04/2022 3:41:54 AM',25.66

8189,'01/04/2022 4:19:43 AM',87.21

8190,'01/04/2022 4:53:47 AM',53.80

8191,'01/04/2022 8:38:53 AM',69.98

8192,'01/04/2022 10:37:52 AM',57.42

8193,'01/04/2022 1:54:10 PM',45.89

8194,'01/04/2022 5:53:23 PM',82.77

8195,'01/04/2022 8:13:26 PM',36.23

8196,'01/04/2022 10:00:49 PM',76.11

8197,'01/05/2022 7:45:37 AM',82.06

8198,'01/05/2022 8:44:36 AM',17.17

```
8199, '01/05/2022 11:26:08 AM', 40.39
8200, '01/05/2022 6:43:08 PM', 37.23
8201, '01/05/2022 10:54:10 PM', 88.27
8202, '01/05/2022 11:09:09 PM', 95.93
];
```

### Risultati

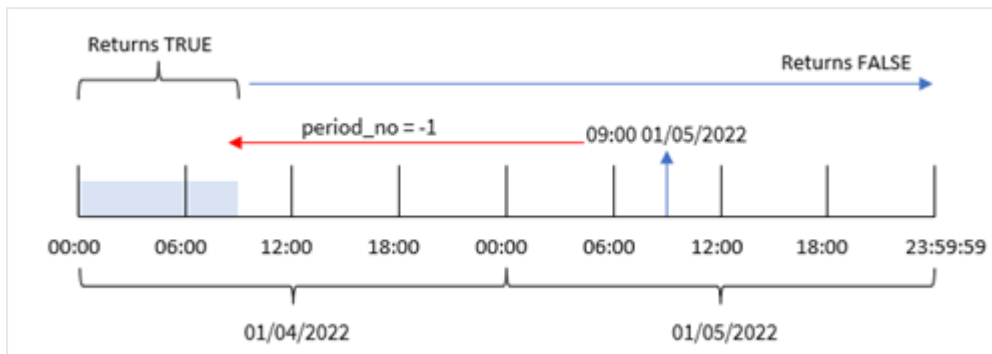
Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- in\_day\_to\_time

Tabella dei risultati

| date                   | in_day_to_time |
|------------------------|----------------|
| 01/04/2022 3:41:54 AM  | -1             |
| 01/04/2022 4:19:43 AM  | -1             |
| 01/04/2022 04:53:47 AM | -1             |
| 01/04/2022 8:38:53 AM  | -1             |
| 01/04/2022 10:37:52 AM | 0              |
| 01/04/2022 1:54:10 PM  | 0              |
| 01/04/2022 5:53:23 PM  | 0              |
| 01/04/2022 8:13:26 PM  | 0              |
| 01/04/2022 10:00:49 PM | 0              |
| 01/05/2022 7:45:37 AM  | 0              |
| 01/05/2022 8:44:36 AM  | 0              |
| 01/05/2022 11:26:08 AM | 0              |
| 01/05/2022 6:43:08 PM  | 0              |
| 01/05/2022 10:54:10 PM | 0              |
| 01/05/2022 11:09:09 PM | 0              |

Schema esempio 2 della funzione `indaytotime` con transazioni dal 4 gennaio.



In questo esempio, dato che un offset di -1 è stato utilizzato come argomento di offset nella funzione `indaytotime()`, la funzione determina se ciascuna data di transazione ha avuto luogo prima delle 9:00 del 4 gennaio. Ciò è verificabile nella tabella di output dove una transazione restituisce un risultato booleano di TRUE.

### Esempio 3 - day\_start

Script di caricamento e risultati

#### Panoramica

Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, in questo esempio, la policy aziendale indica che la giornata lavorativa inizia e termina alle 8:00.

#### Script di caricamento

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

Transactions:

```
Load
*,
indaytotime(date,'01/05/2022 9:00:00 AM', 0,8/24) as in_day_to_time
;

Load
*
Inline
[
id,date,amount
8188,'01/04/2022 3:41:54 AM',25.66
8189,'01/04/2022 4:19:43 AM',87.21
8190,'01/04/2022 4:53:47 AM',53.80
8191,'01/04/2022 8:38:53 AM',69.98
8192,'01/04/2022 10:37:52 AM',57.42
8193,'01/04/2022 1:54:10 PM',45.89
8194,'01/04/2022 5:53:23 PM',82.77
8195,'01/04/2022 8:13:26 PM',36.23
8196,'01/04/2022 10:00:49 PM',76.11
8197,'01/05/2022 7:45:37 AM',82.06
8198,'01/05/2022 8:44:36 AM',17.17
```

```
8199, '01/05/2022 11:26:08 AM', 40.39
8200, '01/05/2022 6:43:08 PM', 37.23
8201, '01/05/2022 10:54:10 PM', 88.27
8202, '01/05/2022 11:09:09 PM', 95.93
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- in\_day\_to\_time

Tabella dei risultati

| date                   | in_day_to_time |
|------------------------|----------------|
| 01/04/2022 3:41:54 AM  | 0              |
| 01/04/2022 4:19:43 AM  | 0              |
| 01/04/2022 04:53:47 AM | 0              |
| 01/04/2022 8:38:53 AM  | 0              |
| 01/04/2022 10:37:52 AM | 0              |
| 01/04/2022 1:54:10 PM  | 0              |
| 01/04/2022 5:53:23 PM  | 0              |
| 01/04/2022 8:13:26 PM  | 0              |
| 01/04/2022 10:00:49 PM | 0              |
| 01/05/2022 7:45:37 AM  | 0              |
| 01/05/2022 8:44:36 AM  | -1             |
| 01/05/2022 11:26:08 AM | 0              |
| 01/05/2022 6:43:08 PM  | 0              |
| 01/05/2022 10:54:10 PM | 0              |
| 01/05/2022 11:09:09 PM | 0              |

Schema esempio 3 della funzione *indaytotime* con transazioni dalle 8:00 alle 9:00.



Poiché nella funzione `indaytotime()` viene utilizzato l'argomento `start_day 8/24`, che equivale alle 8:00 del mattino, ogni giorno inizia e finisce alle 8:00 del mattino. Pertanto, la funzione `indaytotime()` restituirà un risultato booleano di `TRUE` per qualsiasi transazione avvenuta tra le 8:00 e le 9:00 del 5 gennaio.

### Esempio 4 - Oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, in questo esempio, il set di dati è invariato e viene caricato nell'applicazione. Calcolare per determinare se una transazione avviene il 5 gennaio prima delle 9:00 creando una misura in un oggetto grafico.

#### Script di caricamento

```
Transactions:
Load
*
Inline
[
id,date,amount
8188,'01/04/2022 3:41:54 AM',25.66
8189,'01/04/2022 4:19:43 AM',87.21
8190,'01/04/2022 4:53:47 AM',53.80
8191,'01/04/2022 8:38:53 AM',69.98
8192,'01/04/2022 10:37:52 AM',57.42
8193,'01/04/2022 1:54:10 PM',45.89
8194,'01/04/2022 5:53:23 PM',82.77
8195,'01/04/2022 8:13:26 PM',36.23
8196,'01/04/2022 10:00:49 PM',76.11
8197,'01/05/2022 7:45:37 AM',82.06
8198,'01/05/2022 8:44:36 AM',17.17
8199,'01/05/2022 11:26:08 AM',40.39
8200,'01/05/2022 6:43:08 PM',37.23
8201,'01/05/2022 10:54:10 PM',88.27
8202,'01/05/2022 11:09:09 PM',95.93
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

`date.`

Per determinare se una transazione è avvenuta il 5 gennaio prima delle 9:00, creare la seguente misura:

```
=indaytotime(date,'01/05/2022 9:00:00 AM',0)
```

| Tabella dei risultati  |                                              |
|------------------------|----------------------------------------------|
| date                   | =indaytotime(date,'01/05/2022 9:00:00 AM',0) |
| 01/04/2022 3:41:54 AM  | 0                                            |
| 01/04/2022 4:19:43 AM  | 0                                            |
| 01/04/2022 04:53:47 AM | 0                                            |
| 01/04/2022 8:38:53 AM  | 0                                            |
| 01/04/2022 10:37:52 AM | 0                                            |
| 01/04/2022 1:54:10 PM  | 0                                            |
| 01/04/2022 5:53:23 PM  | 0                                            |
| 01/04/2022 8:13:26 PM  | 0                                            |
| 01/04/2022 10:00:49 PM | 0                                            |
| 01/05/2022 7:45:37 AM  | -1                                           |
| 01/05/2022 8:44:36 AM  | -1                                           |
| 01/05/2022 11:26:08 AM | 0                                            |
| 01/05/2022 6:43:08 PM  | 0                                            |
| 01/05/2022 10:54:10 PM | 0                                            |
| 01/05/2022 11:09:09 PM | 0                                            |

La misura `in_day_to_time` viene creata nell'oggetto grafico utilizzando la funzione `indaytotime()` e passando come argomenti della funzione il campo `data`, un timestamp codificato per le 9:00 del 5 gennaio e un offset di 0. Qualsiasi transazione che si verifichi tra la mezzanotte e le 9:00 del 5 gennaio restituisce TRUE. Questo viene convalidato nella tabella dei risultati.

### Esempio 5 - Scenario

Script di caricamento e risultati

#### Panoramica

In questo esempio, un set di dati contenente le vendite di biglietti di un cinema locale viene caricato in una tabella chiamata `Ticket_Sales`. Oggi è il 3 maggio 2022 e sono le 11:00 del mattino.

L'utente desidera un oggetto grafico KPI che mostri i ricavi di tutti gli spettacoli che si sono svolti oggi.

#### Script di caricamento

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

```
Ticket_Sales:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
sale ID, show time, ticket price
1,05/01/2022 09:30:00 AM,10.50
2,05/03/2022 05:30:00 PM,21.00
3,05/03/2022 09:30:00 AM,10.50
4,05/03/2022 09:30:00 AM,31.50
5,05/03/2022 09:30:00 AM,10.50
6,05/03/2022 12:00:00 PM,42.00
7,05/03/2022 12:00:00 PM,10.50
8,05/03/2022 05:30:00 PM,42.00
9,05/03/2022 08:00:00 PM,31.50
10,05/04/2022 10:30:00 AM,31.50
11,05/04/2022 12:00:00 PM,10.50
12,05/04/2022 05:30:00 PM,10.50
13,05/05/2022 05:30:00 PM,21.00
14,05/06/2022 12:00:00 PM,21.00
15,05/07/2022 09:30:00 AM,42.00
16,05/07/2022 10:30:00 AM,42.00
17,05/07/2022 10:30:00 AM,10.50
18,05/07/2022 05:30:00 PM,10.50
19,05/08/2022 05:30:00 PM,21.00
20,05/11/2022 09:30:00 AM,10.50
];
```

### Risultati

Procedere come segue:

1. Creare un oggetto KPI.
2. Creare una misura che mostri la somma di tutte le vendite di biglietti per gli spettacoli che si sono svolti oggi fino ad ora, utilizzando la funzione `indaytotime()`:

```
=sum(if(indaytotime([show time],'05/03/2022 11:00:00 AM',0),[ticket price],0))
```

3. Creare un'etichetta per l'oggetto KPI "Entrate correnti".
4. Impostare la misura **Formattazione numero** su **Denaro**.

Il totale delle vendite dei biglietti fino alle 11:00 del 3 maggio 2022 è di 52,50 dollari.

La funzione `indaytotime()` restituisce un valore booleano quando si confrontano gli orari degli spettacoli di ciascuna vendita di biglietti con l'ora corrente ('05/03/2022 11:00:00 AM'). Per qualsiasi spettacolo del 3 maggio prima delle 11:00, la funzione `indaytotime()` restituisce il valore booleano TRUE e il prezzo del biglietto sarà incluso nella somma totale.

### inlunarweek

Questa funzione determina se **timestamp** ricade all'interno della settimana lunare contenente **base\_date**. Le settimane lunari in Qlik Sense sono definite contando il 1° gennaio come primo giorno della settimana, a parte l'ultima settimana dell'anno, ogni settimana conterrà esattamente sette giorni.

#### Sintassi:

```
InLunarWeek (timestamp, base_date, period_no[, first_week_day])
```

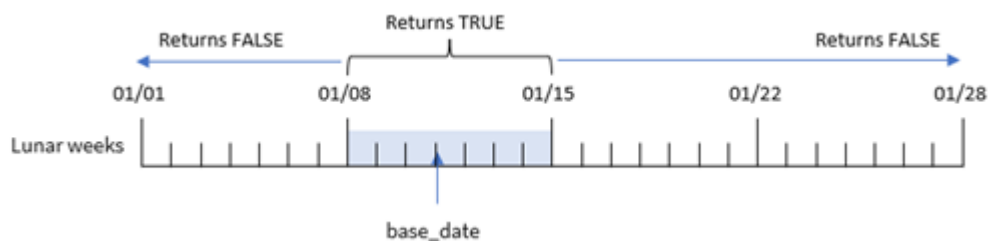
Tipo di dati restituiti: Boolean



*In Qlik Sense, il valore booleano vero è rappresentato da -1 e il valore falso è rappresentato da 0.*

La funzione `inLunarweek()` determina la settimana lunare in cui cade `base_date`. Restituisce quindi un risultato booleano dopo aver determinato se ogni valore di timestamp si verifica nella stessa settimana lunare del valore `base_date`.

Schema della funzione `inLunarweek()`.



## Casi di utilizzo

La funzione `inLunarweek()` restituisce un risultato booleano. In genere, questo tipo di funzione viene utilizzato come condizione in un'espressione IF. Questo restituisce un'aggregazione o un calcolo che dipende dal fatto che la data valutata si sia verificata durante la settimana lunare in questione.

Ad esempio, la funzione `inLunarweek()` può essere utilizzata per identificare tutte le apparecchiature prodotte in una particolare settimana lunare.

### Argomenti

| Argomento             | Descrizione                                                                                                                                                                                                                                                                                                                              |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>timestamp</b>      | La data da confrontare con <b>base_date</b> .                                                                                                                                                                                                                                                                                            |
| <b>base_date</b>      | La data utilizzata per valutare la settimana lunare.                                                                                                                                                                                                                                                                                     |
| <b>period_no</b>      | La settimana lunare può essere differita mediante <b>period_no</b> . <b>period_no</b> è un numero intero, in cui il valore 0 indica la settimana lunare che contiene <b>base_date</b> . I valori negativi di <b>period_no</b> indicano le settimane lunari precedenti, mentre i valori positivi indicano le settimane lunari successive. |
| <b>first_week_day</b> | Un differimento che può essere maggiore o minore di zero. Ciò modifica l'inizio dell'anno in base al numero specificato di giorni e/o frazioni di un giorno.                                                                                                                                                                             |

## Esempi di funzioni

| Esempio                                                    | Risultato                                                                                                                                                                                                                                                                                                                          |
|------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>inlunarweek('01/12/2013', '01/14/2013', 0)</code>    | Restituisce il valore <code>TRUE</code> , dato che il valore di <code>timestamp</code> , 01/12/2013, ricade nella settimana 01/08/2013 a 01/14/2013.                                                                                                                                                                               |
| <code>inlunarweek('01/12/2013', '01/07/2013', 0)</code>    | Restituisce <code>FALSE</code> , dato che il valore di <code>base_date</code> 01/07/2013 è nella settimana lunare definita come 01/01/2013 a 01/07/2013.                                                                                                                                                                           |
| <code>inlunarweek('01/12/2013', '01/14/2013', -1)</code>   | Restituisce <code>FALSE</code> . La specifica di un valore di <code>period_no</code> come -1 fa slittare la settimana a quella precedente, 01/01/2013 a 01/07/2013.                                                                                                                                                                |
| <code>inlunarweek('01/07/2013', '01/14/2013', -1)</code>   | Restituisce <code>TRUE</code> . In confronto con l'esempio precedente, il valore <code>timestamp</code> è nella settimana precedente, dopo aver considerato lo slittamento all'indietro.                                                                                                                                           |
| <code>inlunarweek('01/11/2006', '01/08/2006', 0, 3)</code> | Restituisce <code>FALSE</code> . La specifica di un valore di 3 per <code>first_week_day</code> significa che l'avvio dell'anno è calcolato da 01/04/2013. Pertanto il valore di <code>base_date</code> ricade nella prima settimana mentre il valore di <code>timestamp</code> ricade nella settimana da 01/11/2013 a 01/17/2013. |

La funzione `inlunarweek()` viene spesso utilizzata in combinazione con le seguenti funzioni:

## Funzioni correlate

| Funzione                                                                               | Interazione                                                                                                                       |
|----------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| <a href="#"><u><code>lunarweekname</code></u></a><br><a href="#"><u>(page 972)</u></a> | Questa funzione viene utilizzata per determinare il numero della settimana lunare dell'anno in cui si verifica una data inserita. |

## Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - Nessun argomento aggiuntivo

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati di transazioni per il mese di gennaio, che viene caricato in una tabella chiamata `Transactions`.
- Il campo della data è stato fornito nel formato della variabile di sistema `DateFormat` (MM/GG/AAAA).

Crea un campo, `in_lunar_week`, che determina se le transazioni sono avvenute nella stessa settimana lunare del 10 gennaio.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        inlunarweek(date,'01/10/2022', 0) as in_lunar_week
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8183,'1/5/2022',42.32
```

```
8184,'1/6/2022',68.22
```

```
8185,'1/7/2022',15.25
```

```
8186,'1/8/2022',25.26
```

```
8187,'1/9/2022',37.23
```

```
8188,'1/10/2022',37.23
```

```
8189,'1/11/2022',17.17
```

```
8190,'1/12/2022',88.27
```

```
8191,'1/13/2022',57.42
```

```
8192,'1/14/2022',53.80
```

```
8193,'1/15/2022',82.06
```

```
8194,'1/16/2022',87.21
```

```
8195,'1/17/2022',95.93
```

```
8196,'1/18/2022',45.89
```

```
8197,'1/19/2022',36.23
```

```
8198,'1/20/2022',25.66
```

```
8199,'1/21/2022',82.77
```

```
8200,'1/22/2022',69.98
```

```
8201,'1/23/2022',76.11
```

```
];
```

### Risultati

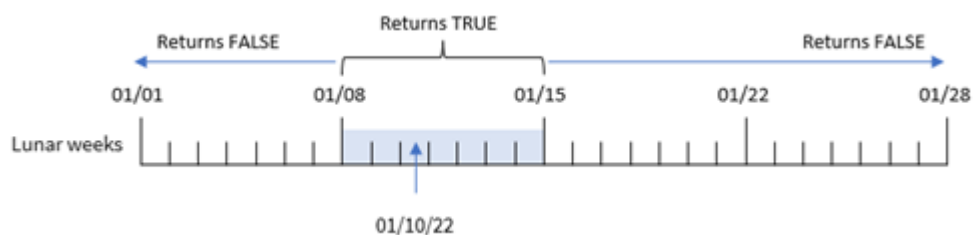
Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- in\_lunar\_week

Tabella dei risultati

| date      | in_lunar_week |
|-----------|---------------|
| 1/5/2022  | 0             |
| 1/6/2022  | 0             |
| 1/7/2022  | 0             |
| 1/8/2022  | -1            |
| 1/9/2022  | -1            |
| 1/10/2022 | -1            |
| 1/11/2022 | -1            |
| 1/12/2022 | -1            |
| 1/13/2022 | -1            |
| 1/14/2022 | -1            |
| 1/15/2022 | 0             |
| 1/16/2022 | 0             |
| 1/17/2022 | 0             |
| 1/18/2022 | 0             |
| 1/19/2022 | 0             |
| 1/20/2022 | 0             |
| 1/21/2022 | 0             |
| 1/22/2022 | 0             |
| 1/23/2022 | 0             |

Funzione `in_lunarweek()`, esempio base



Il campo `in_lunar_week` viene creato nell'istruzione di caricamento precedente utilizzando la funzione `inlunarweek()` e passando i seguenti argomenti alla funzione:

- Il campo `date`
- La data del 10 gennaio, codificata in modo rigido, è `base_date`
- `Aperiod_no` di 0

Poiché le settimane lunari iniziano il 1° gennaio, il 10 gennaio cadrebbe nella settimana lunare che inizia l'8 gennaio e termina il 14 gennaio. Pertanto, tutte le transazioni che si verificano tra queste due date di gennaio restituiranno un valore booleano di `TRUE`. Questo viene convalidato nella tabella dei risultati.

### Esempio 2 - `period_no`

Esempi e risultati:

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.
- Il campo della data è stato fornito nel formato della variabile di sistema `DateFormat` (`MM/GG/AAAA`).

Tuttavia, in questo esempio, il compito è quello di creare un campo, `2_lunar_weeks_later`, che determini se le transazioni sono avvenute o meno due settimane lunari dopo il 10 gennaio.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        inlunarweek(date,'01/10/2022', 2) as [2_lunar_weeks_later]
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8183,'1/5/2022',42.32
```

```
8184,'1/6/2022',68.22
```

```
8185,'1/7/2022',15.25
```

```
8186,'1/8/2022',25.26
```

```
8187,'1/9/2022',37.23
```

```
8188,'1/10/2022',37.23
```

```
8189,'1/11/2022',17.17
```

```
8190,'1/12/2022',88.27
```

```
8191,'1/13/2022',57.42
```

```
8192,'1/14/2022',53.80
```

```
8193,'1/15/2022',82.06
```

```
8194, '1/16/2022', 87.21
8195, '1/17/2022', 95.93
8196, '1/18/2022', 45.89
8197, '1/19/2022', 36.23
8198, '1/20/2022', 25.66
8199, '1/21/2022', 82.77
8200, '1/22/2022', 69.98
8201, '1/23/2022', 76.11
];
```

### Risultati

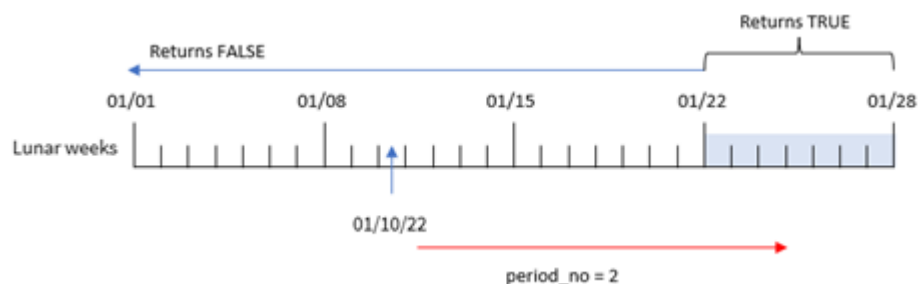
Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- 2\_lunar\_weeks\_later

Tabella dei risultati

| date      | 2_lunar_weeks_later |
|-----------|---------------------|
| 1/5/2022  | 0                   |
| 1/6/2022  | 0                   |
| 1/7/2022  | 0                   |
| 1/8/2022  | 0                   |
| 1/9/2022  | 0                   |
| 1/10/2022 | 0                   |
| 1/11/2022 | 0                   |
| 1/12/2022 | 0                   |
| 1/13/2022 | 0                   |
| 1/14/2022 | 0                   |
| 1/15/2022 | 0                   |
| 1/16/2022 | 0                   |
| 1/17/2022 | 0                   |
| 1/18/2022 | 0                   |
| 1/19/2022 | 0                   |
| 1/20/2022 | 0                   |
| 1/21/2022 | 0                   |
| 1/22/2022 | -1                  |
| 1/23/2022 | -1                  |

Funzione `inLunarweek()`, esempio `period_no`



In questo caso, poiché è stato utilizzato un valore `period_no` di 2 come argomento di offset nella funzione `inLunarweek()`, la funzione definisce la settimana che inizia il 22 gennaio come settimana lunare rispetto alla quale convalidare le transazioni. Pertanto, qualsiasi transazione effettuata tra il 22 e il 28 gennaio restituirà un risultato booleano di `TRUE`.

### Esempio 3 - first\_week\_day

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor di caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento utilizza lo stesso set di dati e lo stesso scenario del primo esempio. Tuttavia, nell'esempio, abbiamo impostato le settimane lunari a partire dal 6 gennaio.

- Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.
- È utilizzata la variabile di sistema predefinita `DateFormat MM/GG/AAAA`.
- Un argomento `first_week_day` di 5. In questo modo le settimane lunari iniziano il 5 gennaio.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
inLunarweek(date,'01/10/2022', 0,5) as in_lunar_week
```

```
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8183,'1/5/2022',42.32
```

```
8184,'1/6/2022',68.22
```

```
8185,'1/7/2022',15.25
```

```
8186,'1/8/2022',25.26
```

```
8187,'1/9/2022',37.23
```

```
8188,'1/10/2022',37.23
```

```
8189, '1/11/2022', 17.17
8190, '1/12/2022', 88.27
8191, '1/13/2022', 57.42
8192, '1/14/2022', 53.80
8193, '1/15/2022', 82.06
8194, '1/16/2022', 87.21
8195, '1/17/2022', 95.93
8196, '1/18/2022', 45.89
8197, '1/19/2022', 36.23
8198, '1/20/2022', 25.66
8199, '1/21/2022', 82.77
8200, '1/22/2022', 69.98
8201, '1/23/2022', 76.11
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

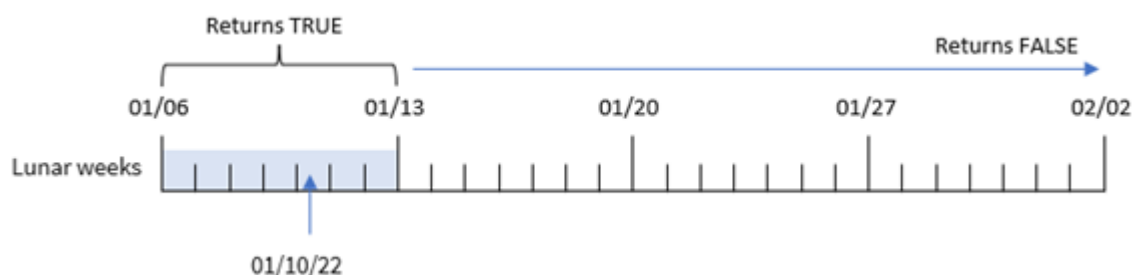
- date
- in\_lunar\_week

Tabella dei risultati

| date      | in_lunar_week |
|-----------|---------------|
| 1/5/2022  | 0             |
| 1/6/2022  | -1            |
| 1/7/2022  | -1            |
| 1/8/2022  | -1            |
| 1/9/2022  | -1            |
| 1/10/2022 | -1            |
| 1/11/2022 | -1            |
| 1/12/2022 | -1            |
| 1/13/2022 | 0             |
| 1/14/2022 | 0             |
| 1/15/2022 | 0             |
| 1/16/2022 | 0             |
| 1/17/2022 | 0             |
| 1/18/2022 | 0             |
| 1/19/2022 | 0             |
| 1/20/2022 | 0             |
| 1/21/2022 | 0             |

| date      | in_lunar_week |
|-----------|---------------|
| 1/22/2022 | 0             |
| 1/23/2022 | 0             |

Funzione `inLunarWeek()`, esempio `first_week_day`



In questo caso, poiché nella funzione `inLunarWeek()` viene utilizzato l'argomento `first_week_date 5`, si sposta l'inizio della settimana lunare al 6 gennaio. Pertanto, il 10 gennaio cade nella settimana lunare che inizia il 6 gennaio e termina il 12 gennaio. Qualsiasi transazione che cade tra queste due date restituirà un valore booleano di `TRUE`.

## Esempio 4 - Oggetto grafico

Script di caricamento ed espressione del grafico:

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.
- Il campo della data è stato fornito nel formato della variabile di sistema `DateFormat (MM/GG/AAAA)`.

Tuttavia, in questo esempio, il set di dati invariato viene caricato nell'applicazione. Il calcolo che determina se le transazioni sono avvenute nella stessa settimana lunare del 10 gennaio viene creato come misura in un oggetto grafico dell'applicazione.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8183,'1/5/2022',42.32
```

```
8184, '1/6/2022', 68.22
8185, '1/7/2022', 15.25
8186, '1/8/2022', 25.26
8187, '1/9/2022', 37.23
8188, '1/10/2022', 37.23
8189, '1/11/2022', 17.17
8190, '1/12/2022', 88.27
8191, '1/13/2022', 57.42
8192, '1/14/2022', 53.80
8193, '1/15/2022', 82.06
8194, '1/16/2022', 87.21
8195, '1/17/2022', 95.93
8196, '1/18/2022', 45.89
8197, '1/19/2022', 36.23
8198, '1/20/2022', 25.66
8199, '1/21/2022', 82.77
8200, '1/22/2022', 69.98
8201, '1/23/2022', 76.11
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: date.

Per calcolare se una transazione avviene nella settimana lunare che contiene il 10 gennaio, creare la seguente misura:

```
= inlunarweek(date, '01/10/2022', 0)
```

Tabella dei risultati

| date      | =inlunarweek(date, '01/10/2022', 0) |
|-----------|-------------------------------------|
| 1/5/2022  | 0                                   |
| 1/6/2022  | 0                                   |
| 1/7/2022  | 0                                   |
| 1/8/2022  | -1                                  |
| 1/9/2022  | -1                                  |
| 1/10/2022 | -1                                  |
| 1/11/2022 | -1                                  |
| 1/12/2022 | -1                                  |
| 1/13/2022 | -1                                  |
| 1/14/2022 | -1                                  |
| 1/15/2022 | 0                                   |
| 1/16/2022 | 0                                   |

| date      | =inlunarweek(date,'01/10/2022', 0) |
|-----------|------------------------------------|
| 1/17/2022 | 0                                  |
| 1/18/2022 | 0                                  |
| 1/19/2022 | 0                                  |
| 1/20/2022 | 0                                  |
| 1/21/2022 | 0                                  |
| 1/22/2022 | 0                                  |
| 1/23/2022 | 0                                  |

## Esempio 5 - Scenario

Script di caricamento ed espressione del grafico:

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella chiamata Products.
- Informazioni composte da ID prodotto, data di produzione e prezzo di costo.

È stato individuato che, a causa di un errore delle apparecchiature, i prodotti fabbricati nella settimana lunare che comprendeva il 12 gennaio erano difettosi. L'utente finale desidera un oggetto grafico che visualizzi, per nome della settimana lunare, lo stato dei prodotti fabbricati "difettosi" o "non difettosi" e il costo dei prodotti fabbricati in quel mese.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
product_id,manufacture_date,cost_price
```

```
8183, '1/5/2022', 42.32
```

```
8184, '1/6/2022', 68.22
```

```
8185, '1/7/2022', 15.25
```

```
8186, '1/8/2022', 25.26
```

```
8187, '1/9/2022', 37.23
```

```
8188, '1/10/2022', 37.23
```

```
8189, '1/11/2022', 17.17
```

```
8190, '1/12/2022', 88.27
```

```
8191, '1/13/2022', 57.42
```

```
8192, '1/14/2022', 53.80
```

```

8193, '1/15/2022', 82.06
8194, '1/16/2022', 87.21
8195, '1/17/2022', 95.93
8196, '1/18/2022', 45.89
8197, '1/19/2022', 36.23
8198, '1/20/2022', 25.66
8199, '1/21/2022', 82.77
8200, '1/22/2022', 69.98
8201, '1/23/2022', 76.11
];

```

## Risultati

Procedere come indicato di seguito:

1. Caricare i dati e aprire un foglio. Creare una nuova tabella.
2. Creare una dimensione per mostrare i nomi dei mesi:  
=lunarweekname(manufacture\_date)
3. Creare una misura per identificare quali prodotti sono difettosi e quali sono privi di difetti utilizzando la funzione inlunarweek():  
=if(only(inlunarweek(manufacture\_date,makedate(2022,01,12),0)), 'Defective','Faultless')
4. Creare una misura per sommare il valore cost\_price dei prodotti:  
=sum(cost\_price)
5. Impostare la **Formattazione numero** della misura su **Denaro**.
6. In **Aspetto**, disattivare **Totals**.

Tabella dei risultati

| lunarweekname<br>(manufacture_date) | =if(only(inlunarweek(manufacture_date,makedate<br>(2022,01,12),0)), 'Defective','Faultless') | =sum(cost_<br>price) |
|-------------------------------------|----------------------------------------------------------------------------------------------|----------------------|
| 2022/01                             | Non difettoso                                                                                | \$125.79             |
| 2022/02                             | Difettoso                                                                                    | \$316.38             |
| 2022/03                             | Non difettoso                                                                                | \$455.75             |
| 2022/04                             | Non difettoso                                                                                | \$146.09             |

La funzione inlunarweek() restituisce un valore booleano quando valuta le date di produzione di ciascun prodotto. Per qualsiasi prodotto fabbricato nella settimana lunare che contiene il 10 gennaio, la funzione inlunarweek() restituisce un valore booleano di TRUE e contrassegna i prodotti come "difettosi". Tutti i prodotti che restituiscono un valore pari a FALSE, e che quindi non sono stati prodotti in quella settimana, vengono contrassegnati come "non difettosi".

## inlunarweektoday

Questa funzione stabilisce se **timestamp** ricade all'interno della parte della settimana lunare fino a includere l'ultimo millisecondo di **base\_date**. Le settimane lunari in Qlik Sense sono definite contando il 1° gennaio come primo giorno della settimana e, a parte l'ultima settimana dell'anno, conterranno esattamente sette giorni.

**Sintassi:**

```
InLunarWeekToDate (timestamp, base_date, period_no [, first_week_day])
```

**Tipo di dati restituiti:** Boolean



*In Qlik Sense, il valore booleano vero è rappresentato da -1 e il valore falso è rappresentato da 0.*

Schema esemplificativo della funzione `inLunarWeekToDate()`



La funzione `inLunarWeekToDate()` funge da punto finale della settimana lunare. In contrasto, la funzione `inLunarWeek()` determina la settimana lunare in cui cade `base_date`. Ad esempio, se il `base_date` fosse il 5 gennaio, qualsiasi `timestamp` tra il 1 e il 5 gennaio restituirebbe un risultato booleano di `TRUE`, mentre le date del 6 e 7 gennaio e successive restituirebbero un risultato booleano di `FALSE`.

## Argomenti

| Argomento             | Descrizione                                                                                                                                                                                                                                                                                                                              |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>timestamp</b>      | La data da confrontare con <b>base_date</b> .                                                                                                                                                                                                                                                                                            |
| <b>base_date</b>      | La data utilizzata per valutare la settimana lunare.                                                                                                                                                                                                                                                                                     |
| <b>period_no</b>      | La settimana lunare può essere differita mediante <b>period_no</b> . <b>period_no</b> è un numero intero, in cui il valore 0 indica la settimana lunare che contiene <b>base_date</b> . I valori negativi di <b>period_no</b> indicano le settimane lunari precedenti, mentre i valori positivi indicano le settimane lunari successive. |
| <b>first_week_day</b> | Un differimento che può essere maggiore o minore di zero. Ciò modifica l'inizio dell'anno in base al numero specificato di giorni e/o frazioni di un giorno.                                                                                                                                                                             |

**Casi di utilizzo**

La funzione `inLunarWeekToDate()` restituisce un risultato booleano. In genere, questo tipo di funzione viene utilizzato come condizione in un'espressione IF. La funzione `inLunarWeekToDate()` viene utilizzata quando l'utente desidera che il calcolo restituisca un'aggregazione o un calcolo, a seconda che la data valutata sia avvenuta in un particolare segmento della settimana in questione.

Ad esempio, la funzione `inLunarWeekToDate()` può essere utilizzata per identificare tutte le apparecchiature prodotte in una determinata settimana fino a una data specifica.

### Esempi di funzioni

| Esempio                                                       | Risultato                                                                                                                                                                                                                                      |
|---------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>inlunarweektodate('01/12/2013', '01/13/2013', 0)</code> | Restituisce <code>TRUE</code> , dato che il valore di <code>timestamp</code> , 01/12/2013, ricade nella parte della settimana da 01/08/2013 a 01/13/2013.                                                                                      |
| <code>inlunarweektodate('01/12/2013', '01/11/2013', 0)</code> | Restituisce <code>FALSE</code> , poiché il valore di <code>timestamp</code> è successivo al valore di <code>base_date</code> , anche se le due date si trovano nella stessa settimana lunare prima di 01/12/2012.                              |
| <code>inlunarweektodate('01/12/2006', '01/05/2006', 1)</code> | Restituisce <code>TRUE</code> . Specificando un lavoro pari a 1 per <code>period_no</code> , <code>base_date</code> slitta in avanti di una settimana, pertanto il valore di <code>timestamp</code> ricade nella parte della settimana lunare. |

La funzione `inlunarweektodate()` viene spesso utilizzata in combinazione con le seguenti funzioni:

### Funzioni correlate

| Funzione                                                                  | Interazione                                                                                                                       |
|---------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| <a href="#"><u>lunarweekname</u></a><br><a href="#"><u>(page 972)</u></a> | Questa funzione viene utilizzata per determinare il numero della settimana lunare dell'anno in cui si verifica una data inserita. |

## Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

## Esempio 1 - Nessun argomento aggiuntivo

Script di caricamento e risultati

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il mese di gennaio, caricato in una tabella denominata `Transactions`. È utilizzata la variabile di sistema predefinita `DateFormat` `MM/GG/AAAA`.
- Creare un campo, `in_lunar_week_to_date`, che determina se le transazioni sono avvenute nella settimana lunare alla data del 10 gennaio.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        inlunarweektoday(date,'01/10/2022', 0) as in_lunar_week_to_date
    ;

Load
*
Inline
[
id,date,amount
8188,'1/10/2022',37.23
8189,'1/17/2022',17.17
8190,'1/26/2022',88.27
8191,'1/12/2022',57.42
8192,'1/19/2022',53.80
8193,'1/21/2022',82.06
8194,'1/1/2022',40.39
8195,'1/27/2022',87.21
8196,'1/11/2022',95.93
8197,'1/29/2022',45.89
8198,'1/31/2022',36.23
8199,'1/18/2022',25.66
8200,'1/23/2022',82.77
8201,'1/15/2022',69.98
8202,'1/4/2022',76.11
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

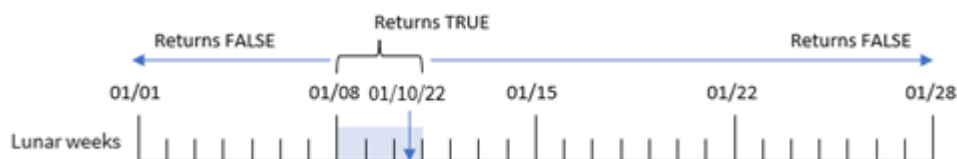
- `date`
- `in_lunar_week_to_date`

Tabella dei risultati

| date      | in_lunar_week_to_date |
|-----------|-----------------------|
| 1/1/2022  | 0                     |
| 1/4/2022  | 0                     |
| 1/10/2022 | -1                    |

| date      | in_lunar_week_to_date |
|-----------|-----------------------|
| 1/11/2022 | 0                     |
| 1/12/2022 | 0                     |
| 1/15/2022 | 0                     |
| 1/17/2022 | 0                     |
| 1/18/2022 | 0                     |
| 1/19/2022 | 0                     |
| 1/21/2022 | 0                     |
| 1/23/2022 | 0                     |
| 1/26/2022 | 0                     |
| 1/27/2022 | 0                     |
| 1/29/2022 | 0                     |
| 1/31/2022 | 0                     |

Funzione `inlunarweektoday()`, senza argomenti aggiuntivi



Il campo `in_lunar_week_to_date` viene creato nell'istruzione di caricamento precedente mediante l'uso della funzione `inlunarweektoday()` e trasferendo il campo `date`, una data con hard coding per il 10 gennaio come `base_date`, e un offset di 0 come argomenti della funzione.

Poiché le settimane lunari iniziano il 1° gennaio, il 10 gennaio cadrebbe nella settimana lunare che inizia l'8 gennaio; e poiché stiamo utilizzando la funzione `inlunarweektoday()`, quella settimana lunare finirebbe il 10. Pertanto, tutte le transazioni che si verificano tra queste due date di gennaio restituiranno un valore booleano di `TRUE`. Questo viene convalidato nella tabella dei risultati.

### Esempio 2 - `period_no`

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor di caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento contiene lo stesso set di dati e lo stesso scenario del primo esempio. Tuttavia, in questo esempio, il compito è quello di creare un campo, `2_lunar_weeks_later`, che determini se le transazioni sono avvenute o meno due settimane dopo la settimana lunare alla data del 1° gennaio.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
Transactions:
    Load
        *,
        inlunarweektoday(date,'01/10/2022', 2) as [2_lunar_weeks_later]
    ;
Load
*
Inline
[
id,date,amount
8188,'1/10/2022',37.23
8189,'1/17/2022',17.17
8190,'1/26/2022',88.27
8191,'1/12/2022',57.42
8192,'1/19/2022',53.80
8193,'1/21/2022',82.06
8194,'1/1/2022',40.39
8195,'1/27/2022',87.21
8196,'1/11/2022',95.93
8197,'1/29/2022',45.89
8198,'1/31/2022',36.23
8199,'1/18/2022',25.66
8200,'1/23/2022',82.77
8201,'1/15/2022',69.98
8202,'1/4/2022',76.11
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

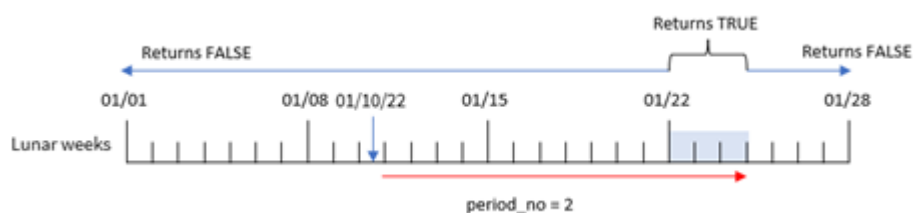
- date
- 2\_lunar\_weeks\_later

Tabella dei risultati

| date      | 2_lunar_weeks_later |
|-----------|---------------------|
| 1/1/2022  | 0                   |
| 1/4/2022  | 0                   |
| 1/10/2022 | 0                   |
| 1/11/2022 | 0                   |

| date      | 2_lunar_weeks_later |
|-----------|---------------------|
| 1/12/2022 | 0                   |
| 1/15/2022 | 0                   |
| 1/17/2022 | 0                   |
| 1/18/2022 | 0                   |
| 1/19/2022 | 0                   |
| 1/21/2022 | 0                   |
| 1/23/2022 | -1                  |
| 1/26/2022 | 0                   |
| 1/27/2022 | 0                   |
| 1/29/2022 | 0                   |
| 1/31/2022 | 0                   |

Funzione `inLunarweektoday()`, esempio `period_no`



In questo caso, la funzione `inLunarweektoday()` determina che la settimana lunare fino al 10 gennaio equivale a tre giorni (8, 9, 10 gennaio). Poiché è stato utilizzato un valore `period_no` di 2 come argomento di offset, questa settimana lunare è spostata di 14 giorni. Pertanto, la settimana lunare di tre giorni comprende il 22, 23 e 24 gennaio. Qualsiasi transazione effettuata tra il 22 e il 24 gennaio restituirà un risultato booleano di `TRUE`.

### Esempio 3 - `first_week_day`

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.
- È utilizzata la variabile di sistema predefinita `DateFormat MM/GG/AAAA`.
- Un argomento `first_week_date` di 3. In questo modo le settimane lunari iniziano il 3 gennaio.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        inlunarweek(date,'01/10/2022', 0,3) as in_lunar_week_to_date
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'1/10/2022',37.23
```

```
8189,'1/17/2022',17.17
```

```
8190,'1/26/2022',88.27
```

```
8191,'1/12/2022',57.42
```

```
8192,'1/19/2022',53.80
```

```
8193,'1/21/2022',82.06
```

```
8194,'1/1/2022',40.39
```

```
8195,'1/27/2022',87.21
```

```
8196,'1/11/2022',95.93
```

```
8197,'1/29/2022',45.89
```

```
8198,'1/31/2022',36.23
```

```
8199,'1/18/2022',25.66
```

```
8200,'1/23/2022',82.77
```

```
8201,'1/15/2022',69.98
```

```
8202,'1/4/2022',76.11
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

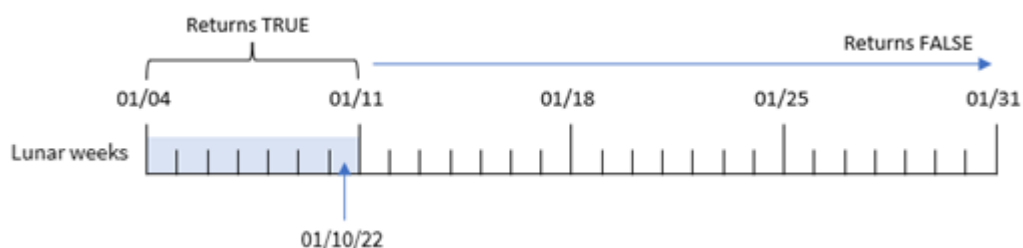
- date
- in\_lunar\_week\_to\_date

Tabella dei risultati

| date      | in_lunar_week_to_date |
|-----------|-----------------------|
| 1/1/2022  | 0                     |
| 1/4/2022  | -1                    |
| 1/10/2022 | -1                    |
| 1/11/2022 | 0                     |
| 1/12/2022 | 0                     |
| 1/15/2022 | 0                     |

| date      | in_lunar_week_to_date |
|-----------|-----------------------|
| 1/17/2022 | 0                     |
| 1/18/2022 | 0                     |
| 1/19/2022 | 0                     |
| 1/21/2022 | 0                     |
| 1/23/2022 | 0                     |
| 1/26/2022 | 0                     |
| 1/27/2022 | 0                     |
| 1/29/2022 | 0                     |
| 1/31/2022 | 0                     |

Funzione `in_lunarweektoday()`, esempio `first_week_day`



In questo caso, poiché nella funzione `in_lunarweek()` viene utilizzato l'argomento 3 the `first_week_date`, la prima settimana lunare andrà dal 3 al 10 gennaio. Poiché il 10 gennaio è anche il `base_date`, qualsiasi transazione che cade tra queste due date restituirà un valore booleano di `TRUE`.

## Esempio 4 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

### Panoramica

Aprire l'editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento contiene lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, in questo esempio, il set di dati invariato viene caricato nell'applicazione. Il calcolo che determina se le transazioni sono avvenute nella stessa settimana lunare fino al 10 gennaio viene creato come misura in un oggetto grafico dell'applicazione.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

Transactions:

Load

\*

Inline

[

id,date,amount

8188,'1/10/2022',37.23

8189,'1/17/2022',17.17

8190,'1/26/2022',88.27

8191,'1/12/2022',57.42

8192,'1/19/2022',53.80

8193,'1/21/2022',82.06

8194,'1/1/2022',40.39

8195,'1/27/2022',87.21

8196,'1/11/2022',95.93

8197,'1/29/2022',45.89

8198,'1/31/2022',36.23

8199,'1/18/2022',25.66

8200,'1/23/2022',82.77

8201,'1/15/2022',69.98

8202,'1/4/2022',76.11

];

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: date.

Creare la seguente misura:

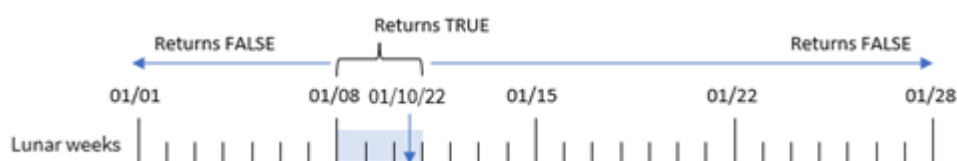
=inlunarweektodate(date,'01/10/2022', 0)

Tabella dei risultati

| date      | =inlunarweektodate(date,'01/10/2022', 0) |
|-----------|------------------------------------------|
| 1/1/2022  | 0                                        |
| 1/4/2022  | 0                                        |
| 1/10/2022 | -1                                       |
| 1/11/2022 | 0                                        |
| 1/12/2022 | 0                                        |
| 1/15/2022 | 0                                        |
| 1/17/2022 | 0                                        |
| 1/18/2022 | 0                                        |
| 1/19/2022 | 0                                        |
| 1/21/2022 | 0                                        |

| date      | =inlunarweektodate(date,'01/10/2022', 0) |
|-----------|------------------------------------------|
| 1/23/2022 | 0                                        |
| 1/26/2022 | 0                                        |
| 1/27/2022 | 0                                        |
| 1/29/2022 | 0                                        |
| 1/31/2022 | 0                                        |

Funzione `inlunarweektodate()`, esempio di oggetto grafico



La misura `in_lunar_week_to_date` viene creata nell'oggetto grafico utilizzando la funzione `inlunarweektodate()` e passando al campo `data`, una data con hard-coding per il 10 gennaio come nostro `base_date`, e un offset di 0 come argomenti della funzione.

Poiché le settimane lunari iniziano il 1° gennaio, il 10 gennaio cadrebbe nella settimana lunare che inizia l'8 gennaio. Inoltre, dal momento che stiamo utilizzando la funzione `inlunarweektodate()`, la settimana lunare terminerebbe il 10. Pertanto, tutte le transazioni che si verificano tra queste due date di gennaio restituiranno un valore booleano di `TRUE`. Questo viene convalidato nella tabella dei risultati.

## Esempio 5 - Scenario

Script di caricamento ed espressioni del grafico

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella chiamata `Products`.
- Informazioni composte da ID prodotto, data di produzione e prezzo di costo.

Si è determinato che, a causa di un errore delle apparecchiature, i prodotti fabbricati nella settimana lunare del 12 gennaio erano difettosi. Il problema è stato risolto il 13 gennaio. L'utente finale desidera un oggetto grafico che visualizzi, per settimana, se i prodotti fabbricati sono "difettosi" o "non difettosi" e il costo dei prodotti fabbricati quella settimana.

### Script di caricamento

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff]';
```

```
Products:
Load
*
Inline
[
product_id,manufacture_date,cost_price
8188,'01/02/2022 12:22:06',37.23
8189,'01/05/2022 01:02:30',17.17
8190,'01/06/2022 15:36:20',88.27
8191,'01/08/2022 10:58:35',57.42
8192,'01/09/2022 08:53:32',53.80
8193,'01/10/2022 21:13:01',82.06
8194,'01/11/2022 00:57:13',40.39
8195,'01/12/2022 09:26:02',87.21
8196,'01/13/2022 15:05:09',95.93
8197,'01/14/2022 18:44:57',45.89
8198,'01/15/2022 06:10:46',36.23
8199,'01/16/2022 06:39:27',25.66
8200,'01/17/2022 10:44:16',82.77
8201,'01/18/2022 18:48:17',69.98
8202,'01/26/2022 04:36:03',76.11
8203,'01/27/2022 08:07:49',25.12
8204,'01/28/2022 12:24:29',46.23
8205,'01/30/2022 11:56:56',84.21
8206,'01/30/2022 14:40:19',96.24
8207,'01/31/2022 05:28:21',67.67
];
```

### Risultati

Procedere come indicato di seguito:

1. Caricare i dati e aprire un foglio. Creare una nuova tabella.
2. Creare una dimensione per visualizzare i nomi delle settimane:  
=weekname(manufacture\_date)
3. Quindi, creare una dimensione che utilizzi la funzione inlunarweektodate() per identificare quali prodotti sono difettosi e quali non lo sono:  
=if(inlunarweektodate(manufacture\_date,makedate(2022,01,12),0),'Defective','Faultless')
4. Creare una misura per sommare il valore cost\_price dei prodotti:  
=sum(cost\_price)
5. Impostare la **Formattazione numero** della misura su **Denaro**.

Tabella dei risultati

| =lunarweekname<br>(manufacture_date) | =if(InLunarWeekToDate(manufacture_date,makedate<br>(2022,01,12),0),'Defective','Faultless') | =Sum(cost_<br>price) |
|--------------------------------------|---------------------------------------------------------------------------------------------|----------------------|
| 2022/01                              | Non difettoso                                                                               | \$142.67             |
| 2022/02                              | Difettoso                                                                                   | \$320.88             |
| 2022/02                              | Non difettoso                                                                               | \$141.82             |

| =lunarweekname<br>(manufacture_date) | =if(InLunarWeekToDate(manufacture_date,makedate<br>(2022,01,12),0),'Defective','Faultless') | =Sum(cost_<br>price) |
|--------------------------------------|---------------------------------------------------------------------------------------------|----------------------|
| 2022/03                              | Non difettoso                                                                               | \$214.64             |
| 2022/04                              | Non difettoso                                                                               | \$147.46             |
| 2022/05                              | Non difettoso                                                                               | \$248.12             |

La funzione `inlunarweektoday()` restituisce un valore booleano quando valuta le date di produzione di ciascun prodotto. Per coloro che restituiscono il valore booleano `TRUE`, contrassegna i prodotti come 'defective'. Per qualsiasi prodotto che restituisce il valore `FALSE`, e quindi non è stato realizzato nella settimana lunare fino al 12 gennaio, contrassegna i prodotti come 'Faultless'.

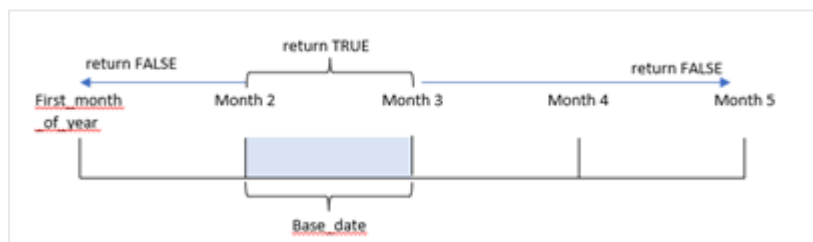
## inmonth

Questa funzione restituisce `True` se **timestamp** ricade all'interno del mese contenente **base\_date**.

**Sintassi:**

**InMonth** (timestamp, base\_date, period\_no)

*Schema della funzione indaytotime.*



In pratica, la funzione `inmonth()` determina se un insieme di date rientra in questo mese e restituisce un valore booleano basato su un valore `base_date` che identifica il mese.

### Casi di utilizzo

La funzione `inmonth()` restituisce un risultato booleano. In genere, questo tipo di funzione viene utilizzato come condizione in un file `if` expression. Restituisce un'aggregazione o un calcolo che dipende se una data si è verificata durante quel mese, inclusa la data in questione.

Ad esempio, la funzione `inmonth()` può essere utilizzata per identificare tutte le apparecchiature prodotte in un determinato giorno.

**Tipo di dati restituiti:** Boolean

In Qlik Sense, il valore booleano vero è rappresentato da -1 e il valore falso è rappresentato da 0.

### Argomenti

| Argomento              | Descrizione                                                                                                                                                                                                                                                                                                    |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| data e ora             | La data da confrontare con <code>base_date</code> .                                                                                                                                                                                                                                                            |
| <code>base_date</code> | La data utilizzata per valutare il mese. È importante notare che <code>base_date</code> può essere qualsiasi giorno entro un mese.                                                                                                                                                                             |
| <code>period_no</code> | Il mese può essere differito mediante <code>period_no</code> . <code>period_no</code> è un numero intero, in cui il valore 0 indica il mese che contiene <code>base_date</code> . I valori negativi di <code>period_no</code> indicano i mesi precedenti, mentre i valori positivi indicano i mesi successivi. |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempi di funzioni

| Esempio                                               | Risultato         |
|-------------------------------------------------------|-------------------|
| <code>inmonth ('25/01/2013', '01/01/2013', 0)</code>  | Restituisce True  |
| <code>inmonth ('25/01/2013', '23/04/2013', 0)</code>  | Restituisce False |
| <code>inmonth ('25/01/2013', '01/01/2013', -1)</code> | Restituisce False |
| <code>inmonth ('25/12/2012', '17/01/2013', -1)</code> | Restituisce True  |

### Esempio 1 - Nessun argomento aggiuntivo

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per la prima metà del 2022.
- Un'istruzione di caricamento precedente con una variabile aggiuntiva, 'in\_month', che determina se le transazioni sono state effettuate nel mese di aprile.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        inmonth(date,'04/01/2022', 0) as in_month
    ;
Load
*
Inline
[
id,date,amount
8188,'1/10/2022',37.23
8189,'1/14/2022',17.17
8190,'1/20/2022',88.27
8191,'1/22/2022',57.42
8192,'2/1/2022',53.80
8193,'2/2/2022',82.06
8194,'2/20/2022',40.39
8195,'4/11/2022',87.21
8196,'4/13/2022',95.93
8197,'4/15/2022',45.89
8198,'4/25/2022',36.23
8199,'5/20/2022',25.66
8200,'5/22/2022',82.77
8201,'6/19/2022',69.98
8202,'6/22/2022',76.11
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- in\_month

Esempi di funzioni

| date      | in_month |
|-----------|----------|
| 1/10/2022 | 0        |
| 1/14/2022 | 0        |
| 1/20/2022 | 0        |
| 1/22/2022 | 0        |

| date      | in_month |
|-----------|----------|
| 2/1/2022  | 0        |
| 2/2/2022  | 0        |
| 2/20/2022 | 0        |
| 4/11/2022 | -1       |
| 4/13/2022 | -1       |
| 4/15/2022 | -1       |
| 4/25/2022 | -1       |
| 5/20/2022 | 0        |
| 5/22/2022 | 0        |
| 6/19/2022 | 0        |
| 6/22/2022 | 0        |

Il campo 'in\_month' viene creato nell'istruzione di caricamento precedente mediante l'uso della funzione `inmonth()` e trasferendo il campo della data, una data con hard coding per il primo aprile `base_date` e un `period_no` di 0 come argomenti della funzione.

`base_date` identifica il mese che restituirà un risultato booleano TRUE. Pertanto, tutte le transazioni avvenute ad aprile restituiscono il valore TRUE, che viene convalidato nella tabella dei risultati.

### Esempio 2 - period\_no

Script di caricamento e risultati

#### Panoramica

Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, in questo esempio, bisogna creare un campo, "2\_months\_prior", che determina se le transazioni sono state effettuate due mesi prima di aprile.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';

Transactions:
Load
    *,
    inmonth(date,'04/01/2022', -2) as [2_months_prior]
Inline
[
id,date,amount
8188,'1/10/2022',37.23
8189,'1/14/2022',17.17
8190,'1/20/2022',88.27
8191,'1/22/2022',57.42
```

```
8192, '2/1/2022', 53.80
8193, '2/2/2022', 82.06
8194, '2/20/2022', 40.39
8195, '4/11/2022', 87.21
8196, '4/13/2022', 95.93
8197, '4/15/2022', 45.89
8198, '4/25/2022', 36.23
8199, '5/20/2022', 25.66
8200, '5/22/2022', 82.77
8201, '6/19/2022', 69.98
8202, '6/22/2022', 76.11
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- 2\_months\_prior

#### Esempi di funzioni

| date      | 2_months_prior |
|-----------|----------------|
| 1/10/2022 | 0              |
| 1/14/2022 | 0              |
| 1/20/2022 | 0              |
| 1/22/2022 | 0              |
| 2/1/2022  | -1             |
| 2/2/2022  | -1             |
| 2/20/2022 | -1             |
| 4/11/2022 | 0              |
| 4/13/2022 | 0              |
| 4/15/2022 | 0              |
| 4/25/2022 | 0              |
| 5/20/2022 | 0              |
| 5/22/2022 | 0              |
| 6/19/2022 | 0              |
| 6/22/2022 | 0              |

L'utilizzo di -2 come argomento `period_no` nella funzione `inmonth()` sposta il mese definito dall'argomento `base_date` due mesi prima. In questo esempio, il mese definito viene cambiato da aprile a febbraio.

Pertanto, qualsiasi transazione avvenuta a febbraio restituirà un risultato booleano TRUE.

### Esempio 3 - Oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, in questo esempio, il set di dati è invariato e viene caricato nell'applicazione. Il calcolo che determina se le transazioni sono avvenute tra ad aprile viene creato come misura in un oggetto grafico dell'applicazione.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'1/10/2022',37.23
```

```
8189,'1/14/2022',17.17
```

```
8190,'1/20/2022',88.27
```

```
8191,'1/22/2022',57.42
```

```
8192,'2/1/2022',53.80
```

```
8193,'2/2/2022',82.06
```

```
8194,'2/20/2022',40.39
```

```
8195,'4/11/2022',87.21
```

```
8196,'4/13/2022',95.93
```

```
8197,'4/15/2022',45.89
```

```
8198,'4/25/2022',36.23
```

```
8199,'5/20/2022',25.66
```

```
8200,'5/22/2022',82.77
```

```
8201,'6/19/2022',69.98
```

```
8202,'6/22/2022',76.11
```

```
];
```

#### Oggetto grafico

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

date

Per calcolare se una transazione è avvenuta ad aprile, creare la seguente misura:

```
=inmonth(date,'04/01/2022', 0)
```

### Risultati

|           | Esempi di funzioni             |
|-----------|--------------------------------|
| date      | =inmonth(date,'04/01/2022', 0) |
| 1/10/2022 | 0                              |
| 1/14/2022 | 0                              |
| 1/20/2022 | 0                              |
| 1/22/2022 | 0                              |
| 2/1/2022  | 0                              |
| 2/2/2022  | 0                              |
| 2/20/2022 | 0                              |
| 4/11/2022 | -1                             |
| 4/13/2022 | -1                             |
| 4/15/2022 | -1                             |
| 4/25/2022 | -1                             |
| 5/20/2022 | 0                              |
| 5/22/2022 | 0                              |
| 6/19/2022 | 0                              |
| 6/22/2022 | 0                              |

### Esempio 4 - Scenario

Script di caricamento e risultati

#### Panoramica

In questo esempio, un set di dati è caricato in una tabella denominata "Products". La tabella contiene i seguenti campi:

- ID prodotto
- Data di produzione
- Prezzo di costo

A causa di un errore dell'attrezzatura, i prodotti fabbricati nel mese di luglio 2022 erano difettosi. Il problema è stato risolto il 27 luglio 2022.

L'utente finale desidera un grafico che visualizzi, per mese, lo stato dei prodotti fabbricati "difettosi" (valore booleano TRUE) o "non difettosi" (valore booleano FALSE), oltre al il costo dei prodotti fabbricati quel mese.

## Script di caricamento

```

Products:
Load
*
Inline
[
product_id,manufacture_date,cost_price
8188,'1/19/2022',37.23
8189,'1/7/2022',17.17
8190,'2/28/2022',88.27
8191,'2/5/2022',57.42
8192,'3/16/2022',53.80
8193,'4/1/2022',82.06
8194,'5/7/2022',40.39
8195,'5/16/2022',87.21
8196,'6/15/2022',95.93
8197,'6/26/2022',45.89
8198,'7/9/2022',36.23
8199,'7/22/2022',25.66
8200,'7/23/2022',82.77
8201,'7/27/2022',69.98
8202,'8/2/2022',76.11
8203,'8/8/2022',25.12
8204,'8/19/2022',46.23
8205,'9/26/2022',84.21
8206,'10/14/2022',96.24
8207,'10/29/2022',67.67
];

```

## Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

=monthname(manufacture\_date)

Creare le seguenti misure:

- =sum(cost\_price)
- =if(only(inmonth(manufacture\_date,makedate(2022,07,01),0)),'Defective','Faultless')

1. Impostare la misura **Formattazione numero** su **Denaro**.
2. In **Aspetto**, disattivare **Totals**.

Tabella dei risultati

| monthname<br>(manufacture_date) | =if(only(inmonth(manufacture_date,makedate<br>(2022,07,01),0)),'Defective','Faultless') | =sum(cost_<br>price) |
|---------------------------------|-----------------------------------------------------------------------------------------|----------------------|
| Gen 2022                        | Non difettoso                                                                           | \$54.40              |
| Feb 2022                        | Non difettoso                                                                           | \$145.69             |
| Mar 2022                        | Non difettoso                                                                           | \$53.80              |

| monthname<br>(manufacture_date) | =if(only(inmonth(manufacture_date,makedate<br>(2022,07,01),0)), 'Defective', 'Faultless') | =sum(cost_<br>price) |
|---------------------------------|-------------------------------------------------------------------------------------------|----------------------|
| Apr 2022                        | Non difettoso                                                                             | \$82.06              |
| Mag 2022                        | Non difettoso                                                                             | \$127.60             |
| Giu 2022                        | Non difettoso                                                                             | \$141.82             |
| Lug 2022                        | Difettoso                                                                                 | \$214.64             |
| Ago 2022                        | Non difettoso                                                                             | \$147.46             |
| Set 2022                        | Non difettoso                                                                             | \$84.21              |
| Ott 2022                        | Non difettoso                                                                             | USD 163.91           |

La funzione `inmonth()` restituisce un valore booleano quando valuta le date di produzione di ciascun prodotto. Per qualsiasi prodotto fabbricato nel mese di luglio 2022, la funzione `inmonth()` restituisce un valore booleano TRUE e contrassegna i prodotti come "difettosi". I prodotti che restituiscono un valore FALSE, e che quindi non sono stati fabbricati a luglio, vengono contrassegnati come "non difettosi".

## inmonths

Questa funzione consente di verificare se un indicatore temporale cade nello stesso mese, bimestre, trimestre, quadrimestre o semestre di una data base. È inoltre possibile stabilire se l'indicatore temporale ricade all'interno di un periodo di tempo precedente o successivo.

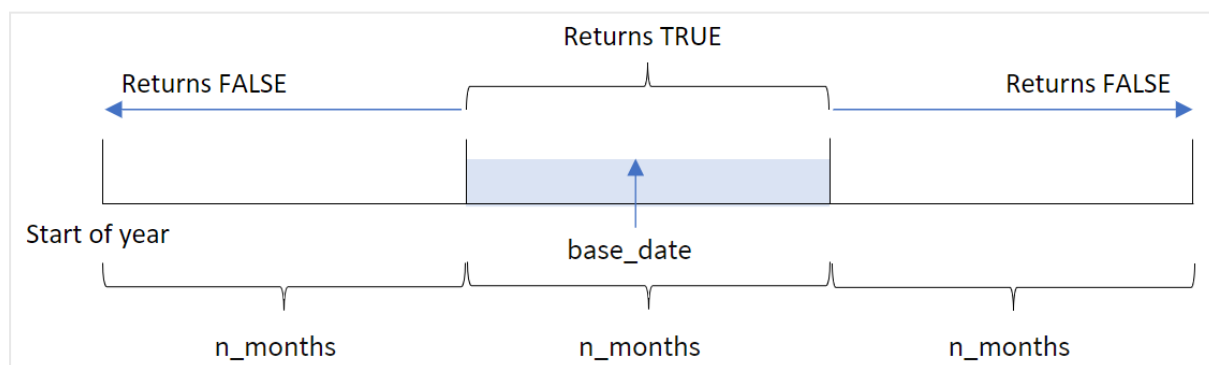
### Sintassi:

```
InMonths (n_months, timestamp, base_date, period_no [, first_month_of_year])
```

**Tipo di dati restituiti:** Boolean

In Qlik Sense, il valore booleano vero è rappresentato da -1 e il valore falso è rappresentato da 0.

*Schema della funzione inmonths().*



La funzione `inmonths()` divide l'anno in segmenti in base all'argomento `n_months` fornito. Quindi determina se ogni timestamp valutato rientra nello stesso segmento dell'argomento `base_date`. Se, invece, viene fornito un argomento `period_no`, la funzione determina se i timestamp cadono in un periodo precedente o successivo al periodo `base_date`.

I seguenti segmenti dell'anno sono disponibili nella funzione come argomenti `n_month`.

Argomenti `n_month`

| Periodo      | Numero di mesi |
|--------------|----------------|
| mese         | 1              |
| bimestre     | 2              |
| trimestre    | 3              |
| quattro mesi | 4              |
| semestre     | 6              |

### Casi di utilizzo

La funzione `inmonths()` restituisce un risultato booleano. In genere, questo tipo di funzione viene utilizzato come condizione in un file `if expression`. Utilizzando la funzione `inmonths()`, è possibile selezionare il periodo che si desidera valutare. Ad esempio, l'utente può identificare i prodotti fabbricati nel mese, trimestre o semestre di un determinato periodo.

**Tipo di dati restituiti:** Boolean

In Qlik Sense, il valore booleano vero è rappresentato da -1 e il valore falso è rappresentato da 0.

Argomenti

| Argomento                  | Descrizione                                                                                                                                                                                                                                                                                                                             |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>n_months</b>            | Il numero di mesi che definisce il periodo. Un numero intero o un'espressione la cui risoluzione è un numero intero corrispondente a: 1 (equivalente alla funzione <code>inmonth()</code> ), 2 (bimestre), 3 (equivalente alla funzione <code>inquarter()</code> ), 4 (quadrimestre) o 6 (semestre).                                    |
| <b>timestamp</b>           | La data da confrontare con <b>base_date</b> .                                                                                                                                                                                                                                                                                           |
| <b>base_date</b>           | La data utilizzata per valutare il periodo.                                                                                                                                                                                                                                                                                             |
| <b>period_no</b>           | Il periodo può essere differito mediante <b>period_no</b> , un numero intero, o un'espressione la cui risoluzione è un numero intero, in cui il valore 0 indica il periodo che contiene <b>base_date</b> . I valori negativi di <b>period_no</b> indicano i periodi precedenti, mentre i valori positivi indicano i periodi successivi. |
| <b>first_month_of_year</b> | Se si intende utilizzare anni (fiscali) che non iniziano a gennaio, indicare un valore compreso tra 2 e 12 in <b>first_month_of_year</b> .                                                                                                                                                                                              |

È possibile utilizzare i seguenti valori per impostare il primo mese dell'anno nell'argomento `first_month_of_year`:

valori first\_month\_of\_year

| Mese      | Valore |
|-----------|--------|
| Febbraio  | 2      |
| Marzo     | 3      |
| Aprile    | 4      |
| Maggio    | 5      |
| Giugno    | 6      |
| Luglio    | 7      |
| Agosto    | 8      |
| Settembre | 9      |
| Ottobre   | 10     |
| Novembre  | 11     |
| Dicembre  | 12     |

## Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempi di funzioni

| Esempio                                                 | Risultato                                                                                                                                                                                                |
|---------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>inmonths(4, '01/25/2013', '04/25/2013', 0)</code> | Restituisce TRUE. Perché il valore del timestamp, 25/01/2013, si trova nel periodo di quattro mesi dal 01/01/2013 al 30/04/2013, in cui si trova il valore di <code>base_date</code> , resta 25/04/2013. |
| <code>inmonths(4, '05/25/2013', '04/25/2013', 0)</code> | Restituisce FALSE. Perché il 25/05/2013 non rientra nello stesso periodo dell'esempio precedente.                                                                                                        |

| Esempio                                                                | Risultato                                                                                                                                                                                                                                    |
|------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>inmonths(4,<br/>'11/25/2012',<br/>'02/01/2013', -1<br/>)</code>  | Restituisce TRUE. Poiché il valore di <code>period_no</code> , -1, sposta il periodo di ricerca indietro di un periodo di quattro mesi (il valore di <code>n-mesi</code> ), il che rende il periodo di ricerca dal 01/09/2012 al 31/12/2012. |
| <code>inmonths(4,<br/>'05/25/2006',<br/>'03/01/2006', 0,<br/>3)</code> | Restituisce TRUE. Perché il valore di <code>first_month_of_year</code> è impostato su 3, che rende il periodo di ricerca dal 01/03/2006 al 30/07/2006 invece che dal 01/01/2006 al 30/04/2006.                                               |

## Esempio 1 - Nessun argomento aggiuntivo

Script di caricamento e risultati

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2022 viene caricato in una tabella denominata 'Transactions'.
- Un caricamento precedente con una variabile aggiuntiva, 'in\_months', che determina quali transazioni hanno avuto luogo nello stesso trimestre del 15 maggio 2022.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
    *,
    inmonths(3,date,'05/15/2022', 0) as in_months
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'2/19/2022',37.23
```

```
8189,'3/7/2022',17.17
```

```
8190,'3/30/2022',88.27
```

```
8191,'4/5/2022',57.42
```

```
8192,'4/16/2022',53.80
```

```
8193,'5/1/2022',82.06
```

```
8194,'5/7/2022',40.39
```

```
8195,'5/22/2022',87.21
```

```
8196,'6/15/2022',95.93
```

```
8197,'6/26/2022',45.89
```

```
8198,'7/9/2022',36.23
```

```
8199,'7/22/2022',25.66
```

```
8200, '7/23/2022', 82.77
8201, '7/27/2022', 69.98
8202, '8/2/2022', 76.11
8203, '8/8/2022', 25.12
8204, '8/19/2022', 46.23
8205, '9/26/2022', 84.21
8206, '10/14/2022', 96.24
8207, '10/29/2022', 67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

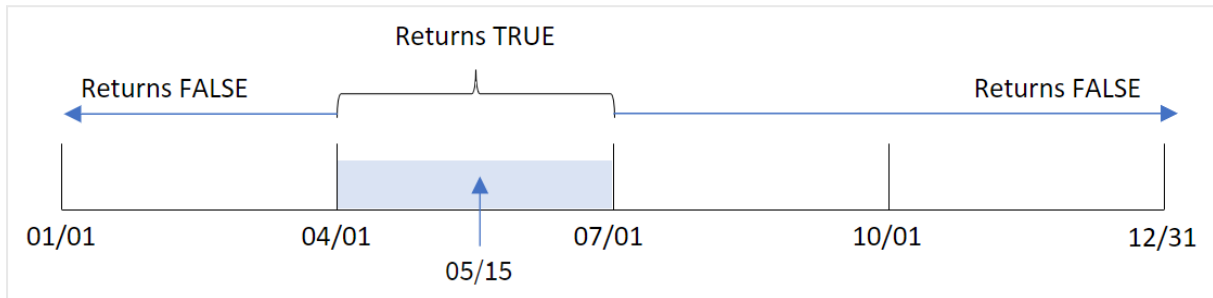
- date
- in\_months

Tabella dei risultati

| date       | in_months |
|------------|-----------|
| 2/19/2022  | 0         |
| 3/7/2022   | 0         |
| 3/30/2022  | 0         |
| 4/5/2022   | -1        |
| 4/16/2022  | -1        |
| 5/1/2022   | -1        |
| 5/7/2022   | -1        |
| 5/22/2022  | -1        |
| 6/15/2022  | -1        |
| 6/26/2022  | -1        |
| 7/9/2022   | 0         |
| 7/22/2022  | 0         |
| 7/23/2022  | 0         |
| 7/27/2022  | 0         |
| 8/2/2022   | 0         |
| 8/8/2022   | 0         |
| 8/19/2022  | 0         |
| 9/26/2022  | 0         |
| 10/14/2022 | 0         |
| 10/29/2022 | 0         |

Il campo 'in\_months' viene creato nell'istruzione di caricamento precedente utilizzando la funzione `inmonths()`. Il primo argomento fornito è 3, che divide l'anno in segmenti di trimestre. Il secondo argomento identifica il campo da valutare, in questo esempio il campo `date`. Il terzo argomento è una data con hard coding per il 15 maggio, che è `base_date` e `period_no` di 0 è l'argomento finale.

*Schema della funzione `inmonths()` con segmenti trimestrali*



Maggio rientra nel secondo trimestre dell'anno. Pertanto, qualsiasi transazione effettuata tra il 1° aprile e il 30 giugno restituirà un risultato booleano pari a TRUE. Questo viene convalidato nella tabella dei risultati.

### Esempio 2 - `period_no`

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2022 viene caricato in una tabella denominata 'Transactions'.
- Un caricamento precedente con una variabile aggiuntiva, 'previous\_quarter', che determina se le transazioni hanno avuto luogo nel trimestre precedente al 15 maggio 2022.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
inmonths(3,date,'05/15/2022', -1) as previous_quarter
```

```
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'2/19/2022',37.23
```

```
8189,'3/7/2022',17.17
```

```
8190,'3/30/2022',88.27
```

```
8191, '4/5/2022', 57.42
8192, '4/16/2022', 53.80
8193, '5/1/2022', 82.06
8194, '5/7/2022', 40.39
8195, '5/22/2022', 87.21
8196, '6/15/2022', 95.93
8197, '6/26/2022', 45.89
8198, '7/9/2022', 36.23
8199, '7/22/2022', 25.66
8200, '7/23/2022', 82.77
8201, '7/27/2022', 69.98
8202, '8/2/2022', 76.11
8203, '8/8/2022', 25.12
8204, '8/19/2022', 46.23
8205, '9/26/2022', 84.21
8206, '10/14/2022', 96.24
8207, '10/29/2022', 67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- previous\_quarter

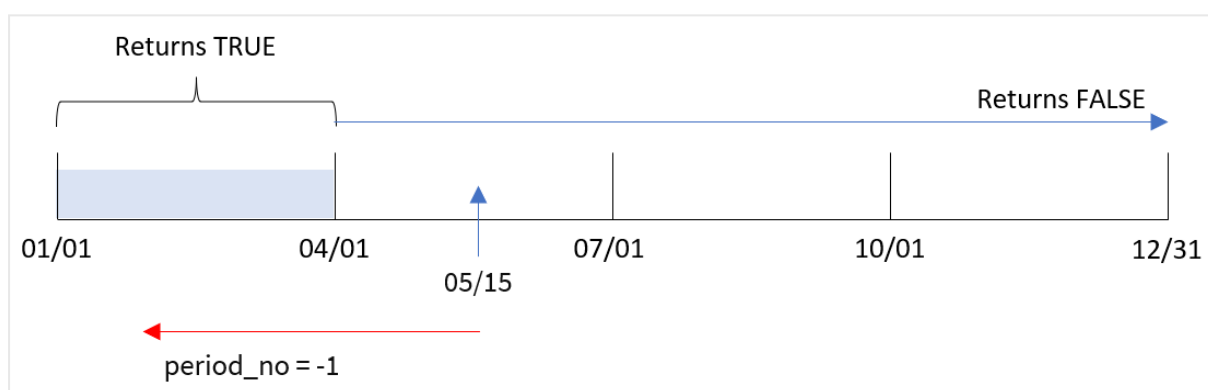
Tabella dei risultati

| date      | trimestre precedente |
|-----------|----------------------|
| 2/19/2022 | -1                   |
| 3/7/2022  | -1                   |
| 3/30/2022 | -1                   |
| 4/5/2022  | 0                    |
| 4/16/2022 | 0                    |
| 5/1/2022  | 0                    |
| 5/7/2022  | 0                    |
| 5/22/2022 | 0                    |
| 6/15/2022 | 0                    |
| 6/26/2022 | 0                    |
| 7/9/2022  | 0                    |
| 7/22/2022 | 0                    |
| 7/23/2022 | 0                    |
| 7/27/2022 | 0                    |

| date       | trimestre precedente |
|------------|----------------------|
| 8/2/2022   | 0                    |
| 8/8/2022   | 0                    |
| 8/19/2022  | 0                    |
| 9/26/2022  | 0                    |
| 10/14/2022 | 0                    |
| 10/29/2022 | 0                    |

La funzione valuta se le transazioni si sono verificate nel primo trimestre dell'anno utilizzando -1 come argomento `period_no` della funzione `inmonths()`. Il 15 maggio è il `base_date` e ricade nel secondo trimestre dell'anno (aprile-giugno).

Schema della funzione `inmonths()` con segmenti trimestrali e `period_no` impostato su -1



Pertanto, qualsiasi transazione che si verifichi tra gennaio e marzo restituirà un risultato booleano di TRUE.

### Esempio 3 - first\_month\_of\_year

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2022 viene caricato in una tabella denominata 'Transactions'.
- Un caricamento precedente con una variabile aggiuntiva, 'in\_months', che determina quali transazioni hanno avuto luogo nello stesso trimestre del 15 maggio 2022.

In questo esempio, il criterio organizzativo prevede che marzo sia il primo mese dell'anno finanziario.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        inmonths(3,date,'05/15/2022', 0, 3) as in_months
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'2/19/2022',37.23
```

```
8189,'3/7/2022',17.17
```

```
8190,'3/30/2022',88.27
```

```
8191,'4/5/2022',57.42
```

```
8192,'4/16/2022',53.80
```

```
8193,'5/1/2022',82.06
```

```
8194,'5/7/2022',40.39
```

```
8195,'5/22/2022',87.21
```

```
8196,'6/15/2022',95.93
```

```
8197,'6/26/2022',45.89
```

```
8198,'7/9/2022',36.23
```

```
8199,'7/22/2022',25.66
```

```
8200,'7/23/2022',82.77
```

```
8201,'7/27/2022',69.98
```

```
8202,'8/2/2022',76.11
```

```
8203,'8/8/2022',25.12
```

```
8204,'8/19/2022',46.23
```

```
8205,'9/26/2022',84.21
```

```
8206,'10/14/2022',96.24
```

```
8207,'10/29/2022',67.67
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- in\_months

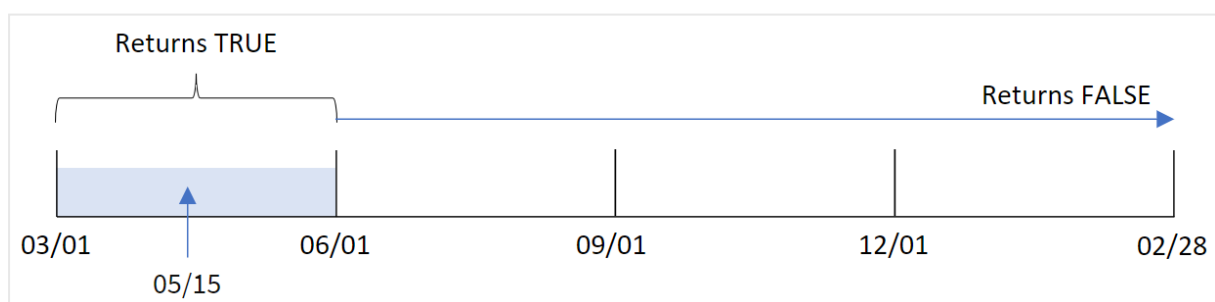
Tabella dei risultati

| date      | in_months |
|-----------|-----------|
| 2/19/2022 | 0         |
| 3/7/2022  | -1        |
| 3/30/2022 | -1        |
| 4/5/2022  | -1        |

| date       | in_months |
|------------|-----------|
| 4/16/2022  | -1        |
| 5/1/2022   | -1        |
| 5/7/2022   | -1        |
| 5/22/2022  | -1        |
| 6/15/2022  | 0         |
| 6/26/2022  | 0         |
| 7/9/2022   | 0         |
| 7/22/2022  | 0         |
| 7/23/2022  | 0         |
| 7/27/2022  | 0         |
| 8/2/2022   | 0         |
| 8/8/2022   | 0         |
| 8/19/2022  | 0         |
| 9/26/2022  | 0         |
| 10/14/2022 | 0         |
| 10/29/2022 | 0         |

Utilizzando 3 come argomento `first_month_of_year` nella funzione `inmonths()`, la funzione inizia l'anno il 1° marzo. La funzione `inmonths()` divide quindi l'anno in trimestri: mar-mag, giu-ago, set-nov, dic-feb. Pertanto, il 15 maggio cade nel primo trimestre dell'anno (marzo-maggio).

*Schema della funzione `inmonths()` con marzo impostato come primo mese dell'anno*



Qualsiasi transazione che si verifica in questi mesi restituirà un risultato booleano pari a TRUE.

### Esempio 4 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, in questo esempio, il set di dati è invariato e viene caricato nell'applicazione. Il calcolo che determina se le transazioni sono avvenute nello stesso trimestre del 15 maggio 2022 viene creato come misura in un grafico dell'app.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'2/19/2022',37.23
```

```
8189,'3/7/2022',17.17
```

```
8190,'3/30/2022',88.27
```

```
8191,'4/5/2022',57.42
```

```
8192,'4/16/2022',53.80
```

```
8193,'5/1/2022',82.06
```

```
8194,'5/7/2022',40.39
```

```
8195,'5/22/2022',87.21
```

```
8196,'6/15/2022',95.93
```

```
8197,'6/26/2022',45.89
```

```
8198,'7/9/2022',36.23
```

```
8199,'7/22/2022',25.66
```

```
8200,'7/23/2022',82.77
```

```
8201,'7/27/2022',69.98
```

```
8202,'8/2/2022',76.11
```

```
8203,'8/8/2022',25.12
```

```
8204,'8/19/2022',46.23
```

```
8205,'9/26/2022',84.21
```

```
8206,'10/14/2022',96.24
```

```
8207,'10/29/2022',67.67
```

```
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- date

Per calcolare se le transazioni sono avvenute nello stesso trimestre del 15 maggio, creare la seguente misura:

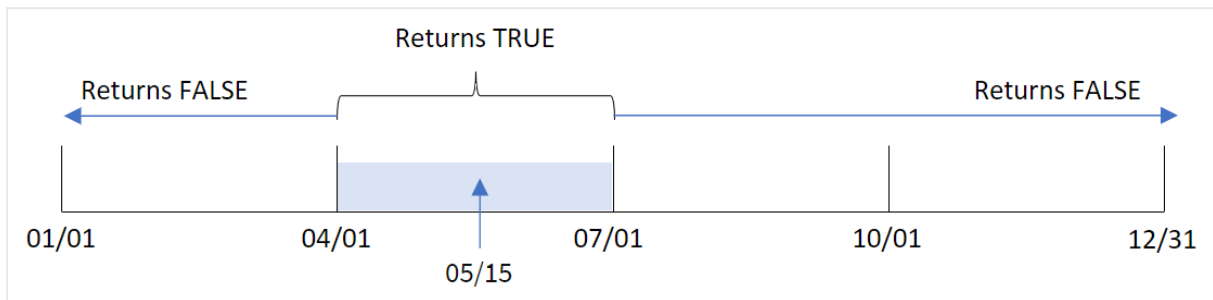
```
=inmonths(3,date,'05/15/2022', 0)
```

Tabella dei risultati

| date       | =inmonths(3,date,'05/15/2022', 0) |
|------------|-----------------------------------|
| 2/19/2022  | 0                                 |
| 3/7/2022   | 0                                 |
| 3/30/2022  | 0                                 |
| 4/5/2022   | -1                                |
| 4/16/2022  | -1                                |
| 5/1/2022   | -1                                |
| 5/7/2022   | -1                                |
| 5/22/2022  | -1                                |
| 6/15/2022  | -1                                |
| 6/26/2022  | -1                                |
| 7/9/2022   | 0                                 |
| 7/22/2022  | 0                                 |
| 7/23/2022  | 0                                 |
| 7/27/2022  | 0                                 |
| 8/2/2022   | 0                                 |
| 8/8/2022   | 0                                 |
| 8/19/2022  | 0                                 |
| 9/26/2022  | 0                                 |
| 10/14/2022 | 0                                 |
| 10/29/2022 | 0                                 |

Il campo 'in\_months' viene creato nel grafico mediante l'utilizzo della funzione `inmonths()`. Il primo argomento fornito è 3, che divide l'anno in segmenti di trimestre. Il secondo argomento identifica il campo da valutare, in questo esempio il campo `data`. Il terzo argomento è una data con hard coding per il 15 maggio, che è `base_date` e `period_no` di 0 è l'argomento finale.

Schema della funzione `inmonths()` con segmenti trimestrali



Maggio rientra nel secondo trimestre dell'anno. Pertanto, qualsiasi transazione effettuata tra il 1° aprile e il 30 giugno restituirà un risultato booleano pari a `TRUE`. Questo viene convalidato nella tabella dei risultati.

### Esempio 5 - Scenario

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella chiamata 'Products'.
- La tabella contiene i seguenti campi:
  - ID prodotto
  - tipo di prodotto
  - data di produzione
  - prezzo di costo

L'utente finale desidera un grafico che visualizzi, per tipologia di prodotto, il costo dei prodotti fabbricati nel primo segmento del 2021. L'utente vorrebbe poter definire la lunghezza di questo segmento.

#### Script di caricamento

```
SET vPeriod = 1;
```

```
Products:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
product_id,product_type,manufacture_date,cost_price
```

```
8188,product A,'2/19/2022',37.23
```

```
8189,product D,'3/7/2022',17.17
```

```
8190,product C,'3/30/2022',88.27
```

```
8191,product B,'4/5/2022',57.42
```

```
8192,product D,'4/16/2022',53.80
```

```
8193,product D,'5/1/2022',82.06
```

```
8194,product A,'5/7/2022',40.39
```

```
8195,product B,'5/22/2022',87.21
8196,product C,'6/15/2022',95.93
8197,product B,'6/26/2022',45.89
8198,product C,'7/9/2022',36.23
8199,product D,'7/22/2022',25.66
8200,product D,'7/23/2022',82.77
8201,product A,'7/27/2022',69.98
8202,product A,'8/2/2022',76.11
8203,product B,'8/8/2022',25.12
8204,product B,'8/19/2022',46.23
8205,product B,'9/26/2022',84.21
8206,product C,'10/14/2022',96.24
8207,product D,'10/29/2022',67.67
];
```

## Risultati

Caricare i dati e aprire un foglio.

All'inizio dello script di caricamento, viene creata una variabile, `vPeriod`, legata al controllo dell'input variabile.

Procedere come segue:

1. Nel pannello delle risorse, fare clic su **Oggetti personalizzati**.
2. Selezionare **Qlik Dashboard bundle** e creare un oggetto **Input variabile**.
3. Immettere un titolo per l'oggetto grafico.
4. In **Variabile**, selezionare **vPeriod** come nome e impostare l'oggetto in modo che venga visualizzato come **Elenco a discesa**.
5. Sotto **Valori**, fare clic sui valori **Dinamici**. Inserire quanto segue:  
`= '1~month|2~bi-month|3~quarter|4~tertiar|6~half-year'`.
6. Aggiungere una nuova tabella al foglio.
7. Nella sezione **Dati** nel pannello delle proprietà, aggiungere `product_type` come dimensione.
8. Aggiungere l'espressione seguente come misura:  
`=sum(if(inmonths($(vPeriod),manufacture_date,makedate(2022,01,01),0),cost_price,0))`
9. Impostare la misura **Formattazione numero** su **Denaro**.

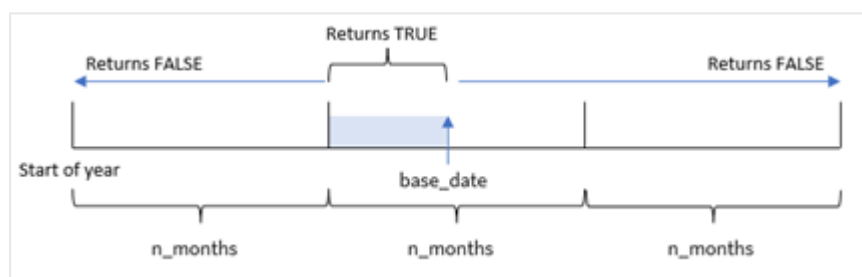
Tabella dei risultati

| tipo_<br>prodotto | =sum(if(inmonths(\$(vPeriod),manufacture_date,makedate(2022,01,01),0),cost_price,0)) |
|-------------------|--------------------------------------------------------------------------------------|
| prodotto A        | \$88.27                                                                              |
| prodotto B        | \$37.23                                                                              |
| prodotto C        | \$17.17                                                                              |
| prodotto D        | \$0.00                                                                               |

Questa funzione stabilisce se un indicatore temporale ricade all'interno della parte di un periodo di un mese, bimestre, trimestre, quadrimestre o semestre fino a includere l'ultimo millisecondo di `base_date`. È inoltre possibile stabilire se l'indicatore temporale ricade all'interno di un periodo di tempo precedente o successivo.

**InMonths** (n months, timestamp, base date, period no[, first month of year ])

*Schema della funzione inmonthstodate.*

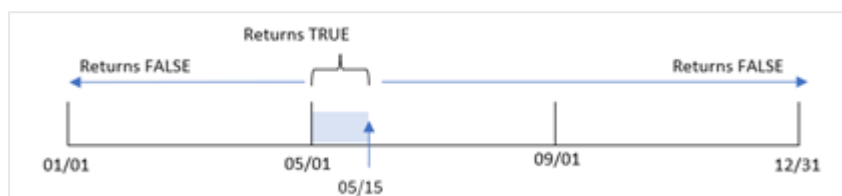


| Argomento                  | Descrizione                                                                                                                                                                                                                                                                                                                             |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>n_months</b>            | Il numero di mesi che definisce il periodo. Un numero intero o un'espressione la cui risoluzione è un numero intero corrispondente a: 1 (equivalente alla funzione inmonth()), 2 (bimestre), 3 (equivalente alla funzione inquarter()), 4 (quadrimestre) o 6 (semestre).                                                                |
| <b>timestamp</b>           | La data da confrontare con <b>base_date</b> .                                                                                                                                                                                                                                                                                           |
| <b>base_date</b>           | La data utilizzata per valutare il periodo.                                                                                                                                                                                                                                                                                             |
| <b>period_no</b>           | Il periodo può essere differito mediante <b>period_no</b> , un numero intero, o un'espressione la cui risoluzione è un numero intero, in cui il valore 0 indica il periodo che contiene <b>base_date</b> . I valori negativi di <b>period_no</b> indicano i periodi precedenti, mentre i valori positivi indicano i periodi successivi. |
| <b>first_month_of_year</b> | Se si intende utilizzare anni (fiscali) che non iniziano a gennaio, indicare un valore compreso tra 2 e 12 in <b>first_month_of_year</b> .                                                                                                                                                                                              |

Nella funzione `inmonthstodate()`, `base_date` agisce come punto finale del particolare segmento di anno di cui fa parte.

Ad esempio, se l'anno è stato suddiviso in segmenti terziari e la `base_date` era il 15 maggio, qualsiasi timestamp tra l'inizio di gennaio e la fine di aprile restituirà un risultato booleano di FALSE. Le date comprese tra il 1° e il 15 maggio restituirebbero TRUE. Il resto dell'anno restituirebbe FALSE.

*Schema dell'intervallo di risultati booleani della funzione `inmonthstodate`.*



I seguenti segmenti dell'anno sono disponibili nella funzione come argomenti `n_month`.

Argomenti `n_month`

| Periodo      | Numero di mesi |
|--------------|----------------|
| mese         | 1              |
| bimestre     | 2              |
| trimestre    | 3              |
| quadrimestre | 4              |
| semestre     | 6              |

### Casi di utilizzo

La funzione `inmonthstodate()` restituisce un risultato booleano. In genere, questo tipo di funzione viene utilizzato come condizione in un file `if expression`. Utilizzando la funzione `inmonthstodate()`, è possibile selezionare il periodo che si desidera valutare. Ad esempio, fornendo una variabile di input che consenta all'utente di identificare i prodotti fabbricati nel mese, trimestre o semestre di un periodo, fino a una certa data.

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è

impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

Esempi di funzioni

| Esempio                                                          | Risultato                                                                                                                                                                                                                  |
|------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>inmonthstodate(4, '01/25/2013', '04/25/2013', 0)</code>    | Restituisce il valore di True, perché il valore di timestamp, 01/25/2013 ricade entro il periodo di quattro mesi di 01/01/2013 fino alla fine di 04/25/2013, in cui ricade il valore di base_date, 04/25/2013.             |
| <code>inmonthstodate(4, '04/26/2013', '04/25/2006', 0)</code>    | Restituisce False, perché 04/26/2013 non ricade nello stesso periodo dell'esempio precedente.                                                                                                                              |
| <code>inmonthstodate(4, '09/25/2005', '02/01/2006', -1)</code>   | Restituisce True, perché il valore di period_no, -1, fa slittare il periodo della ricerca indietro di un periodo di quattro mesi (il valore di n-months) e questo cambia il periodo di ricerca da 01/09/2005 a 02/01/2006. |
| <code>inmonthstodate(4, '04/25/2006', '06/01/2006', 0, 3)</code> | Restituisce True, perché il valore di first_month_of_year è impostato su 3 e ciò cambia il periodo di ricerca da 03/01/2006 a 06/01/2006 anziché da 05/01/2006 in 06/01/2006.                                              |

### Esempio 1 - Nessun argomento aggiuntivo

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2022 viene caricato in una tabella denominata 'Transactions'.
- Un campo data nel formato DateFormat della variabile di sistema (MM/DD/YYYY).
- Un'istruzione di caricamento precedente contenente:
  - La funzione `inmonthstodate()` impostata come campo, 'in\_months\_to\_date'. Questo determina quali transazioni sono state effettuate nel trimestre fino al 15 maggio 2022.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
    *
    inmonthstodate(3,date,'05/15/2022', 0) as in_months_to_date
    ;
Load
```

```
*
inline
[
id,date,amount
8188,'1/19/2022',37.23
8189,'1/7/2022',17.17
8190,'2/28/2022',88.27
8191,'2/5/2022',57.42
8192,'3/16/2022',53.80
8193,'4/1/2022',82.06
8194,'5/7/2022',40.39
8195,'5/16/2022',87.21
8196,'6/15/2022',95.93
8197,'6/26/2022',45.89
8198,'7/9/2022',36.23
8199,'7/22/2022',25.66
8200,'7/23/2022',82.77
8201,'7/27/2022',69.98
8202,'8/2/2022',76.11
8203,'8/8/2022',25.12
8204,'8/19/2022',46.23
8205,'9/26/2022',84.21
8206,'10/14/2022',96.24
8207,'10/29/2022',67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- in\_months\_to\_date

Tabella dei risultati

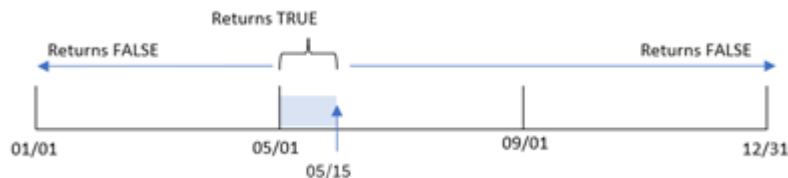
| date      | in_months_to_date |
|-----------|-------------------|
| 1/7/2022  | 0                 |
| 1/19/2022 | 0                 |
| 2/5/2022  | 0                 |
| 2/28/2022 | 0                 |
| 3/16/2022 | 0                 |
| 4/1/2022  | -1                |
| 5/7/2022  | -1                |
| 5/16/2022 | 0                 |
| 6/15/2022 | 0                 |
| 6/26/2022 | 0                 |

| date       | in_months_to_date |
|------------|-------------------|
| 7/9/2022   | 0                 |
| 7/22/2022  | 0                 |
| 7/23/2022  | 0                 |
| 7/27/2022  | 0                 |
| 8/2/2022   | 0                 |
| 8/8/2022   | 0                 |
| 8/19/2022  | 0                 |
| 9/26/2022  | 0                 |
| 10/14/2022 | 0                 |
| 10/29/2022 | 0                 |

Il campo 'in\_months\_to\_date' viene creato nell'istruzione LOAD precedente utilizzando la funzione `inmonthstodate()`.

Il primo argomento fornito è 3, che divide l'anno in segmenti di trimestre. Il secondo argomento identifica il campo da valutare. Il terzo argomento è una data codificata per il 15 maggio, ovvero il `base_date` che definisce il confine finale del segmento. Un `period_no` di 0 rappresenta l'argomento finale.

*Schema della funzione `inmonthstodate` senza argomenti aggiuntivi.*



Qualsiasi transazione avvenuta tra il 1° aprile e il 15 maggio restituisce il risultato booleano di TRUE. Le date di transazione al di fuori di tale periodo restituiscono FALSE.

### Esempio 2 - `period_no`

Script di caricamento e risultati

#### Panoramica

Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, in questo esempio, il compito è quello di creare un campo, 'previous\_qtr\_to\_date' che determini se le transazioni hanno avuto luogo un trimestre prima del 15 maggio.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
    *,
    inmonthstodate(3,date,'05/15/2022', -1) as previous_qtr_to_date
    ;
Load
*
Inline
[
id,date,amount
8188,'1/19/2022',37.23
8189,'1/7/2022',17.17
8190,'2/28/2022',88.27
8191,'2/5/2022',57.42
8192,'3/16/2022',53.80
8193,'4/1/2022',82.06
8194,'5/7/2022',40.39
8195,'5/16/2022',87.21
8196,'6/15/2022',95.93
8197,'6/26/2022',45.89
8198,'7/9/2022',36.23
8199,'7/22/2022',25.66
8200,'7/23/2022',82.77
8201,'7/27/2022',69.98
8202,'8/2/2022',76.11
8203,'8/8/2022',25.12
8204,'8/19/2022',46.23
8205,'9/26/2022',84.21
8206,'10/14/2022',96.24
8207,'10/29/2022',67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- previous\_qtr\_to\_date

Tabella dei risultati

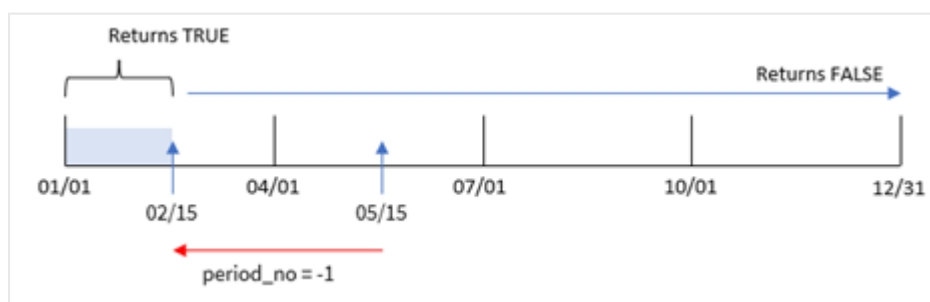
| date      | trimestre_precedente_a_oggi |
|-----------|-----------------------------|
| 1/7/2022  | -1                          |
| 1/19/2022 | -1                          |
| 2/5/2022  | -1                          |
| 2/28/2022 | 0                           |

| date       | trimestre_precedente_a_oggi |
|------------|-----------------------------|
| 3/16/2022  | 0                           |
| 4/1/2022   | 0                           |
| 5/7/2022   | 0                           |
| 5/16/2022  | 0                           |
| 6/15/2022  | 0                           |
| 6/26/2022  | 0                           |
| 7/9/2022   | 0                           |
| 7/22/2022  | 0                           |
| 7/23/2022  | 0                           |
| 7/27/2022  | 0                           |
| 8/2/2022   | 0                           |
| 8/8/2022   | 0                           |
| 8/19/2022  | 0                           |
| 9/26/2022  | 0                           |
| 10/14/2022 | 0                           |
| 10/29/2022 | 0                           |

Utilizzando -1 come argomento `period_no` della funzione `inmonthstodate()`, la funzione sposta di un trimestre i confini del segmento dell'anno di confronto.

Il 15 maggio rientra nel secondo trimestre dell'anno, pertanto il segmento inizialmente corrisponde al periodo compreso tra il 1° aprile e il 15 maggio. L'argomento `period_no` compensa questo segmento con un valore negativo di tre mesi. I limiti di data diventano dal 1° gennaio al 15 febbraio.

*Schema della funzione `inmonthstodate` con un valore `period_no` impostato a -1.*



Pertanto, qualsiasi transazione che si verifichi tra il 1° gennaio e il 15 febbraio restituirà il risultato booleano di TRUE.

### Esempio 3 - `first_month_of_year`

Script di caricamento e risultati

### Panoramica

Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.

In questo esempio, il criterio organizzativo prevede che marzo sia il primo mese dell'anno finanziario.

Crea un campo, 'in\_months\_to\_date', che determina quali transazioni sono avvenute nello stesso trimestre fino al 15 maggio 2022.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        inmonthstodate(3,date,'05/15/2022', 0,3) as in_months_to_date
    ;
Load
*
Inline
[
id,date,amount
8188,'1/19/2022',37.23
8189,'1/7/2022',17.17
8190,'2/28/2022',88.27
8191,'2/5/2022',57.42
8192,'3/16/2022',53.80
8193,'4/1/2022',82.06
8194,'5/7/2022',40.39
8195,'5/16/2022',87.21
8196,'6/15/2022',95.93
8197,'6/26/2022',45.89
8198,'7/9/2022',36.23
8199,'7/22/2022',25.66
8200,'7/23/2022',82.77
8201,'7/27/2022',69.98
8202,'8/2/2022',76.11
8203,'8/8/2022',25.12
8204,'8/19/2022',46.23
8205,'9/26/2022',84.21
8206,'10/14/2022',96.24
8207,'10/29/2022',67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- in\_months\_to\_date

Tabella dei risultati

| date       | trimestre_precedente_a_oggi |
|------------|-----------------------------|
| 1/7/2022   | 0                           |
| 1/19/2022  | 0                           |
| 2/5/2022   | 0                           |
| 2/28/2022  | 0                           |
| 3/16/2022  | -1                          |
| 4/1/2022   | -1                          |
| 5/7/2022   | -1                          |
| 5/16/2022  | 0                           |
| 6/15/2022  | 0                           |
| 6/26/2022  | 0                           |
| 7/9/2022   | 0                           |
| 7/22/2022  | 0                           |
| 7/23/2022  | 0                           |
| 7/27/2022  | 0                           |
| 8/2/2022   | 0                           |
| 8/8/2022   | 0                           |
| 8/19/2022  | 0                           |
| 9/26/2022  | 0                           |
| 10/14/2022 | 0                           |
| 10/29/2022 | 0                           |

Utilizzando 3 come argomento `first_month_of_year` nella funzione `inmonthstodate()`, questa inizia l'anno il 1° marzo e poi divide l'anno in trimestri in base al primo argomento fornito. Pertanto, i segmenti del trimestre sono:

- Mag-Mag
- Giu-Ago
- Set-Nov
- Dic-Feb

Il `base_date` del 15 maggio segmenta quindi il trimestre da marzo a maggio impostando il limite finale al 15 maggio.

Schema della funzione `inmonthstodate` con marzo impostato come primo mese dell'anno.



Pertanto, qualsiasi transazione che si verifichi tra il 1° marzo e il 15 maggio restituirà un risultato booleano di TRUE, mentre le transazioni con date al di fuori di questi confini restituiranno un valore di FALSE.

### Esempio 4 - Esempio grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.

In questo esempio, il set di dati è invariato e viene caricato nell'app. L'attività richiede di creare un calcolo che determina se le transazioni sono avvenute nello stesso trimestre del 15 maggio come misura in un grafico dell'app.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

Transactions:

Load

\*

Inline

[

id,date,amount

8188,'1/19/2022',37.23

8189,'1/7/2022',17.17

8190,'2/28/2022',88.27

8191,'2/5/2022',57.42

8192,'3/16/2022',53.80

8193,'4/1/2022',82.06

8194,'5/7/2022',40.39

8195,'5/16/2022',87.21

8196,'6/15/2022',95.93

8197,'6/26/2022',45.89

8198,'7/9/2022',36.23

8199,'7/22/2022',25.66

8200,'7/23/2022',82.77

8201,'7/27/2022',69.98

8202,'8/2/2022',76.11

8203,'8/8/2022',25.12

8204,'8/19/2022',46.23

8205,'9/26/2022',84.21

```
8206, '10/14/2022', 96.24  
8207, '10/29/2022', 67.67  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

date

Per calcolare se le transazioni sono avvenute nello stesso trimestre del 15 maggio, creare la seguente misura:

```
=inmonthstodate(3,date,'05/15/2022', 0)
```

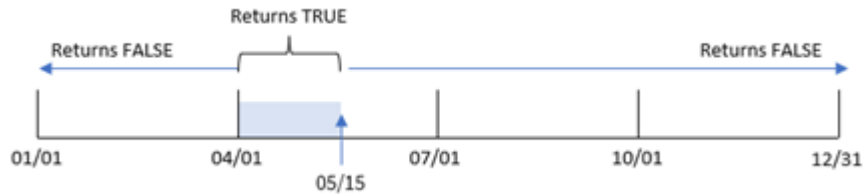
Tabella dei risultati

| date       | =inmonthstodate(3,date,'05/15/2022', 0) |
|------------|-----------------------------------------|
| 1/7/2022   | 0                                       |
| 1/19/2022  | 0                                       |
| 2/5/2022   | 0                                       |
| 2/28/2022  | 0                                       |
| 3/16/2022  | 0                                       |
| 4/1/2022   | -1                                      |
| 5/7/2022   | -1                                      |
| 5/16/2022  | 0                                       |
| 6/15/2022  | 0                                       |
| 6/26/2022  | 0                                       |
| 7/9/2022   | 0                                       |
| 7/22/2022  | 0                                       |
| 7/23/2022  | 0                                       |
| 7/27/2022  | 0                                       |
| 8/2/2022   | 0                                       |
| 8/8/2022   | 0                                       |
| 8/19/2022  | 0                                       |
| 9/26/2022  | 0                                       |
| 10/14/2022 | 0                                       |
| 10/29/2022 | 0                                       |

La misura ''in\_months\_to\_date' viene creata nell'oggetto grafico utilizzando la funzione inmonthstodate().

Il primo argomento fornito è 3, che divide l'anno in segmenti di trimestre. Il secondo argomento identifica il campo da valutare. Il terzo argomento è una data codificata per il 15 maggio, ovvero il `base_date`, che definisce il confine finale del segmento. Un `period_no` di 0 rappresenta l'argomento finale.

*Schema della funzione `inmonthstodate` con segmenti trimestrali.*



Qualsiasi transazione avvenuta tra il 1° aprile e il 15 maggio restituirà il risultato booleano di TRUE. Le date di transazione al di fuori di tale segmento restituiscono FALSE.

### Esempio 5 - Scenario

Script di caricamento e risultati

#### Panoramica

In questo esempio, un set di dati è caricato in una tabella denominata "sales". La tabella contiene i seguenti campi:

- ID prodotto
- Tipo di prodotto
- Data di vendita
- Prezzo di vendita

L'utente finale desidera un grafico che visualizzi, per tipo di prodotto, le vendite dei prodotti venduti nel periodo precedente il 24 dicembre 2022. L'utente vorrebbe poter definire la lunghezza di questo periodo.

#### Script di caricamento

```
SET vPeriod = 1;
```

```
Products:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
product_id,product_type,sales_date,sales_price
```

```
8188,product A,'9/19/2022',37.23
```

```
8189,product D,'10/27/2022',17.17
```

```
8190,product C,'10/30/2022',88.27
```

```
8191,product B,'10/31/2022',57.42
```

```
8192,product D,'11/16/2022',53.80
```

```
8193,product D,'11/28/2022',82.06
```

```
8194,product A,'12/2/2022',40.39
```

```
8195,product B,'12/5/2022',87.21
```

```
8196,product C,'12/15/2022',95.93
8197,product B,'12/16/2022',45.89
8198,product C,'12/19/2022',36.23
8199,product D,'12/22/2022',25.66
8200,product D,'12/23/2022',82.77
8201,product A,'12/24/2022',69.98
8202,product A,'12/24/2022',76.11
8203,product B,'12/26/2022',25.12
8204,product B,'12/27/2022',46.23
8205,product B,'12/27/2022',84.21
8206,product C,'12/28/2022',96.24
8207,product D,'12/29/2022',67.67
];
```

## Risultati

Caricare i dati e aprire un foglio.

All'inizio dello script di caricamento, viene creata una variabile, `vPeriod`, legata al controllo dell'input variabile.

Procedere come segue:

1. Nel pannello delle risorse, fare clic su **Oggetti personalizzati**.
2. Selezionare **Qlik Dashboard bundle** e creare un oggetto **Input variabile**.
3. Inserire un titolo per il grafico.
4. In **Variabile**, selezionare **vPeriod** come nome e impostare l'oggetto in modo che venga visualizzato come **Elenco a discesa**.
5. Sotto **Valori**, fare clic sui valori **Dinamici**. Inserire quanto segue:  
`= '1~month|2~bi-month|3~quarter|4~tertiar|6~half-year'`.
6. Aggiungere una nuova tabella al foglio.
7. Nella sezione **Dati** nel pannello delle proprietà, aggiungere `product_type` come dimensione.
8. Aggiungere l'espressione seguente come misura:  
`=sum(if(inmonthstodate($(vPeriod),sales_date,makedate(2022,12,24),0),sales_price,0))`
9. Impostare la misura **Formattazione numero** su **Denaro**.

Tabella dei risultati

| product_type | =sum(if(inmonthstodate(\$(vPeriod),sales_date,makedate(2022,12,24),0),sales_price,0)) |
|--------------|---------------------------------------------------------------------------------------|
| prodotto A   | \$186.48                                                                              |
| prodotto B   | \$190.52                                                                              |
| prodotto C   | \$220.43                                                                              |
| prodotto D   | \$261.46                                                                              |

La funzione `inmonthstodate()` utilizza l'input dell'utente come argomento per definire la dimensione del segmento iniziale dell'anno.

La funzione inserisce la data di vendita di ciascun prodotto come secondo argomento della funzione `inmonthstodate()`. Utilizzando il 24 dicembre come terzo argomento della funzione `inmonthstodate()`, i prodotti con date di vendita che si verificano nel periodo definito fino al 24 dicembre incluso restituiscono un valore booleano di TRUE. La funzione somma aggiunge le vendite di questi prodotti.

### inmonthtodate

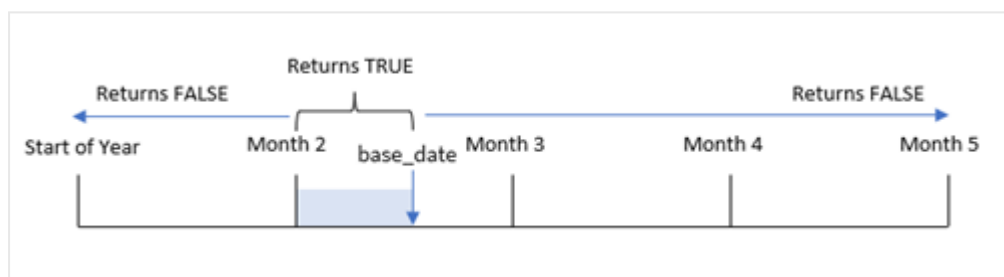
Restituisce True se **date** ricade nella parte di mese contenente **basedate** fino a includere l'ultimo millisecondo di **basedate**.

Sintassi:

```
InMonthToDate (timestamp, base_date, period_no)
```

Tipo di dati restituiti: Boolean

Schema della funzione `inmonthtodate`.



La funzione `inmonthtodate()` identifica un mese selezionato come un segmento. Il limite di partenza è l'inizio del mese. Il limite finale può essere impostato come una data successiva del mese. Determina quindi se un insieme di date rientra o meno in questo segmento, restituendo un valore booleano TRUE o FALSE.

#### Argomenti

| Argomento        | Descrizione                                                                                                                                                                                                                                                                            |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>timestamp</b> | La data da confrontare con <b>base_date</b> .                                                                                                                                                                                                                                          |
| <b>base_date</b> | La data utilizzata per valutare il mese.                                                                                                                                                                                                                                               |
| <b>period_no</b> | Il mese può essere differito mediante <b>period_no</b> . <b>period_no</b> è un numero intero, in cui il valore 0 indica il mese che contiene <b>base_date</b> . I valori negativi di <b>period_no</b> indicano i mesi precedenti, mentre i valori positivi indicano i mesi successivi. |

#### Casi di utilizzo

La funzione `inmonthtodate()` restituisce un risultato booleano. In genere, questo tipo di funzione viene utilizzato come condizione in un file `if` expression. La funzione `inmonthtodate()` restituisce un'aggregazione o un calcolo che dipende dal fatto che una data valutata si sia verificata in un determinato

mese fino a includere la data in questione.

Ad esempio, la funzione `inmonthtoday()` può essere utilizzata per identificare tutte le apparecchiature prodotte in un mese fino a una data specifica.

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

Esempi di funzioni

| Esempio                                                    | Risultato         |
|------------------------------------------------------------|-------------------|
| <code>inmonthtoday ('01/25/2013', '25/01/2013', 0)</code>  | Restituisce True  |
| <code>inmonthtoday ('01/25/2013', '24/01/2013', 0)</code>  | Restituisce False |
| <code>inmonthtoday ('01/25/2013', '28/02/2013', -1)</code> | Restituisce True  |

### Esempio 1 - Nessun argomento aggiuntivo

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2022 viene caricato in una tabella denominata 'Transactions'.
- Un campo data viene fornito nel formato `DateFormat` della variabile di sistema (MM/DD/YYYY).
- Un'istruzione di caricamento precedente contenente:
  - La funzione `inmonthtoday()` impostata come campo, `'in_month_to_date'`. Ciò determina quali transazioni sono state effettuate tra il 1° luglio e il 26 luglio 2022.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

Transactions:

```
Load
*,
inmonthtoday(date,'07/26/2022', 0) as in_month_to_date
;
Load
*
Inline
[
id,date,amount
8188,'1/19/2022',37.23
8189,'1/7/2022',17.17
8190,'2/28/2022',88.27
8191,'2/5/2022',57.42
8192,'3/16/2022',53.80
8193,'4/1/2022',82.06
8194,'5/7/2022',40.39
8195,'5/16/2022',87.21
8196,'6/15/2022',95.93
8197,'6/26/2022',45.89
8198,'7/9/2022',36.23
8199,'7/22/2022',25.66
8200,'7/23/2022',82.77
8201,'7/27/2022',69.98
8202,'8/2/2022',76.11
8203,'8/8/2022',25.12
8204,'8/19/2022',46.23
8205,'9/26/2022',84.21
8206,'10/14/2022',96.24
8207,'10/29/2022',67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- in\_month\_to\_date

Tabella dei risultati

| date      | in_month_to_date |
|-----------|------------------|
| 1/7/2022  | 0                |
| 1/19/2022 | 0                |
| 2/5/2022  | 0                |
| 2/28/2022 | 0                |
| 3/16/2022 | 0                |
| 4/1/2022  | 0                |

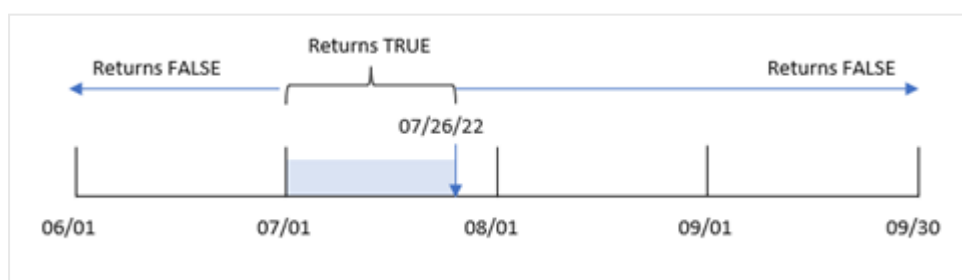
| date       | in_month_to_date |
|------------|------------------|
| 5/7/2022   | 0                |
| 5/16/2022  | 0                |
| 6/15/2022  | 0                |
| 6/26/2022  | 0                |
| 7/9/2022   | -1               |
| 7/22/2022  | -1               |
| 7/23/2022  | -1               |
| 7/27/2022  | 0                |
| 8/2/2022   | 0                |
| 8/8/2022   | 0                |
| 8/19/2022  | 0                |
| 9/26/2022  | 0                |
| 10/14/2022 | 0                |
| 10/29/2022 | 0                |

Il campo 'in\_month\_to\_date' viene creato nell'istruzione LOAD precedente utilizzando la funzione inmonthtoday().

Il primo argomento identifica il campo da valutare. Il secondo argomento è una data codificata, il 26 luglio, che è il base\_date. Questo argomento base\_date identifica quale mese è segmentato e il limite finale di tale segmento.

Un period\_no di 0 è l'argomento finale, a significare che la funzione non confronta i mesi precedenti o successivi al mese segmentato.

*Schema della funzione inmonthtoday senza argomenti aggiuntivi.*



Di conseguenza, qualsiasi transazione effettuata tra il 1° e il 26 luglio restituisce un risultato booleano di TRUE. Qualsiasi transazione che si verifichi nel mese di luglio dopo il 26 luglio restituisce un risultato booleano di FALSE, così come qualsiasi transazione in qualsiasi altro mese dell'anno.

### Esempio 2 - period\_no

Script di caricamento e risultati

#### Panoramica

Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.

In questo esempio, il compito è quello di creare un campo, 'six\_months\_prior', che determini quali transazioni hanno avuto luogo sei mesi prima del 1° luglio e del 26 luglio.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
,
```

```
inmonthtoday(date,'07/26/2022', -6) as six_months_prior
```

```
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'1/19/2022',37.23
```

```
8189,'1/7/2022',17.17
```

```
8190,'2/28/2022',88.27
```

```
8191,'2/5/2022',57.42
```

```
8192,'3/16/2022',53.80
```

```
8193,'4/1/2022',82.06
```

```
8194,'5/7/2022',40.39
```

```
8195,'5/16/2022',87.21
```

```
8196,'6/15/2022',95.93
```

```
8197,'6/26/2022',45.89
```

```
8198,'7/9/2022',36.23
```

```
8199,'7/22/2022',25.66
```

```
8200,'7/23/2022',82.77
```

```
8201,'7/27/2022',69.98
```

```
8202,'8/2/2022',76.11
```

```
8203,'8/8/2022',25.12
```

```
8204,'8/19/2022',46.23
```

```
8205,'9/26/2022',84.21
```

```
8206,'10/14/2022',96.24
```

```
8207,'10/29/2022',67.67
```

```
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

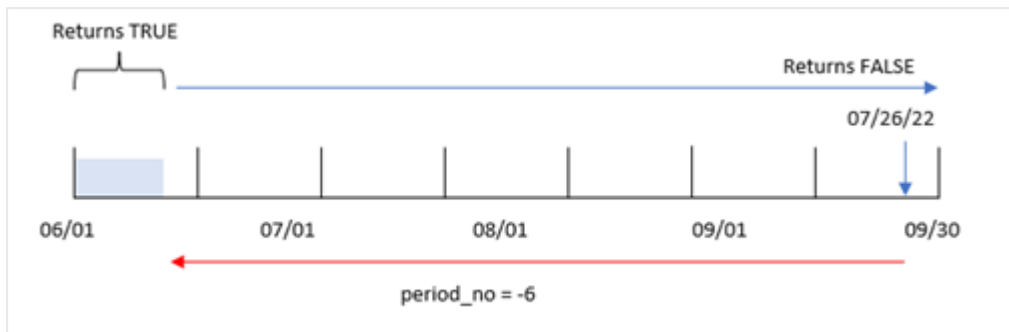
- `date`
- `six_months_prior`

Tabella dei risultati

| <code>date</code> | <code>six_months_prior</code> |
|-------------------|-------------------------------|
| 1/7/2022          | -1                            |
| 1/19/2022         | -1                            |
| 2/5/2022          | 0                             |
| 2/28/2022         | 0                             |
| 3/16/2022         | 0                             |
| 4/1/2022          | 0                             |
| 5/7/2022          | 0                             |
| 5/16/2022         | 0                             |
| 6/15/2022         | 0                             |
| 6/26/2022         | 0                             |
| 7/9/2022          | 0                             |
| 7/22/2022         | 0                             |
| 7/23/2022         | 0                             |
| 7/27/2022         | 0                             |
| 8/2/2022          | 0                             |
| 8/8/2022          | 0                             |
| 8/19/2022         | 0                             |
| 9/26/2022         | 0                             |
| 10/14/2022        | 0                             |
| 10/29/2022        | 0                             |

Utilizzando -6 come argomento `period_no` nella funzione `inmonthtoday()`, i confini del segmento del mese di confronto si spostano di sei mesi. Inizialmente il segmento del mese corrisponde a un periodo compreso tra il 1° e il 26 luglio. `period_no` viene quindi compensato con un valore negativo di sei mesi e i confini della data vengono spostati e cadono tra il 1° e il 26 gennaio.

Schema della funzione `inmonthtoday` con un valore `period_no` impostato a -6.



Di conseguenza, qualsiasi transazione effettuata tra il 1° e il 26 gennaio restituirà un risultato booleano di TRUE.

### Esempio 3 - Esempio di grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.

In questo esempio, il set di dati è invariato e viene caricato nell'app. L'attività richiede di creare un calcolo che determina se le transazioni sono avvenute tra il 1 e il 26 luglio come misura in un oggetto grafico dell'applicazione.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

Transactions:

Load

\*

Inline

[

id,date,amount

8188,'1/19/2022',37.23

8189,'1/7/2022',17.17

8190,'2/28/2022',88.27

8191,'2/5/2022',57.42

8192,'3/16/2022',53.80

8193,'4/1/2022',82.06

8194,'5/7/2022',40.39

8195,'5/16/2022',87.21

8196,'6/15/2022',95.93

8197,'6/26/2022',45.89

8198,'7/9/2022',36.23

8199,'7/22/2022',25.66

8200,'7/23/2022',82.77

8201,'7/27/2022',69.98

8202,'8/2/2022',76.11

```
8203, '8/8/2022', 25.12  
8204, '8/19/2022', 46.23  
8205, '9/26/2022', 84.21  
8206, '10/14/2022', 96.24  
8207, '10/29/2022', 67.67  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

date

Per calcolare se le transazioni sono avvenute tra il 1° luglio e il 26 luglio, creare la seguente misura:

```
=inmonthtodate(date, '07/26/2022', 0)
```

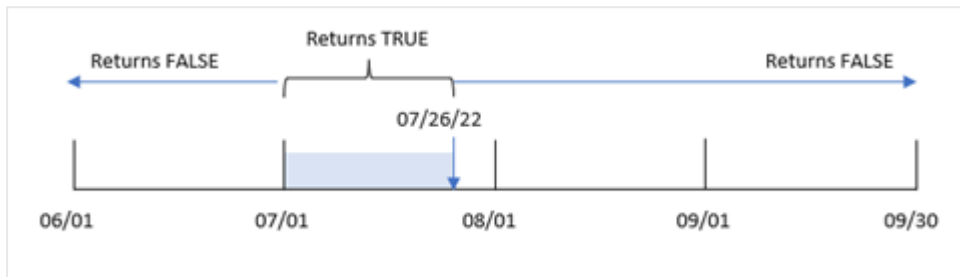
Tabella dei risultati

| date       | =inmonthtodate(date, '07/26/2022', 0) |
|------------|---------------------------------------|
| 1/7/2022   | 0                                     |
| 1/19/2022  | 0                                     |
| 2/5/2022   | 0                                     |
| 2/28/2022  | 0                                     |
| 3/16/2022  | 0                                     |
| 4/1/2022   | 0                                     |
| 5/7/2022   | 0                                     |
| 5/16/2022  | 0                                     |
| 6/15/2022  | 0                                     |
| 6/26/2022  | 0                                     |
| 7/9/2022   | -1                                    |
| 7/22/2022  | -1                                    |
| 7/23/2022  | -1                                    |
| 7/27/2022  | 0                                     |
| 8/2/2022   | 0                                     |
| 8/8/2022   | 0                                     |
| 8/19/2022  | 0                                     |
| 9/26/2022  | 0                                     |
| 10/14/2022 | 0                                     |
| 10/29/2022 | 0                                     |

La misura del campo 'in\_month\_to\_date' viene creata nel grafico mediante l'utilizzo della funzione `inmonthtodate()`.

Il primo argomento identifica il campo da valutare. Il secondo argomento è una data codificata, il 26 luglio, che è il `base_date`. Questo argomento `base_date` identifica quale mese è segmentato e il limite finale di tale segmento. Un `period_no` di 0 rappresenta l'argomento finale. Ciò significa che la funzione non confronta i mesi precedenti o successivi al mese segmentato.

*Schema della funzione `inmonthtodate` senza argomenti aggiuntivi.*



Di conseguenza, qualsiasi transazione effettuata tra il 1° e il 26 luglio restituisce un risultato booleano di TRUE. Qualsiasi transazione che si verifichi nel mese di luglio dopo il 26 luglio restituisce un risultato booleano di FALSE, così come qualsiasi transazione in qualsiasi altro mese dell'anno.

### Esempio 4 - Scenario

Script di caricamento e risultati

#### Panoramica

In questo esempio, un set di dati è caricato in una tabella denominata "Products". La tabella contiene i seguenti campi:

- ID prodotto
- Data di produzione
- Prezzo di costo

A causa di un errore dell'attrezzatura, i prodotti fabbricati nel mese di luglio 2022 erano difettosi. Il problema è stato risolto il 27 luglio 2022.

L'utente finale desidera un grafico che visualizzi, per mese, lo stato dei prodotti fabbricati "difettosi" (valore booleano TRUE) o "non difettosi" (valore booleano FALSE), oltre al costo dei prodotti fabbricati quel mese.

#### Script di caricamento

```
Products:
Load
*
Inline
[
product_id,manufacture_date,cost_price
```

```
8188, '1/19/2022', 37.23
8189, '1/7/2022', 17.17
8190, '2/28/2022', 88.27
8191, '2/5/2022', 57.42
8192, '3/16/2022', 53.80
8193, '4/1/2022', 82.06
8194, '5/7/2022', 40.39
8195, '5/16/2022', 87.21
8196, '6/15/2022', 95.93
8197, '6/26/2022', 45.89
8198, '7/9/2022', 36.23
8199, '7/22/2022', 25.66
8200, '7/23/2022', 82.77
8201, '7/27/2022', 69.98
8202, '8/2/2022', 76.11
8203, '8/8/2022', 25.12
8204, '8/19/2022', 46.23
8205, '9/26/2022', 84.21
8206, '10/14/2022', 96.24
8207, '10/29/2022', 67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- =monthname(manufacture\_date)
- =if(Inmonthtodate(manufacture\_date,makedate(2022,07,26),0),'Defective','Faultless')

Per calcolare la somma dei costi dei prodotti, creare questa misura:

```
=sum(cost_price)
```

Impostare la misura **Formattazione numero** su **Denaro**.

Tabella dei risultati

| monthname<br>(manufacture_date) | if(Inmonthtodate(manufacture_date,makedate<br>(2022,07,26),0),'Defective','Faultless') | =sum(cost_<br>price) |
|---------------------------------|----------------------------------------------------------------------------------------|----------------------|
| Gen 2022                        | Non difettoso                                                                          | \$54.40              |
| Feb 2022                        | Non difettoso                                                                          | \$145.69             |
| Mar 2022                        | Non difettoso                                                                          | \$53.80              |
| Apr 2022                        | Non difettoso                                                                          | \$82.06              |
| Mag 2022                        | Non difettoso                                                                          | \$127.60             |
| Giu 2022                        | Non difettoso                                                                          | \$141.82             |
| Lug 2022                        | Difettoso                                                                              | \$144.66             |
| Lug 2022                        | Non difettoso                                                                          | \$69.98              |

| monthname<br>(manufacture_date) | if(Inmonthtodate(manufacture_date,makedate<br>(2022,07,26),0),'Defective','Faultless') | =sum(cost_<br>price) |
|---------------------------------|----------------------------------------------------------------------------------------|----------------------|
| Ago 2022                        | Non difettoso                                                                          | \$147.46             |
| Set 2022                        | Non difettoso                                                                          | \$84.21              |
| Ott 2022                        | Non difettoso                                                                          | USD 163.91           |

La funzione `inmonthtodate()` restituisce un valore booleano quando valuta le date di produzione di ciascun prodotto.

Per le date che restituiscono un valore booleano di TRUE, il prodotto è contrassegnato come 'difettoso'. Tutti i prodotti che restituiscono un valore di FALSE, e che quindi non sono stati realizzati nel mese fino al 26 luglio compreso, vengono contrassegnati come 'non difettosi'.

## inquarter

Questa funzione restituisce True se **timestamp** ricade all'interno del trimestre contenente **base\_date**.

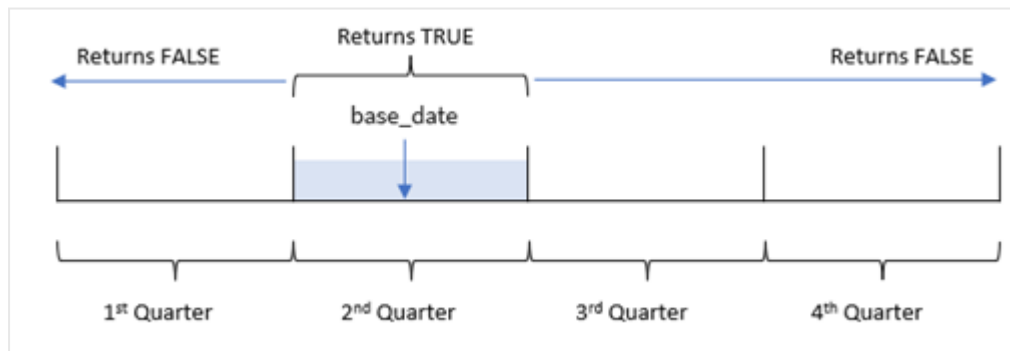
**Sintassi:**

```
InQuarter (timestamp, base_date, period_no[, first_month_of_year])
```

**Tipo di dati restituiti:** Boolean

In Qlik Sense, il valore booleano vero è rappresentato da -1 e il valore falso è rappresentato da 0.

*Schema dell'intervallo della funzione inquarter()*



In altre parole, la funzione `inquarter()` divide l'anno in quattro trimestri uguali tra il 1° gennaio e il 31 dicembre. È possibile utilizzare l'argomento `first_month_of_year` per modificare il mese che viene considerato il primo nell'app e i trimestri cambieranno in base a tale argomento. `base_date`, la funzione identifica quale trimestre deve essere utilizzato come comparatore per la funzione. Infine, la funzione restituisce un risultato booleano quando si confrontano i valori della data con il segmento del trimestre.

### Casi di utilizzo

La funzione `inquarter()` restituisce un risultato booleano. In genere, questo tipo di funzione viene utilizzato come condizione in un file `if expression`. Ciò restituisce un'aggregazione o un calcolo che dipende dalla presenza di una data nel trimestre selezionato.

Ad esempio, la funzione `inquarter()` può essere utilizzata per identificare tutte le apparecchiature prodotte in un segmento trimestrale in base alle date di produzione delle apparecchiature.

#### Argomenti

| Argomento                  | Descrizione                                                                                                                                                                                                                                                                                                |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>timestamp</b>           | La data da confrontare con <b>base_date</b> .                                                                                                                                                                                                                                                              |
| <b>base_date</b>           | La data utilizzata per valutare il trimestre.                                                                                                                                                                                                                                                              |
| <b>period_no</b>           | Il trimestre può essere differito mediante <b>period_no</b> . <b>period_no</b> è un numero intero, in cui il valore 0 indica il trimestre che contiene <b>base_date</b> . I valori negativi di <b>period_no</b> indicano i trimestri precedenti, mentre i valori positivi indicano i trimestri successivi. |
| <b>first_month_of_year</b> | Se si intende utilizzare anni (fiscali) che non iniziano a gennaio, indicare un valore compreso tra 2 e 12 in <b>first_month_of_year</b> .                                                                                                                                                                 |

È possibile utilizzare i seguenti valori per impostare il primo mese dell'anno nell'argomento `first_month_of_year`:

valori `first_month_of_year`

| Mese      | Valore |
|-----------|--------|
| Febbraio  | 2      |
| Marzo     | 3      |
| Aprile    | 4      |
| Maggio    | 5      |
| Giugno    | 6      |
| Luglio    | 7      |
| Agosto    | 8      |
| Settembre | 9      |
| Ottobre   | 10     |
| Novembre  | 11     |
| Dicembre  | 12     |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

Esempi di funzioni

| Esempio                                                   | Risultato         |
|-----------------------------------------------------------|-------------------|
| <code>inquarter ('01/25/2013', '01/01/2013', 0)</code>    | Restituisce TRUE  |
| <code>inquarter ('01/25/2013', '04/01/2013', 0)</code>    | Restituisce FALSE |
| <code>inquarter ('01/25/2013', '01/01/2013', -1)</code>   | Restituisce FALSE |
| <code>inquarter ('12/25/2012', '01/01/2013', -1)</code>   | Restituisce TRUE  |
| <code>inquarter ('01/25/2013', '03/01/2013', 0, 3)</code> | Restituisce FALSE |
| <code>inquarter ('03/25/2013', '03/01/2013', 0, 3)</code> | Restituisce TRUE  |

### Esempio 1 - Nessun argomento aggiuntivo

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2022, caricato in una tabella denominata 'Transactions'.
- Un caricamento precedente che contiene la funzione `inquarter()` impostata come campo 'in\_quarter' e determina quali transazioni hanno avuto luogo nello stesso trimestre del 15 maggio 2022.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

Transactions:

```
Load
    *,
    inquarter (date,'05/15/2022', 0) as in_quarter
;

Load
*
Inline
[
id,date,amount
8188,'1/19/2022',37.23
8189,'1/7/2022',17.17
8190,'2/28/2022',88.27
8191,'2/5/2022',57.42
8192,'3/16/2022',53.80
8193,'4/1/2022',82.06
8194,'5/7/2022',40.39
8195,'5/16/2022',87.21
8196,'6/15/2022',95.93
8197,'6/26/2022',45.89
8198,'7/9/2022',36.23
8199,'7/22/2022',25.66
8200,'7/23/2022',82.77
8201,'7/27/2022',69.98
8202,'8/2/2022',76.11
8203,'8/8/2022',25.12
8204,'8/19/2022',46.23
8205,'9/26/2022',84.21
8206,'10/14/2022',96.24
8207,'10/29/2022',67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- in\_quarter

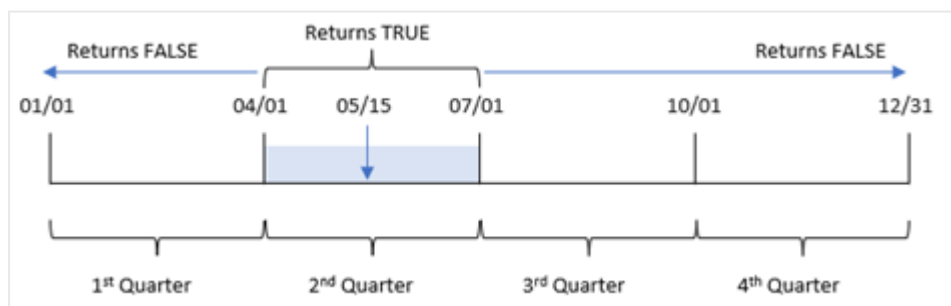
Tabella dei risultati

| date      | in_quarter |
|-----------|------------|
| 1/7/2022  | 0          |
| 1/19/2022 | 0          |
| 2/5/2022  | 0          |
| 2/28/2022 | 0          |
| 3/16/2022 | 0          |
| 4/1/2022  | -1         |

| date       | in_quarter |
|------------|------------|
| 5/7/2022   | -1         |
| 5/16/2022  | -1         |
| 6/15/2022  | -1         |
| 6/26/2022  | -1         |
| 7/9/2022   | 0          |
| 7/22/2022  | 0          |
| 7/23/2022  | 0          |
| 7/27/2022  | 0          |
| 8/2/2022   | 0          |
| 8/8/2022   | 0          |
| 8/19/2022  | 0          |
| 9/26/2022  | 0          |
| 10/14/2022 | 0          |
| 10/29/2022 | 0          |

Il campo 'in\_quarter' viene creato nell'istruzione LOAD precedente utilizzando la funzione `inquarter()`. Il primo argomento identifica il campo da valutare. Il secondo argomento è una data codificata per il 15 maggio che identifica il trimestre da definire come comparatore. Un `period_no` di 0 è l'argomento finale e garantisce che la funzione `inquarter()` non confronti i trimestri precedenti o successivi al trimestre segmentato.

*Schema della funzione `inquarter()` con data base 15 maggio*



Qualsiasi transazione che si verifica tra il 1° aprile e la fine del 30 giugno restituisce un risultato booleano pari a TRUE.

### Esempio 2 - period\_no

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2022, caricato in una tabella denominata 'Transactions'.
- Un caricamento precedente che contiene la funzione `inquarter()` impostata come campo 'previous\_quarter' e determina quali transazioni hanno avuto luogo nello trimestre precedente quello del 15 maggio 2022.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*,
```

```
inquarter (date,'05/15/2022', -1) as previous_qtr
```

```
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'1/19/2022',37.23
```

```
8189,'1/7/2022',17.17
```

```
8190,'2/28/2022',88.27
```

```
8191,'2/5/2022',57.42
```

```
8192,'3/16/2022',53.80
```

```
8193,'4/1/2022',82.06
```

```
8194,'5/7/2022',40.39
```

```
8195,'5/16/2022',87.21
```

```
8196,'6/15/2022',95.93
```

```
8197,'6/26/2022',45.89
```

```
8198,'7/9/2022',36.23
```

```
8199,'7/22/2022',25.66
```

```
8200,'7/23/2022',82.77
```

```
8201,'7/27/2022',69.98
```

```
8202,'8/2/2022',76.11
```

```
8203,'8/8/2022',25.12
```

```
8204,'8/19/2022',46.23
```

```
8205,'9/26/2022',84.21
```

```
8206,'10/14/2022',96.24
```

```
8207,'10/29/2022',67.67
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

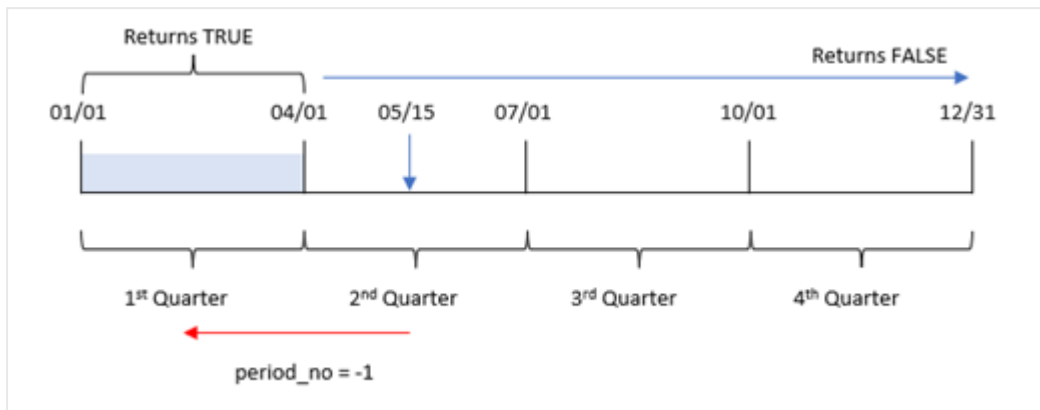
- date
- previous\_qtr

Tabella dei risultati

| date       | previous_qtr |
|------------|--------------|
| 1/7/2022   | -1           |
| 1/19/2022  | -1           |
| 2/5/2022   | -1           |
| 2/28/2022  | -1           |
| 3/16/2022  | -1           |
| 4/1/2022   | 0            |
| 5/7/2022   | 0            |
| 5/16/2022  | 0            |
| 6/15/2022  | 0            |
| 6/26/2022  | 0            |
| 7/9/2022   | 0            |
| 7/22/2022  | 0            |
| 7/23/2022  | 0            |
| 7/27/2022  | 0            |
| 8/2/2022   | 0            |
| 8/8/2022   | 0            |
| 8/19/2022  | 0            |
| 9/26/2022  | 0            |
| 10/14/2022 | 0            |
| 10/29/2022 | 0            |

L'utilizzo di -1 come argomento `period_no` nella funzione `inquarter()` sposta i confini del trimestre del comparatore indietro di un intero trimestre. Il 15 maggio cade nel secondo trimestre dell'anno e quindi il segmento equivale inizialmente al trimestre dal 1° aprile al 30 giugno. `period_no` compensa questo segmento con un valore negativo di tre mesi e fa sì che i confini della data diventino dal 1° gennaio al 30 marzo.

Schema della funzione `inquarter()` con data base 15 maggio



Pertanto, qualsiasi transazione che si verifichi tra il 1° gennaio e il 30 marzo restituirà un risultato booleano di TRUE.

### Esempio 3 - `first_month_of_year`

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2022, caricato in una tabella denominata 'Transactions'.
- Un caricamento precedente che contiene la funzione `inquarter()` impostata come campo 'in\_quarter' e determina quali transazioni hanno avuto luogo nello stesso trimestre del 15 maggio 2022.

Tuttavia, in questo esempio, il criterio organizzativo prevede che marzo sia il primo mese dell'anno finanziario.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
    *,
    inquarter (date,'05/15/2022', 0, 3) as in_quarter
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'1/19/2022',37.23
```

```
8189, '1/7/2022', 17.17
8190, '2/28/2022', 88.27
8191, '2/5/2022', 57.42
8192, '3/16/2022', 53.80
8193, '4/1/2022', 82.06
8194, '5/7/2022', 40.39
8195, '5/16/2022', 87.21
8196, '6/15/2022', 95.93
8197, '6/26/2022', 45.89
8198, '7/9/2022', 36.23
8199, '7/22/2022', 25.66
8200, '7/23/2022', 82.77
8201, '7/27/2022', 69.98
8202, '8/2/2022', 76.11
8203, '8/8/2022', 25.12
8204, '8/19/2022', 46.23
8205, '9/26/2022', 84.21
8206, '10/14/2022', 96.24
8207, '10/29/2022', 67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- previous\_qtr

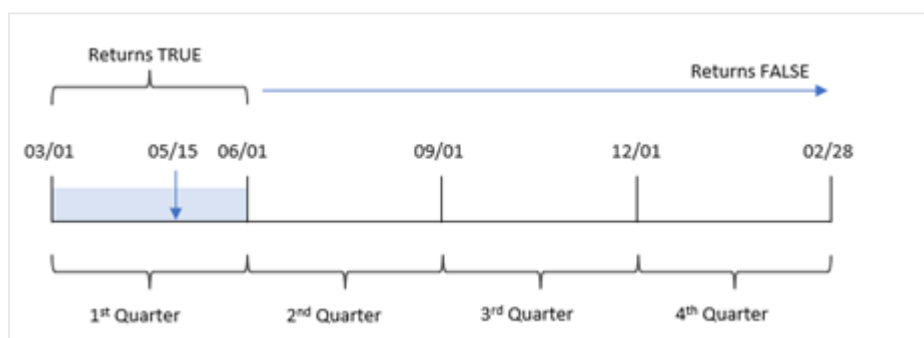
Tabella dei risultati

| date      | previous_qtr |
|-----------|--------------|
| 1/7/2022  | 0            |
| 1/19/2022 | 0            |
| 2/5/2022  | 0            |
| 2/28/2022 | 0            |
| 3/16/2022 | -1           |
| 4/1/2022  | -1           |
| 5/7/2022  | -1           |
| 5/16/2022 | -1           |
| 6/15/2022 | 0            |
| 6/26/2022 | 0            |
| 7/9/2022  | 0            |
| 7/22/2022 | 0            |
| 7/23/2022 | 0            |

| date       | previous_qtr |
|------------|--------------|
| 7/27/2022  | 0            |
| 8/2/2022   | 0            |
| 8/8/2022   | 0            |
| 8/19/2022  | 0            |
| 9/26/2022  | 0            |
| 10/14/2022 | 0            |
| 10/29/2022 | 0            |

Utilizzando 3 come argomento `first_month_of_year` nella funzione `inquarter()`, si imposta il 1° marzo come inizio dell'anno e poi si divide l'anno in trimestri. Pertanto, i segmenti trimestrali sono marzo-maggio, giugno-agosto, settembre-novembre, dicembre-febbraio. `base_date` del 15 maggio stabilisce il trimestre marzo-maggio come trimestre di confronto per la funzione.

Lo schema della funzione `inquarter()` con marzo impostato come primo mese dell'anno



Pertanto, qualsiasi transazione che si verifichi tra il 1° marzo e il 31 marzo restituirà un risultato booleano di TRUE.

## Esempio 4 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2022, caricato in una tabella denominata 'Transactions'.
- Un caricamento precedente che contiene la funzione `inquarter()` impostata come campo 'in\_quarter' e determina quali transazioni hanno avuto luogo nello stesso trimestre del 15 maggio 2022.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'1/19/2022',37.23
```

```
8189,'1/7/2022',17.17
```

```
8190,'2/28/2022',88.27
```

```
8191,'2/5/2022',57.42
```

```
8192,'3/16/2022',53.80
```

```
8193,'4/1/2022',82.06
```

```
8194,'5/7/2022',40.39
```

```
8195,'5/16/2022',87.21
```

```
8196,'6/15/2022',95.93
```

```
8197,'6/26/2022',45.89
```

```
8198,'7/9/2022',36.23
```

```
8199,'7/22/2022',25.66
```

```
8200,'7/23/2022',82.77
```

```
8201,'7/27/2022',69.98
```

```
8202,'8/2/2022',76.11
```

```
8203,'8/8/2022',25.12
```

```
8204,'8/19/2022',46.23
```

```
8205,'9/26/2022',84.21
```

```
8206,'10/14/2022',96.24
```

```
8207,'10/29/2022',67.67
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- date

Creare la seguente misura per calcolare se le transazioni sono avvenute nello stesso trimestre del 15 maggio:

```
=inquarter(date,'05/15/2022', 0)
```

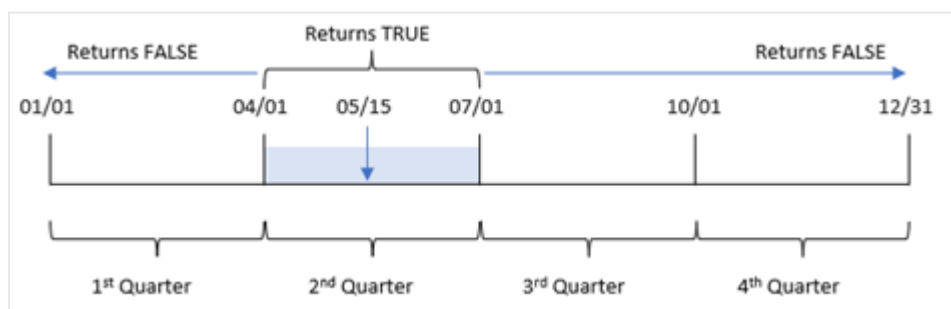
Tabella dei risultati

| date      | in_quarter |
|-----------|------------|
| 1/7/2022  | 0          |
| 1/19/2022 | 0          |
| 2/5/2022  | 0          |
| 2/28/2022 | 0          |

| date       | in_quarter |
|------------|------------|
| 3/16/2022  | 0          |
| 4/1/2022   | -1         |
| 5/7/2022   | -1         |
| 5/16/2022  | -1         |
| 6/15/2022  | -1         |
| 6/26/2022  | -1         |
| 7/9/2022   | 0          |
| 7/22/2022  | 0          |
| 7/23/2022  | 0          |
| 7/27/2022  | 0          |
| 8/2/2022   | 0          |
| 8/8/2022   | 0          |
| 8/19/2022  | 0          |
| 9/26/2022  | 0          |
| 10/14/2022 | 0          |
| 10/29/2022 | 0          |

La misura 'in\_quarter' viene creata nell'oggetto grafico utilizzando la funzione `inquarter()`. Il primo argomento identifica il campo da valutare. Il secondo argomento è una data codificata per il 15 maggio che identifica il trimestre da definire come comparatore. Un `period_no` di 0 è l'argomento finale e garantisce che la funzione `inquarter()` non confronti i trimestri precedenti o successivi al trimestre segmentato.

*Schema della funzione `inquarter()` con data base 15 maggio*



Qualsiasi transazione che si verifica tra il 1° aprile e la fine del 30 giugno restituisce un risultato booleano pari a TRUE.

### Esempio 5 - Scenario

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella chiamata 'Products'.
- La tabella contiene i seguenti campi:
  - ID prodotto
  - tipo di prodotto
  - data di produzione
  - prezzo di costo

Si è determinato che, a causa di un errore delle apparecchiature, i prodotti fabbricati nella settimana nel trimestre del 15 maggio 2022 erano difettosi. L'utente finale vorrebbe un grafico che visualizzi, per nome del trimestre, lo stato dei prodotti fabbricati "difettosi" o "non difettosi" e il costo dei prodotti fabbricati in quel trimestre.

#### Script di caricamento

Products:

Load

\*

Inline

[

product\_id,manufacture\_date,cost\_price

8188, '1/19/2022', 37.23

8189, '1/7/2022', 17.17

8190, '2/28/2022', 88.27

8191, '2/5/2022', 57.42

8192, '3/16/2022', 53.80

8193, '4/1/2022', 82.06

8194, '5/7/2022', 40.39

8195, '5/16/2022', 87.21

8196, '6/15/2022', 95.93

8197, '6/26/2022', 45.89

8198, '7/9/2022', 36.23

8199, '7/22/2022', 25.66

8200, '7/23/2022', 82.77

8201, '7/27/2022', 69.98

8202, '8/2/2022', 76.11

8203, '8/8/2022', 25.12

8204, '8/19/2022', 46.23

8205, '9/26/2022', 84.21

8206, '10/14/2022', 96.24

8207, '10/29/2022', 67.67

];

## Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

=quartername(manufacture\_date)

Creare le seguenti misure:

- =if(only(InQuarter(manufacture\_date,makedate(2022,05,15),0)), 'Defective', 'Faultless'), per identificare quali prodotti sono difettosi e quali non lo sono utilizzando la funzione inquarter().
- =sum(cost\_price), per mostrare la somma dei costi di ciascun prodotto.

Procedere come indicato di seguito:

1. Impostare la misura **Formattazione numero** su **Denaro**.
2. In **Aspetto**, disattivare **Totals**.

Tabella dei risultati

| quartername<br>(manufacture_date) | =if(only(InQuarter(manufacture_date,makedate<br>(2022,05,15),0)), 'Defective', 'Faultless') | =sum(cost_<br>price) |
|-----------------------------------|---------------------------------------------------------------------------------------------|----------------------|
| gen-mar 2022                      | Non difettoso                                                                               | 253.89               |
| apr-giu 2022                      | Difettoso                                                                                   | 351.48               |
| lug-set 2022                      | Non difettoso                                                                               | 446.31               |
| ott-dic 2022                      | Non difettoso                                                                               | 163.91               |

La funzione inquarter() restituisce un valore booleano quando valuta le date di produzione di ciascun prodotto. Per qualsiasi prodotto fabbricato nel trimestre che contiene il 15 maggio, la funzione inquarter() restituisce un valore booleano di TRUE e contrassegna i prodotti come 'difettosi'. Qualsiasi prodotto che restituisce un valore di FALSE, e che quindi non è stato fabbricato nel trimestre, viene contrassegnato come "non difettoso".

## inquartertoday

Questa funzione restituisce True se **timestamp** ricade all'interno della parte del trimestre contenente **base\_date** fino a includere l'ultimo millisecondo di **base\_date**.

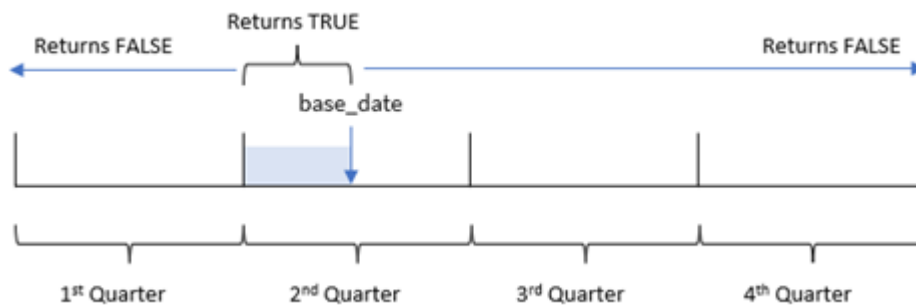
Sintassi:

**InQuarterToDate** (timestamp, base\_date, period\_no [, first\_month\_of\_year])

Tipo di dati restituiti: Boolean



*In Qlik Sense, il valore booleano vero è rappresentato da -1 e il valore falso è rappresentato da 0.*

Diagramma della funzione *inquartertodate*

La funzione `inquartertodate()` divide l'anno in quattro trimestri uguali tra il 1 gennaio e il 31 dicembre (o l'inizio dell'anno definito dall'utente e la data di fine corrispondente). Usando `base_date`, la funzione segmenterà quindi un particolare trimestre e `base_date` identificherà sia il trimestre che la data massima consentita per quel segmento di trimestre. Infine, la funzione restituisce un risultato booleano quando si confrontano i valori di data prescritti con quel segmento.

## Argomenti

| Argomento                  | Descrizione                                                                                                                                                                                                                                                                                                |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>timestamp</b>           | La data da confrontare con <b>base_date</b> .                                                                                                                                                                                                                                                              |
| <b>base_date</b>           | La data utilizzata per valutare il trimestre.                                                                                                                                                                                                                                                              |
| <b>period_no</b>           | Il trimestre può essere differito mediante <b>period_no</b> . <b>period_no</b> è un numero intero, in cui il valore 0 indica il trimestre che contiene <b>base_date</b> . I valori negativi di <b>period_no</b> indicano i trimestri precedenti, mentre i valori positivi indicano i trimestri successivi. |
| <b>first_month_of_year</b> | Se si intende utilizzare anni (fiscali) che non iniziano a gennaio, indicare un valore compreso tra 2 e 12 in <b>first_month_of_year</b> .                                                                                                                                                                 |

## Casi di utilizzo

La funzione `inquartertodate()` restituisce un risultato booleano. In genere, questo tipo di funzione viene utilizzato come condizione in un'espressione `if`. La funzione `inquartertodate()` restituisce un'aggregazione o un calcolo che dipende dal fatto che una data valutata si sia verificata in un determinato trimestre fino a includere la data in questione.

Ad esempio, la funzione `inquartertodate()` può essere utilizzata per identificare tutte le apparecchiature prodotte in un determinato trimestre fino a una data specifica.

## Esempi di funzioni

| Esempio                                                     | Risultato                                                                                                                                                                                                                                |
|-------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>inquartertodate('01/25/2013', '03/25/2013', 0)</code> | Restituisce <code>TRUE</code> , poiché il valore di <code>timestamp</code> , 25/01/2013, rientra nel periodo di tre mesi dal giorno 01/01/2013 al giorno 25/03/2013, in cui il valore di <code>base_date</code> , 25/03/2013, è incluso. |

| Esempio                                                        | Risultato                                                                                                                                                                                                                                                      |
|----------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>inquartertoday ('04/26/2013', '03/25/2013', 0)</code>    | Restituisce <code>FALSE</code> , poiché la data 26/04/2013 non ricade nello stesso periodo dell'esempio precedente.                                                                                                                                            |
| <code>inquartertoday ('02/25/2013', '06/09/2013', -1)</code>   | Restituisce <code>TRUE</code> , poiché il valore di <code>period_no</code> , -1, sposta il periodo di ricerca indietro per un intervallo di tempo di tre mesi (un quarto dell'anno). Questo include il periodo di ricerca tra le date 01/01/2013 e 09/03/2013. |
| <code>inquartertoday ('03/25/2006', '04/15/2006', 0, 2)</code> | Restituisce <code>TRUE</code> , poiché il valore di <code>first_month_of_year</code> è impostato su 2, che fa rientrare il periodo di ricerca tra il 01/02/2006 e il 15/04/2006 anziché dal 01/04/2006 al 15/04/2006.                                          |

## Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

## Esempio 1 - Nessun argomento aggiuntivo

Script di caricamento e risultati

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2022, caricato in una tabella denominata 'Transactions'.
- Il campo della data fornito nel formato della variabile di sistema `DateFormat` (MM/GG/AAAA).
- La creazione di un campo, `in_quarter_to_date`, che determina quali transazioni sono avvenute nel trimestre fino al 15 maggio 2022.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

Transactions:

```
Load
    *,
    inquartertoday(date,'05/15/2022', 0) as in_quarter_to_date
;

Load
*
Inline
[
id,date,amount
8188,'1/19/2022',37.23
8189,'1/7/2022',17.17
8190,'2/28/2022',88.27
8191,'2/5/2022',57.42
8192,'3/16/2022',53.80
8193,'4/1/2022',82.06
8194,'5/7/2022',40.39
8195,'5/16/2022',87.21
8196,'6/15/2022',95.93
8197,'6/26/2022',45.89
8198,'7/9/2022',36.23
8199,'7/22/2022',25.66
8200,'7/23/2022',82.77
8201,'7/27/2022',69.98
8202,'8/2/2022',76.11
8203,'8/8/2022',25.12
8204,'8/19/2022',46.23
8205,'9/26/2022',84.21
8206,'10/14/2022',96.24
8207,'10/29/2022',67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- in\_quarter\_to\_date

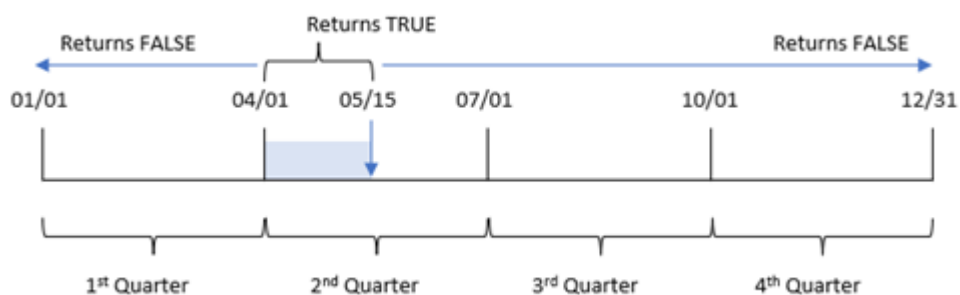
Tabella dei risultati

| date      | nel_trimestre_a_oggi |
|-----------|----------------------|
| 1/7/2022  | 0                    |
| 1/19/2022 | 0                    |
| 2/5/2022  | 0                    |
| 2/28/2022 | 0                    |
| 3/16/2022 | 0                    |
| 4/1/2022  | -1                   |

| date       | nel_trimestre_a_oggi |
|------------|----------------------|
| 5/7/2022   | -1                   |
| 5/16/2022  | 0                    |
| 6/15/2022  | 0                    |
| 6/26/2022  | 0                    |
| 7/9/2022   | 0                    |
| 7/22/2022  | 0                    |
| 7/23/2022  | 0                    |
| 7/27/2022  | 0                    |
| 8/2/2022   | 0                    |
| 8/8/2022   | 0                    |
| 8/19/2022  | 0                    |
| 9/26/2022  | 0                    |
| 10/14/2022 | 0                    |
| 10/29/2022 | 0                    |

Il campo `in_quarter_to_date` viene creato nell'istruzione caricamento precedente utilizzando la funzione `inquartertodate()`. Il primo argomento fornito identifica il campo da valutare. Il secondo argomento è una data codificata per il 15 maggio, che è il `base_date` che identifica il trimestre da segmentare e definisce il limite finale di quel segmento. Un `period_no` di 0 è l'argomento finale, il che significa che la funzione non confronta i trimestri precedenti o successivi al trimestre segmentato.

*Diagramma della funzione `inquartertodate`, senza argomenti aggiuntivi*



Qualsiasi transazione avvenuta tra il 1 aprile e il 15 maggio restituisce il risultato booleano `TRUE`. Le date delle transazioni del 16 maggio e successive restituiranno `FALSE`, così come tutte le transazioni precedenti al 1 aprile.

### Esempio 2 - period\_no

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.
- La creazione di un campo, `previous_qtr_to_date`, che determina quali transazioni sono avvenute nel trimestre precedente al segmento del trimestre che termina il 16 maggio 2022.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        inquartertodate(date,'05/15/2022', -1) as previous_qtr_to_date
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
id,date,amount
8188,'1/19/2022',37.23
8189,'1/7/2022',17.17
8190,'2/28/2022',88.27
8191,'2/5/2022',57.42
8192,'3/16/2022',53.80
8193,'4/1/2022',82.06
8194,'5/7/2022',40.39
8195,'5/16/2022',87.21
8196,'6/15/2022',95.93
8197,'6/26/2022',45.89
8198,'7/9/2022',36.23
8199,'7/22/2022',25.66
8200,'7/23/2022',82.77
8201,'7/27/2022',69.98
8202,'8/2/2022',76.11
8203,'8/8/2022',25.12
8204,'8/19/2022',46.23
8205,'9/26/2022',84.21
8206,'10/14/2022',96.24
8207,'10/29/2022',67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

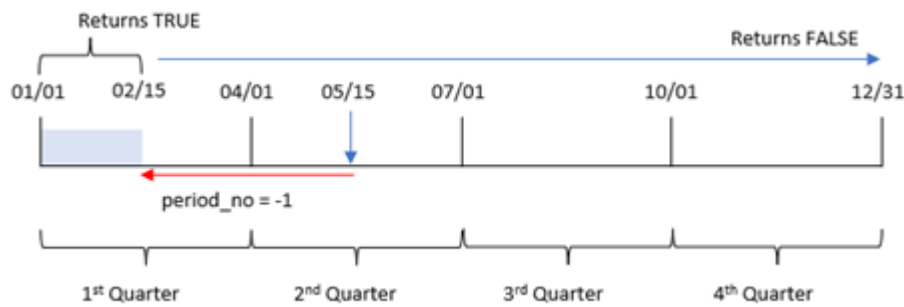
- date
- previous\_qtr\_to\_date

Tabella dei risultati

| date       | trimestre_precedente_a_oggi |
|------------|-----------------------------|
| 1/7/2022   | -1                          |
| 1/19/2022  | -1                          |
| 2/5/2022   | -1                          |
| 2/28/2022  | 0                           |
| 3/16/2022  | 0                           |
| 4/1/2022   | 0                           |
| 5/7/2022   | 0                           |
| 5/16/2022  | 0                           |
| 6/15/2022  | 0                           |
| 6/26/2022  | 0                           |
| 7/9/2022   | 0                           |
| 7/22/2022  | 0                           |
| 7/23/2022  | 0                           |
| 7/27/2022  | 0                           |
| 8/2/2022   | 0                           |
| 8/8/2022   | 0                           |
| 8/19/2022  | 0                           |
| 9/26/2022  | 0                           |
| 10/14/2022 | 0                           |
| 10/29/2022 | 0                           |

Un valore period\_no di -1 indica che la funzione `inquartertodate ()` confronta il segmento del trimestre di input con il trimestre precedente. Il 15 maggio rientra nel secondo trimestre dell'anno, quindi il segmento inizialmente corrisponde al periodo compreso tra il 1 aprile e il 15 maggio. period\_no quindi compensa questo segmento di tre mesi prima, facendo sì che i limiti della data diventino dal 1 gennaio al 15 febbraio.

Diagramma della funzione *inquartertodate*, esempio di *period\_no*



Pertanto, qualsiasi transazione che si verifichi tra il 1 gennaio e il 15 febbraio restituirà il risultato booleano TRUE.

### Esempio 3 - *first\_month\_of\_year*

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.
- La creazione di un campo, *in\_quarter\_to\_date*, che determina quali transazioni sono avvenute nello stesso trimestre fino al 15 maggio 2022.

In questo esempio, marzo è stato impostato come primo mese dell'anno fiscale.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

Transactions:

```
Load
    *,
    inquartertodate(date,'05/15/2022', 0,3) as in_quarter_to_date
;

Load
*

Inline
[
id,date,amount
8188,'1/19/2022',37.23
8189,'1/7/2022',17.17
8190,'2/28/2022',88.27
8191,'2/5/2022',57.42
8192,'3/16/2022',53.80
8193,'4/1/2022',82.06
```

```
8194, '5/7/2022', 40.39
8195, '5/16/2022', 87.21
8196, '6/15/2022', 95.93
8197, '6/26/2022', 45.89
8198, '7/9/2022', 36.23
8199, '7/22/2022', 25.66
8200, '7/23/2022', 82.77
8201, '7/27/2022', 69.98
8202, '8/2/2022', 76.11
8203, '8/8/2022', 25.12
8204, '8/19/2022', 46.23
8205, '9/26/2022', 84.21
8206, '10/14/2022', 96.24
8207, '10/29/2022', 67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- in\_quarter\_to\_date

Tabella dei risultati

| date      | nel_trimestre_a_oggi |
|-----------|----------------------|
| 1/7/2022  | 0                    |
| 1/19/2022 | 0                    |
| 2/5/2022  | 0                    |
| 2/28/2022 | 0                    |
| 3/16/2022 | -1                   |
| 4/1/2022  | -1                   |
| 5/7/2022  | -1                   |
| 5/16/2022 | 0                    |
| 6/15/2022 | 0                    |
| 6/26/2022 | 0                    |
| 7/9/2022  | 0                    |
| 7/22/2022 | 0                    |
| 7/23/2022 | 0                    |
| 7/27/2022 | 0                    |
| 8/2/2022  | 0                    |
| 8/8/2022  | 0                    |

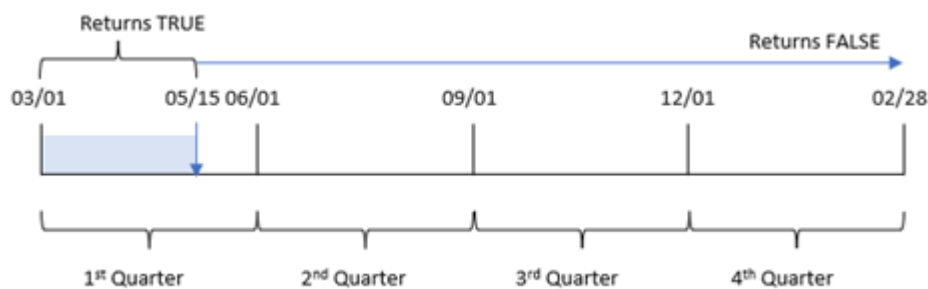
| date       | nel_trimestre_a_oggi |
|------------|----------------------|
| 8/19/2022  | 0                    |
| 9/26/2022  | 0                    |
| 10/14/2022 | 0                    |
| 10/29/2022 | 0                    |

Utilizzando 3 come argomento `first_month_of_year` nella funzione `inquartertodate()`, la funzione inizia l'anno il 1 marzo, quindi divide l'anno in trimestri. Pertanto, i segmenti del trimestre sono:

- Da marzo a maggio
- Da giugno ad agosto
- Da settembre a novembre
- Da dicembre a febbraio

Il `base_date` del 15 maggio segmenta quindi il trimestre da marzo a maggio impostando il limite finale al 15 maggio.

*Diagramma della funzione `inquartertodate`, esempio di `first_month_of_year`*



Pertanto, qualsiasi transazione che si verifichi tra il 1 marzo e il 15 maggio restituirà un risultato booleano TRUE, mentre le transazioni con una data che non rientra in questi limiti restituiranno il valore FALSE.

### Esempio 4 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor di caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento contiene lo stesso set di dati e lo stesso scenario del primo esempio. Tuttavia, in questo esempio, il set di dati invariato viene caricato nell'applicazione. Il calcolo che determina quali transazioni sono avvenute lo stesso trimestre del 15 maggi viene creato come misura in un oggetto grafico.

**Script di caricamento**

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'1/19/2022',37.23
```

```
8189,'1/7/2022',17.17
```

```
8190,'2/28/2022',88.27
```

```
8191,'2/5/2022',57.42
```

```
8192,'3/16/2022',53.80
```

```
8193,'4/1/2022',82.06
```

```
8194,'5/7/2022',40.39
```

```
8195,'5/16/2022',87.21
```

```
8196,'6/15/2022',95.93
```

```
8197,'6/26/2022',45.89
```

```
8198,'7/9/2022',36.23
```

```
8199,'7/22/2022',25.66
```

```
8200,'7/23/2022',82.77
```

```
8201,'7/27/2022',69.98
```

```
8202,'8/2/2022',76.11
```

```
8203,'8/8/2022',25.12
```

```
8204,'8/19/2022',46.23
```

```
8205,'9/26/2022',84.21
```

```
8206,'10/14/2022',96.24
```

```
8207,'10/29/2022',67.67
```

```
];
```

**Risultati**

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:date.

Creare la seguente misura:

```
=inquartertodate(date,'05/15/2022',0)
```

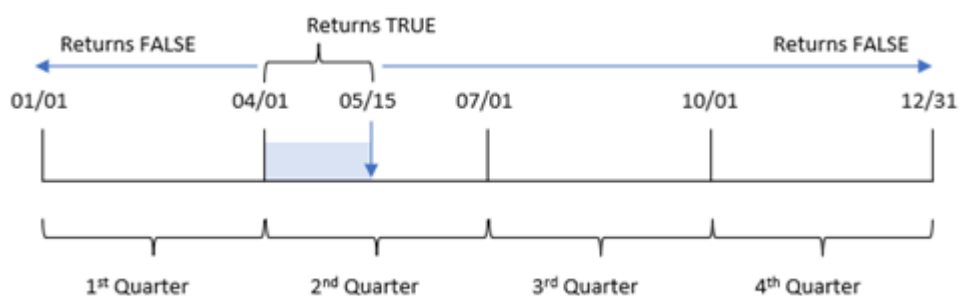
Tabella dei risultati

| date      | =inquartertodate(date,'15/05/2022',0) |
|-----------|---------------------------------------|
| 1/7/2022  | 0                                     |
| 1/19/2022 | 0                                     |
| 2/5/2022  | 0                                     |
| 2/28/2022 | 0                                     |
| 3/16/2022 | 0                                     |

| date       | =inquartertodate(date,'15/05/2022', 0) |
|------------|----------------------------------------|
| 4/1/2022   | -1                                     |
| 5/7/2022   | -1                                     |
| 5/16/2022  | 0                                      |
| 6/15/2022  | 0                                      |
| 6/26/2022  | 0                                      |
| 7/9/2022   | 0                                      |
| 7/22/2022  | 0                                      |
| 7/23/2022  | 0                                      |
| 7/27/2022  | 0                                      |
| 8/2/2022   | 0                                      |
| 8/8/2022   | 0                                      |
| 8/19/2022  | 0                                      |
| 9/26/2022  | 0                                      |
| 10/14/2022 | 0                                      |
| 10/29/2022 | 0                                      |

La misura `in_quarter_to_date` viene creata in un oggetto grafico utilizzando la funzione `inquartertodate()`. Il primo argomento è il campo della data da valutare. Il secondo argomento è una data codificata per il 15 maggio, che è il valore `base_date` che identifica il trimestre da segmentare e definisce il limite finale di quel segmento. Un `period_no` di 0 è l'argomento finale, il che significa che la funzione non confronta i trimestri precedenti o successivi al trimestre segmentato.

*Diagramma della funzione `inquartertodate`, esempio di oggetto grafico*



Qualsiasi transazione avvenuta tra il 1 aprile e il 15 maggio restituisce il risultato booleano `TRUE`. Le transazioni del 16 maggio e successive restituiranno `FALSE`, così come tutte le transazioni precedenti al 1 aprile.

### Esempio 5 - Scenario

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella chiamata `Products`.
- Informazioni relative all'ID prodotto, alla data di produzione e al prezzo di costo.

Il 15 maggio 2022, un errore dell'attrezzatura è stato identificato nel processo di produzione e risolto. I prodotti fabbricati in quel trimestre fino a questa data saranno difettosi. L'utente finale desidera un oggetto grafico che visualizzi, in base al nome del trimestre, se lo stato dei prodotti è "difettoso" o "senza difetti" e il costo dei prodotti fabbricati in quel trimestre fino a quella data.

#### Script di caricamento

```
Products:
Load
*
Inline
[
product_id,manufacture_date,cost_price
8188,'1/19/2022',37.23
8189,'1/7/2022',17.17
8190,'2/28/2022',88.27
8191,'2/5/2022',57.42
8192,'3/16/2022',53.80
8193,'4/1/2022',82.06
8194,'5/7/2022',40.39
8195,'5/16/2022',87.21
8196,'6/15/2022',95.93
8197,'6/26/2022',45.89
8198,'7/9/2022',36.23
8199,'7/22/2022',25.66
8200,'7/23/2022',82.77
8201,'7/27/2022',69.98
8202,'8/2/2022',76.11
8203,'8/8/2022',25.12
8204,'8/19/2022',46.23
8205,'9/26/2022',84.21
8206,'10/14/2022',96.24
8207,'10/29/2022',67.67
];
```

## Risultati

Procedere come indicato di seguito:

1. Caricare i dati e aprire un foglio. Creare una nuova tabella. Creare una dimensione per mostrare i nomi dei trimestri:  
=quartername(manufacture\_date)
2. Quindi, creare una dimensione per identificare quali prodotti sono difettosi e quali senza difetti:  
=if(inquartertodate(manufacture\_date,makedate(2022,05,15),0),'Defective','Faultless')
3. Creare una misura per sommare il valore cost\_price dei prodotti:  
=sum(cost\_price)
4. Impostare la **Formattazione numero** della misura su **Denaro**.

Tabella dei risultati

| quartername(data_di_produzione) | if(inquartertodate(data_di_produzione,makedate(2022,05,15),0),'Difettoso','Senza difetti') | =sum(cost_price) |
|---------------------------------|--------------------------------------------------------------------------------------------|------------------|
| gen-mar 2022                    | Non difettoso                                                                              | 253.89 \$        |
| apr-giu 2022                    | Non difettoso                                                                              | \$229.03         |
| apr-giu 2022                    | Difettoso                                                                                  | \$122.45         |
| lug-set 2022                    | Non difettoso                                                                              | USD 446.31       |
| ott-dic 2022                    | Non difettoso                                                                              | USD 163.91       |

La funzione inquartertodate() restituisce un valore booleano quando valuta le date di produzione di ciascun prodotto. Per coloro che restituiscono il valore booleano TRUE, contrassegna i prodotti come 'Defective'. Per qualsiasi prodotto che restituisce il valore FALSE, e quindi non realizzato nel trimestre fino al 15 maggio incluso, contrassegna i prodotti come 'Faultless'.

## inweek

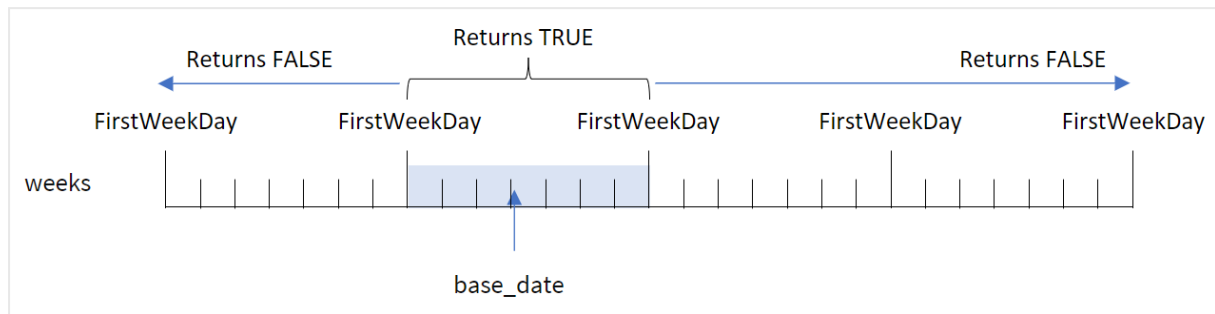
Questa funzione restituisce True se **timestamp** ricade all'interno della settimana contenente **base\_date**.

Sintassi:

```
InWeek (timestamp, base_date, period_no[, first_week_day])
```

Tipo di dati restituiti: Boolean

In Qlik Sense, il valore booleano vero è rappresentato da -1 e il valore falso è rappresentato da 0.

Schema dell'intervallo della funzione `inweek()`

La funzione `inweek()` utilizza l'argomento `base_date` per identificare in quale periodo di sette giorni cade la data. Il giorno iniziale della settimana si basa sulla variabile di sistema `FirstWeekDay`. Tuttavia, è possibile anche modificare il primo giorno della settimana utilizzando l'argomento `first_week_day` nella funzione `inweek()`.

Dopo aver definito la settimana selezionata, la funzione restituisce risultati booleani quando si confrontano i valori delle date prescritte con quel segmento di settimana.

### Casi di utilizzo

La funzione `Inweek` restituisce un risultato booleano. In genere, questo tipo di funzione viene utilizzato come condizione in un file `if expression`. La funzione `Inweek` restituisce un'aggregazione o un calcolo che dipende dal fatto che una data valutata si sia verificata nella settimana con la data selezionata dell'argomento `base_date`.

Ad esempio, la funzione `Inweek` può essere utilizzata per identificare tutte le apparecchiature prodotte in una determinata settimana.

### Argomenti

| Argomento             | Descrizione                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>timestamp</b>      | La data da confrontare con <b>base_date</b> .                                                                                                                                                                                                                                                                                                                                                      |
| <b>base_date</b>      | La data utilizzata per valutare la settimana.                                                                                                                                                                                                                                                                                                                                                      |
| <b>period_no</b>      | La settimana può essere differita mediante <b>period_no</b> . <b>period_no</b> è un numero intero, in cui il valore 0 indica la settimana che contiene <b>base_date</b> . I valori negativi di <b>period_no</b> indicano le settimane precedenti, mentre i valori positivi indicano le settimane successive.                                                                                       |
| <b>first_week_day</b> | Per impostazione predefinita, il primo giorno della settimana è domenica (come determinato dalla variabile di sistema <code>FirstWeekDay</code> ), a partire dalla mezzanotte tra sabato e domenica. Il parametro <b>first_week_day</b> sostituisce la variabile <b>FirstWeekDay</b> . Per indicare un altro giorno per l'inizio della settimana, specificare un contrassegno con valore da 0 a 6. |

valori first\_week\_day

| Giorno    | Valore |
|-----------|--------|
| Lunedì    | 0      |
| Martedì   | 1      |
| Mercoledì | 2      |
| Giovedì   | 3      |
| Venerdì   | 4      |
| Sabato    | 5      |
| Domenica  | 6      |

## Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempi di funzioni

| Esempio                                                              | Risultato         |
|----------------------------------------------------------------------|-------------------|
| <code>inweek<br/>( '01/12/2006' ,<br/>'01/14/2006' , 0 )</code>      | Restituisce TRUE  |
| <code>inweek<br/>( '01/12/2006' ,<br/>'01/20/2006' , 0 )</code>      | Restituisce FALSE |
| <code>inweek<br/>( '01/12/2006' ,<br/>'01/14/2006' , -1<br/>)</code> | Restituisce FALSE |
| <code>inweek<br/>( '01/07/2006' ,<br/>'01/14/2006' , -1 )</code>     | Restituisce TRUE  |

| Esempio                                               | Risultato                                                                                                                                                                               |
|-------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| inweek<br>( '01/12/2006' ,<br>'01/09/2006' , 0,<br>3) | Restituisce FALSE perché first_week_day è specificato come 3 (giovedì), il che rende il 12/01/2006 il primo giorno della settimana successiva a quella contenente il giorno 09/01/2006. |

I seguenti argomenti possono aiutarti a lavorare con questa funzione:

#### Argomenti correlati

| Argomento                               | Contrassegno predefinito / Valore | Descrizione                                      |
|-----------------------------------------|-----------------------------------|--------------------------------------------------|
| <a href="#">FirstWeekDay (page 242)</a> | 6 / Domenica                      | Definisce il giorno di inizio di ogni settimana. |

## Esempio 1 - Nessun argomento aggiuntivo

Script di caricamento e risultati

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il mese di gennaio 2022, caricato in una tabella denominata 'Transactions'.
- La variabile di sistema Firstweekday che è impostata su 6 (domenica).
- Un'istruzione di caricamento precedente che contiene i seguenti elementi:
  - La funzione inweek(), impostata come campo 'in\_week' che determina quali transazioni hanno avuto luogo nella settimana del 14 gennaio 2022.
  - La funzione weekday(), impostata come campo 'week\_day' che mostra quale giorno della settimana corrisponde a ciascuna data.

### Script di caricamento

```
SET FirstWeekDay=6;
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
    Load
        *,
        weekday(date) as week_day,
        inweek(date, '01/14/2022', 0) as in_week
    ;
Load
*
Inline
[
```

```
id,date,amount
8188,'01/02/2022',37.23
8189,'01/05/2022',17.17
8190,'01/06/2022',88.27
8191,'01/08/2022',57.42
8192,'01/09/2022',53.80
8193,'01/10/2022',82.06
8194,'01/11/2022',40.39
8195,'01/12/2022',87.21
8196,'01/13/2022',95.93
8197,'01/14/2022',45.89
8198,'01/15/2022',36.23
8199,'01/16/2022',25.66
8200,'01/17/2022',82.77
8201,'01/18/2022',69.98
8202,'01/26/2022',76.11
8203,'01/27/2022',25.12
8204,'01/28/2022',46.23
8205,'01/29/2022',84.21
8206,'01/30/2022',96.24
8207,'01/31/2022',67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- week\_day
- in\_week

Tabella dei risultati

| date       | week_day | nella settimana |
|------------|----------|-----------------|
| 01/02/2022 | Sun      | 0               |
| 01/05/2022 | Wed      | 0               |
| 01/06/2022 | Thu      | 0               |
| 01/08/2022 | Sat      | 0               |
| 09/01/2022 | Sun      | -1              |
| 10/01/2022 | Mon      | -1              |
| 01/11/2022 | Tue      | -1              |
| 12/01/2022 | Wed      | -1              |
| 13/01/2022 | Thu      | -1              |
| 01/14/2022 | Ven      | -1              |
| 15/01/2022 | Sat      | -1              |

| date       | week_day | nella settimana |
|------------|----------|-----------------|
| 16/01/2022 | Sun      | 0               |
| 17/01/2022 | Mon      | 0               |
| 18/01/2022 | Tue      | 0               |
| 26/01/2022 | Wed      | 0               |
| 27/01/2022 | Thu      | 0               |
| 28/01/2022 | Ven      | 0               |
| 29/01/2022 | Sat      | 0               |
| 30/01/2022 | Sun      | 0               |
| 01/31/2022 | Mon      | 0               |

Il campo 'in\_week' viene creato nell'istruzione LOAD precedente utilizzando la funzione `inweek()`. Il primo argomento identifica il campo da valutare. Il secondo argomento è una data codificata, il 14 gennaio, che è il `base_date`. L'argomento `base_date` collabora con la variabile di sistema `FirstWeekDay` per identificare la settimana del comparatore. Un `period_no` di 0 – significa che la funzione non sta confrontando le settimane precedenti o successive alla settimana segmentata - è l'argomento finale.

La variabile di sistema `FirstWeekDay` determina che le settimane iniziano di domenica e terminano di sabato. Pertanto, gennaio viene suddiviso in settimane secondo lo schema seguente, con le date comprese tra il 9 e il 15 gennaio che forniscono il periodo valido per il calcolo `inweek()`:

Schema del calendario con evidenziato l'intervallo della funzione `inweek()`

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     |     |     |     |     | 1   |
| 2   | 3   | 4   | 5   | 6   | 7   | 8   |
| 9   | 10  | 11  | 12  | 13  | 14  | 15  |
| 16  | 17  | 18  | 19  | 20  | 21  | 22  |
| 23  | 24  | 25  | 26  | 27  | 28  | 29  |
| 30  | 31  |     |     |     |     |     |

Qualsiasi transazione avvenuta tra il 9 e il 15 gennaio restituisce il risultato booleano di `TRUE`.

### Esempio 2 - `period_no`

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Lo stesso set di dati contenente un insieme di transazioni per il 2022 viene caricato in una tabella denominata 'Transactions'.
- La variabile di sistema `Firstweekday` che è impostata su 6 (domenica).
- Un'istruzione `LOAD` precedente che contiene i seguenti elementi:
  - La funzione `inweek()`, impostata come campo 'prev\_week' che determina quali transazioni hanno avuto luogo in un'intera settimana prima della settimana del 14 gennaio 2022.

- La funzione `weekday()`, impostata come campo `'week_day'` che mostra quale giorno della settimana corrisponde a ciascuna data.

### Script di caricamento

```
SET FirstWeekDay=6;
SET DateFormat='MM/DD/YYYY';

Transactions:
    Load
        *,
        weekday(date) as week_day,
        inweek(date,'01/14/2022', -1) as prev_week
    ;

Load
*
Inline
[
id,date,amount
8188,'01/02/2022',37.23
8189,'01/05/2022',17.17
8190,'01/06/2022',88.27
8191,'01/08/2022',57.42
8192,'01/09/2022',53.80
8193,'01/10/2022',82.06
8194,'01/11/2022',40.39
8195,'01/12/2022',87.21
8196,'01/13/2022',95.93
8197,'01/14/2022',45.89
8198,'01/15/2022',36.23
8199,'01/16/2022',25.66
8200,'01/17/2022',82.77
8201,'01/18/2022',69.98
8202,'01/26/2022',76.11
8203,'01/27/2022',25.12
8204,'01/28/2022',46.23
8205,'01/29/2022',84.21
8206,'01/30/2022',96.24
8207,'01/31/2022',67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- week\_day
- prev\_week

Tabella dei risultati

| date       | week_day | settimana precedente |
|------------|----------|----------------------|
| 01/02/2022 | Sun      | -1                   |
| 01/05/2022 | Wed      | -1                   |
| 01/06/2022 | Thu      | -1                   |
| 01/08/2022 | Sat      | -1                   |
| 09/01/2022 | Sun      | 0                    |
| 10/01/2022 | Mon      | 0                    |
| 01/11/2022 | Tue      | 0                    |
| 12/01/2022 | Wed      | 0                    |
| 13/01/2022 | Thu      | 0                    |
| 01/14/2022 | Ven      | 0                    |
| 15/01/2022 | Sat      | 0                    |
| 16/01/2022 | Sun      | 0                    |
| 17/01/2022 | Mon      | 0                    |
| 18/01/2022 | Tue      | 0                    |
| 26/01/2022 | Wed      | 0                    |
| 27/01/2022 | Thu      | 0                    |
| 28/01/2022 | Ven      | 0                    |
| 29/01/2022 | Sat      | 0                    |
| 30/01/2022 | Sun      | 0                    |
| 01/31/2022 | Mon      | 0                    |

L'utilizzo di -1 come argomento `period_no` nella funzione `inweek()` sposta i confini della settimana del comparatore indietro di sette giorni interi. Con un `period_no` di 0 la settimana sarebbe compresa tra il 9 e il 15 gennaio. Eppure in questo esempio, il `period_no` di -1 sposta il confine iniziale e finale di questo segmento indietro di una settimana. I limiti di data diventano dal 2 gennaio all'8 gennaio.

Schema del calendario con evidenziato l'intervallo della funzione `inweek()`

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     |     |     |     |     | 1   |
| 2   | 3   | 4   | 5   | 6   | 7   | 8   |
| 9   | 10  | 11  | 12  | 13  | 14  | 15  |
| 16  | 17  | 18  | 19  | 20  | 21  | 22  |
| 23  | 24  | 25  | 26  | 27  | 28  | 29  |
| 30  | 31  |     |     |     |     |     |

Pertanto, qualsiasi transazione che si verifichi tra il 2 gennaio e l'8 gennaio restituirà il risultato booleano di TRUE.

### Esempio 3 - first\_week\_day

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Lo stesso set di dati contenente un insieme di transazioni per il 2022 viene caricato in una tabella denominata 'Transactions'.
- La variabile di sistema `Firstweekday` che è impostata su 6 (domenica).
- Un'istruzione `LOAD` precedente che contiene i seguenti elementi:
  - La funzione `inweek()`, impostata come campo 'in\_week' che determina quali transazioni hanno avuto luogo nella settimana del 14 gennaio 2022.

- La funzione `weekday()`, impostata come campo `'week_day'` che mostra quale giorno della settimana corrisponde a ciascuna data.

### Script di caricamento

```
SET FirstWeekDay=6;
SET DateFormat='MM/DD/YYYY';

Transactions:
    Load
        *,
        weekday(date) as week_day,
        inweek(date,'01/14/2022', 0, 0) as in_week
    ;

Load
*
Inline
[
id,date,amount
8188,'01/02/2022',37.23
8189,'01/05/2022',17.17
8190,'01/06/2022',88.27
8191,'01/08/2022',57.42
8192,'01/09/2022',53.80
8193,'01/10/2022',82.06
8194,'01/11/2022',40.39
8195,'01/12/2022',87.21
8196,'01/13/2022',95.93
8197,'01/14/2022',45.89
8198,'01/15/2022',36.23
8199,'01/16/2022',25.66
8200,'01/17/2022',82.77
8201,'01/18/2022',69.98
8202,'01/26/2022',76.11
8203,'01/27/2022',25.12
8204,'01/28/2022',46.23
8205,'01/29/2022',84.21
8206,'01/30/2022',96.24
8207,'01/31/2022',67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- week\_day
- in\_week

Tabella dei risultati

| date       | week_day | nella settimana |
|------------|----------|-----------------|
| 01/02/2022 | Sun      | 0               |
| 01/05/2022 | Wed      | 0               |
| 01/06/2022 | Thu      | 0               |
| 01/08/2022 | Sat      | 0               |
| 09/01/2022 | Sun      | 0               |
| 10/01/2022 | Mon      | -1              |
| 01/11/2022 | Tue      | -1              |
| 12/01/2022 | Wed      | -1              |
| 13/01/2022 | Thu      | -1              |
| 01/14/2022 | Ven      | -1              |
| 15/01/2022 | Sat      | -1              |
| 16/01/2022 | Sun      | -1              |
| 17/01/2022 | Mon      | 0               |
| 18/01/2022 | Tue      | 0               |
| 26/01/2022 | Wed      | 0               |
| 27/01/2022 | Thu      | 0               |
| 28/01/2022 | Ven      | 0               |
| 29/01/2022 | Sat      | 0               |
| 30/01/2022 | Sun      | 0               |
| 01/31/2022 | Mon      | 0               |

Utilizzando 0 come argomento `first_week_day` nella funzione `inweek()` si sostituisce la variabile di sistema `FirstWeekDay` e si imposta lunedì come primo giorno della settimana.

*Schema del calendario con evidenziato l'intervallo della funzione `inweek()`*

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     |     |     |     |     | 1   |
| 2   | 3   | 4   | 5   | 6   | 7   | 8   |
| 9   | 10  | 11  | 12  | 13  | 14  | 15  |
| 16  | 17  | 18  | 19  | 20  | 21  | 22  |
| 23  | 24  | 25  | 26  | 27  | 28  | 29  |
| 30  | 31  |     |     |     |     |     |

Pertanto, qualsiasi transazione che si verifichi tra il 10 e il 16 gennaio restituirà un risultato booleano di TRUE.

### Esempio 4 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, in questo esempio, il set di dati è invariato e viene caricato nell'applicazione. Creare una misura nella tabella dei risultati per determinare quali transazioni sono state effettuate nella settimana del 14 gennaio 2022.

#### Script di caricamento

```
SET FirstWeekDay=6;  
SET DateFormat='MM/DD/YYYY';
```

Transactions:

```
Load
*
Inline
[
id,date,amount
8188,'01/02/2022',37.23
8189,'01/05/2022',17.17
8190,'01/06/2022',88.27
8191,'01/08/2022',57.42
8192,'01/09/2022',53.80
8193,'01/10/2022',82.06
8194,'01/11/2022',40.39
8195,'01/12/2022',87.21
8196,'01/13/2022',95.93
8197,'01/14/2022',45.89
8198,'01/15/2022',36.23
8199,'01/16/2022',25.66
8200,'01/17/2022',82.77
8201,'01/18/2022',69.98
8202,'01/26/2022',76.11
8203,'01/27/2022',25.12
8204,'01/28/2022',46.23
8205,'01/29/2022',84.21
8206,'01/30/2022',96.24
8207,'01/31/2022',67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- date

Creare le seguenti misure:

- =inweek (date, '01/14/2022', 0), per calcolare se le transazioni sono avvenute nella stessa settimana del 14 gennaio.
- =weekday(date), per mostrare quale giorno della settimana corrisponde a ciascuna data.

Tabella dei risultati

| date       | week_day | =inweek (date,'01/14/2022',0) |
|------------|----------|-------------------------------|
| 01/02/2022 | Sun      | 0                             |
| 01/05/2022 | Wed      | 0                             |
| 01/06/2022 | Thu      | 0                             |
| 01/08/2022 | Sat      | 0                             |
| 09/01/2022 | Sun      | -1                            |
| 10/01/2022 | Mon      | -1                            |

| date       | week_day | =inweek (date,'01/14/2022',0) |
|------------|----------|-------------------------------|
| 01/11/2022 | Tue      | -1                            |
| 12/01/2022 | Wed      | -1                            |
| 13/01/2022 | Thu      | -1                            |
| 01/14/2022 | Ven      | -1                            |
| 15/01/2022 | Sat      | -1                            |
| 16/01/2022 | Sun      | 0                             |
| 17/01/2022 | Mon      | 0                             |
| 18/01/2022 | Tue      | 0                             |
| 26/01/2022 | Wed      | 0                             |
| 27/01/2022 | Thu      | 0                             |
| 28/01/2022 | Ven      | 0                             |
| 29/01/2022 | Sat      | 0                             |
| 30/01/2022 | Sun      | 0                             |
| 01/31/2022 | Mon      | 0                             |

La misura 'in\_week' viene creata nel grafico utilizzando la funzione `inweek()`. Il primo argomento identifica il campo da valutare. Il secondo argomento è una data codificata, il 14 gennaio, che è il `base_date`.

L'argomento `base_date` collabora con la variabile di sistema `FirstWeekDay` per identificare la settimana del comparatore. Un `period_no` di 0 rappresenta l'argomento finale.

La variabile di sistema `determina` che le settimane iniziano di domenica e terminano di sabato. `FirstWeekDay` Pertanto, gennaio viene suddiviso in settimane secondo lo schema seguente, con le date comprese tra il 9 e il 15 gennaio che forniscono il periodo valido per il calcolo `inweek()`:

Schema del calendario con evidenziato l'intervallo della funzione `inweek()`

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     |     |     |     |     | 1   |
| 2   | 3   | 4   | 5   | 6   | 7   | 8   |
| 9   | 10  | 11  | 12  | 13  | 14  | 15  |
| 16  | 17  | 18  | 19  | 20  | 21  | 22  |
| 23  | 24  | 25  | 26  | 27  | 28  | 29  |
| 30  | 31  |     |     |     |     |     |

Qualsiasi transazione avvenuta tra il 9 e il 15 gennaio restituisce il risultato booleano di `TRUE`.

### Esempio 5 - Scenario

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella chiamata 'Products'.
- La tabella contiene i seguenti campi:
  - ID prodotto
  - tipo di prodotto
  - data di produzione
  - prezzo di costo

Si è determinato che, a causa di un errore delle apparecchiature, i prodotti fabbricati nella settimana del 12 gennaio erano difettosi. L'utente finale vorrebbe un grafico che visualizzi, per settimana, lo stato dei prodotti fabbricati "difettosi" o "non difettosi" e il costo dei prodotti fabbricati in quella settimana.

### Script di caricamento

```
Products:
Load
*
Inline
[
product_id,manufacture_date,cost_price
8188,'01/02/2022',37.23
8189,'01/05/2022',17.17
8190,'01/06/2022',88.27
8191,'01/08/2022',57.42
8192,'01/09/2022',53.80
8193,'01/10/2022',82.06
8194,'01/11/2022',40.39
8195,'01/12/2022',87.21
8196,'01/13/2022',95.93
8197,'01/14/2022',45.89
8198,'01/15/2022',36.23
8199,'01/16/2022',25.66
8200,'01/17/2022',82.77
8201,'01/18/2022',69.98
8202,'01/26/2022',76.11
8203,'01/27/2022',25.12
8204,'01/28/2022',46.23
8205,'01/29/2022',84.21
8206,'01/30/2022',96.24
8207,'01/31/2022',67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `=weekname(manufacture_date)`

Creare le seguenti misure:

- `=if(only(inweek(manufacture_date,makedate(2022,01,12),0)), 'Defective', 'Faultless')`, per identificare quali prodotti sono difettosi e quali non lo sono utilizzando la funzione `inweek()`.
- `=sum(cost_price)`, per mostrare la somma dei costi di ciascun prodotto.

**Procedere come indicato di seguito:**

1. Impostare la misura **Formattazione numero** su **Denaro**.
2. In **Aspetto**, disattivare **Totals**.

Tabella dei risultati

| nomesettimana(data_<br>di_produzione) | =se(solo(insettimana(data_di_produzione,creadata<br>(2022,01,12),0)), 'Difettoso','Senza_difetti') | =sum<br>(cost_<br>price) |
|---------------------------------------|----------------------------------------------------------------------------------------------------|--------------------------|
| 2022/02                               | Non difettoso                                                                                      | 200.09                   |
| 2022/03                               | Difettoso                                                                                          | 441.51                   |
| 2022/04                               | Non difettoso                                                                                      | 178.41                   |
| 2022/05                               | Non difettoso                                                                                      | 231.67                   |
| 2022/06                               | Non difettoso                                                                                      | 163.91                   |

La funzione `inweek()` restituisce un valore booleano quando valuta le date di produzione di ciascun prodotto. Per qualsiasi prodotto fabbricato nella settimana del 12 gennaio, la funzione `inweek()` restituisce un valore booleano TRUE e contrassegna i prodotti come "difettosi". Per tutti i prodotti che restituiscono un valore pari a FALSE, e che quindi non sono stati prodotti in quella settimana, vengono contrassegnati come "non difettosi".

## inweektoday

Questa funzione restituisce True se **timestamp** ricade all'interno della parte della settimana contenente **base\_date** fino a includere l'ultimo millisecondo di **base\_date**.

**Sintassi:**

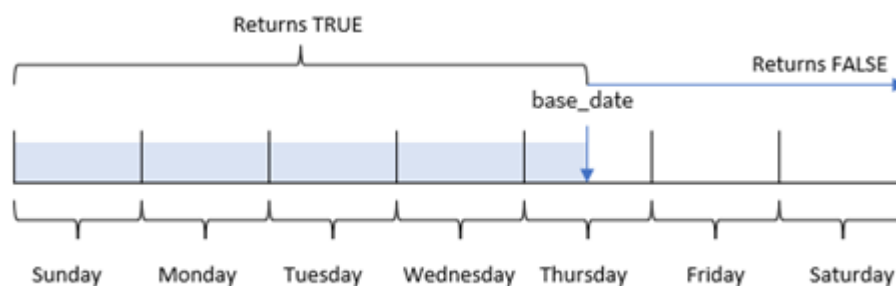
**InWeekToDate** (timestamp, base\_date, period\_no [, first\_week\_day])

**Tipo di dati restituiti:** Boolean



*In Qlik Sense, il valore booleano vero è rappresentato da -1 e il valore falso è rappresentato da 0.*

Diagramma della funzione `inweektoday`



La funzione `inweektoday()` utilizza il parametro `base_date` per identificare una data limite massima di un segmento settimanale, nonché la data corrispondente per l'inizio della settimana, che si basa sulla variabile di sistema `FirstWeekDay` (o sul parametro `first_week_day` definito dall'utente). Una volta che questo segmento di settimana viene definito, la funzione restituisce un risultato booleano quando si confrontano i valori di data prescritti con quel segmento.

### Casi di utilizzo

La funzione `inweektoday()` restituisce un risultato booleano. In genere, questo tipo di funzione viene utilizzato come condizione in un'espressione `if`. Questa restituisce un'aggregazione o un calcolo che dipende dal fatto che una data valutata si sia verificata nella settimana in questione fino a includere una data specifica.

Ad esempio, la funzione `inweektoday()` può essere utilizzata per calcolare tutte le vendite effettuate durante una settimana specificata fino a una data particolare.

Argomenti

| Argomento             | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>timestamp</b>      | La data da confrontare con <b>base_date</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>base_date</b>      | La data utilizzata per valutare la settimana.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>period_no</b>      | La settimana può essere differita mediante <b>period_no</b> . <b>period_no</b> è un numero intero, in cui il valore 0 indica la settimana che contiene <b>base_date</b> . I valori negativi di <b>period_no</b> indicano le settimane precedenti, mentre i valori positivi indicano le settimane successive.                                                                                                                                                                                                                                                                                                  |
| <b>first_week_day</b> | <p>Per impostazione predefinita, il primo giorno della settimana è domenica (come determinato dalla variabile di sistema <code>FirstWeekDay</code>), a partire dalla mezzanotte tra sabato e domenica. Il parametro <b>first_week_day</b> sostituisce la variabile <b>FirstWeekDay</b>. Per indicare un altro giorno per l'inizio della settimana, specificare un contrassegno con valore da 0 a 6.</p> <p>Per una settimana che inizia il lunedì e termina la domenica, usare il contrassegno 0 per lunedì, 1 per martedì, 2 per mercoledì, 3 per giovedì, 4 per venerdì, 5 per sabato e 6 per domenica.</p> |

Esempi di funzioni

| Esempio                                                 | Interazione                      |
|---------------------------------------------------------|----------------------------------|
| <code>inweektoday('01/12/2006', '01/12/2006', 0)</code> | Restituisce <code>TRUE</code> .  |
| <code>inweektoday('01/12/2006', '01/11/2006', 0)</code> | Restituisce <code>FALSE</code> . |

| Esempio                                                                   | Interazione                                                                                                                                                                                           |
|---------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>inweektoday<br/>( '01/12/2006',<br/>'01/18/2006', -1)</code>        | Restituisce FALSE.<br>Perché <code>period_no</code> è specificato come -1, la data effettiva con cui viene misurato <code>timestamp</code> è 01/11/2006.                                              |
| <code>inweektoday<br/>( '01/11/2006',<br/>'01/12/2006', 0, 3<br/>)</code> | Restituisce FALSE, poiché il valore <code>first_week_day</code> è specificato come 3 (giovedì), rendendo 01/12/2006 il primo giorno della settimana successiva alla settimana che include 01/12/2006. |

I seguenti argomenti possono aiutarti a lavorare con questa funzione:

#### Argomenti correlati

| Argomento                               | Contrassegno predefinito / Valore | Descrizione                                      |
|-----------------------------------------|-----------------------------------|--------------------------------------------------|
| <a href="#">FirstWeekDay (page 242)</a> | 6 / Domenica                      | Definisce il giorno di inizio di ogni settimana. |

## Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

## Esempio 1 - Nessun argomento aggiuntivo

Script di caricamento e risultati

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il mese di gennaio 2022, caricato in una tabella denominata `Transactions`.
- Il campo dati fornito nel formato `TimestampFormat='M/D/YYYY h:mm:ss[.fff]'`.

- La creazione di un campo, `in_week_to_date`, che determina quali transazioni sono avvenute nella settimana fino al 14 gennaio 2022.
- La creazione di un campo aggiuntivo, denominato `weekday`, utilizzando la funzione `weekday()`. Questo nuovo campo viene creato per mostrare quale giorno della settimana corrisponde a ciascuna data.

### Script di caricamento

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff]';
SET FirstWeekDay=6;
Transactions:
    Load
        *,
        weekday(date) as week_day,
        inweektoday(date,'01/14/2022', 0) as in_week_to_date
    ;

Load
*
Inline
[
id,date,amount
8188,'2022-01-02 12:22:06',37.23
8189,'2022-01-05 01:02:30',17.17
8190,'2022-01-06 15:36:20',88.27
8191,'2022-01-08 10:58:35',57.42
8192,'2022-01-09 08:53:32',53.80
8193,'2022-01-10 21:13:01',82.06
8194,'2022-01-11 00:57:13',40.39
8195,'2022-01-12 09:26:02',87.21
8196,'2022-01-13 15:05:09',95.93
8197,'2022-01-14 18:44:57',45.89
8198,'2022-01-15 06:10:46',36.23
8199,'2022-01-16 06:39:27',25.66
8200,'2022-01-17 10:44:16',82.77
8201,'2022-01-18 18:48:17',69.98
8202,'2022-01-26 04:36:03',76.11
8203,'2022-01-27 08:07:49',25.12
8204,'2022-01-28 12:24:29',46.23
8205,'2022-01-30 11:56:56',84.21
8206,'2022-01-30 14:40:19',96.24
8207,'2022-01-31 05:28:21',67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `date`
- `week_day`
- `in_week_to_date`

Tabella dei risultati

| date                | week_day | in_week_to_date |
|---------------------|----------|-----------------|
| 2022-01-02 12:22:06 | Sun      | 0               |
| 2022-01-05 01:02:30 | Wed      | 0               |
| 2022-01-06 15:36:20 | Thu      | 0               |
| 2022-01-08 10:58:35 | Sat      | 0               |
| 2022-01-09 08:53:32 | Sun      | -1              |
| 2022-01-10 21:13:01 | Mon      | -1              |
| 2022-01-11 00:57:13 | Tue      | -1              |
| 2022-01-12 09:26:02 | Wed      | -1              |
| 2022-01-13 15:05:09 | Thu      | -1              |
| 2022-01-14 18:44:57 | Ven      | -1              |
| 2022-01-15 06:10:46 | Sat      | 0               |
| 2022-01-16 06:39:27 | Sun      | 0               |
| 2022-01-17 10:44:16 | Mon      | 0               |
| 2022-01-18 18:48:17 | Tue      | 0               |
| 2022-01-26 04:36:03 | Wed      | 0               |
| 2022-01-27 08:07:49 | Thu      | 0               |
| 2022-01-28 12:24:29 | Ven      | 0               |
| 2022-01-30 11:56:56 | Sun      | 0               |
| 2022-01-30 14:40:19 | Sun      | 0               |
| 2022-01-31 05:28:21 | Mon      | 0               |

Il campo `in_week_to_date` viene creato nell'istruzione caricamento precedente utilizzando la funzione `inweektodate()`. Il primo argomento fornito identifica il campo da valutare. Il secondo argomento è una data codificata per il 14 gennaio, che è il valore `base_date` che identifica la settimana da segmentare e definisce il limite finale di quel segmento. Un valore `period_no` uguale a 0 è l'argomento finale, il che significa che la funzione non confronta le settimane precedenti o successive alla settimana segmentata.

La variabile di sistema `FirstWeekDay` determina che le settimane iniziano di domenica e terminano di sabato. Pertanto, gennaio viene suddiviso in settimane secondo il diagramma seguente, con le date comprese tra il 9 e il 14 gennaio che forniscono il periodo valido per il calcolo `inweektodate()`:

Diagramma del calendario che mostra le date della transazione che restituiscono un risultato booleano TRUE

| Sun | Mon | Tue | Wed | Thur | Fri | Sat |
|-----|-----|-----|-----|------|-----|-----|
|     |     |     |     |      |     | 1   |
| 2   | 3   | 4   | 5   | 6    | 7   | 8   |
| 9   | 10  | 11  | 12  | 13   | 14  | 15  |
| 16  | 17  | 18  | 19  | 20   | 21  | 22  |
| 23  | 24  | 25  | 26  | 27   | 28  | 29  |
| 30  | 31  |     |     |      |     |     |

Qualsiasi transazione avvenuta tra il 9 e il 14 gennaio restituisce il risultato booleano TRUE. Le transazioni prima e dopo quelle date restituiscono il risultato booleano FALSE.

### Esempio 2 - period\_no

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.
- La creazione di un campo, `prev_week_to_date`, che determina quali transazioni sono avvenute nella settimana precedente al segmento della settimana che termina il 14 gennaio 2022.
- La creazione di un campo aggiuntivo, denominato `weekday`, utilizzando la funzione `weekday()`. Questo mostra quale giorno della settimana corrisponde a ciascuna data.

#### Script di caricamento

```
SET FirstWeekDay=6;
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff]';
Transactions:
    Load
        *,
        weekday(date) as week_day,
        inweektodate(date,'01/14/2022', -1) as prev_week_to_date
    ;
Load
*
Inline
[
id,date,amount
```

```
8188, '2022-01-02 12:22:06', 37.23
8189, '2022-01-05 01:02:30', 17.17
8190, '2022-01-06 15:36:20', 88.27
8191, '2022-01-08 10:58:35', 57.42
8192, '2022-01-09 08:53:32', 53.80
8193, '2022-01-10 21:13:01', 82.06
8194, '2022-01-11 00:57:13', 40.39
8195, '2022-01-12 09:26:02', 87.21
8196, '2022-01-13 15:05:09', 95.93
8197, '2022-01-14 18:44:57', 45.89
8198, '2022-01-15 06:10:46', 36.23
8199, '2022-01-16 06:39:27', 25.66
8200, '2022-01-17 10:44:16', 82.77
8201, '2022-01-18 18:48:17', 69.98
8202, '2022-01-26 04:36:03', 76.11
8203, '2022-01-27 08:07:49', 25.12
8204, '2022-01-28 12:24:29', 46.23
8205, '2022-01-30 11:56:56', 84.21
8206, '2022-01-30 14:40:19', 96.24
8207, '2022-01-31 05:28:21', 67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- week\_day
- prev\_week\_to\_date

Tabella dei risultati

| date                | week_day | prev_week_to_date |
|---------------------|----------|-------------------|
| 2022-01-02 12:22:06 | Sun      | -1                |
| 2022-01-05 01:02:30 | Wed      | -1                |
| 2022-01-06 15:36:20 | Thu      | -1                |
| 2022-01-08 10:58:35 | Sat      | 0                 |
| 2022-01-09 08:53:32 | Sun      | 0                 |
| 2022-01-10 21:13:01 | Mon      | 0                 |
| 2022-01-11 00:57:13 | Tue      | 0                 |
| 2022-01-12 09:26:02 | Wed      | 0                 |
| 2022-01-13 15:05:09 | Thu      | 0                 |
| 2022-01-14 18:44:57 | Ven      | 0                 |
| 2022-01-15 06:10:46 | Sat      | 0                 |

| date                | week_day | prev_week_to_date |
|---------------------|----------|-------------------|
| 2022-01-16 06:39:27 | Sun      | 0                 |
| 2022-01-17 10:44:16 | Mon      | 0                 |
| 2022-01-18 18:48:17 | Tue      | 0                 |
| 2022-01-26 04:36:03 | Wed      | 0                 |
| 2022-01-27 08:07:49 | Thu      | 0                 |
| 2022-01-28 12:24:29 | Ven      | 0                 |
| 2022-01-30 11:56:56 | Sun      | 0                 |
| 2022-01-30 14:40:19 | Sun      | 0                 |
| 2022-01-31 05:28:21 | Mon      | 0                 |

Il valore `period_no` di -1 indica che la funzione `inweektoday` () confronta il segmento del trimestre di input con la settimana precedente. Il segmento della settimana inizialmente corrisponde a una data compresa tra il 9 e il 14 gennaio. Il valore `period_no` quindi sposta sia il limite di inizio che di fine di questo segmento a una settimana prima, facendo sì che i limiti della data siano definiti tra il 2 e il 7 gennaio.

*Diagramma del calendario che mostra le date della transazione che restituiscono un risultato booleano TRUE*

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     |     |     |     |     | 1   |
| 2   | 3   | 4   | 5   | 6   | 7   | 8   |
| 9   | 10  | 11  | 12  | 13  | 14  | 15  |
| 16  | 17  | 18  | 19  | 20  | 21  | 22  |
| 23  | 24  | 25  | 26  | 27  | 28  | 29  |
| 30  | 31  |     |     |     |     |     |

Pertanto, qualsiasi transazione che si verifichi tra il 2 e l'8 gennaio (non includendo lo stesso 8 gennaio) restituirà il risultato booleano `TRUE`.

### Esempio 3 - first\_week\_day

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.
- La creazione di un campo, `in_week_to_date`, che determina quali transazioni sono avvenute nella settimana fino al 14 gennaio 2022.
- La creazione di un campo aggiuntivo, denominato `weekday`, utilizzando la funzione `weekday()`. Questo mostra quale giorno della settimana corrisponde a ciascuna data.

In questo esempio viene utilizzato il domenica come primo giorno della settimana.

### Script di caricamento

```
SET FirstWeekDay=6;
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff]';

Transactions:
    Load
        *,
        weekday(date) as week_day,
        inweektodate(date,'01/14/2022', 0, 0) as in_week_to_date
    ;
Load
*
Inline
[
id,date,amount
8188,'2022-01-02 12:22:06',37.23
8189,'2022-01-05 01:02:30',17.17
8190,'2022-01-06 15:36:20',88.27
8191,'2022-01-08 10:58:35',57.42
8192,'2022-01-09 08:53:32',53.80
8193,'2022-01-10 21:13:01',82.06
8194,'2022-01-11 00:57:13',40.39
8195,'2022-01-12 09:26:02',87.21
8196,'2022-01-13 15:05:09',95.93
8197,'2022-01-14 18:44:57',45.89
8198,'2022-01-15 06:10:46',36.23
8199,'2022-01-16 06:39:27',25.66
8200,'2022-01-17 10:44:16',82.77
8201,'2022-01-18 18:48:17',69.98
8202,'2022-01-26 04:36:03',76.11
8203,'2022-01-27 08:07:49',25.12
8204,'2022-01-28 12:24:29',46.23
8205,'2022-01-30 11:56:56',84.21
8206,'2022-01-30 14:40:19',96.24
8207,'2022-01-31 05:28:21',67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- week\_day

- `in_week_to_date`

Tabella dei risultati

| date                | week_day | in_week_to_date |
|---------------------|----------|-----------------|
| 2022-01-02 12:22:06 | Sun      | 0               |
| 2022-01-05 01:02:30 | Wed      | 0               |
| 2022-01-06 15:36:20 | Thu      | 0               |
| 2022-01-08 10:58:35 | Sat      | 0               |
| 2022-01-09 08:53:32 | Sun      | 0               |
| 2022-01-10 21:13:01 | Mon      | -1              |
| 2022-01-11 00:57:13 | Tue      | -1              |
| 2022-01-12 09:26:02 | Wed      | -1              |
| 2022-01-13 15:05:09 | Thu      | -1              |
| 2022-01-14 18:44:57 | Ven      | -1              |
| 2022-01-15 06:10:46 | Sat      | 0               |
| 2022-01-16 06:39:27 | Sun      | 0               |
| 2022-01-17 10:44:16 | Mon      | 0               |
| 2022-01-18 18:48:17 | Tue      | 0               |
| 2022-01-26 04:36:03 | Wed      | 0               |
| 2022-01-27 08:07:49 | Thu      | 0               |
| 2022-01-28 12:24:29 | Ven      | 0               |
| 2022-01-30 11:56:56 | Sun      | 0               |
| 2022-01-30 14:40:19 | Sun      | 0               |
| 2022-01-31 05:28:21 | Mon      | 0               |

Utilizzando 0 come argomento `first_week_day` nella funzione `inweektodate()`, l'argomento della funzione sostituisce la variabile di sistema `Firstweekday` e imposta il lunedì come primo giorno della settimana.

*Diagramma del calendario che mostra le date della transazione che restituiscono un risultato booleano TRUE*

| Mon | Tue | Wed | Thu | Fri | Sat | Sun |
|-----|-----|-----|-----|-----|-----|-----|
|     |     |     |     |     | 1   | 2   |
| 3   | 4   | 5   | 6   | 7   | 8   | 9   |
| 10  | 11  | 12  | 13  | 14  | 15  | 16  |
| 17  | 18  | 19  | 20  | 21  | 22  | 23  |
| 24  | 25  | 26  | 27  | 28  | 29  | 30  |
| 31  |     |     |     |     |     |     |

Pertanto, qualsiasi transazione che si verifichi tra il 10 e 15 gennaio restituirà un risultato booleano `TRUE`, mentre le transazioni con una data che non rientra in questi limiti restituiranno il valore `FALSE`.

### Esempio 4 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor di caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento contiene lo stesso set di dati e lo stesso scenario del primo esempio. Tuttavia, in questo esempio, il set di dati invariato viene caricato nell'applicazione. Il calcolo che determina quali transazioni sono avvenute nella settimana fino al 14 gennaio viene creato come misura in un oggetto grafico.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'2022-01-02 12:22:06',37.23
```

```
8189,'2022-01-05 01:02:30',17.17
```

```
8190,'2022-01-06 15:36:20',88.27
```

```
8191,'2022-01-08 10:58:35',57.42
```

```
8192,'2022-01-09 08:53:32',53.80
```

```
8193,'2022-01-10 21:13:01',82.06
```

```
8194,'2022-01-11 00:57:13',40.39
```

```
8195, '2022-01-12 09:26:02', 87.21
8196, '2022-01-13 15:05:09', 95.93
8197, '2022-01-14 18:44:57', 45.89
8198, '2022-01-15 06:10:46', 36.23
8199, '2022-01-16 06:39:27', 25.66
8200, '2022-01-17 10:44:16', 82.77
8201, '2022-01-18 18:48:17', 69.98
8202, '2022-01-26 04:36:03', 76.11
8203, '2022-01-27 08:07:49', 25.12
8204, '2022-01-28 12:24:29', 46.23
8205, '2022-01-30 11:56:56', 84.21
8206, '2022-01-30 14:40:19', 96.24
8207, '2022-01-31 05:28:21', 67.67
];
```

### Risultati

#### Procedere come indicato di seguito:

1. Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: date.
2. Per calcolare se le transazioni sono avvenute nella stessa settimana fino al 14 gennaio, creare la misura seguente:  
=inweektoday(date, '01/14/2022', 0)
3. Per mostrare quale giorno della settimana corrisponde a ciascuna data, creare una misura aggiuntiva:  
=weekday(date)

Tabella dei risultati

| date                | week_day | in_week_to_date |
|---------------------|----------|-----------------|
| 2022-01-02 12:22:06 | Sun      | 0               |
| 2022-01-05 01:02:30 | Wed      | 0               |
| 2022-01-06 15:36:20 | Thu      | 0               |
| 2022-01-08 10:58:35 | Sat      | 0               |
| 2022-01-09 08:53:32 | Sun      | -1              |
| 2022-01-10 21:13:01 | Mon      | -1              |
| 2022-01-11 00:57:13 | Tue      | -1              |
| 2022-01-12 09:26:02 | Wed      | -1              |
| 2022-01-13 15:05:09 | Thu      | -1              |
| 2022-01-14 18:44:57 | Ven      | -1              |
| 2022-01-15 06:10:46 | Sat      | 0               |
| 2022-01-16 06:39:27 | Sun      | 0               |

| date                | week_day | in_week_to_date |
|---------------------|----------|-----------------|
| 2022-01-17 10:44:16 | Mon      | 0               |
| 2022-01-18 18:48:17 | Tue      | 0               |
| 2022-01-26 04:36:03 | Wed      | 0               |
| 2022-01-27 08:07:49 | Thu      | 0               |
| 2022-01-28 12:24:29 | Ven      | 0               |
| 2022-01-30 11:56:56 | Sun      | 0               |
| 2022-01-30 14:40:19 | Sun      | 0               |
| 2022-01-31 05:28:21 | Mon      | 0               |

Il campo `in_week_to_date` viene creato come misura nell'oggetto grafico utilizzando la funzione `inweektodate()`. Il primo argomento fornito identifica il campo da valutare. Il secondo argomento è una data codificata per il 14 gennaio, che è il valore `base_date` che identifica la settimana da segmentare e definisce il limite finale di quel segmento. Un valore `period_no` uguale a 0 è l'argomento finale, il che significa che la funzione non confronta le settimane precedenti o successive alla settimana segmentata.

La variabile di sistema `determina` che le settimane iniziano di domenica e terminano di sabato. `FirstWeekDay` Pertanto, gennaio viene suddiviso in settimane secondo il diagramma seguente, con le date comprese tra il 9 e il 14 gennaio che forniscono il periodo valido per il calcolo `inweektodate()`:

*Diagramma del calendario che mostra le date della transazione che restituiscono un risultato booleano TRUE*

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     |     |     |     |     | 1   |
| 2   | 3   | 4   | 5   | 6   | 7   | 8   |
| 9   | 10  | 11  | 12  | 13  | 14  | 15  |
| 16  | 17  | 18  | 19  | 20  | 21  | 22  |
| 23  | 24  | 25  | 26  | 27  | 28  | 29  |
| 30  | 31  |     |     |     |     |     |

Qualsiasi transazione avvenuta tra il 9 e il 14 gennaio restituisce il risultato booleano `TRUE`. Le transazioni prima e dopo quelle date restituiscono il risultato booleano `FALSE`.

### Esempio 5 - Scenario

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella chiamata `Products`.
- Informazioni relative all'ID prodotto, alla data di produzione e al prezzo di costo.

Si è determinato che, a causa di un errore delle apparecchiature, i prodotti fabbricati nella settimana del 12 gennaio erano difettosi. Il problema è stato risolto il 13 gennaio. L'utente finale desidera un oggetto grafico che visualizzi, per settimana, se i prodotti fabbricati sono "difettosi" o "senza difetti" e il costo dei prodotti fabbricati quella settimana.

#### Script di caricamento

```
Products:
Load
*
Inline
[
product_id,manufacture_date,cost_price
8188,'2022-01-02 12:22:06',37.23
8189,'2022-01-05 01:02:30',17.17
8190,'2022-01-06 15:36:20',88.27
8191,'2022-01-08 10:58:35',57.42
8192,'2022-01-09 08:53:32',53.80
8193,'2022-01-10 21:13:01',82.06
8194,'2022-01-11 00:57:13',40.39
8195,'2022-01-12 09:26:02',87.21
8196,'2022-01-13 15:05:09',95.93
8197,'2022-01-14 18:44:57',45.89
8198,'2022-01-15 06:10:46',36.23
8199,'2022-01-16 06:39:27',25.66
8200,'2022-01-17 10:44:16',82.77
8201,'2022-01-18 18:48:17',69.98
8202,'2022-01-26 04:36:03',76.11
8203,'2022-01-27 08:07:49',25.12
8204,'2022-01-28 12:24:29',46.23
8205,'2022-01-30 11:56:56',84.21
8206,'2022-01-30 14:40:19',96.24
8207,'2022-01-31 05:28:21',67.67
];
```

### Risultati

Procedere come indicato di seguito:

1. Caricare i dati e aprire un foglio. Creare una nuova tabella. Creare una dimensione per visualizzare i nomi delle settimane:  
`=weekname(manufacture_date)`
2. Quindi, creare una dimensione per identificare quali prodotti sono difettosi e quali senza difetti:  
`=if(inweektodate(manufacture_date,makedate(2022,01,12),0),'Defective','Faultless')`
3. Creare una misura per sommare il valore `cost_price` dei prodotti:  
`=sum(cost_price)`
4. Impostare la **Formattazione numero** della misura su **Denaro**.

Tabella dei risultati

| <code>weekname(manufacture_date)</code> | <code>if(inweektodate(manufacture_date,makedate(2022,01,12),0),'Defective','Faultless')</code> | <code>=sum(cost_price)</code> |
|-----------------------------------------|------------------------------------------------------------------------------------------------|-------------------------------|
| 2022/02                                 | Non difettoso                                                                                  | \$200.09                      |
| 2022/03                                 | Difettoso                                                                                      | \$263.46                      |
| 2022/03                                 | Non difettoso                                                                                  | \$178.05                      |
| 2022/04                                 | Non difettoso                                                                                  | \$178.41                      |
| 2022/05                                 | Non difettoso                                                                                  | \$147.46                      |
| 2022/06                                 | Non difettoso                                                                                  | \$248.12                      |

La funzione `inweektodate()` restituisce un valore booleano quando valuta le date di produzione di ciascun prodotto. Per coloro che restituiscono il valore booleano `TRUE`, contrassegna i prodotti come 'Defective'. Per qualsiasi prodotto che restituisce il valore `FALSE`, e quindi non realizzato nella settimana fino al 12 gennaio, contrassegna i prodotti come 'Faultless'.

### inyear

Questa funzione restituisce `True` se **timestamp** ricade all'interno dell'anno contenente **base\_date**.

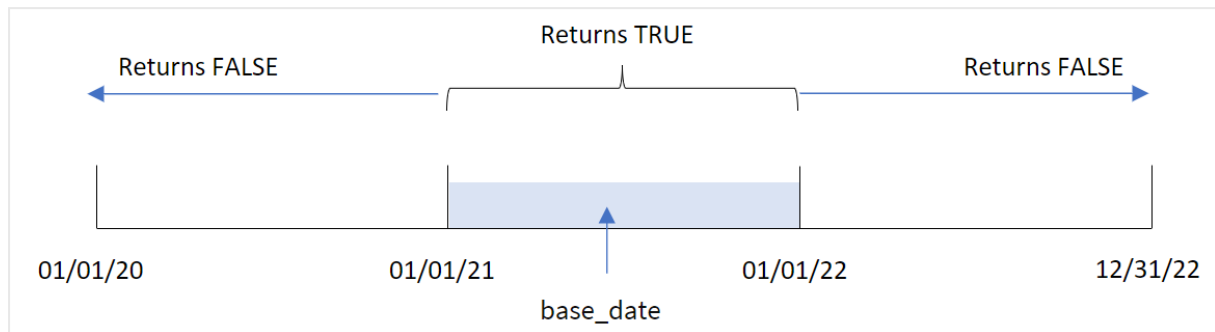
Sintassi:

```
InYear (timestamp, base_date, period_no [, first_month_of_year])
```

Tipo di dati restituiti: Boolean

In Qlik Sense, il valore booleano vero è rappresentato da -1 e il valore falso è rappresentato da 0.

Schema dell'intervallo della funzione `inyear()`



La funzione `inyear()` restituisce un risultato booleano quando si confrontano i valori della data selezionata con un anno definito dall'opzione `base_date`.

### Casi di utilizzo

La funzione `inyear()` restituisce un risultato booleano. In genere, questo tipo di funzione viene utilizzato come condizione in un file `if expression`. Restituisce un'aggregazione o un calcolo che dipende dal fatto che una data valutata si sia verificata nell'anno in questione. Ad esempio, la funzione `inyear()` può essere utilizzata per identificare tutte le vendite avvenute in un determinato anno.

#### Argomenti

| Argomento                  | Descrizione                                                                                                                                                                                                                                                                              |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>timestamp</b>           | La data da confrontare con <b>base_date</b> .                                                                                                                                                                                                                                            |
| <b>base_date</b>           | La data utilizzata per valutare l'anno.                                                                                                                                                                                                                                                  |
| <b>period_no</b>           | L'anno può essere differito mediante <b>period_no</b> . <b>period_no</b> è un numero intero, in cui il valore 0 indica l'anno che contiene <b>base_date</b> . I valori negativi di <b>period_no</b> indicano gli anni precedenti, mentre i valori positivi indicano gli anni successivi. |
| <b>first_month_of_year</b> | Se si intende utilizzare anni (fiscali) che non iniziano a gennaio, indicare un valore compreso tra 2 e 12 in <b>first_month_of_year</b> .                                                                                                                                               |

È possibile utilizzare i seguenti valori per impostare il primo mese dell'anno nell'argomento `first_month_of_year`:

valori `first_month_of_year`

| Mese     | Valore |
|----------|--------|
| Febbraio | 2      |
| Marzo    | 3      |
| Aprile   | 4      |
| Maggio   | 5      |
| Giugno   | 6      |

| Mese      | Valore |
|-----------|--------|
| Luglio    | 7      |
| Agosto    | 8      |
| Settembre | 9      |
| Ottobre   | 10     |
| Novembre  | 11     |
| Dicembre  | 12     |

## Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempi di funzioni

| Esempio                                                 | Risultato                                                                                                                                                                         |
|---------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>inyear ('01/25/2013', '01/01/2013', 0 )</code>    | Restituisce TRUE                                                                                                                                                                  |
| <code>inyear ('01/25/2012', '01/01/2013', 0)</code>     | Restituisce FALSE                                                                                                                                                                 |
| <code>inyear ('01/25/2013', '01/01/2013', -1)</code>    | Restituisce FALSE                                                                                                                                                                 |
| <code>inyear ('01/25/2012', '01/01/2013', -1 )</code>   | Restituisce TRUE                                                                                                                                                                  |
| <code>inyear ('01/25/2013', '01/01/2013', 0, 3)</code>  | Restituisce TRUE<br><br>Il valore di <code>base_date</code> e <code>first_month_of_year</code> specifica che il timestamp deve essere compreso tra il 03/01/2012 e il 28/02/2013. |
| <code>inyear ('03/25/2013', '07/01/2013', 0, 3 )</code> | Restituisce TRUE                                                                                                                                                                  |

### Esempio 1 - Esempio di base

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni tra il 2020 e il 2022, caricato in una tabella denominata 'Transactions'.
- Un caricamento precedente che contiene la funzione `inyear()` impostata come campo 'in\_year' e che determina quali transazioni hanno avuto luogo nello stesso anno del 26 luglio 2021.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
Transactions:
    Load
        *,
        inyear(date,'07/26/2021', 0) as in_year
    ;

Load
*
Inline
[
id,date,amount
8188,'01/13/2020',37.23
8189,'02/26/2020',17.17
8190,'03/27/2020',88.27
8191,'04/16/2020',57.42
8192,'05/21/2020',53.80
8193,'08/14/2020',82.06
8194,'10/07/2020',40.39
8195,'12/05/2020',87.21
8196,'01/22/2021',95.93
8197,'02/03/2021',45.89
8198,'03/17/2021',36.23
8199,'04/23/2021',25.66
8200,'05/04/2021',82.77
8201,'06/30/2021',69.98
8202,'07/26/2021',76.11
8203,'12/27/2021',25.12
8204,'06/06/2022',46.23
8205,'07/18/2022',84.21
8206,'11/14/2022',96.24
8207,'12/12/2022',67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

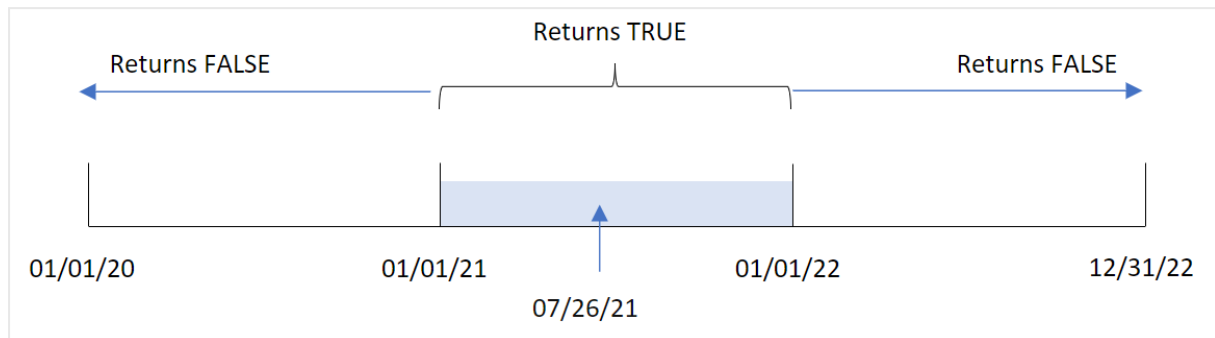
- date
- in\_year

Tabella dei risultati

| date       | in_year |
|------------|---------|
| 01/13/2020 | 0       |
| 02/26/2020 | 0       |
| 03/27/2020 | 0       |
| 04/16/2020 | 0       |
| 05/21/2020 | 0       |
| 08/14/2020 | 0       |
| 10/07/2020 | 0       |
| 12/05/2020 | 0       |
| 01/22/2021 | -1      |
| 02/03/2021 | -1      |
| 03/17/2021 | -1      |
| 04/23/2021 | -1      |
| 05/04/2021 | -1      |
| 06/30/2021 | -1      |
| 07/26/2021 | -1      |
| 12/27/2021 | -1      |
| 06/06/2022 | 0       |
| 07/18/2022 | 0       |
| 11/14/2022 | 0       |
| 12/12/2022 | 0       |

Il campo 'in\_year' viene creato nell'istruzione LOAD precedente utilizzando la funzione `inyear()`. Il primo argomento identifica il campo da valutare. Il secondo argomento è una data codificata per il 26 luglio 2021, ovvero il `base_date` che determina l'anno di confronto. Un `period_no` di 0 è l'argomento finale, il che significa che la funzione `inyear()` non confronta gli anni precedenti o successivi all'anno.

Schema dell'intervallo della funzione `inyear()` con il 26 luglio come data base



Qualsiasi transazione che si verifica nel 2021 restituisce un risultato booleano pari a TRUE.

### Esempio 2 - period\_no

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni tra il 2020 e il 2022, caricato in una tabella denominata Transactions.
- Un caricamento precedente che contiene la funzione `inyear()` impostata come campo 'previous\_year' e determina quali transazioni hanno avuto luogo nell'anno prima dell'anno con 26 luglio 2021.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
Transactions:
  Load
    *,
    inyear(date,'07/26/2021', -1) as previous_year
  ;
Load
*
Inline
[
id,date,amount
8188,'01/13/2020',37.23
8189,'02/26/2020',17.17
8190,'03/27/2020',88.27
8191,'04/16/2020',57.42
8192,'05/21/2020',53.80
8193,'08/14/2020',82.06
8194,'10/07/2020',40.39
8195,'12/05/2020',87.21
8196,'01/22/2021',95.93
8197,'02/03/2021',45.89
```

```
8198, '03/17/2021', 36.23
8199, '04/23/2021', 25.66
8200, '05/04/2021', 82.77
8201, '06/30/2021', 69.98
8202, '07/26/2021', 76.11
8203, '12/27/2021', 25.12
8204, '06/06/2022', 46.23
8205, '07/18/2022', 84.21
8206, '11/14/2022', 96.24
8207, '12/12/2022', 67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- previous\_year

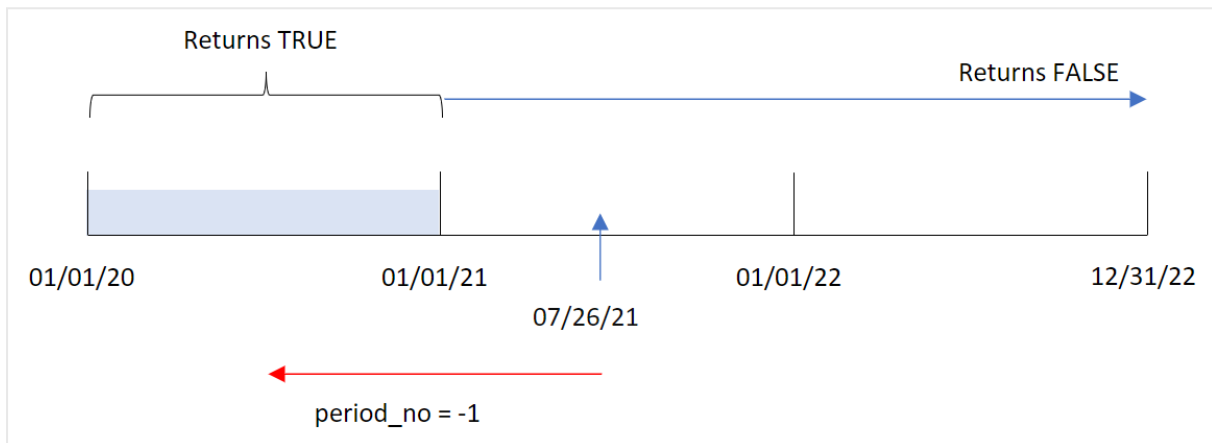
Tabella dei risultati

| date       | previous_year |
|------------|---------------|
| 01/13/2020 | -1            |
| 02/26/2020 | -1            |
| 03/27/2020 | -1            |
| 04/16/2020 | -1            |
| 05/21/2020 | -1            |
| 08/14/2020 | -1            |
| 10/07/2020 | -1            |
| 12/05/2020 | -1            |
| 01/22/2021 | 0             |
| 02/03/2021 | 0             |
| 03/17/2021 | 0             |
| 04/23/2021 | 0             |
| 05/04/2021 | 0             |
| 06/30/2021 | 0             |
| 07/26/2021 | 0             |
| 12/27/2021 | 0             |
| 06/06/2022 | 0             |
| 07/18/2022 | 0             |

| date       | previous_year |
|------------|---------------|
| 11/14/2022 | 0             |
| 12/12/2022 | 0             |

L'utilizzo di -1 come argomento `period_no` della funzione `inyear()` sposta i confini dell'anno di confronto indietro di un intero anno. Il 2021 viene inizialmente identificato come anno di confronto. `period_no` compensa l'anno comparatore di un anno, rendendo il 2020 l'anno di confronto.

*Schema dell'intervallo della funzione `inyear()` con l'argomento `period_no` impostato su -1*



Pertanto, qualsiasi transazione che si verifica nel 2020 restituisce un risultato booleano di TRUE.

### Esempio 3 - first\_month\_of\_year

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni tra il 2020 e il 2022, caricato in una tabella denominata `Transactions`.
- Un caricamento precedente che contiene la funzione `inyear()` impostata come campo `'in_year'` e che determina quali transazioni hanno avuto luogo nello stesso anno del 26 luglio 2021.

Tuttavia, in questo esempio, il criterio organizzativo prevede che marzo sia il primo mese dell'anno finanziario.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
Transactions:
    Load
        *
```

```
        inyear(date,'07/26/2021', 0, 3) as in_year
    ;
Load
*
Inline
[
id,date,amount
8188,'01/13/2020',37.23
8189,'02/26/2020',17.17
8190,'03/27/2020',88.27
8191,'04/16/2020',57.42
8192,'05/21/2020',53.80
8193,'08/14/2020',82.06
8194,'10/07/2020',40.39
8195,'12/05/2020',87.21
8196,'01/22/2021',95.93
8197,'02/03/2021',45.89
8198,'03/17/2021',36.23
8199,'04/23/2021',25.66
8200,'05/04/2021',82.77
8201,'06/30/2021',69.98
8202,'07/26/2021',76.11
8203,'12/27/2021',25.12
8204,'06/06/2022',46.23
8205,'07/18/2022',84.21
8206,'11/14/2022',96.24
8207,'12/12/2022',67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- in\_year

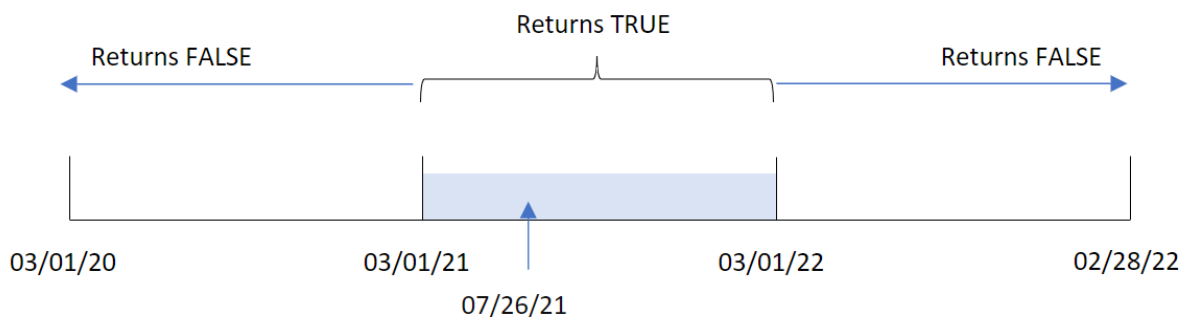
Tabella dei risultati

| date       | in_year |
|------------|---------|
| 01/13/2020 | 0       |
| 02/26/2020 | 0       |
| 03/27/2020 | 0       |
| 04/16/2020 | 0       |
| 05/21/2020 | 0       |
| 08/14/2020 | 0       |
| 10/07/2020 | 0       |
| 12/05/2020 | 0       |

| date       | in_year |
|------------|---------|
| 01/22/2021 | 0       |
| 02/03/2021 | 0       |
| 03/17/2021 | -1      |
| 04/23/2021 | -1      |
| 05/04/2021 | -1      |
| 06/30/2021 | -1      |
| 07/26/2021 | -1      |
| 12/27/2021 | -1      |
| 06/06/2022 | 0       |
| 07/18/2022 | 0       |
| 11/14/2022 | 0       |
| 12/12/2022 | 0       |

Utilizzando 3 come argomento `first_month_of_year` nella funzione `inyear()`, l'anno inizia il 1° marzo e termina alla fine di febbraio.

*Schema dell'intervallo della funzione `inyear()` con marzo impostato come primo mese dell'anno*



Pertanto, qualsiasi transazione che si verifichi tra il 1° marzo 2021 e il 1° marzo 2022 restituirà un risultato booleano di TRUE.

### Esempio 4 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, in questo esempio, il set di dati è invariato e viene caricato nell'applicazione. Il calcolo che determina se le transazioni sono avvenute nello stesso anno del 26 luglio 2021 viene creato come misura in un oggetto grafico dell'applicazione.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
Transactions:
Load
*
Inline
[
id,date,amount
8188,'01/13/2020',37.23
8189,'02/26/2020',17.17
8190,'03/27/2020',88.27
8191,'04/16/2020',57.42
8192,'05/21/2020',53.80
8193,'08/14/2020',82.06
8194,'10/07/2020',40.39
8195,'12/05/2020',87.21
8196,'01/22/2021',95.93
8197,'02/03/2021',45.89
8198,'03/17/2021',36.23
8199,'04/23/2021',25.66
8200,'05/04/2021',82.77
8201,'06/30/2021',69.98
8202,'07/26/2021',76.11
8203,'12/27/2021',25.12
8204,'06/06/2022',46.23
8205,'07/18/2022',84.21
8206,'11/14/2022',96.24
8207,'12/12/2022',67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- date

Per calcolare se le transazioni sono avvenute nello stesso anno del 26 luglio 2021, creare la seguente misura:

- =inyear(date,'07/26/2021', 0)

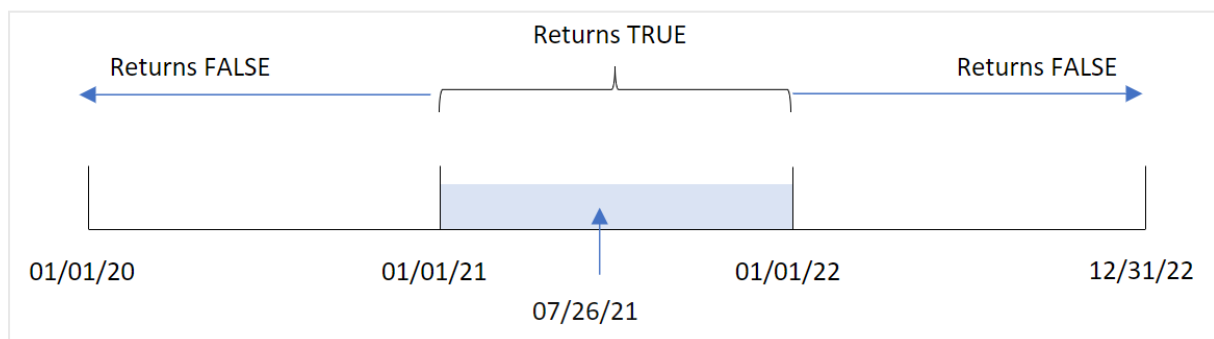
Tabella dei risultati

| date       | =inyear(date,'07/26/2021',0) |
|------------|------------------------------|
| 01/13/2020 | 0                            |
| 02/26/2020 | 0                            |

| date       | =inyear(date,'07/26/2021',0) |
|------------|------------------------------|
| 03/27/2020 | 0                            |
| 04/16/2020 | 0                            |
| 05/21/2020 | 0                            |
| 08/14/2020 | 0                            |
| 10/07/2020 | 0                            |
| 12/05/2020 | 0                            |
| 01/22/2021 | -1                           |
| 02/03/2021 | -1                           |
| 03/17/2021 | -1                           |
| 04/23/2021 | -1                           |
| 05/04/2021 | -1                           |
| 06/30/2021 | -1                           |
| 07/26/2021 | -1                           |
| 12/27/2021 | -1                           |
| 06/06/2022 | 0                            |
| 07/18/2022 | 0                            |
| 11/14/2022 | 0                            |
| 12/12/2022 | 0                            |

Il campo 'in\_year' viene creato nel grafico mediante l'utilizzo della funzione `inyear()`. Il primo argomento identifica il campo da valutare. Il secondo argomento è una data codificata per il 26 luglio 2021, ovvero il `base_date` che determina l'anno di confronto. Un `period_no` di 0 è l'argomento finale, il che significa che la funzione `inyear()` non confronta gli anni precedenti o successivi all'anno.

*Schema dell'intervallo della funzione `inyear()` con il 27 luglio come data base*



Qualsiasi transazione che si verifica nel 2021 restituisce un risultato booleano pari a TRUE.

### Esempio 5 - Scenario

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella chiamata 'Products'.
- La tabella contiene i seguenti campi:
  - ID prodotto
  - tipo di prodotto
  - data di produzione
  - prezzo di costo

L'utente finale vuole un oggetto grafico che mostri, per tipo di prodotto, il costo dei prodotti fabbricati nel 2021.

#### Script di caricamento

```
Products:
Load
*
Inline
[
product_id,product_type,manufacture_date,cost_price
8188,product A,'01/13/2020',37.23
8189,product B,'02/26/2020',17.17
8190,product B,'03/27/2020',88.27
8191,product C,'04/16/2020',57.42
8192,product D,'05/21/2020',53.80
8193,product D,'08/14/2020',82.06
8194,product C,'10/07/2020',40.39
8195,product B,'12/05/2020',87.21
8196,product A,'01/22/2021',95.93
8197,product B,'02/03/2021',45.89
8198,product C,'03/17/2021',36.23
8199,product C,'04/23/2021',25.66
8200,product B,'05/04/2021',82.77
8201,product D,'06/30/2021',69.98
8202,product D,'07/26/2021',76.11
8203,product D,'12/27/2021',25.12
8204,product C,'06/06/2022',46.23
8205,product C,'07/18/2022',84.21
8206,product A,'11/14/2022',96.24
8207,product B,'12/12/2022',67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- product\_type

Per calcolare la somma di ogni prodotto fabbricato nel 2021, creare la seguente misura:

- `=sum(if(InYear(manufacture_date,makedate(2021,01,01),0),cost_price,0))`

Procedere come indicato di seguito:

1. Impostare la misura **Formattazione numero** su **Denaro**.
2. In **Aspetto**, disattivare **Totals**.

Tabella dei risultati

| product_type | =sum(if(InYear(manufacture_date,makedate(2021,01,01),0),cost_price,0)) |
|--------------|------------------------------------------------------------------------|
| prodotto A   | \$95.93                                                                |
| prodotto B   | \$128.66                                                               |
| prodotto C   | \$61.89                                                                |
| prodotto D   | \$171.21                                                               |

La funzione `inyear()` restituisce un valore booleano quando valuta le date di produzione di ciascun prodotto. Per qualsiasi prodotto fabbricato nel 2021, la funzione `inyear()` restituisce il valore booleano TRUE e mostra la somma di `cost_price`.

### inyeartodate

Questa funzione restituisce True se **timestamp** ricade all'interno della parte dell'anno contenente **base\_date** fino a includere l'ultimo millisecondo di **base\_date**.

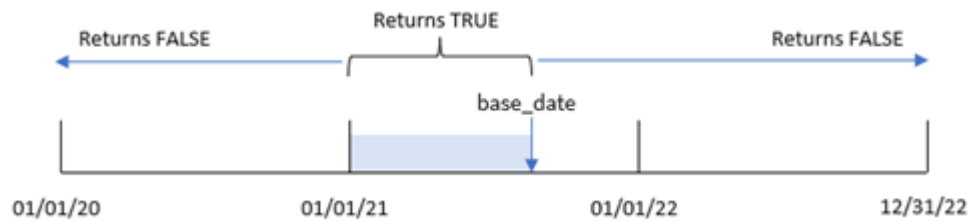
Sintassi:

```
InYearToDate (timestamp, base_date, period_no[, first_month_of_year])
```

Tipo di dati restituiti: Boolean



*In Qlik Sense, il valore booleano vero è rappresentato da -1 e il valore falso è rappresentato da 0.*

Diagramma della funzione *inyeartodate*

La funzione *inyeartodate()* segmenterà una particolare parte dell'anno con *base\_date*, identificando la data massima consentita per quel segmento di anno. La funzione valuta quindi se un campo data o un valore rientrano in tale segmento e restituisce un risultato booleano.

## Argomenti

| Argomento                  | Descrizione                                                                                                                                                                                                                                                                              |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>timestamp</b>           | La data da confrontare con <b>base_date</b> .                                                                                                                                                                                                                                            |
| <b>base_date</b>           | La data utilizzata per valutare l'anno.                                                                                                                                                                                                                                                  |
| <b>period_no</b>           | L'anno può essere differito mediante <b>period_no</b> . <b>period_no</b> è un numero intero, in cui il valore 0 indica l'anno che contiene <b>base_date</b> . I valori negativi di <b>period_no</b> indicano gli anni precedenti, mentre i valori positivi indicano gli anni successivi. |
| <b>first_month_of_year</b> | Se si intende utilizzare anni (fiscali) che non iniziano a gennaio, indicare un valore compreso tra 2 e 12 in <b>first_month_of_year</b> .                                                                                                                                               |

## Casi di utilizzo

La funzione *inyeartodate()* restituisce un risultato booleano. In genere, questo tipo di funzione viene utilizzato come condizione in un'espressione *if*. Questa restituisce un'aggregazione o un calcolo che dipende dal fatto che una data valutata si sia verificata in quell'anno fino a includere la data in questione.

Ad esempio, la funzione *inyeartodate()* può essere utilizzata per identificare tutte le apparecchiature prodotte in un determinato anno fino a una data specifica.

In questi esempi viene utilizzato il formato della data (MM/GG/AAAA). Il formato della data viene specificato nell'istruzione *SET DateFormat* sopra lo script di caricamento dei dati. Modificare il formato negli esempi in base alle proprie necessità.

## Esempi di funzioni

| Esempio                                                    | Risultato          |
|------------------------------------------------------------|--------------------|
| <i>inyeartodate</i><br>( '01/25/2013',<br>'02/01/2013', 0) | Restituisce TRUE.  |
| <i>inyeartodate</i><br>( '01/25/2012',<br>'01/01/2013', 0) | Restituisce FALSE. |

| Esempio                                                             | Risultato                                                                                                                                                                                                                    |
|---------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>inyeartodate</code><br>( '01/25/2012',<br>'02/01/2013', -1)   | Restituisce TRUE.                                                                                                                                                                                                            |
| <code>inyeartodate</code><br>( '11/25/2012',<br>'01/31/2013', 0, 4) | Restituisce TRUE.<br>Il valore di <code>timestamp</code> ricade nell'anno fiscale che inizia nel quarto mese e prima del valore di <code>base_date</code> .                                                                  |
| <code>inyeartodate</code><br>( '3/31/2013',<br>'01/31/2013', 0, 4 ) | Restituisce FALSE.<br>Rispetto all'esempio precedente, il valore di <code>timestamp</code> ricade ancora nell'anno fiscale, ma viene dopo il valore di <code>base_date</code> , pertanto ricade fuori dalla parte dell'anno. |

## Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

## Esempio 1 - Nessun argomento aggiuntivo

Script di caricamento e risultati

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni tra il 2020 e il 2022, caricato in una tabella denominata `Transactions`.
- Il campo della data fornito nel formato della variabile di sistema `DateFormat` (MM/GG/AAAA).
- La creazione di un campo, `in_year_to_date`, che determina quali transazioni sono avvenute durante l'anno fino al 26 luglio 2021.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        inyeartodate(date,'07/26/2021', 0) as in_year_to_date
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'01/13/2020',37.23
```

```
8189,'02/26/2020',17.17
```

```
8190,'03/27/2020',88.27
```

```
8191,'04/16/2020',57.42
```

```
8192,'05/21/2020',53.80
```

```
8193,'06/14/2020',82.06
```

```
8194,'08/07/2020',40.39
```

```
8195,'09/05/2020',87.21
```

```
8196,'01/22/2021',95.93
```

```
8197,'02/03/2021',45.89
```

```
8198,'03/17/2021',36.23
```

```
8199,'04/23/2021',25.66
```

```
8200,'05/04/2021',82.77
```

```
8201,'06/30/2021',69.98
```

```
8202,'07/26/2021',76.11
```

```
8203,'07/27/2021',25.12
```

```
8204,'06/06/2022',46.23
```

```
8205,'07/18/2022',84.21
```

```
8206,'11/14/2022',96.24
```

```
8207,'12/12/2022',67.67
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- in\_year\_to\_date

Tabella dei risultati

| date       | nell'anno_fino_ad_oggi |
|------------|------------------------|
| 01/13/2020 | 0                      |
| 02/26/2020 | 0                      |
| 03/27/2020 | 0                      |
| 04/16/2020 | 0                      |

| date       | nell'anno_fino_ad_oggi |
|------------|------------------------|
| 05/21/2020 | 0                      |
| 14/06/2020 | 0                      |
| 07/08/2020 | 0                      |
| 05/09/2020 | 0                      |
| 01/22/2021 | -1                     |
| 02/03/2021 | -1                     |
| 03/17/2021 | -1                     |
| 04/23/2021 | -1                     |
| 05/04/2021 | -1                     |
| 06/30/2021 | -1                     |
| 07/26/2021 | -1                     |
| 27/07/2021 | 0                      |
| 06/06/2022 | 0                      |
| 07/18/2022 | 0                      |
| 11/14/2022 | 0                      |
| 12/12/2022 | 0                      |

Il campo `in_year_to_date` viene creato nell'istruzione caricamento precedente utilizzando la funzione `inyeartodate()`. Il primo argomento fornito identifica il campo da valutare.

Il secondo argomento è una data codificata per il 26 luglio 2021, che è il valore `base_date` che identifica il limite finale del segmento di anno. Un valore `period_no` uguale a 0 è l'argomento finale, il che significa che la funzione non confronta gli anni precedenti o successivi all'anno segmentato.

*Diagramma della funzione `inyeartodate`, senza argomenti aggiuntivi*



Qualsiasi transazione avvenuta tra il 1 gennaio e il 26 luglio restituisce il risultato booleano `TRUE`. Le date delle transazioni anteriori al 2021 e successive al 26 luglio 2021 restituiscono il valore `FALSE`.

### Esempio 2 - period\_no

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.
- La creazione di un campo, `previous_year_to_date`, che determina quali transazioni sono avvenute un anno prima del segmento di anno che termina il 26 luglio 2021.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        inyeartodate(date,'07/26/2021', -1) as previous_year_to_date
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
id,date,amount
8188,'01/13/2020',37.23
8189,'02/26/2020',17.17
8190,'03/27/2020',88.27
8191,'04/16/2020',57.42
8192,'05/21/2020',53.80
8193,'06/14/2020',82.06
8194,'08/07/2020',40.39
8195,'09/05/2020',87.21
8196,'01/22/2021',95.93
8197,'02/03/2021',45.89
8198,'03/17/2021',36.23
8199,'04/23/2021',25.66
8200,'05/04/2021',82.77
8201,'06/30/2021',69.98
8202,'07/26/2021',76.11
8203,'07/27/2021',25.12
8204,'06/06/2022',46.23
8205,'07/18/2022',84.21
8206,'11/14/2022',96.24
8207,'12/12/2022',67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

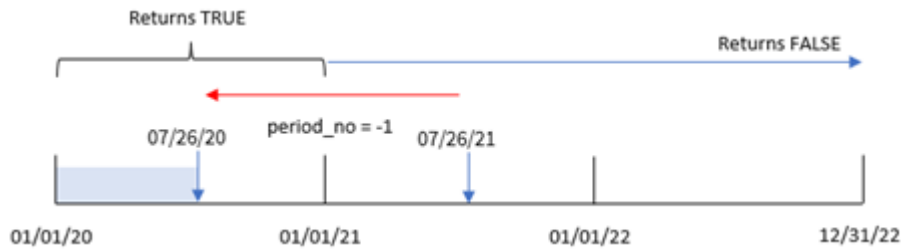
- date
- previous\_year\_to\_date

Tabella dei risultati

| date       | anno_precedente_fino_ad_oggi |
|------------|------------------------------|
| 01/13/2020 | -1                           |
| 02/26/2020 | -1                           |
| 03/27/2020 | -1                           |
| 04/16/2020 | -1                           |
| 05/21/2020 | -1                           |
| 14/06/2020 | -1                           |
| 07/08/2020 | 0                            |
| 05/09/2020 | 0                            |
| 01/22/2021 | 0                            |
| 02/03/2021 | 0                            |
| 03/17/2021 | 0                            |
| 04/23/2021 | 0                            |
| 05/04/2021 | 0                            |
| 06/30/2021 | 0                            |
| 07/26/2021 | 0                            |
| 27/07/2021 | 0                            |
| 06/06/2022 | 0                            |
| 07/18/2022 | 0                            |
| 11/14/2022 | 0                            |
| 12/12/2022 | 0                            |

Il valore period\_no di -1 indica che la funzione `inyeartodate` () confronta il segmento del trimestre di input con l'anno precedente. Con la data di input del 26 luglio 2021, il segmento dal 1 gennaio 2021 al 26 luglio 2021 è stato inizialmente identificato come year-to-date. period\_no quindi compensa questo segmento facendolo precedere di un intero anno, facendo sì che i limiti della data diventino dal 1 gennaio al 26 luglio 2020.

Diagramma della funzione `inyeartodate`, esempio di `period_no`



Pertanto, qualsiasi transazione che si verifichi tra il 1 gennaio e il 26 luglio 2020 restituirà il risultato booleano `TRUE`.

### Esempio 3 - `first_month_of_year`

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.
- La creazione di un campo, `in_year_to_date`, che determina quali transazioni sono avvenute nello stesso anno fino al 26 luglio 2021.

In questo esempio, marzo è stato impostato come primo mese dell'anno fiscale.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

Transactions:

```
Load
    *,
    inyeartodate(date,'07/26/2021', 0,3) as in_year_to_date
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'01/13/2020',37.23
```

```
8189,'02/26/2020',17.17
```

```
8190,'03/27/2020',88.27
```

```
8191,'04/16/2020',57.42
```

```
8192,'05/21/2020',53.80
```

```
8193,'06/14/2020',82.06
```

```
8194,'08/07/2020',40.39
```

```
8195,'09/05/2020',87.21
```

```
8196,'01/22/2021',95.93
```

```
8197, '02/03/2021', 45.89
8198, '03/17/2021', 36.23
8199, '04/23/2021', 25.66
8200, '05/04/2021', 82.77
8201, '06/30/2021', 69.98
8202, '07/26/2021', 76.11
8203, '07/27/2021', 25.12
8204, '06/06/2022', 46.23
8205, '07/18/2022', 84.21
8206, '11/14/2022', 96.24
8207, '12/12/2022', 67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- in\_year\_to\_date

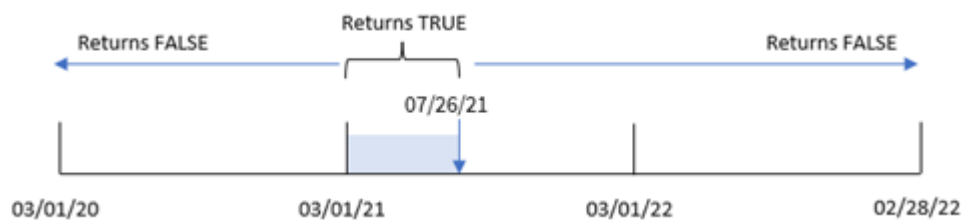
Tabella dei risultati

| date       | nell'anno_fino_ad_oggi |
|------------|------------------------|
| 01/13/2020 | 0                      |
| 02/26/2020 | 0                      |
| 03/27/2020 | 0                      |
| 04/16/2020 | 0                      |
| 05/21/2020 | 0                      |
| 14/06/2020 | 0                      |
| 07/08/2020 | 0                      |
| 05/09/2020 | 0                      |
| 01/22/2021 | 0                      |
| 02/03/2021 | 0                      |
| 03/17/2021 | -1                     |
| 04/23/2021 | -1                     |
| 05/04/2021 | -1                     |
| 06/30/2021 | -1                     |
| 07/26/2021 | -1                     |
| 27/07/2021 | 0                      |
| 06/06/2022 | 0                      |

| date       | nell'anno_fino_ad_oggi |
|------------|------------------------|
| 07/18/2022 | 0                      |
| 11/14/2022 | 0                      |
| 12/12/2022 | 0                      |

Utilizzando 3 come argomento `first_month_of_year` nella funzione `inyeartodate()`, la funzione fa iniziare l'anno il 1 marzo. Il valore `base_date` del 26 luglio 2021 imposta quindi la data di fine per quel segmento di anno.

Diagramma della funzione `inyeartodate`, esempio di `first_month_of_year`



Pertanto, qualsiasi transazione che si verifichi tra il 1 marzo e il 26 luglio 2001 restituirà un risultato booleano `TRUE`, mentre le transazioni con una data che non rientra in questi limiti restituiranno il valore `FALSE`.

### Esempio 4 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor di caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento contiene lo stesso set di dati e lo stesso scenario del primo esempio. Tuttavia, in questo esempio, il set di dati invariato viene caricato nell'applicazione. Il calcolo che determina quali transazioni sono avvenute nello stesso anno fino al 26 luglio 2021 viene creato come misura in un oggetto grafico dell'applicazione.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'01/13/2020',37.23
```

```
8189,'02/26/2020',17.17
```

```
8190, '03/27/2020', 88.27
8191, '04/16/2020', 57.42
8192, '05/21/2020', 53.80
8193, '06/14/2020', 82.06
8194, '08/07/2020', 40.39
8195, '09/05/2020', 87.21
8196, '01/22/2021', 95.93
8197, '02/03/2021', 45.89
8198, '03/17/2021', 36.23
8199, '04/23/2021', 25.66
8200, '05/04/2021', 82.77
8201, '06/30/2021', 69.98
8202, '07/26/2021', 76.11
8203, '07/27/2021', 25.12
8204, '06/06/2022', 46.23
8205, '07/18/2022', 84.21
8206, '11/14/2022', 96.24
8207, '12/12/2022', 67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:date.

Creare la seguente misura:

```
=inyeartodate(date, '07/26/2021', 0)
```

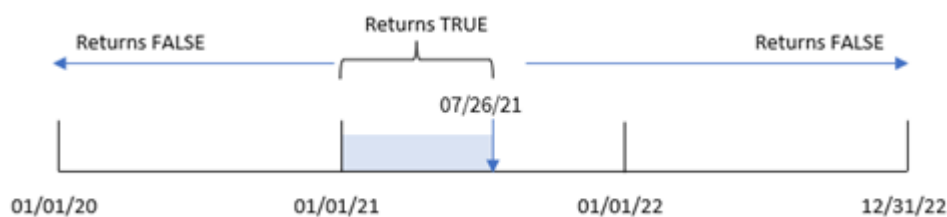
Tabella dei risultati

| date       | =inyeartodate(date,'26/07/2021', 0) |
|------------|-------------------------------------|
| 01/13/2020 | 0                                   |
| 02/26/2020 | 0                                   |
| 03/27/2020 | 0                                   |
| 04/16/2020 | 0                                   |
| 05/21/2020 | 0                                   |
| 14/06/2020 | 0                                   |
| 07/08/2020 | 0                                   |
| 05/09/2020 | 0                                   |
| 01/22/2021 | -1                                  |
| 02/03/2021 | -1                                  |
| 03/17/2021 | -1                                  |
| 04/23/2021 | -1                                  |

| date       | =inyearToDate(date,'26/07/2021', 0) |
|------------|-------------------------------------|
| 05/04/2021 | -1                                  |
| 06/30/2021 | -1                                  |
| 07/26/2021 | -1                                  |
| 27/07/2021 | 0                                   |
| 06/06/2022 | 0                                   |
| 07/18/2022 | 0                                   |
| 11/14/2022 | 0                                   |
| 12/12/2022 | 0                                   |

La misura `in_year_to_date` viene creata nell'oggetto grafico utilizzando la funzione `inyearToDate()`. Il primo argomento fornito identifica il campo da valutare. Il secondo argomento è una data codificata per il 26 luglio 2021, che è il valore `base_date` che identifica il limite finale del segmento di anno di confronto. Un valore `period_no` uguale a 0 è l'argomento finale, il che significa che la funzione non confronta gli anni precedenti o successivi all'anno segmentato.

*Diagramma della funzione `inyearToDate`, esempio di oggetto grafico*



Qualsiasi transazione avvenuta tra il 1 gennaio e il 26 luglio 2021 restituisce il risultato booleano `TRUE`. Le date delle transazioni anteriori al 2021 e successive al 26 luglio 2021 restituiscono il valore `FALSE`.

### Esempio 5 - Scenario

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella chiamata `Products`.
- Informazioni relative all'ID prodotto, al tipo di prodotto, alla data di produzione e al prezzo di costo.

L'utente finale vuole un oggetto grafico che mostri, per tipo di prodotto, il costo dei prodotti fabbricati nel 2021 fino al 26 luglio.

### Script di caricamento

```
Products:
Load
*
Inline
[
product_id,product_type,manufacture_date,cost_price
8188,product A,'01/13/2020',37.23
8189,product B,'02/26/2020',17.17
8190,product B,'03/27/2020',88.27
8191,product C,'04/16/2020',57.42
8192,product D,'05/21/2020',53.80
8193,product D,'08/14/2020',82.06
8194,product C,'10/07/2020',40.39
8195,product B,'12/05/2020',87.21
8196,product A,'01/22/2021',95.93
8197,product B,'02/03/2021',45.89
8198,product C,'03/17/2021',36.23
8199,product C,'04/23/2021',25.66
8200,product B,'05/04/2021',82.77
8201,product D,'06/30/2021',69.98
8202,product D,'07/26/2021',76.11
8203,product D,'12/27/2021',25.12
8204,product C,'06/06/2022',46.23
8205,product C,'07/18/2022',84.21
8206,product A,'11/14/2022',96.24
8207,product B,'12/12/2022',67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:product\_type.

Creare una misura che calcoli la somma di ogni prodotto che è stato prodotto nel 2021 prima del 27 luglio:

```
=sum(if(inyearthdate(manufacture_date,makedate(2021,07,26),0),cost_price,0))
```

Impostare la **Formattazione numero** della misura su **Denaro**.

Tabella dei risultati

| tipo_prodotto | =somma(se(da_inizio_anno_a_data(data_produzione,creadata(2021,07,26),0),prezzo_costo,0)) |
|---------------|------------------------------------------------------------------------------------------|
| prodotto A    | \$95.93                                                                                  |
| prodotto B    | \$128.66                                                                                 |
| prodotto C    | \$61.89                                                                                  |
| prodotto D    | \$146.09                                                                                 |

La funzione `inyeartodate()` restituisce un valore booleano quando valuta le date di produzione di ciascun prodotto. Per qualsiasi prodotto fabbricato nel 2021 prima del 27 luglio, la funzione `inyeartodate()` restituisce un valore booleano `TRUE` e somma il valore `cost_price`.

Il prodotto D è l'unico prodotto fabbricato anche dopo il 26 luglio 2021. La voce con `product_ID` 8203 è stata prodotta il 27 dicembre e costa \$25.12. Pertanto, questo costo non è stato incluso nel totale del Prodotto D nell'oggetto grafico.

### lastworkdate

La funzione **lastworkdate** restituisce la data di fine più prossima per ottenere **no\_of\_workdays** (dal lunedì al venerdì) se si inizia dalla data **start\_date** tenendo in considerazione tutte le festività **holiday** eventualmente in calendario. **start\_date** e **holiday** devono essere date o indicatori temporali validi.

Sintassi:

```
lastworkdate(start_date, no_of_workdays {, holiday})
```

Tipo di dati restituiti: numero intero

*Un calendario che mostra come viene utilizzata la funzione lastworkdate()*

| Sun | Mon | Tue | Wed              | Thu | Fri            | Sat |
|-----|-----|-----|------------------|-----|----------------|-----|
|     | 1   | 2   | 3                | 4   | 5              | 6   |
| 7   | 8   | 9   | 10<br>start_date | 11  | 12             | 13  |
| 14  | 15  | 16  | 17               | 18  | 19             | 20  |
| 21  | 22  | 23  | 24               | 25  | 26<br>end_date | 27  |
| 28  | 29  | 30  | 31               |     |                |     |

### Limitazioni

Non esiste un metodo per modificare la funzione `lastworkdate()` per le regioni o gli scenari che prevedono qualcosa di diverso da una settimana lavorativa che inizia il lunedì e termina il venerdì.

Il parametro vacanza deve essere una costante stringa. Non accetta un'espressione.

### Casi di utilizzo

La funzione `Lastworkdate()` viene comunemente utilizzata come parte di un'espressione quando l'utente desidera calcolare la data di fine proposta di un progetto o di un incarico, in base alla data di inizio del progetto e alle vacanze che si verificheranno in quel periodo.

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

#### Argomenti

| Argomento             | Descrizione                                                                                                                                                                                                                                                         |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>start_date</b>     | La data di inizio da valutare.                                                                                                                                                                                                                                      |
| <b>no_of_workdays</b> | Il numero di giorni lavorativi da raggiungere.                                                                                                                                                                                                                      |
| <b>holiday</b>        | Periodi di vacanza da escludere dai giorni lavorativi. Una vacanza è definita come una stringa di data costante. È possibile specificare più periodi di vacanza, separati da virgole.<br><br><b>Esempio:</b> '12/25/2013', '12/26/2013', '12/31/2013', '01/01/2014' |

### Esempio 1 - Esempio di base

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente gli ID dei progetti, le date di inizio e l'impegno stimato, in giorni, richiesto per i progetti. Il set di dati viene caricato in una tabella chiamata 'Projects'.

- Un caricamento precedente che contiene la funzione `Lastworkdate()` impostata come campo 'end\_date' e che identifica quando è prevista la fine di ogni progetto.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Projects:
```

```
    Load
        *,
        LastWorkDate(start_date,effort) as end_date
    ;
```

```
Load
```

```
id,
```

```
start_date,
```

```
effort
```

```
Inline
```

```
[
```

```
id,start_date,effort
```

```
1,01/01/2022,14
```

```
2,02/10/2022,17
```

```
3,05/17/2022,5
```

```
4,06/01/2022,12
```

```
5,08/10/2022,26
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- start\_date
- effort
- end\_date

Tabella dei risultati

| id | start_date | effort | end_date   |
|----|------------|--------|------------|
| 1  | 01/01/2022 | 14     | 01/20/2022 |
| 2  | 10/02/2022 | 17     | 03/04/2022 |
| 3  | 05/17/2022 | 5      | 05/23/2022 |
| 4  | 06/01/2022 | 12     | 06/16/2022 |
| 5  | 10/08/2022 | 26     | 09/14/2022 |

Poiché non ci sono giorni festivi programmati, la funzione aggiunge il numero definito di giorni lavorativi, dal lunedì al venerdì, alla data di inizio per trovare la prima data di fine possibile.

Il calendario seguente mostra la data di inizio e di fine del progetto 3, con i giorni lavorativi evidenziati in verde.

*Un calendario che mostra la data di inizio e di fine del progetto 3*

| Sun | Mon            | Tue                 | Wed | Thu | Fri | Sat |
|-----|----------------|---------------------|-----|-----|-----|-----|
| 1   | 2              | 3                   | 4   | 5   | 6   | 7   |
| 8   | 9              | 10                  | 11  | 12  | 13  | 14  |
| 15  | 16             | 17<br>Start<br>Date | 18  | 19  | 20  | 21  |
| 22  | 23<br>End Date | 24                  | 25  | 26  | 27  | 28  |
| 29  | 30             | 31                  |     |     |     |     |

### Esempio 2 - Singolo giorno festivo

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente gli ID dei progetti, le date di inizio e l'impegno stimato, in giorni, richiesto per i progetti. Il set di dati viene caricato in una tabella chiamata 'Projects'.
- Un caricamento precedente che contiene la funzione `Lastworkdate()` impostata come campo 'end\_date' e che identifica quando è prevista la fine di ogni progetto.

Tuttavia, è prevista una festività il 18 maggio 2022. La funzione `Lastworkdate()` nel caricamento precedente include il giorno festivo nel suo terzo argomento per identificare quando è prevista la fine di ogni progetto.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';

Projects:
    Load
        *,
        LastWorkDate(start_date,effort, '05/18/2022') as end_date
    ;
Load
id,
start_date,
effort
Inline
[
id,start_date,effort
1,01/01/2022,14
2,02/10/2022,17
3,05/17/2022,5
4,06/01/2022,12
5,08/10/2022,26
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- start\_date
- effort
- end\_date

Tabella dei risultati

| id | start_date | effort | end_date   |
|----|------------|--------|------------|
| 1  | 01/01/2022 | 14     | 01/20/2022 |
| 2  | 10/02/2022 | 17     | 03/04/2022 |
| 3  | 05/17/2022 | 5      | 05/24/2022 |
| 4  | 06/01/2022 | 12     | 06/16/2022 |
| 5  | 10/08/2022 | 26     | 09/14/2022 |

La singola vacanza programmata viene inserita come terzo argomento della funzione `Lastworkdate()`. Di conseguenza, la data di fine del progetto 3 viene spostata di un giorno perché la vacanza si svolge in uno dei giorni lavorativi precedenti la data di fine.

Il calendario seguente mostra la data di inizio e di fine del progetto 3 e mostra che la vacanza modifica la data di fine del progetto di un giorno.

*Un calendario che mostra la data di inizio e fine del progetto 3 con una vacanza il 18 maggio.*

| Sun | Mon | Tue                 | Wed           | Thu | Fri | Sat |
|-----|-----|---------------------|---------------|-----|-----|-----|
| 1   | 2   | 3                   | 4             | 5   | 6   | 7   |
| 8   | 9   | 10                  | 11            | 12  | 13  | 14  |
| 15  | 16  | 17<br>Start<br>Date | 18<br>Holiday | 19  | 20  | 21  |
| 22  | 23  | 24<br>End Date      | 25            | 26  | 27  | 28  |
| 29  | 30  | 31                  |               |     |     |     |

### Esempio 3 - Vacanze multiple

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente gli ID dei progetti, le date di inizio e l'impegno stimato, in giorni, richiesto per i progetti. Il set di dati viene caricato in una tabella chiamata 'Projects'.
- Un caricamento precedente che contiene la funzione `lastworkdate()` impostata come campo 'end\_date' e che identifica quando è prevista la fine di ogni progetto.

Tuttavia, sono previste quattro festività: il 19, 20, 21 e 22 maggio. La funzione `Lastworkdate()` nel caricamento precedente include ciascuno dei giorni festivi nel suo terzo argomento per identificare quando è prevista la fine di ogni progetto.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';

Projects:
    Load
        *,
        LastWorkDate(start_date,effort, '05/19/2022','05/20/2022','05/21/2022','05/22/2022') as
    end_date
    ;
Load
id,
start_date,
effort
Inline
[
id,start_date,effort
1,01/01/2022,14
2,02/10/2022,17
3,05/17/2022,5
4,06/01/2022,12
5,08/10/2022,26
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- start\_date
- effort
- end\_date

Tabella dei risultati

| id | start_date | effort | end_date   |
|----|------------|--------|------------|
| 1  | 01/01/2022 | 14     | 01/20/2022 |
| 2  | 10/02/2022 | 17     | 03/04/2022 |
| 3  | 05/17/2022 | 5      | 05/25/2022 |
| 4  | 06/01/2022 | 12     | 06/16/2022 |
| 5  | 10/08/2022 | 26     | 09/14/2022 |

I quattro giorni festivi vengono inseriti come elenco di argomenti nella funzione `Lastworkdate()` dopo la data di inizio e il numero di giorni lavorativi.

Il calendario seguente mostra la data di inizio e di fine del progetto 3 e mostra che le vacanze modificano la data di fine del progetto di tre giorni.

*Un calendario che mostra la data di inizio e fine del progetto 3 con le vacanze dal 19 al 22 maggio.*

| Sun           | Mon | Tue                 | Wed            | Thu           | Fri           | Sat           |
|---------------|-----|---------------------|----------------|---------------|---------------|---------------|
| 1             | 2   | 3                   | 4              | 5             | 6             | 7             |
| 8             | 9   | 10                  | 11             | 12            | 13            | 14            |
| 15            | 16  | 17<br>Start<br>Date | 18             | 19<br>Holiday | 20<br>Holiday | 21<br>Holiday |
| 22<br>Holiday | 23  | 24                  | 25<br>End Date | 26            | 27            | 28            |
| 29            | 30  | 31                  |                |               |               |               |

### Esempio 4 - Vacanza singola (grafico)

Script di caricamento ed espressione del grafico

#### Panoramica

Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, in questo esempio, il set di dati è invariato e viene caricato nell'app. Il campo end\_date viene calcolato come misura in un oggetto grafico.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

Projects:

Load

id,

start\_date,

```
effort
inline
[
id,start_date,effort
1,01/01/2022,14
2,02/10/2022,17
3,05/17/2022,5
4,06/01/2022,12
5,08/10/2022,26
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- start\_date
- effort

Per calcolare la end\_date, creare la seguente misura:

- =LastWorkDate(start\_date,effort,'05/18/2022')

Tabella dei risultati

| id | start_date | effort | =LastWorkDate(start_date,effort,'05/18/2022') |
|----|------------|--------|-----------------------------------------------|
| 1  | 01/01/2022 | 14     | 01/20/2022                                    |
| 2  | 10/02/2022 | 17     | 03/04/2022                                    |
| 3  | 05/17/2022 | 5      | 05/23/2022                                    |
| 4  | 06/01/2022 | 12     | 06/16/2022                                    |
| 5  | 10/08/2022 | 26     | 09/14/2022                                    |

La singola festività programmata viene inserita come misura nel grafico. Di conseguenza, la data di fine del progetto 3 viene spostata di un giorno perché la vacanza si svolge in uno dei giorni lavorativi precedenti la data di fine.

Il calendario seguente mostra la data di inizio e di fine del progetto 3 e mostra che la vacanza modifica la data di fine del progetto di un giorno.

Un calendario che mostra la data di inizio e fine del progetto 3 con una vacanza il 18 maggio.

| Sun | Mon | Tue                 | Wed           | Thu | Fri | Sat |
|-----|-----|---------------------|---------------|-----|-----|-----|
| 1   | 2   | 3                   | 4             | 5   | 6   | 7   |
| 8   | 9   | 10                  | 11            | 12  | 13  | 14  |
| 15  | 16  | 17<br>Start<br>Date | 18<br>Holiday | 19  | 20  | 21  |
| 22  | 23  | 24<br>End Date      | 25            | 26  | 27  | 28  |
| 29  | 30  | 31                  |               |     |     |     |

### localtime

Questa funzione restituisce un indicatore temporale dell'ora attuale fornita per il fuso orario specificato.

**Sintassi:**

```
LocalTime([timezone [, ignoreDST ]])
```

Tipo di dati restituiti: duale

#### Argomenti

| Argomento        | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>timezone</b>  | <p>Il valore <b>timezone</b> viene specificato come stringa contenente una qualsiasi delle località geografiche elencate in <b>Time Zone</b> nel <b>Windows Control Panel</b> per <b>Date and Time</b> o come stringa nel formato 'GMT+hh:mm'. Un elenco delle località accettate e dei fusi orari viene presentato anche nella tabella sottostante.</p> <p>Se non è specificato alcun fuso orario, viene restituita l'ora locale.</p> <div>  <p>Se si utilizza un differimento dalla DST (ovvero, si specifica un valore argomento <b>ignoreDST</b> che restituisce <b>False</b>), è necessario specificare una località, piuttosto che un differimento dall'ora GMT, nell'argomento <b>place</b>. Infatti, un adattamento in funzione dell'ora legale richiede informazioni latitudinali oltre a quelle longitudinali offerte da un differimento dall'ora GMT. Per ulteriori informazioni, vedere <a href="#">Utilizzo dei differimenti dall'ora GMT in combinazione con la DST (page 959)</a></p> </div> |
| <b>ignoreDST</b> | <p>Se l'argomento restituisce <b>True</b>, la DST (ora legale) viene ignorata. I valori argomento validi che restituiscono <b>True</b> includono <b>-1</b> e <b>True()</b>.</p> <p>Se questo argomento restituisce <b>False</b>, l'indicatore temporale viene adattato in funzione dell'ora legale. I valori argomento validi che restituiscono <b>False</b> includono <b>0</b> e <b>False()</b>.</p> <p>Se il valore argomento <b>ignoreDST</b> non è valido, la funzione valuta l'espressione come se il valore <b>ignore_dst</b> restituisse <b>True</b>. Se il valore argomento <b>ignoreDST</b> non è specificato, la funzione valuta l'espressione come se il valore <b>ignore_dst</b> restituisse <b>False</b>.</p>                                                                                                                                                                                                                                                                                                                                                                   |

#### Località e fusi orari validi

| A-C       | D-K                        | L-R       | S-Z          |
|-----------|----------------------------|-----------|--------------|
| Abu Dhabi | Darwin                     | La Paz    | Samoa        |
| Adelaide  | Dhaka                      | Lima      | Santiago     |
| Alaska    | Eastern Time (US & Canada) | Lisbon    | Sapporo      |
| Amsterdam | Edinburgh                  | Ljubljana | Sarajevo     |
| Arizona   | Ekaterinburg               | London    | Saskatchewan |
| Astana    | Fiji                       | Madrid    | Seoul        |
| Athens    | Georgetown                 | Magadan   | Singapore    |

## 8 Funzioni per script e grafici

| A-C                           | D-K                             | L-R                            | S-Z                    |
|-------------------------------|---------------------------------|--------------------------------|------------------------|
| Atlantic Time (Canada)        | Greenland                       | Mazatlan                       | Skopje                 |
| Auckland                      | Greenwich Mean Time :<br>Dublin | Melbourne                      | Sofia                  |
| Azores                        | Guadalajara                     | Mexico City                    | Solomon Is.            |
| Baghdad                       | Guam                            | Mid-Atlantic                   | Sri<br>Jayawardenepura |
| Baku                          | Hanoi                           | Minsk                          | St. Petersburg         |
| Bangkok                       | Harare                          | Monrovia                       | Stockholm              |
| Beijing                       | Hawaii                          | Monterrey                      | Sydney                 |
| Belgrade                      | Helsinki                        | Moscow                         | Taipei                 |
| Berlin                        | Hobart                          | Mountain Time (US &<br>Canada) | Tallinn                |
| Bern                          | Hong Kong                       | Mumbai                         | Tashkent               |
| Bogota                        | Indiana (East)                  | Muscat                         | Tbilisi                |
| Brasilia                      | International Date Line<br>West | Nairobi                        | Tehran                 |
| Bratislava                    | Irkutsk                         | New Caledonia                  | Tokyo                  |
| Brisbane                      | Islamabad                       | New Delhi                      | Urumqi                 |
| Brussels                      | Istanbul                        | Newfoundland                   | Warsaw                 |
| Bucharest                     | Jakarta                         | Novosibirsk                    | Wellington             |
| Budapest                      | Jerusalem                       | Nuku'alofa                     | West Central Africa    |
| Buenos Aires                  | Kabul                           | Osaka                          | Vienna                 |
| Cairo                         | Kamchatka                       | Pacific Time (US &<br>Canada)  | Vilnius                |
| Canberra                      | Karachi                         | Paris                          | Vladivostok            |
| Cape Verde Is.                | Kathmandu                       | Perth                          | Volgograd              |
| Caracas                       | Kolkata                         | Port Moresby                   | Yakutsk                |
| Casablanca                    | Krasnoyarsk                     | Prague                         | Yerevan                |
| Central America               | Kuala Lumpur                    | Pretoria                       | Zagreb                 |
| Central Time (US &<br>Canada) | Kuwait                          | Quito                          | -                      |
| Chennai                       | Kyiv                            | Riga                           | -                      |

| A-C        | D-K | L-R    | S-Z |
|------------|-----|--------|-----|
| Chihuahua  | -   | Riyadh | -   |
| Chongqing  | -   | Rome   | -   |
| Copenhagen | -   | -      | -   |

**Esempi e risultati:**

Gli esempi sottostanti si basano su una funzione chiamata all'ora locale 2023-08-14 08:39:47, con un fuso orario locale dell'ambiente desktop o server pari a GMT-05:00 e in una regione che ha implementato l'ora legale a partire dalla data elencata.

## Esempi di script

| Esempio                                   | Risultato                                                                                                                                                                                                                  |
|-------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>localtime ()</code>                 | Restituisce l'ora locale 2023-08-14 08:39:47.                                                                                                                                                                              |
| <code>localtime ('London')</code>         | Restituisce l'ora locale a Londra, 2023-08-14 13:39:47.                                                                                                                                                                    |
| <code>localtime ('GMT+02:00')</code>      | Restituisce l'ora locale nel fuso orario di GMT+02:00, 2023-08-14 14:39:47. Non viene effettuato nessun adattamento in funzione dell'ora legale perché è specificato il differimento dall'ora GMT, invece di una località. |
| <code>localtime ('Paris', -1)</code>      | Restituisce l'ora locale di Parigi e l'ora legale viene ignorata, 2023-08-14 13:39:47.                                                                                                                                     |
| <code>localtime ('Paris', True())</code>  | Restituisce l'ora locale di Parigi e l'ora legale viene ignorata, 2023-08-14 13:39:47.                                                                                                                                     |
| <code>localtime ('Paris', 0)</code>       | Restituisce l'ora locale di Parigi tenendo in considerazione l'ora legale, 2023-08-14 14:39:47.                                                                                                                            |
| <code>localtime ('Paris', False())</code> | Restituisce l'ora locale di Parigi tenendo in considerazione l'ora legale, 2023-08-14 14:39:47.                                                                                                                            |

**Utilizzo dei differimenti dall'ora GMT in combinazione con la DST**

In seguito all'implementazione delle librerie ICU (International Components for Unicode) in Qlik Sense, l'utilizzo di differimenti dall'ora GMT (ora media di Greenwich) in combinazione con la DST (ora legale) richiede informazioni latitudinali aggiuntive.

L'ora GMT è un differimento longitudinale (est-ovest), mentre la DST è un differimento latitudinale (nord-sud). Per esempio, Helsinki (Finlandia) e Johannesburg (Sudafrica) condividono lo stesso differimento GMT+02:00, ma non condividono lo stesso differimento dalla DST. Ciò significa che, ulteriormente al differimento dall'ora GMT, qualsiasi differimento dalla DST richiede informazioni sulla posizione latitudinale del fuso orario locale (input del fuso orario geografico) per avere informazioni complete sulle condizioni relative alla DST locale.

## lunarweekend

Questa funzione restituisce un valore corrispondente a un indicatore temporale recante l'ultimo millisecondo dell'ultimo giorno della settimana lunare contenente **date**. Le settimane lunari in Qlik Sense sono definite contando il 1° gennaio come primo giorno della settimana e, a parte l'ultima settimana dell'anno, conterranno esattamente sette giorni.

### Sintassi:

```
LunarweekEnd(date[, period_no[, first_week_day]])
```

Tipo di dati restituiti: duale

Schema esemplificativo della funzione `Lunarweekend()`



La funzione `Lunarweekend()` determina la settimana lunare in cui cade `date`. Quindi, restituisce un timestamp, nel formato data, per l'ultimo millisecondo di quella settimana.

### Argomenti

| Argomento             | Descrizione                                                                                                                                                                                                                                                                                                                     |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>date</b>           | La data o la data e ora da valutare.                                                                                                                                                                                                                                                                                            |
| <b>period_no</b>      | <b>period_no</b> è un numero intero o un'espressione la cui risoluzione è un numero intero, in cui il valore 0 indica la settimana lunare che contiene il valore <b>date</b> . I valori negativi di <b>period_no</b> indicano le settimane lunari precedenti, mentre i valori positivi indicano le settimane lunari successive. |
| <b>first_week_day</b> | Un differimento che può essere maggiore o minore di zero. Ciò modifica l'inizio dell'anno in base al numero specificato di giorni e/o frazioni di un giorno.                                                                                                                                                                    |

## Casi di utilizzo

La funzione `Lunarweekend()` viene comunemente utilizzata come parte di un'espressione quando l'utente desidera che il calcolo utilizzi la frazione di settimana non ancora trascorsa. A differenza della funzione `weekend()`, l'ultima settimana lunare di ogni anno solare terminerà il 31 dicembre. Ad esempio, la funzione `Lunarweekend()` può essere utilizzata per calcolare gli interessi non ancora maturati durante la settimana.

### Esempi di funzioni

| Esempio                                 | Risultato                        |
|-----------------------------------------|----------------------------------|
| <code>Lunarweekend('01/12/2013')</code> | Restituisce 01/14/2013 23:59:59. |

| Esempio                                       | Risultato                        |
|-----------------------------------------------|----------------------------------|
| <code>lunarweekend('01/12/2013', -1)</code>   | Restituisce 01/07/2013 23:59:59. |
| <code>lunarweekend('01/12/2013', 0, 1)</code> | Restituisce 01/15/2013 23:59:59. |

## Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

## Esempio 1 - Nessun argomento aggiuntivo

Script di caricamento e risultati

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2022, caricato in una tabella denominata 'Transactions'.
- Il campo della data fornito nel formato della variabile di sistema `DateFormat` (MM/GG/AAAA).
- La creazione di un campo, `end_of_week`, che restituisce un timestamp per la fine della settimana lunare in cui sono avvenute le transazioni.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
  Load
    *,
    lunarweekend(date) as end_of_week,
    timestamp(lunarweekend(date)) as end_of_week_timestamp
  ;
```

```
Load
*
```

```
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- end\_of\_week
- end\_of\_week\_timestamp

Tabella dei risultati

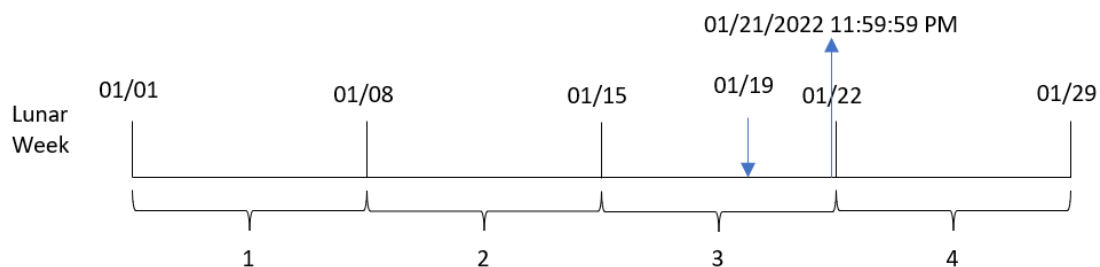
| date      | end_of_week | end_of_week_timestamp |
|-----------|-------------|-----------------------|
| 1/7/2022  | 01/07/2022  | 1/7/2022 11:59:59 PM  |
| 1/19/2022 | 01/21/2022  | 1/21/2022 11:59:59 PM |
| 2/5/2022  | 02/11/2022  | 2/11/2022 11:59:59 PM |
| 2/28/2022 | 03/04/2022  | 3/4/2022 11:59:59 PM  |
| 3/16/2022 | 03/18/2022  | 3/18/2022 11:59:59 PM |
| 4/1/2022  | 04/01/2022  | 4/1/2022 11:59:59 PM  |
| 5/7/2022  | 05/13/2022  | 5/13/2022 11:59:59 PM |
| 5/16/2022 | 05/20/2022  | 5/20/2022 11:59:59 PM |
| 6/15/2022 | 06/17/2022  | 6/17/2022 11:59:59 PM |

| date       | end_of_week | end_of_week_timestamp  |
|------------|-------------|------------------------|
| 6/26/2022  | 07/01/2022  | 7/1/2022 11:59:59 PM   |
| 7/9/2022   | 07/15/2022  | 7/15/2022 11:59:59 PM  |
| 7/22/2022  | 07/22/2022  | 7/22/2022 11:59:59 PM  |
| 7/23/2022  | 07/29/2022  | 7/29/2022 11:59:59 PM  |
| 7/27/2022  | 07/29/2022  | 7/29/2022 11:59:59 PM  |
| 8/2/2022   | 08/05/2022  | 8/5/2022 11:59:59 PM   |
| 8/8/2022   | 08/12/2022  | 8/12/2022 11:59:59 PM  |
| 8/19/2022  | 08/19/2022  | 8/19/2022 11:59:59 PM  |
| 9/26/2022  | 09/30/2022  | 9/30/2022 11:59:59 PM  |
| 10/14/2022 | 10/14/2022  | 10/14/2022 11:59:59 PM |
| 10/29/2022 | 11/04/2022  | 11/4/2022 11:59:59 PM  |

Il campo `end_of_week` viene creato nell'istruzione di caricamento precedente mediante l'uso della funzione `1unarweekend()` e trasferendo il campo `date` come argomento della funzione.

La funzione `1unarweekend()` identifica in quale settimana lunare cade il valore della data, restituendo un timestamp per l'ultimo millisecondo di quella settimana.

*Schema della funzione `1unarweekend()`, esempio senza argomenti aggiuntivi*



La transazione 8189 è avvenuta il 19 gennaio. La funzione `1unarweekend()` identifica che la settimana lunare inizia il 15 gennaio. Pertanto, il valore `end_of_week` per questa transazione restituisce l'ultimo millisecondo della settimana lunare, ovvero il 21 gennaio alle 23:59:59.

### Esempio 2 - period\_no

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.
- La creazione di un campo, `previous_lunar_week_end`, che restituisce il timestamp per la fine della settimana lunare prima che avvenga la transazione.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
```

```
        *,
```

```
        lunarweekend(date,-1) as previous_lunar_week_end,
```

```
        timestamp(lunarweekend(date,-1)) as previous_lunar_week_end_timestamp
```

```
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
```

```
8189,1/19/2022,37.23
```

```
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
```

```
8192,3/16/2022,53.80
```

```
8193,4/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/16/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
```

```
8200,7/23/2022,82.77
```

```
8201,7/27/2022,69.98
```

```
8202,8/2/2022,76.11
```

```
8203,8/8/2022,25.12
```

```
8204,8/19/2022,46.23
```

```
8205,9/26/2022,84.21
```

```
8206,10/14/2022,96.24
```

```
8207,10/29/2022,67.67
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

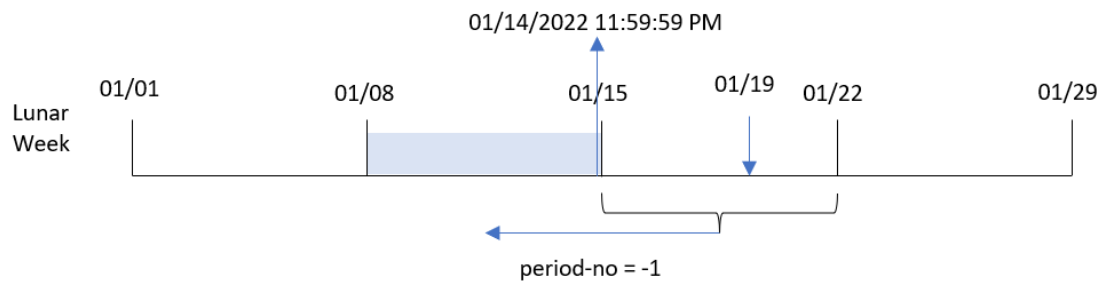
- date
- previous\_lunar\_week\_end
- previous\_lunar\_week\_end\_timestamp

Tabella dei risultati

| date       | previous_lunar_week_end | previous_lunar_week_end_timestamp |
|------------|-------------------------|-----------------------------------|
| 1/7/2022   | 12/31/2021              | 12/31/2021 11:59:59 PM            |
| 1/19/2022  | 01/14/2022              | 1/14/2022 11:59:59 PM             |
| 2/5/2022   | 02/04/2022              | 2/4/2022 11:59:59 PM              |
| 2/28/2022  | 02/25/2022              | 2/25/2022 11:59:59 PM             |
| 3/16/2022  | 03/11/2022              | 3/18/2022 11:59:59 PM             |
| 4/1/2022   | 03/25/2022              | 3/25/2022 11:59:59 PM             |
| 5/7/2022   | 05/06/2022              | 5/6/2022 11:59:59 PM              |
| 5/16/2022  | 05/13/2022              | 5/13/2022 11:59:59 PM             |
| 6/15/2022  | 06/10/2022              | 6/10/2022 11:59:59 PM             |
| 6/26/2022  | 06/24/2022              | 6/24/2022 11:59:59 PM             |
| 7/9/2022   | 07/08/2022              | 7/8/2022 11:59:59 PM              |
| 7/22/2022  | 07/15/2022              | 7/15/2022 11:59:59 PM             |
| 7/23/2022  | 07/22/2022              | 7/22/2022 11:59:59 PM             |
| 7/27/2022  | 07/22/2022              | 7/22/2022 11:59:59 PM             |
| 8/2/2022   | 07/29/2022              | 7/29/2022 11:59:59 PM             |
| 8/8/2022   | 08/05/2022              | 8/5/2022 11:59:59 PM              |
| 8/19/2022  | 08/12/2022              | 8/12/2022 11:59:59 PM             |
| 9/26/2022  | 09/23/2022              | 9/23/2022 11:59:59 PM             |
| 10/14/2022 | 10/07/2022              | 10/7/2022 11:59:59 PM             |
| 10/29/2022 | 10/28/2022              | 10/28/2022 11:59:59 PM            |

In questo caso, poiché è stato utilizzato un valore `period_no` di -1 come argomento di offset nella funzione `1unarweekend()`, la funzione identifica innanzitutto la settimana lunare in cui sono avvenute le transazioni. Si sposta poi una settimana prima e identifica il millisecondo finale di quella settimana lunare.

Schema della funzione `lunarweekend()`, esempio di `period_no`



La transazione 8189 è avvenuta il 19 gennaio. La funzione `lunarweekend()` identifica che la settimana lunare inizia il 15 gennaio. Pertanto, la settimana lunare precedente è iniziata l'8 gennaio ed è terminata il 14 gennaio alle 23:59:59; questo è il valore restituito per il campo `previous_lunar_week_end`.

### Esempio 3 - `first_week_day`

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor di caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento contiene lo stesso set di dati e lo stesso scenario del primo esempio. In questo esempio, abbiamo impostato che le settimane lunari inizino il 5 gennaio.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';

Transactions:
  Load
    *,
    lunarweekend(date,0,4) as end_of_week,
    timestamp(lunarweekend(date,0,4)) as end_of_week_timestamp
  ;
Load
  *
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- end\_of\_week
- end\_of\_week\_timestamp

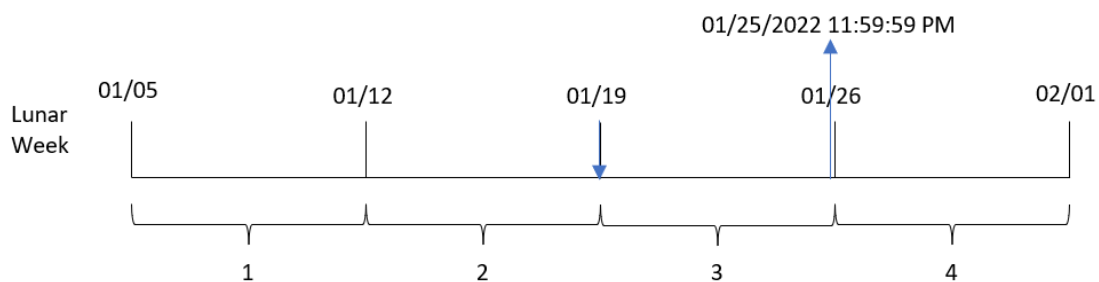
Tabella dei risultati

| date      | end_of_week | end_of_week_timestamp |
|-----------|-------------|-----------------------|
| 1/7/2022  | 01/11/2022  | 1/11/2022 11:59:59 PM |
| 1/19/2022 | 01/25/2022  | 1/25/2022 11:59:59 PM |
| 2/5/2022  | 02/08/2022  | 2/8/2022 11:59:59 PM  |
| 2/28/2022 | 03/01/2022  | 3/1/2022 11:59:59 PM  |
| 3/16/2022 | 03/22/2022  | 3/22/2022 11:59:59 PM |
| 4/1/2022  | 04/05/2022  | 4/5/2022 11:59:59 PM  |
| 5/7/2022  | 05/10/2022  | 5/10/2022 11:59:59 PM |
| 5/16/2022 | 05/17/2022  | 5/17/2022 11:59:59 PM |
| 6/15/2022 | 06/21/2022  | 6/21/2022 11:59:59 PM |
| 6/26/2022 | 06/28/2022  | 6/28/2022 11:59:59 PM |
| 7/9/2022  | 07/12/2022  | 7/12/2022 11:59:59 PM |
| 7/22/2022 | 07/26/2022  | 7/26/2022 11:59:59 PM |
| 7/23/2022 | 07/26/2022  | 7/26/2022 11:59:59 PM |
| 7/27/2022 | 08/02/2022  | 8/2/2022 11:59:59 PM  |
| 8/2/2022  | 08/02/2022  | 8/2/2022 11:59:59 PM  |
| 8/8/2022  | 08/09/2022  | 8/9/2022 11:59:59 PM  |
| 8/19/2022 | 08/23/2022  | 8/23/2022 11:59:59 PM |

| date       | end_of_week | end_of_week_timestamp  |
|------------|-------------|------------------------|
| 9/26/2022  | 09/27/2022  | 9/27/2022 11:59:59 PM  |
| 10/14/2022 | 10/18/2022  | 10/18/2022 11:59:59 PM |
| 10/29/2022 | 11/01/2022  | 11/1/2022 11:59:59 PM  |

In questo caso, poiché nella funzione `1unarweekend()` viene utilizzato l'argomento `first_week_date` di 4, l'inizio dell'anno viene spostato dal 1° al 5 gennaio.

*Schema della funzione `1unarweekend()`, esempio `first_week_day`*



La transazione 8189 è avvenuta il 19 gennaio. Poiché le settimane lunari iniziano il 5 gennaio, la funzione `1unarweekend()` identifica che la settimana lunare contenente il 19 gennaio inizia anch'essa il 19 gennaio. Pertanto, la fine di quella settimana lunare avviene il 25 gennaio alle 23:59:59; questo è il valore restituito per il campo `end_of_week`.

### Esempio 4 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento contiene lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, in questo esempio, il set di dati invariato viene caricato nell'applicazione. Il calcolo che restituisce un timestamp per la fine della settimana lunare in cui sono avvenute le transazioni viene creato come misura in un oggetto grafico dell'applicazione.

#### Script di caricamento

```
Transactions:
Load
*
Inline
[
id,date,amount
8188,1/7/2022,17.17
```

```
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: date.

Aggiungere le seguenti misure:

=lunarweekend(date)

=timestamp(lunarweekend(date))

Tabella dei risultati

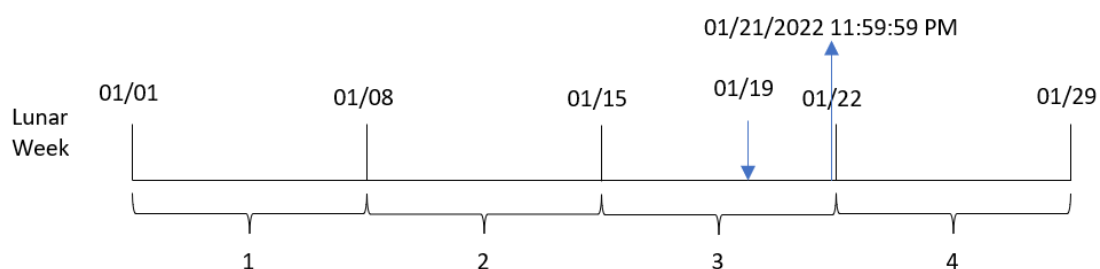
| date      | =lunarweekend(date) | =timestamp(lunarweekend(date)) |
|-----------|---------------------|--------------------------------|
| 1/7/2022  | 01/07/2022          | 1/7/2022 11:59:59 PM           |
| 1/19/2022 | 01/21/2022          | 1/21/2022 11:59:59 PM          |
| 2/5/2022  | 02/11/2022          | 2/11/2022 11:59:59 PM          |
| 2/28/2022 | 03/04/2022          | 3/4/2022 11:59:59 PM           |
| 3/16/2022 | 03/18/2022          | 3/18/2022 11:59:59 PM          |
| 4/1/2022  | 04/01/2022          | 4/1/2022 11:59:59 PM           |
| 5/7/2022  | 05/13/2022          | 5/13/2022 11:59:59 PM          |
| 5/16/2022 | 05/20/2022          | 5/20/2022 11:59:59 PM          |
| 6/15/2022 | 06/17/2022          | 6/17/2022 11:59:59 PM          |
| 6/26/2022 | 07/01/2022          | 7/1/2022 11:59:59 PM           |
| 7/9/2022  | 07/15/2022          | 7/15/2022 11:59:59 PM          |

| date       | =lunarweekend(date) | =timestamp(lunarweekend(date)) |
|------------|---------------------|--------------------------------|
| 7/22/2022  | 07/22/2022          | 7/22/2022 11:59:59 PM          |
| 7/23/2022  | 07/29/2022          | 7/29/2022 11:59:59 PM          |
| 7/27/2022  | 07/29/2022          | 7/29/2022 11:59:59 PM          |
| 8/2/2022   | 08/05/2022          | 8/5/2022 11:59:59 PM           |
| 8/8/2022   | 08/12/2022          | 8/12/2022 11:59:59 PM          |
| 8/19/2022  | 08/19/2022          | 8/19/2022 11:59:59 PM          |
| 9/26/2022  | 09/30/2022          | 9/30/2022 11:59:59 PM          |
| 10/14/2022 | 10/14/2022          | 10/14/2022 11:59:59 PM         |
| 10/29/2022 | 11/04/2022          | 11/4/2022 11:59:59 PM          |

La misura `end_of_week` viene creata nell'oggetto grafico mediante l'utilizzo della funzione `lunarweekend()` e trasferendo il campo `date` come argomento della funzione.

La funzione `lunarweekend()` identifica in quale settimana lunare cade il valore della data, restituendo un timestamp per l'ultimo millisecondo di quella settimana.

*Schema della funzione `lunarweekend()`, esempio di oggetto grafico*



La transazione 8189 è avvenuta il 19 gennaio. La funzione `lunarweekend()` identifica che la settimana lunare inizia il 15 gennaio. Pertanto, il valore `end_of_week` per questa transazione restituisce l'ultimo millisecondo della settimana lunare, ovvero il 21 gennaio alle 23:59:59.

### Esempio 5 - Scenario

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella chiamata `Employee_Expenses`.
- Gli ID dei dipendenti, il nome del dipendente e le spese medie giornaliere dichiarate da ciascun dipendente.

L'utente finale desidera un oggetto grafico che visualizzi, in base all'ID dipendente e al nome del dipendente, le richieste di rimborso spese stimate ancora da sostenere per il resto della settimana lunare.

### Script di caricamento

```
Employee_Expenses:
Load
*
Inline
[
employee_id,employee_name,avg_daily_claim
182,Mark, $15
183,Deryck, $12.5
184,Dexter, $12.5
185,Sydney,$27
186,Agatha,$18
];
```

### Risultati

#### Procedere come indicato di seguito:

1. Caricare i dati e aprire un foglio. Creare una nuova tabella.
2. Aggiungere i seguenti campi come dimensioni:
  - `employee_id`
  - `employee_name`
3. Quindi, creare la seguente misura per calcolare gli interessi accumulati:  
`=(lunarweekend(today(1))-today(1))*avg_daily_claim`
4. Impostare la **Formattazione numero** della misura su **Denaro**.

Tabella dei risultati

| employee_id | employee_name | =(lunarweekend(today(1))-today(1))*avg_daily_claim |
|-------------|---------------|----------------------------------------------------|
| 182         | Contrassegno  | \$75.00                                            |
| 183         | Deryck        | \$62.50                                            |
| 184         | Dexter        | \$62.50                                            |
| 185         | Sydney        | \$135.00                                           |
| 186         | Agatha        | \$90.00                                            |

La funzione `1unarweekend()`, utilizzando come unico argomento la data odierna, restituisce la data finale della settimana lunare corrente. Quindi, sottraendo la data odierna dalla data di fine settimana lunare, l'espressione restituisce il numero di giorni rimanenti di questa settimana.

Questo valore viene quindi moltiplicato per la media delle richieste di rimborso spese giornaliere di ciascun dipendente per calcolare il valore stimato delle richieste che ogni dipendente dovrebbe presentare durante il periodo rimanente della settimana lunare.

## 1unarweekname

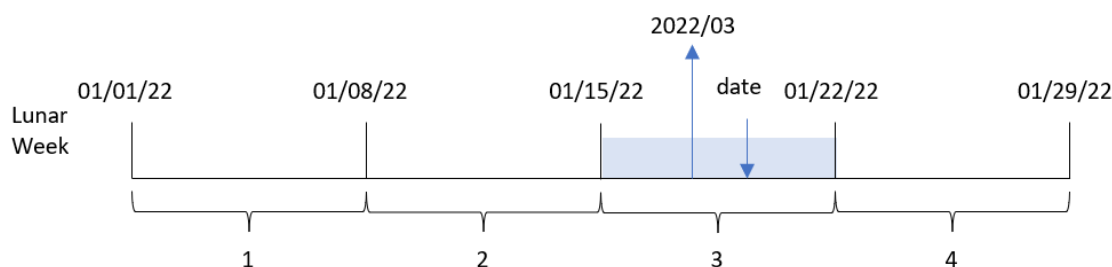
Questa funzione restituisce un valore di visualizzazione che mostra l'anno e il numero della settimana lunare corrispondente a un indicatore temporale del primo millisecondo del primo giorno della settimana lunare contenente **date**. Le settimane lunari in Qlik Sense sono definite contando il 1° gennaio come primo giorno della settimana e, a parte l'ultima settimana dell'anno, conterranno esattamente sette giorni.

### Sintassi:

```
1unarWeekName (date [, period_no[, first_week_day]])
```

**Tipo di dati restituiti:** duale

*Schema esemplificativo della funzione 1unarweekname()*



La funzione `1unarweekname()` determina in quale settimana lunare cade la data, iniziando il conteggio delle settimane dal 1° gennaio. Restituisce quindi un valore composto da year/weekcount.

### Argomenti

| Argomento             | Descrizione                                                                                                                                                                                                                                                                                                                     |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>date</b>           | La data o la data e ora da valutare.                                                                                                                                                                                                                                                                                            |
| <b>period_no</b>      | <b>period_no</b> è un numero intero o un'espressione la cui risoluzione è un numero intero, in cui il valore 0 indica la settimana lunare che contiene il valore <b>date</b> . I valori negativi di <b>period_no</b> indicano le settimane lunari precedenti, mentre i valori positivi indicano le settimane lunari successive. |
| <b>first_week_day</b> | Un differimento che può essere maggiore o minore di zero. Ciò modifica l'inizio dell'anno in base al numero specificato di giorni e/o frazioni di un giorno.                                                                                                                                                                    |

### Casi di utilizzo

La funzione `1unarweekname()` è utile quando si desidera confrontare le aggregazioni per settimane lunari. Ad esempio, la funzione potrebbe essere utilizzata per determinare le vendite totali dei prodotti per settimana lunare. Le settimane lunari sono utili quando si vuole garantire che tutti i valori contenuti nella prima settimana dell'anno contengano solo valori a partire dal 1° gennaio.

È possibile creare queste dimensioni nello script di caricamento utilizzando la funzione che consente di creare un campo in una tabella Calendario principale. La funzione può anche essere utilizzata direttamente in un grafico come dimensione calcolata.

Esempi di funzioni

| Esempio                                        | Risultato            |
|------------------------------------------------|----------------------|
| <code>1unarweekname('01/12/2013')</code>       | Restituisce 2006/02. |
| <code>1unarweekname('01/12/2013', -1)</code>   | Restituisce 2006/01. |
| <code>1unarweekname('01/12/2013', 0, 1)</code> | Restituisce 2006/02. |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - data senza alcun argomento aggiuntivo

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2022, caricato in una tabella denominata 'Transactions'.
- Il campo della data fornito nel formato della variabile di sistema `DateFormat` (MM/GG/AAAA).

- La creazione di un campo, `lunar_week_name`, che restituisce l'anno e il numero della settimana lunare in cui sono avvenute le transazioni.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        lunarweekname(date) as lunar_week_name
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `date`
- `lunar_week_name`

Tabella dei risultati

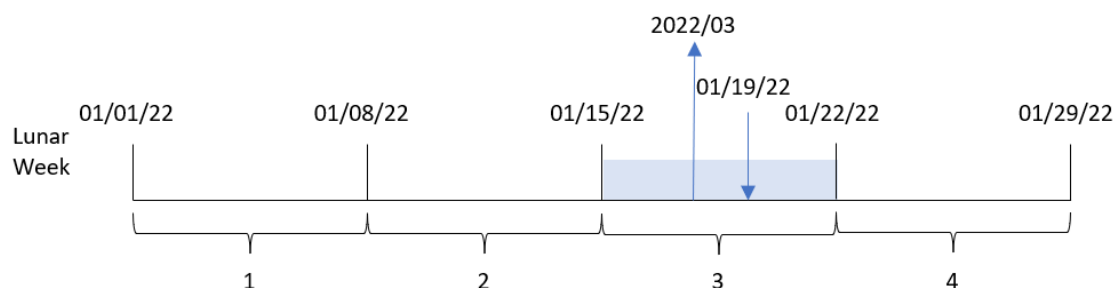
| date      | nome della settimana lunare |
|-----------|-----------------------------|
| 1/7/2022  | 2022/01                     |
| 1/19/2022 | 2022/03                     |

| date       | nome della settimana lunare |
|------------|-----------------------------|
| 2/5/2022   | 2022/06                     |
| 2/28/2022  | 2022/09                     |
| 3/16/2022  | 2022/11                     |
| 4/1/2022   | 2022/13                     |
| 5/7/2022   | 2022/19                     |
| 5/16/2022  | 2022/20                     |
| 6/15/2022  | 2022/24                     |
| 6/26/2022  | 2022/26                     |
| 7/9/2022   | 2022/28                     |
| 7/22/2022  | 2022/29                     |
| 7/23/2022  | 2022/30                     |
| 7/27/2022  | 2022/30                     |
| 8/2/2022   | 2022/31                     |
| 8/8/2022   | 2022/32                     |
| 8/19/2022  | 2022/33                     |
| 9/26/2022  | 2022/39                     |
| 10/14/2022 | 2022/41                     |
| 10/29/2022 | 2022/44                     |

Il campo `lunar_week_name` viene creato nell'istruzione di caricamento precedente mediante l'uso della funzione `lunarweekname()` e trasferendo il campo `date` come argomento della funzione.

La funzione `lunarweekname()` identifica la settimana lunare in cui cade il valore della data, restituendo l'anno e il numero della settimana di quella data.

*Schema della funzione `lunarweekname()`, esempio senza argomenti aggiuntivi*



La transazione 8189 è avvenuta il 19 gennaio. La funzione `lunarweekname()` identifica che questa data cade nella settimana lunare che inizia il 15 gennaio; si tratta della terza settimana lunare dell'anno. Pertanto, il valore `lunar_week_name` restituito per tale transazione è 2022/03.

### Esempio 2 - data con argomento `period_no`

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.
- La creazione di un campo, `previous_lunar_week_name`, che restituisce l'anno e il numero di settimana per la settimana lunare prima di quella in cui sono avvenute le transazioni.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
```

```
        *,
```

```
        lunarweekname(date,-1) as previous_lunar_week_name
```

```
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
```

```
8189,1/19/2022,37.23
```

```
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
```

```
8192,3/16/2022,53.80
```

```
8193,4/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/16/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
```

```
8200,7/23/2022,82.77
```

```
8201,7/27/2022,69.98
```

```
8202,8/2/2022,76.11
```

```
8203,8/8/2022,25.12
```

```
8204,8/19/2022,46.23
```

```
8205,9/26/2022,84.21
```

```
8206,10/14/2022,96.24
```

```
8207,10/29/2022,67.67
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

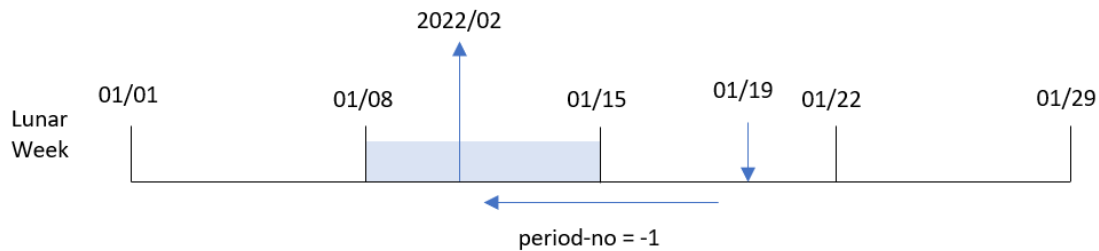
- date
- previous\_lunar\_week\_name

Tabella dei risultati

| date       | Nome settimana lunare precedente |
|------------|----------------------------------|
| 1/7/2022   | 2021/52                          |
| 1/19/2022  | 2022/02                          |
| 2/5/2022   | 2022/05                          |
| 2/28/2022  | 2022/08                          |
| 3/16/2022  | 2022/10                          |
| 4/1/2022   | 2022/12                          |
| 5/7/2022   | 2022/18                          |
| 5/16/2022  | 2022/19                          |
| 6/15/2022  | 2022/23                          |
| 6/26/2022  | 2022/25                          |
| 7/9/2022   | 2022/27                          |
| 7/22/2022  | 2022/28                          |
| 7/23/2022  | 2022/29                          |
| 7/27/2022  | 2022/29                          |
| 8/2/2022   | 2022/30                          |
| 8/8/2022   | 2022/31                          |
| 8/19/2022  | 2022/32                          |
| 9/26/2022  | 2022/38                          |
| 10/14/2022 | 2022/40                          |
| 10/29/2022 | 2022/43                          |

In questo caso, poiché è stato utilizzato un valore `period_no` di -1 come argomento di offset nella funzione `Tlunarweekname()`, la funzione identifica innanzitutto la settimana lunare in cui sono avvenute le transazioni. Quindi restituisce l'anno e il numero di una settimana prima.

Schema della funzione `lunarweekname()`, esempio di `period_no`



La transazione 8189 è avvenuta il 19 gennaio. La funzione `lunarweekname()` identifica che questa transazione è avvenuta nella terza settimana lunare dell'anno, quindi restituisce l'anno e il valore di una settimana prima, 2022/02, per il campo `previous_lunar_week_name`.

### Esempio 3 - data con argomento `first_week_day`

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor di caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento contiene lo stesso set di dati e lo stesso scenario del primo esempio. In questo esempio, abbiamo impostato che le settimane lunari inizino il 5 gennaio.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*,  
lunarweekname(date,0,4) as lunar_week_name  
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
```

```
8189,1/19/2022,37.23
```

```
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
```

```
8192,3/16/2022,53.80
```

```
8193,4/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/16/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Risultati

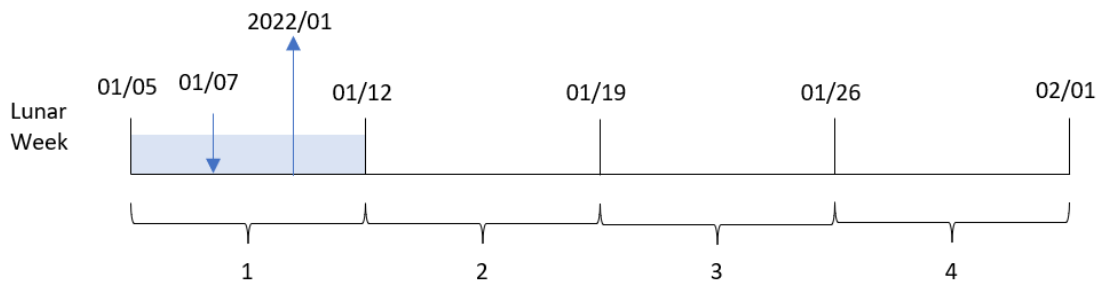
Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- Lunar\_week\_name

Tabella dei risultati

| date       | nome della settimana lunare |
|------------|-----------------------------|
| 1/7/2022   | 2022/01                     |
| 1/19/2022  | 2022/03                     |
| 2/5/2022   | 2022/05                     |
| 2/28/2022  | 2022/08                     |
| 3/16/2022  | 2022/11                     |
| 4/1/2022   | 2022/13                     |
| 5/7/2022   | 2022/18                     |
| 5/16/2022  | 2022/19                     |
| 6/15/2022  | 2022/24                     |
| 6/26/2022  | 2022/25                     |
| 7/9/2022   | 2022/27                     |
| 7/22/2022  | 2022/29                     |
| 7/23/2022  | 2022/29                     |
| 7/27/2022  | 2022/30                     |
| 8/2/2022   | 2022/30                     |
| 8/8/2022   | 2022/31                     |
| 8/19/2022  | 2022/33                     |
| 9/26/2022  | 2022/38                     |
| 10/14/2022 | 2022/41                     |
| 10/29/2022 | 2022/43                     |

Schema della funzione `lunarweekname()`, esempio `first_week_day`



In questo caso, poiché nella funzione `lunarweekname()` viene utilizzato l'argomento `first_week_date` di 4, si sposta l'inizio delle settimane lunari dal 1° gennaio al 5 gennaio.

La transazione 8188 è avvenuta il 7 gennaio. Poiché le settimane lunari iniziano il 5 gennaio, la funzione `lunarweekname()` identifica che la settimana lunare contenente il 7 gennaio è la prima settimana lunare dell'anno. Pertanto, il valore `lunar_week_name` restituito per quella transazione è 2022/01.

### Esempio 4 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento contiene lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, in questo esempio, il set di dati invariato viene caricato nell'applicazione. Il calcolo che restituisce il numero della settimana lunare e l'anno in cui sono avvenute le transazioni viene creato come misura in un oggetto grafico dell'applicazione.

#### Script di caricamento

```
Transactions:
Load
*
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: date.

Per calcolare la data di inizio della settimana lunare in cui avviene una transazione, creare la seguente misura:

```
=lunarweekname(date)
```

Tabella dei risultati

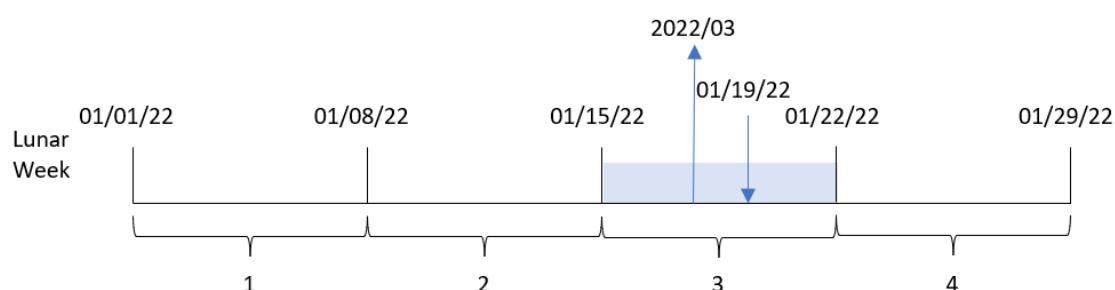
| date      | =lunarweekname(date) |
|-----------|----------------------|
| 1/7/2022  | 2022/01              |
| 1/19/2022 | 2022/03              |
| 2/5/2022  | 2022/06              |
| 2/28/2022 | 2022/09              |
| 3/16/2022 | 2022/11              |
| 4/1/2022  | 2022/13              |
| 5/7/2022  | 2022/19              |
| 5/16/2022 | 2022/20              |
| 6/15/2022 | 2022/24              |
| 6/26/2022 | 2022/26              |
| 7/9/2022  | 2022/28              |
| 7/22/2022 | 2022/29              |
| 7/23/2022 | 2022/30              |
| 7/27/2022 | 2022/30              |
| 8/2/2022  | 2022/31              |
| 8/8/2022  | 2022/32              |

| date       | =lunarweekname(date) |
|------------|----------------------|
| 8/19/2022  | 2022/33              |
| 9/26/2022  | 2022/39              |
| 10/14/2022 | 2022/41              |
| 10/29/2022 | 2022/44              |

La misura `lunar_week_name` viene creata nell'oggetto grafico mediante l'utilizzo della funzione `lunarweekname()` e trasferendo il campo `date` come argomento della funzione.

La funzione `lunarweekname()` identifica la settimana lunare in cui cade il valore della data, restituendo l'anno e il numero della settimana di quella data.

*Schema della funzione `lunarweekname()`, esempio di oggetto grafico*



La transazione 8189 è avvenuta il 19 gennaio. La funzione `lunarweekname()` identifica che questa data cade nella settimana lunare che inizia il 15 gennaio; si tratta della terza settimana lunare dell'anno. Pertanto, il valore `lunar_week_name` per tale transazione è 2022/03.

## Esempio 5 - Scenario

Script di caricamento ed espressione del grafico

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2022, caricato in una tabella denominata `Transactions`.
- Il campo della data fornito nel formato della variabile di sistema `DateFormat (MM/GG/AAAA)`.

L'utente finale desidera un oggetto grafico che presenti le vendite totali per settimana per l'anno in corso. La settimana 1, con una durata di sette giorni, dovrebbe iniziare il 1° gennaio. È possibile ottenere questo risultato anche quando questa dimensione non è disponibile nel modello di dati, utilizzando la funzione `lunarweekname()` come dimensione calcolata nel grafico.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
```

```
8189,1/19/2022,37.23
```

```
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
```

```
8192,3/16/2022,53.80
```

```
8193,4/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/16/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
```

```
8200,7/23/2022,82.77
```

```
8201,7/27/2022,69.98
```

```
8202,8/2/2022,76.11
```

```
8203,8/8/2022,25.12
```

```
8204,8/19/2022,46.23
```

```
8205,9/26/2022,84.21
```

```
8206,10/14/2022,96.24
```

```
8207,10/29/2022,67.67
```

```
];
```

### Risultati

Procedere come indicato di seguito:

1. Caricare i dati e aprire un foglio. Creare una nuova tabella.
2. Creare una dimensione calcolata utilizzando la seguente espressione:  
=lunarweekname(date)
3. Calcolare le vendite totali utilizzando la seguente misura di aggregazione:  
=sum(amount)
4. Impostare la **Formattazione numero** della misura su **Denaro**.

Tabella dei risultati

| =lunarweekname(date) | sum(amount) |
|----------------------|-------------|
| 2022/01              | \$17.17     |
| 2022/03              | \$37.23     |
| 2022/06              | \$57.42     |

| =lunarweekname(date) | sum(amount) |
|----------------------|-------------|
| 2022/09              | \$88.27     |
| 2022/11              | \$53.80     |
| 2022/13              | \$82.06     |
| 2022/19              | \$40.39     |
| 2022/20              | \$87.21     |
| 2022/24              | \$95.93     |
| 2022/26              | \$45.89     |
| 2022/28              | \$36.23     |
| 2022/29              | \$25.66     |
| 2022/30              | \$152.75    |
| 2022/31              | \$76.11     |
| 2022/32              | \$25.12     |
| 2022/33              | \$46.23     |
| 2022/39              | \$84.21     |
| 2022/41              | \$96.24     |
| 2022/44              | \$67.67     |

## lunarweekstart

Questa funzione restituisce un valore corrispondente a un indicatore temporale del primo millisecondo del primo giorno della settimana lunare contenente **date**. Le settimane lunari in Qlik Sense sono definite contando il 1° gennaio come primo giorno della settimana e, a parte l'ultima settimana dell'anno, conterranno esattamente sette giorni.

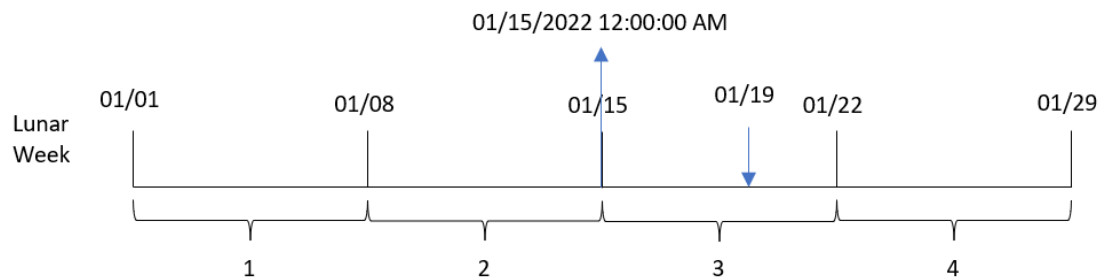
### Sintassi:

```
LunarweekStart(date[, period_no[, first_week_day]])
```

**Tipo di dati restituiti:** duale

La funzione `lunarweekstart()` determina la settimana lunare in cui cade `date`. Quindi, restituisce un timestamp, nel formato data, per il primo millisecondo di quella settimana.

Schema esemplificativo della funzione `lunarweekstart()`



#### Argomenti

| Argomento             | Descrizione                                                                                                                                                                                                                                                                                                                     |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>date</b>           | La data o la data e ora da valutare.                                                                                                                                                                                                                                                                                            |
| <b>period_no</b>      | <b>period_no</b> è un numero intero o un'espressione la cui risoluzione è un numero intero, in cui il valore 0 indica la settimana lunare che contiene il valore <b>date</b> . I valori negativi di <b>period_no</b> indicano le settimane lunari precedenti, mentre i valori positivi indicano le settimane lunari successive. |
| <b>first_week_day</b> | Un differimento che può essere maggiore o minore di zero. Ciò modifica l'inizio dell'anno in base al numero specificato di giorni e/o frazioni di un giorno.                                                                                                                                                                    |

### Casi di utilizzo

La funzione `lunarweekstart()` viene comunemente utilizzata come parte di un'espressione quando l'utente desidera che il calcolo utilizzi la frazione della settimana trascorsa finora. A differenza della funzione `weekstart()`, all'inizio di ogni nuovo anno solare, la settimana inizia il 1° gennaio e ogni settimana successiva inizia sette giorni dopo. La funzione `lunarweekstart()` non è interessata dalla variabile di sistema `FirstWeekDay`.

Ad esempio, `lunarweekstart()` è utilizzabile per calcolare gli interessi accumulati in una settimana fino ad oggi.

#### Esempi di funzioni

| Esempio                                         | Risultato                                                                                                                                       |
|-------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>lunarweekstart('01/12/2013')</code>       | Restituisce 01/08/2013.                                                                                                                         |
| <code>lunarweekstart('01/12/2013', -1)</code>   | Restituisce 01/01/2013.                                                                                                                         |
| <code>lunarweekstart('01/12/2013', 0, 1)</code> | Restituisce 01/09/2013, perché l'impostazione <code>first_week_day</code> su 1 significa che l'inizio dell'anno è stato cambiato in 01/02/2013. |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - Nessun argomento aggiuntivo

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2022, caricato in una tabella denominata 'Transactions'.
- Il campo della data fornito nel formato della variabile di sistema `DateFormat` (MM/GG/AAAA).
- La creazione di un campo, `start_of_week`, che restituisce un timestamp per l'inizio della settimana lunare in cui sono avvenute le transazioni.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        lunarweekstart(date) as start_of_week,
        timestamp(lunarweekstart(date)) as start_of_week_timestamp
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
```

```
8189,1/19/2022,37.23
```

```
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- start\_of\_week
- start\_of\_week\_timestamp

Tabella dei risultati

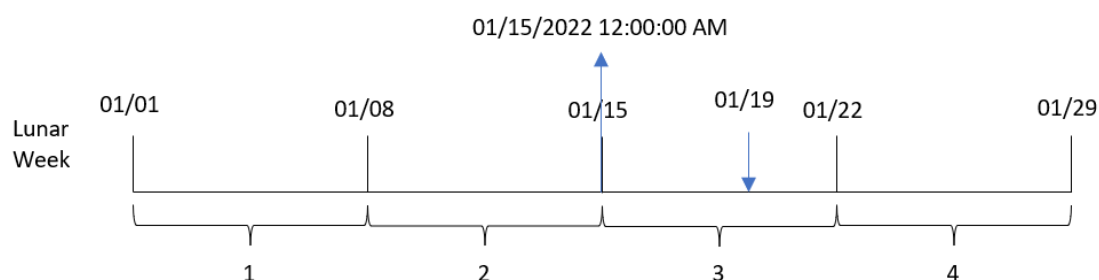
| date      | inizio_settimana | inizio_settimana_timestamp |
|-----------|------------------|----------------------------|
| 1/7/2022  | 01/01/2022       | 1/1/2022 00:00:00          |
| 1/19/2022 | 15/01/2022       | 15/01/2022 00:00:00        |
| 2/5/2022  | 05/02/2022       | 5/2/2022 00:00:00          |
| 2/28/2022 | 02/26/2022       | 26/2/2022 00:00:00         |
| 3/16/2022 | 03/12/2022       | 12/3/2022 00:00:00         |
| 4/1/2022  | 26/03/2022       | 26/03/2022 00:00:00        |
| 5/7/2022  | 07/05/2022       | 07/05/2022 00:00:00        |
| 5/16/2022 | 14/05/2022       | 14/05/2022 00:00:00        |
| 6/15/2022 | 11/06/2022       | 11/06/2022 00:00:00        |
| 6/26/2022 | 25/06/2022       | 25/06/2022 00:00:00        |
| 7/9/2022  | 09/07/2022       | 09/07/2022 00:00:00        |
| 7/22/2022 | 16/07/2022       | 16/07/2022 00:00:00        |
| 7/23/2022 | 23/07/2022       | 23/07/2022 00:00:00        |

| date       | inizio_settimana | inizio_settimana_timestamp |
|------------|------------------|----------------------------|
| 7/27/2022  | 23/07/2022       | 23/07/2022 00:00:00        |
| 8/2/2022   | 30/07/2022       | 30/07/2022 12:00:00 AM     |
| 8/8/2022   | 06/08/2022       | 6/8/2022 12:00:00 AM       |
| 8/19/2022  | 13/08/2022       | 13/8/2022 12:00:00 AM      |
| 9/26/2022  | 24/09/2022       | 24/09/2022 00:00:00        |
| 10/14/2022 | 08/10/2022       | 08/10/2022 00:00:00        |
| 10/29/2022 | 10/29/2022       | 29/10/2022 00:00:00        |

Il campo `start_of_week` viene creato nell'istruzione di caricamento precedente mediante l'uso della funzione `lunarweekstart()` e trasferendo il campo `date` come argomento della funzione.

La funzione `lunarweekstart()` identifica la settimana lunare in cui cade la data, restituendo un timestamp per il primo millisecondo di quella settimana.

*Schema della funzione `lunarweekstart()`, esempio senza argomenti aggiuntivi*



La transazione 8189 è avvenuta il 19 gennaio. La funzione `lunarweekstart()` identifica che la settimana lunare inizia il 15 gennaio. Pertanto, il valore `start_of_week` per questa transazione restituisce il primo millisecondo di quel giorno, ovvero il 15 gennaio alle 12:00:00 AM.

### Esempio 2 - `period_no`

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.
- La creazione di un campo, `previous_lunar_week_start`, che restituisce un timestamp per l'inizio della settimana lunare prima che avvenga la transazione.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        lunarweekstart(date,-1) as previous_lunar_week_start,
        timestamp(lunarweekstart(date,-1)) as previous_lunar_week_start_timestamp
    ;

Load
*
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Risultati

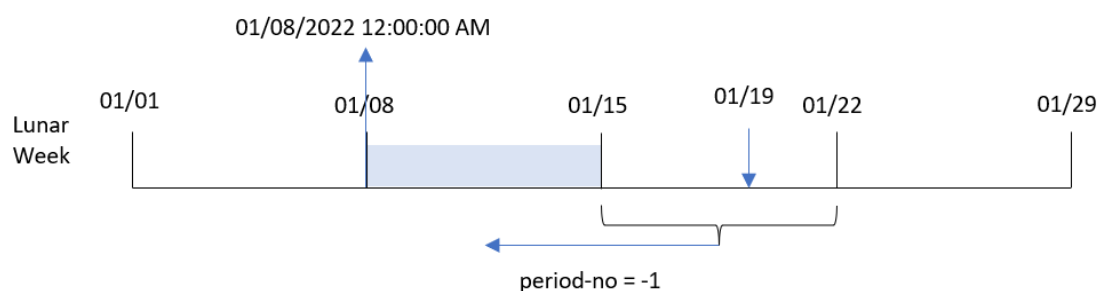
Tabella dei risultati

| date      | inizio_settimana_lunare_precedente | timestamp_inizio_settimana_lunare_precedente |
|-----------|------------------------------------|----------------------------------------------|
| 1/7/2022  | 12/24/2021                         | 24/12/2021 12:00:00 AM                       |
| 1/19/2022 | 01/08/2022                         | 8/1/2022 12:00:00 AM                         |
| 2/5/2022  | 01/29/2022                         | 29/1/2022 12:00:00 AM                        |
| 2/28/2022 | 02/19/2022                         | 19/2/2022 00:00:00                           |
| 3/16/2022 | 03/05/2022                         | 5/3/2022 00:00:00                            |

| date       | inizio_settimana_lunare_precedente | timestamp_inizio_settimana_lunare_precedente |
|------------|------------------------------------|----------------------------------------------|
| 4/1/2022   | 03/19/2022                         | 19/3/2022 00:00:00                           |
| 5/7/2022   | 04/30/2022                         | 30/04/2022 00:00:00                          |
| 5/16/2022  | 07/05/2022                         | 07/05/2022 00:00:00                          |
| 6/15/2022  | 06/04/2022                         | 4/6/2022 00:00:00                            |
| 6/26/2022  | 06/18/2022                         | 18/6/2022 00:00:00                           |
| 7/9/2022   | 07/02/2022                         | 2/7/2022 00:00:00                            |
| 7/22/2022  | 09/07/2022                         | 9/7/2022 12:00:00 AM                         |
| 7/23/2022  | 16/07/2022                         | 16/07/2022 00:00:00                          |
| 7/27/2022  | 16/07/2022                         | 16/07/2022 00:00:00                          |
| 8/2/2022   | 23/07/2022                         | 23/07/2022 12:00:00 AM                       |
| 8/8/2022   | 30/07/2022                         | 30/07/2022 00:00:00                          |
| 8/19/2022  | 06/08/2022                         | 06/08/2022 00:00:00                          |
| 9/26/2022  | 09/17/2022                         | 17/09/2022 00:00:00                          |
| 10/14/2022 | 01/10/2022                         | 10/1/2022 12:00:00 AM                        |
| 10/29/2022 | 10/22/2022                         | 22/10/2022 00:00:00                          |

In questo caso, poiché il valore `period_no` di -1 è stato utilizzato come argomento `offset` nella funzione `lunarweekstart()`, la funzione per prima cosa identifica la settimana lunare in cui avvengono le transazioni. Si sposta poi una settimana prima e identifica il primo millisecondo di quella settimana lunare.

*Schema della funzione `lunarweekstart()`, esempio di `period_no`*



La transazione 8189 è avvenuta il 19 gennaio. La funzione `lunarweekstart()` identifica che la settimana lunare inizia il 15 gennaio. Pertanto, la settimana lunare precedente è iniziata l'8 gennaio alle 12:00:00; questo è il valore restituito per il campo `previous_lunar_week_start`.

### Esempio 3 - first\_week\_day

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor di caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento contiene lo stesso set di dati e lo stesso scenario del primo esempio. In questo esempio, abbiamo impostato che le settimane lunari inizino il 5 gennaio.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*,
    lunarweekstart(date,0,4) as start_of_week,
    timestamp(lunarweekstart(date,0,4)) as start_of_week_timestamp
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
```

```
8189,1/19/2022,37.23
```

```
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
```

```
8192,3/16/2022,53.80
```

```
8193,4/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/16/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
```

```
8200,7/23/2022,82.77
```

```
8201,7/27/2022,69.98
```

```
8202,8/2/2022,76.11
```

```
8203,8/8/2022,25.12
```

```
8204,8/19/2022,46.23
```

```
8205,9/26/2022,84.21
```

```
8206,10/14/2022,96.24
```

```
8207,10/29/2022,67.67
```

```
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

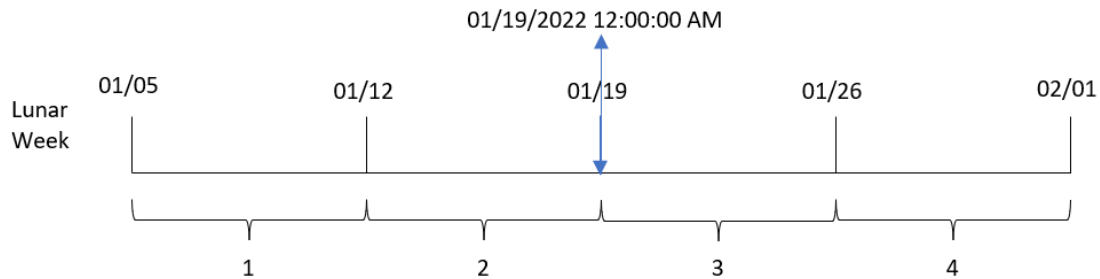
- date
- start\_of\_week
- start\_of\_week\_timestamp

Tabella dei risultati

| date       | inizio_settimana | inizio_settimana_timestamp |
|------------|------------------|----------------------------|
| 1/7/2022   | 01/05/2022       | 05/01/2022 00:00:00        |
| 1/19/2022  | 01/19/2022       | 19/01/2022 00:00:00        |
| 2/5/2022   | 02/02/2022       | 02/02/2022 00:00:00        |
| 2/28/2022  | 02/23/2022       | 23/02/2022 00:00:00        |
| 3/16/2022  | 03/16/2022       | 16/03/2022 00:00:00        |
| 4/1/2022   | 03/30/2022       | 30/03/2022 00:00:00        |
| 5/7/2022   | 05/04/2022       | 04/05/2022 00:00:00        |
| 5/16/2022  | 05/11/2022       | 11/5/2022 00:00:00         |
| 6/15/2022  | 06/15/2022       | 15/6/2022 00:00:00         |
| 6/26/2022  | 06/22/2022       | 22/6/2022 00:00:00         |
| 7/9/2022   | 07/06/2022       | 06/07/2022 00:00:00        |
| 7/22/2022  | 20/07/2022       | 20/07/2022 00:00:00        |
| 7/23/2022  | 20/07/2022       | 20/07/2022 00:00:00        |
| 7/27/2022  | 27/07/2022       | 27/07/2022 00:00:00        |
| 8/2/2022   | 27/07/2022       | 27/7/2022 00:00:00         |
| 8/8/2022   | 08/03/2022       | 3/8/2022 00:00:00          |
| 8/19/2022  | 08/17/2022       | 17/8/2022 00:00:00         |
| 9/26/2022  | 09/21/2022       | 21/09/2022 00:00:00        |
| 10/14/2022 | 10/12/2022       | 12/10/2022 00:00:00        |
| 10/29/2022 | 10/26/2022       | 26/10/2022 00:00:00        |

In questo caso, poiché nella funzione `1unarweekstart()` viene utilizzato l'argomento `first_week_date` di 4, l'inizio dell'anno viene spostato dal 1° al 5 gennaio.

Schema della funzione `Lunarweekstart()`, esempio `first_week_day`



La transazione 8189 è avvenuta il 19 gennaio. Poiché le settimane lunari iniziano il 5 gennaio, la funzione `Lunarweekstart()` identifica che la settimana lunare contenente il 19 gennaio inizia anch'essa il 19 gennaio alle 12:00:00 AM. Pertanto, ciò corrisponde al valore restituito per il campo `start_of_week`.

### Esempio 4 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento contiene lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, in questo esempio, il set di dati invariato viene caricato nell'applicazione. Il calcolo che restituisce un timestamp per l'inizio della settimana lunare in cui sono avvenute le transazioni viene creato come misura in un oggetto grafico dell'applicazione.

#### Script di caricamento

```
Transactions:
Load
*
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
```

```
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: date.

Aggiungere le seguenti misure:

```
=lunarweekstart(date)
```

```
=timestamp(lunarweekstart(date))
```

Tabella dei risultati

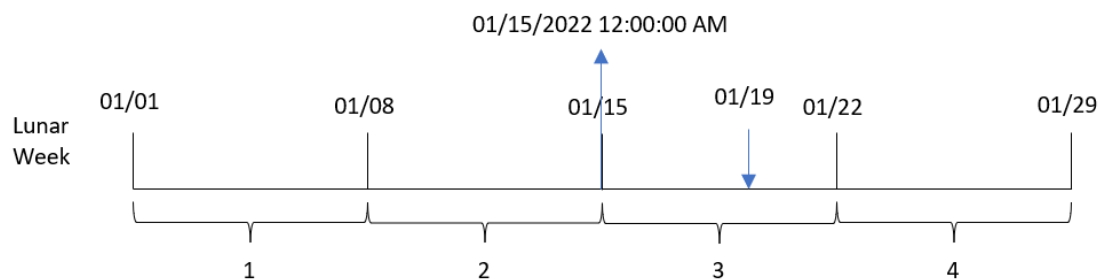
| date      | =lunarweekstart(data) | =timestamp(lunarweekstart(data)) |
|-----------|-----------------------|----------------------------------|
| 1/7/2022  | 01/01/2022            | 1/1/2022 00:00:00                |
| 1/19/2022 | 15/01/2022            | 15/01/2022 00:00:00              |
| 2/5/2022  | 05/02/2022            | 5/2/2022 00:00:00                |
| 2/28/2022 | 02/26/2022            | 26/2/2022 00:00:00               |
| 3/16/2022 | 03/12/2022            | 12/3/2022 00:00:00               |
| 4/1/2022  | 26/03/2022            | 26/03/2022 00:00:00              |
| 5/7/2022  | 07/05/2022            | 07/05/2022 00:00:00              |
| 5/16/2022 | 14/05/2022            | 14/05/2022 00:00:00              |
| 6/15/2022 | 11/06/2022            | 11/06/2022 00:00:00              |
| 6/26/2022 | 25/06/2022            | 25/06/2022 00:00:00              |
| 7/9/2022  | 09/07/2022            | 09/07/2022 00:00:00              |
| 7/22/2022 | 16/07/2022            | 16/07/2022 00:00:00              |
| 7/23/2022 | 23/07/2022            | 23/07/2022 00:00:00              |
| 7/27/2022 | 23/07/2022            | 23/07/2022 00:00:00              |
| 8/2/2022  | 30/07/2022            | 30/07/2022 12:00:00 AM           |
| 8/8/2022  | 06/08/2022            | 6/8/2022 12:00:00 AM             |
| 8/19/2022 | 13/08/2022            | 13/8/2022 12:00:00 AM            |
| 9/26/2022 | 24/09/2022            | 24/09/2022 00:00:00              |

| date       | =lunarweekstart(data) | =timestamp(lunarweekstart(data)) |
|------------|-----------------------|----------------------------------|
| 10/14/2022 | 08/10/2022            | 08/10/2022 00:00:00              |
| 10/29/2022 | 10/29/2022            | 29/10/2022 00:00:00              |

La misura `start_of_week` viene creata nell'oggetto grafico mediante l'utilizzo della funzione `lunarweekstart()` e trasferendo il campo `data` come argomento della funzione.

La funzione `lunarweekstart()` identifica in quale settimana lunare cade il valore della data, restituendo un timestamp per l'ultimo millisecondo di quella settimana.

*Schema della funzione `lunarweekstart()`, esempio di oggetto grafico*



La transazione 8189 è avvenuta il 19 gennaio. La funzione `lunarweekstart()` identifica che la settimana lunare inizia il 15 gennaio. Pertanto, il valore `start_of_week` per questa transazione corrisponde al primo millisecondo di quel giorno, ovvero il 15 gennaio alle 12:00:00 AM.

### Esempio 5 - Scenario

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di saldi di prestiti, che viene caricato in una tabella chiamata `Loans`.
- Dati costituiti dagli ID dei prestiti, dal saldo all'inizio della settimana e dal tasso di interesse semplice annuo applicato a ciascun prestito.

L'utente finale desidera un oggetto grafico che mostri, in base all'ID del prestito, gli interessi correnti che sono stati maturati per ciascun prestito nella settimana in corso.

#### Script di caricamento

```
Loans:
Load
*
```

```

inline
[
loan_id,start_balance,rate
8188,$10000.00,0.024
8189,$15000.00,0.057
8190,$17500.00,0.024
8191,$21000.00,0.034
8192,$90000.00,0.084
];

```

## Risultati

Procedere come indicato di seguito:

1. Caricare i dati e aprire un foglio. Creare una nuova tabella.
2. Aggiungere i seguenti campi come dimensioni:
  - loan\_id
  - start\_balance
3. Quindi, creare la seguente misura per calcolare gli interessi accumulati:  

$$=start\_balance*(rate*(today(1)-lunarweekstart(today(1)))/365)$$
4. Impostare la **Formattazione numero** della misura su **Denaro**.

Tabella dei risultati

| loan_id | saldo_iniziale | =saldo_iniziale*(tasso*(today(1)- lunarweekstart(today(1)))/365) |
|---------|----------------|------------------------------------------------------------------|
| 8188    | \$10000.00     | \$15.07                                                          |
| 8189    | \$15000.00     | \$128.84                                                         |
| 8190    | \$17500.00     | \$63.29                                                          |
| 8191    | \$21000.00     | \$107.59                                                         |
| 8192    | \$90000.00     | \$1139.18                                                        |

Utilizzando la data odierna come unico argomento, la funzione `lunarweekstart()` restituisce la data di inizio dell'anno corrente. Sottraendo tale risultato dalla data corrente, l'espressione restituisce il numero di giorni trascorsi fino ad ora questa settimana.

Questo valore viene quindi moltiplicato per il tasso di interesse e diviso per 365, per restituire il tasso di interesse effettivo incorso per questo periodo. Il risultato viene poi moltiplicato per il saldo iniziale del prestito per ottenere gli interessi maturati fino a questo momento della settimana.

## makedate

Questa funzione restituisce una data calcolata dall'anno **YYYY**, dal mese **MM** e dal giorno **DD**.

Sintassi:

```
MakeDate (YYYY [ , MM [ , DD ] ] )
```

**Tipo di dati restituiti:** duale

#### Argomenti

| Argomento | Descrizione                                                                                              |
|-----------|----------------------------------------------------------------------------------------------------------|
| YYYY      | L'anno è un numero intero.                                                                               |
| MM        | Il mese è un numero intero. Se non si indica il mese, verrà utilizzato come valore 1 (gennaio).          |
| DD        | Il giorno è un numero intero. Se non si indica un giorno, verrà utilizzato come valore 1 (primo giorno). |

### Casi di utilizzo

La funzione `makedate()` viene comunemente utilizzata nello script per la generazione dei dati per generare un calendario. Ciò potrebbe essere usato anche quando il campo data non è direttamente disponibile come data, ma necessita di alcune trasformazioni per estrarre i componenti di anno, mese e giorno.

In questi esempi viene utilizzato il formato della data (MM/GG/AAAA). Il formato della data viene specificato nell'istruzione `SET DateFormat` nella parte superiore dello script di caricamento dei dati. Modificare il formato negli esempi in base alle proprie necessità.

#### Esempi di funzioni

| Esempio                            | Risultato               |
|------------------------------------|-------------------------|
| <code>makedate(2012)</code>        | Restituisce 01/01/2012. |
| <code>makedate(12)</code>          | Restituisce 01/01/2012. |
| <code>makedate(2012, 12)</code>    | Restituisce 12/01/2012. |
| <code>makedate(2012, 2, 14)</code> | Restituisce 02/14/2012. |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - Esempio di base

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2018, caricato in una tabella denominata `Transactions`.
- Il campo della data fornito nel formato della variabile di sistema `DateFormat` (MM/GG/AAAA).
- La creazione di un campo, `transaction_date`, che restituisce una data nel formato MM/GG/AAAA.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
  Load
    *,
    makedate(transaction_year, transaction_month, transaction_day) as transaction_date
  ;
Load * Inline [
transaction_id, transaction_year, transaction_month, transaction_day, transaction_amount,
transaction_quantity, customer_id
3750, 2018, 08, 30, 12423.56, 23, 2038593
3751, 2018, 09, 07, 5356.31, 6, 203521
3752, 2018, 09, 16, 15.75, 1, 5646471
3753, 2018, 09, 22, 1251, 7, 3036491
3754, 2018, 09, 22, 21484.21, 1356, 049681
3756, 2018, 09, 22, -59.18, 2, 2038593
3757, 2018, 09, 23, 3177.4, 21, 203521
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `transaction_year`
- `transaction_month`
- `transaction_day`
- `transaction_date`

Tabella dei risultati

| <code>transaction_year</code> | <code>transaction_month</code> | <code>transaction_day</code> | <code>transaction_date</code> |
|-------------------------------|--------------------------------|------------------------------|-------------------------------|
| 2018                          | 08                             | 30                           | 08/30/2018                    |

| transaction_year | transaction_month | transaction_day | transaction_date |
|------------------|-------------------|-----------------|------------------|
| 2018             | 09                | 07              | 09/07/2018       |
| 2018             | 09                | 16              | 09/16/2018       |
| 2018             | 09                | 22              | 09/22/2018       |
| 2018             | 09                | 23              | 09/23/2018       |

Il campo `transaction_date` viene creato nell'istruzione di caricamento precedente utilizzando la funzione `makedate()` e passando i campi `anno`, `mese` e `giorno` come argomenti della funzione.

La funzione combina e converte questi valori in un campo `data`, restituendo i risultati nel formato della variabile di sistema `DateFormat`.

### Esempio 2 - DateFormat modificato

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.
- La creazione di un campo, `transaction_date`, nel formato `GG/MM/AAAA` senza modificare la variabile di sistema `DateFormat`.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        date(makedate(transaction_year, transaction_month, transaction_day), 'DD/MM/YYYY') as
transaction_date
    ;
Load * Inline [
transaction_id, transaction_year, transaction_month, transaction_day, transaction_amount,
transaction_quantity, customer_id
3750, 2018, 08, 30, 12423.56, 23, 2038593
3751, 2018, 09, 07, 5356.31, 6, 203521
3752, 2018, 09, 16, 15.75, 1, 5646471
3753, 2018, 09, 22, 1251, 7, 3036491
3754, 2018, 09, 22, 21484.21, 1356, 049681
3756, 2018, 09, 22, -59.18, 2, 2038593
3757, 2018, 09, 23, 3177.4, 21, 203521
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- transaction\_year
- transaction\_month
- transaction\_day
- transaction\_date

Tabella dei risultati

| transaction_year | transaction_month | transaction_day | transaction_date |
|------------------|-------------------|-----------------|------------------|
| 2018             | 08                | 30              | 30/08/2018       |
| 2018             | 09                | 07              | 07/09/2018       |
| 2018             | 09                | 16              | 16/09/2018       |
| 2018             | 09                | 22              | 22/09/2018       |
| 2018             | 09                | 23              | 23/09/2018       |

In questo caso, la funzione `makedate()` è nidificata all'interno della funzione `date()`. Il secondo argomento della funzione `date()` imposta il formato dei risultati della funzione `makedate()` come GG/MM/AAAA.

### Esempio 3 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2018, caricato in una tabella denominata `Transactions`.
- Le date delle transazioni sono fornite in due campi: `year` e `month`.

Creare una misura dell'oggetto grafico, `transaction_date`, che restituisca una data nel formato MM/GG/AAAA.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load * Inline [
```

```
transaction_id, transaction_year, transaction_month, transaction_amount, transaction_quantity,  
customer_id
```

```
3750, 2018, 08, 12423.56, 23, 2038593
```

```
3751, 2018, 09, 5356.31, 6, 203521
3752, 2018, 09, 15.75, 1, 5646471
3753, 2018, 09, 1251, 7, 3036491
3754, 2018, 09, 21484.21, 1356, 049681
3756, 2018, 09, -59.18, 2, 2038593
3757, 2018, 09, 3177.4, 21, 203521
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- year
- month

Per determinare il valore transaction\_date, creare questa misura:

```
=makedate(transaction_year,transaction_month)
```

Tabella dei risultati

| transaction_year | transaction_month | transaction_date |
|------------------|-------------------|------------------|
| 2018             | 08                | 08/01/2018       |
| 2018             | 09                | 09/01/2018       |

La misura transaction\_date viene creata nell'oggetto grafico utilizzando la funzione makedate() e passando i campi anno e mese come argomenti della funzione.

La funzione combina quindi questi valori e il valore presunto del giorno, pari a 01. Questi valori vengono poi convertiti in un campo data, restituendo i risultati nel formato della variabile di sistema DateFormat.

### Esempio 4 - Scenario

Script di caricamento ed espressione del grafico

#### Panoramica

Creare un set di dati di calendario per l'anno solare 2022.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Calendar:
```

```
load
```

```
*
```

```
where year(date)=2022;
```

```
load
```

```
date(recno()+makedate(2021,12,31)) as date
```

```
AutoGenerate 400;
```

### Risultati

Tabella dei risultati

| date                  |
|-----------------------|
| 01/01/2022            |
| 01/02/2022            |
| 01/03/2022            |
| 01/04/2022            |
| 01/05/2022            |
| 01/06/2022            |
| 01/07/2022            |
| 01/08/2022            |
| 01/09/2022            |
| 01/10/2022            |
| 01/11/2022            |
| 01/12/2022            |
| 01/13/2022            |
| 01/14/2022            |
| 01/15/2022            |
| 01/16/2022            |
| 01/17/2022            |
| 01/18/2022            |
| 01/19/2022            |
| 01/20/2022            |
| 01/21/2022            |
| 01/22/2022            |
| 01/23/2022            |
| 01/24/2022            |
| 01/25/2022            |
| + 340 ulteriori righe |

La funzione `makedate()` crea un valore di data per il 31 dicembre 2021. La funzione `recno()` fornisce il numero di record del record correntemente caricato nella tabella, a partire da 1. Pertanto, il primo record ha la data del 1° gennaio 2022. A ogni valore `recno()` successivo, la data viene incrementata di 1. Questa espressione è racchiusa in una funzione `date()` che converte il valore in una data. Questo processo viene

ripetuto 400 volte dalla funzione autogenerate. Infine, sfruttando un caricamento precedente, è possibile utilizzare una condizione where per caricare solo le date a partire dall'anno 2022. Questo script genera un calendario contenente tutte le date del 2022.

### maketime

Questa funzione restituisce una data calcolata dall'ora **hh**, dal minuto **mm** e dal secondo **ss**.

#### Sintassi:

```
MakeTime (hh [ , mm [ , ss ] ])
```

Tipo di dati restituiti: duale

#### Argomenti

| Argomento | Descrizione                                                                                    |
|-----------|------------------------------------------------------------------------------------------------|
| hh        | L'ora è un numero intero.                                                                      |
| mm        | Il minuto è un numero intero.<br>Se non si indicano i minuti, verrà utilizzato il valore 00.   |
| ss        | Il secondo è un numero intero.<br>Se non si indicano i secondi, verrà utilizzato il valore 00. |

### Casi di utilizzo

La funzione `maketime()` viene comunemente utilizzata nello script per la generazione dei dati per generare un campo temporale. A volte, quando il campo dell'ora è derivato dal testo in ingresso, questa funzione può essere utilizzata per costruire l'ora utilizzando i suoi componenti.

Questi esempi utilizzano il formato orario `h:mm:ss`. Il formato della data viene specificato nell'istruzione `SET TimeFormat` sopra lo script di caricamento dei dati. Modificare il formato negli esempi in base ai propri requisiti.

#### Esempi di funzioni

| Esempio                          | Risultato             |
|----------------------------------|-----------------------|
| <code>maketime(22)</code>        | Restituisce 22:00:00. |
| <code>maketime(22, 17)</code>    | Restituisce 22:17:00. |
| <code>maketime(22,17,52 )</code> | Restituisce 22:17:52. |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: `MM/GG/AAAA`. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema,

a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - maketime()

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni, caricato in una tabella denominata Transactions.
- Gli orari delle transazioni sono forniti in tre campi: hours, minutes e seconds.
- La creazione di un campo, transaction\_time, che restituisce l'ora nel formato della variabile di sistema TimeFormat.

#### Script di caricamento

```
SET TimeFormat='h:mm:ss TT';
```

```
Transactions:
```

```
    Load
        *,
        maketime(transaction_hour, transaction_minute, transaction_second) as transaction_time
    ;
Load * Inline [
transaction_id, transaction_hour, transaction_minute, transaction_second, transaction_amount,
transaction_quantity, customer_id
3750, 18, 43, 30, 12423.56, 23, 2038593
3751, 6, 32, 07, 5356.31, 6, 203521
3752, 12, 09, 16, 15.75, 1, 5646471
3753, 21, 43, 41, 7, 3036491
3754, 17, 55, 22, 21484.21, 1356, 049681
3756, 2, 52, 22, -59.18, 2, 2038593
3757, 9, 25, 23, 3177.4, 21, 203521
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- transaction\_hour
- transaction\_minute
- transaction\_second
- transaction\_time

Tabella dei risultati

| ora_transazione | minuto_transazione | secondo_transazione | tempo_transazione |
|-----------------|--------------------|---------------------|-------------------|
| 2               | 52                 | 22                  | 2:52:22 AM        |
| 6               | 32                 | 07                  | 6:32:07 AM        |
| 9               | 25                 | 23                  | 9:25:23 AM        |
| 12              | 09                 | 16                  | 12:09:16 PM       |
| 17              | 55                 | 22                  | 5:55:22 PM        |
| 18              | 43                 | 30                  | 6:43:30 PM        |
| 21              | 43                 | 41                  | 9:43:41 PM        |

Il campo `transaction_time` viene creato nell'istruzione di caricamento precedente utilizzando la funzione `maketime()` e passando i campi ora, minuti e secondi come argomenti della funzione.

La funzione combina e converte questi valori in un campo temporale, restituendo i risultati nel formato orario della variabile di sistema `TimeFormat`.

### Esempio 2 - time() function

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.
- La creazione di un campo, `transaction_time`, che ci permetterà di mostrare i risultati in formato orario 24 ore senza modificare la variabile di sistema `TimeFormat`.

#### Script di caricamento

```
SET TimeFormat='h:mm:ss TT';
```

```
Transactions:
```

```
    Load
        *,
        time(maketime(transaction_hour, transaction_minute, transaction_second), 'h:mm:ss') as
    transaction_time
    ;
```

```
Load * Inline [  
transaction_id, transaction_hour, transaction_minute, transaction_second, transaction_amount,  
transaction_quantity, customer_id  
3750, 18, 43, 30, 12423.56, 23, 2038593  
3751, 6, 32, 07, 5356.31, 6, 203521  
3752, 12, 09, 16, 15.75, 1, 5646471  
3753, 21, 43, 41, 7, 3036491  
3754, 17, 55, 22, 21484.21, 1356, 049681  
3756, 2, 52, 22, -59.18, 2, 2038593  
3757, 9, 25, 23, 3177.4, 21, 203521  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- transaction\_hour
- transaction\_minute
- transaction\_second
- transaction\_time

Tabella dei risultati

| ora_transazione | minuto_transazione | secondo_transazione | tempo_transazione |
|-----------------|--------------------|---------------------|-------------------|
| 2               | 52                 | 22                  | 2:52:22           |
| 6               | 32                 | 07                  | 6:32:07           |
| 9               | 25                 | 23                  | 9:25:23           |
| 12              | 09                 | 16                  | 12:09:16          |
| 17              | 55                 | 22                  | 17:55:22          |
| 18              | 43                 | 30                  | 18:43:30          |
| 21              | 43                 | 41                  | 21:43:41          |

In questo caso, la funzione `maketime()` è nidificata all'interno della funzione `time()`. Il secondo argomento della funzione `time()` imposta il formato dei risultati della funzione `maketime()` come il valore `h:mm:ss` richiesto.

### Esempio 3 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni, caricato in una tabella denominata Transactions.
- Tempi di transazione forniti in due campi: hours e minutes.
- La creazione di un campo, transaction\_time, che restituisce l'ora nel formato della variabile di sistema TimeFormat.

Creare una misura dell'oggetto grafico, transaction\_time, che restituisca un orario nel formato h:mm:ss TT.

### Script di caricamento

```
SET TimeFormat='h:mm:ss TT';
```

```
Transactions:
```

```
Load * Inline [  
transaction_id, transaction_hour, transaction_minute, transaction_amount, transaction_  
quantity, customer_id  
3750, 18, 43, 12423.56, 23, 2038593  
3751, 6, 32, 5356.31, 6, 203521  
3752, 12, 09, 15.75, 1, 5646471  
3753, 21, 43, 7, 3036491  
3754, 17, 55, 21484.21, 1356, 049681  
3756, 2, 52, -59.18, 2, 2038593  
3757, 9, 25, 3177.4, 21, 203521  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- transaction\_hour
- transaction\_minute

Per calcolare il valore transaction\_time, creare questa misura:

```
=maketime(transaction_hour,transaction_minute)
```

Tabella dei risultati

| ora_transazione | minuto_transazione | =maketime(transaction_hour, transaction_minute) |
|-----------------|--------------------|-------------------------------------------------|
| 2               | 52                 | 2:52:00 AM                                      |
| 6               | 32                 | 6:32:00 AM                                      |
| 9               | 25                 | 9:25:00 AM                                      |
| 12              | 09                 | 12:09:00 PM                                     |
| 17              | 55                 | 5:55:00 PM                                      |
| 18              | 43                 | 18:43:00                                        |
| 21              | 43                 | 21:43:00                                        |

La misura `transaction_time` viene creata nell'oggetto grafico utilizzando la funzione `maketime()` e passando i campi `anno` e `minuto` come argomenti della funzione.

La funzione combina quindi questi valori e i secondi vengono assunti pari a 00. Questi valori vengono poi convertiti in un campo temporale, restituendo i risultati nel formato della variabile di sistema `TimeFormat`.

### Esempio 4 - Scenario

Script di caricamento ed espressione del grafico

#### Panoramica

Creare un set di dati di calendario per il mese di gennaio 2022, suddiviso in incrementi di otto ore.

#### Script di caricamento

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';

tmpCalendar:
    load
        *
        where year(date)=2022;
load
    date(recno()+makedate(2021,12,31)) as date
AutoGenerate 31;

Left join(tmpCalendar)
load
    maketime((recno()-1)*8,00,00) as time
autogenerate 3;

Calendar:
load
    timestamp(date + time) as timestamp
resident tmpCalendar;

drop table tmpCalendar;
```

#### Risultati

Tabella dei risultati

| data e ora           |
|----------------------|
| 1/1/2022 12:00:00 AM |
| 01/01/2022 08:00:00  |
| 01/01/2022 16:00:00  |
| 02/01/2022 00:00:00  |
| 02/01/2022 08:00:00  |

| data e ora             |
|------------------------|
| 02/01/2022 16:00:00    |
| 03/01/2022 00:00:00    |
| 03/01/2022 08:00:00    |
| 03/01/2022 16:00:00    |
| 1/4/2022 12:00:00 a.m. |
| 1/4/2022 8:00:00 a.m.  |
| 1/4/2022 4:00:00 p.m.  |
| 05/01/2022 00:00:00    |
| 05/01/2022 08:00:00    |
| 05/01/2022 16:00:00    |
| 06/01/2022 00:00:00    |
| 06/01/2022 08:00:00    |
| 06/01/2022 16:00:00    |
| 07/01/2022 00:00:00    |
| 07/01/2022 08:00:00    |
| 07/01/2022 16:00:00    |
| 08/01/2022 00:00:00    |
| 08/01/2022 08:00:00    |
| 08/01/2022 16:00:00    |
| 09/01/2022 00:00:00    |
| + 68 ulteriori righe   |

La funzione iniziale autogenerate crea un calendario contenente tutte le date di gennaio in una tabella chiamata `tmpCalendar`.

Viene creata una seconda tabella, contenente tre record. Per ogni record, viene preso il valore `recno()` - 1 (valori 0, 1, 2) e il risultato viene moltiplicato per 8. Di conseguenza, si ottengono i valori 0, 8 e 16. Questi valori vengono utilizzati come parametro dell'ora in una funzione `maketime()`, con i valori dei minuti e dei secondi pari a 0. Di conseguenza, la tabella contiene tre campi orari: 12:00:00 AM, 8:00:00 AM e 4:00:00 PM.

Questa tabella è unita alla tabella `tmpCalendar`. Poiché non ci sono campi corrispondenti tra le due tabelle per il join, le righe dell'ora vengono aggiunte a ciascuna riga della data. Di conseguenza, ogni riga di data viene ripetuta tre volte per ogni valore temporale.

Infine, la tabella Calendario viene creata da un caricamento residente della tabella tmpCalendar. I campi data e ora vengono concatenati e avvolti nella funzione timestamp() per creare il campo timestamp.

La tabella tmpCalendar viene quindi rilasciata.

### makeweekdate

Questa funzione restituisce una data calcolata dall'anno, dal numero di settimana e dal giorno della settimana.

#### Sintassi:

```
MakeWeekDate (weekyear [, week [, weekday [, first_week_day [, broken_weeks [, reference_day]]]])
```

**Tipo di dati restituiti:** duale

La funzione makeweekdate() è disponibile sia come script che come grafico. La funzione calcola la data in base ai parametri inseriti nella funzione.

#### Argomenti

| Argomento       | Descrizione                                                                                                                                                                                                                                                                                                                                                                             |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>weekyear</b> | L'anno definito dalla funzione weekYear() per la data specifica, ovvero l'anno a cui appartiene il numero della settimana.<br><div> <i>In alcuni casi l'anno della settimana può essere diverso dall'anno solare, ad esempio se la settimana 1 inizia già a dicembre dell'anno precedente.</i></div> |
| <b>week</b>     | Il numero della settimana definito dalla funzione week() per la data specifica.<br>Se non viene dichiarato alcun numero di settimana, viene utilizzato -1.                                                                                                                                                                                                                              |

| Argomento             | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>weekday</b>        | <p>Il giorno della settimana definito dalla funzione <code>weekday()</code> per la data in questione. 0 è il primo giorno della settimana e 6 è l'ultimo giorno della settimana.</p> <p>Se non si indica alcun giorno della settimana, verrà utilizzato il valore 0.</p> <div>  <p><i>Anche se 0 significa sempre primo giorno della settimana e 6 è sempre l'ultimo, il parametro <b>first_week_day</b> determina a quali giorni della settimana corrisponde. Se omissso, viene utilizzato il valore della variabile <b>FirstWeekDay</b>.</i></p> </div> <p>Se si utilizzano settimane parziali, insieme a una combinazione impossibile di parametri, si può ottenere un risultato che non appartiene all'anno scelto.</p> <p><b>Esempio:</b></p> <p><code>MakeWeekDate(2021,1,0,6,1)</code><br/> Restituisce "27 dicembre 2020", poiché questo giorno è il primo (la domenica) della settimana specificata. Il 1° gennaio 2021 era un venerdì.</p> |
| <b>first_week_day</b> | <p>Specifica il giorno di inizio della settimana. Se omissso, viene utilizzato il valore della variabile <b>FirstWeekDay</b>.</p> <p>I valori possibili per <b>first_week_day</b> sono 0 per lunedì, 1 per martedì, 2 per mercoledì, 3 per giovedì, 4 per venerdì, 5 per sabato e 6 per domenica.</p> <p>Per maggiori informazioni sulle variabili di sistema, vedere <a href="#">FirstWeekDay (page 242)</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>broken_weeks</b>   | <p>Se non si specifica <b>broken_weeks</b>, il valore della variabile <b>BrokenWeeks</b> verrà utilizzato per definire se le settimane sono parziali o meno.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>reference_day</b>  | <p>Se non si specifica <b>reference_day</b>, il valore della variabile <b>ReferenceDay</b> verrà utilizzato per definire quale giorno di gennaio impostare come giorno di riferimento per definire la settimana 1.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

## Casi di utilizzo

La funzione `makeweekdate()` viene comunemente utilizzata nello script per la generazione di dati per generare un elenco di date o per costruire date quando l'anno, la settimana e il giorno della settimana sono forniti nei dati di input.

Gli esempi seguenti presuppongono:

```
SET FirstWeekDay=0;
SET BrokenWeeks=0;
SET ReferenceDay=4;
```

### Esempi di funzioni

| Esempio                             | Risultato                                                             |
|-------------------------------------|-----------------------------------------------------------------------|
| <code>makeweekdate(2014,6,6)</code> | restituisce 02/09/2014                                                |
| <code>makeweekdate(2014,6,1)</code> | restituisce 02/04/2014                                                |
| <code>makeweekdate(2014,6)</code>   | restituisce 02/03/2014 (viene utilizzato il giorno della settimana 0) |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - giorno incluso

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente il totale delle vendite settimanali per il 2022 in una tabella chiamata `sa1es`.
- Le date delle transazioni sono fornite in due campi: `year`, `week` e `sa1es`.
- Un caricamento precedente, utilizzato per creare una misura, `end_of_week`, utilizza la funzione `makeweekdate()` per restituire la data del venerdì di quella settimana nel formato MM/GG/AAAA.

Per dimostrare che la data restituita è un venerdì, l'espressione `end_of_week` viene anche racchiusa nella funzione `weekday()` per mostrare il giorno della settimana.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
SET FirstWeekDay=0;
SET BrokenWeeks=0;
SET ReferenceDay=4;
```

Transactions:

```
Load
    *,
    makeweekdate(transaction_year, transaction_week,4) as end_of_week,
    weekday(makeweekdate(transaction_year, transaction_week,4)) as week_day
;
Load * Inline [
transaction_year, transaction_week, sales
2022, 01, 10000
2022, 02, 11250
2022, 03, 9830
2022, 04, 14010
2022, 05, 28402
2022, 06, 9992
2022, 07, 7292
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- transaction\_year
- transaction\_week
- end\_of\_week
- week\_day

Tabella dei risultati

| anno_transazione | settimana_transazione | end_of_week | week_day |
|------------------|-----------------------|-------------|----------|
| 2022             | 01                    | 01/07/2022  | Ven      |
| 2022             | 02                    | 01/14/2022  | Ven      |
| 2022             | 03                    | 01/21/2022  | Ven      |
| 2022             | 04                    | 01/28/2022  | Ven      |
| 2022             | 05                    | 02/04/2022  | Ven      |
| 2022             | 06                    | 02/11/2022  | Ven      |
| 2022             | 07                    | 18/02/2022  | Ven      |

Il campo 'end\_of\_week' viene creato nell'istruzione di caricamento precedente utilizzando la funzione `makeweekdate()`. I campi `transaction_year`, `transaction_week` vengono passati alla funzione come argomenti dell'anno e della settimana. Il valore 4 viene utilizzato per l'argomento giorno.

La funzione combina e converte questi valori in un campo data, restituendo i risultati nel formato della variabile di sistema `DateFormat`.

La funzione `makeweekdate()` e i relativi argomenti sono anche avvolti in una funzione `weekday()` per restituire il campo `week_day`; e come si può vedere nella tabella precedente, il campo `week_day` mostra che queste date si verificano di venerdì.

### Esempio 2 - giorno escluso

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente i totali delle vendite settimanali per il 2022 in una tabella chiamata `sales`.
- Le date delle transazioni sono fornite in due campi: `year`, `week` e `sales`.
- Un caricamento precedente, che viene sfruttato per creare una misura, `first_day_of_week`, utilizzando la funzione `makeweekdate()`. Questo restituirà la data del lunedì di quella settimana nel formato MM/GG/AAAA.

Per dimostrare che la data restituita è un lunedì, l'espressione `first_day_of_week` viene anche racchiusa nella funzione `weekday()` per mostrare il giorno della settimana.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
SET FirstWeekDay=0;
SET BrokenWeeks=0;
SET ReferenceDay=4;
```

Transactions:

```
Load
    *,
    makeweekdate(transaction_year, transaction_week) as first_day_of_week,
    weekday(makeweekdate(transaction_year, transaction_week)) as week_day
;
Load * Inline [
transaction_year, transaction_week, sales
2022, 01, 10000
2022, 02, 11250
2022, 03, 9830
2022, 04, 14010
2022, 05, 28402
2022, 06, 9992
2022, 07, 7292
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `transaction_year`
- `transaction_week`

- `first_day_of_week`
- `week_day`

Tabella dei risultati

| anno_transazione | settimana_transazione | primo_giorno_della_settimana | week_day |
|------------------|-----------------------|------------------------------|----------|
| 2022             | 01                    | 01/03/2022                   | Mon      |
| 2022             | 02                    | 01/10/2022                   | Mon      |
| 2022             | 03                    | 01/17/2022                   | Mon      |
| 2022             | 04                    | 01/24/2022                   | Mon      |
| 2022             | 05                    | 01/31/2022                   | Mon      |
| 2022             | 06                    | 02/07/2022                   | Mon      |
| 2022             | 07                    | 02/14/2022                   | Mon      |

Il campo `first_day_of_week` viene creato nell'istruzione `LOAD` precedente utilizzando la funzione `makeweekdate()`. I parametri `transaction_year` e `transaction_week` vengono passati come argomenti della funzione, mentre il parametro `giorno` viene lasciato vuoto.

La funzione combina e converte questi valori in un campo `data`, restituendo i risultati nel formato della variabile di sistema `DateFormat`.

La funzione `makeweekdate()` e i relativi argomenti sono anche avvolti in una funzione `weekday()` che restituisce il campo `week_day`. Come si può vedere nella tabella precedente, il campo `week_day` restituisce Lunedì in tutti i casi, poiché questo parametro è stato lasciato vuoto nella funzione `makeweekdate()`, che ha come valore predefinito 0 (primo giorno della settimana), e il primo giorno della settimana è impostato su Lunedì dalla variabile di sistema `FirstWeekDay`.

### Esempio 3 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente i totali delle vendite settimanali per il 2022 in una tabella chiamata `sa1es`.
- Le date delle transazioni sono fornite in due campi: `year`, `week` e `sa1es`.

In questo esempio, si utilizzerà un oggetto grafico per creare una misura equivalente al calcolo `end_of_week` del primo esempio. Questa misura utilizzerà la funzione `makeweekdate()` per restituire la data del venerdì di quella settimana nel formato `MM/GG/AAAA`.

Per dimostrare che la data restituita è un venerdì, viene creata una seconda misura per restituire il giorno della settimana.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
SET FirstWeekDay=0;
SET BrokenWeeks=0;
SET ReferenceDay=4;

Master_Calendar:
Load * Inline [
transaction_year, transaction_week, sales
2022, 01, 10000
2022, 02, 11250
2022, 03, 9830
2022, 04, 14010
2022, 05, 28402
2022, 06, 9992
2022, 07, 7292
];
```

### Risultati

#### Procedere come indicato di seguito:

1. Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:
  - transaction\_year
  - transaction\_week
2. Per eseguire il calcolo equivalente a quello del end\_of\_week campo del primo esempio, creare la seguente misura:  
=makeweekdate(transaction\_year, transaction\_week, 4)
3. Per calcolare il giorno della settimana per ogni transazione, creare la seguente misura:  
=weekday(makeweekdate(transaction\_year, transaction\_week, 4))

Tabella dei risultati

| anno_<br>transazione | settimana_<br>transazione | =makeweekdate(anno_<br>transazione, settimana_<br>transazione, 4) | =GIORNO.SETTIMANA<br>(makeweekdate(anno_<br>transazione, settimana_<br>transazione, 4)) |
|----------------------|---------------------------|-------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
| 2022                 | 01                        | 01/07/2022                                                        | Ven                                                                                     |
| 2022                 | 02                        | 01/14/2022                                                        | Ven                                                                                     |
| 2022                 | 03                        | 01/21/2022                                                        | Ven                                                                                     |
| 2022                 | 04                        | 01/28/2022                                                        | Ven                                                                                     |

| anno_<br>transazione | settimana_<br>transazione | =makeweekdate(anno_<br>transazione,settimana_<br>transazione,4) | =GIORNO.SETTIMANA<br>(makeweekdate(anno_<br>transazione,settimana_<br>transazione,4)) |
|----------------------|---------------------------|-----------------------------------------------------------------|---------------------------------------------------------------------------------------|
| 2022                 | 05                        | 02/04/2022                                                      | Ven                                                                                   |
| 2022                 | 06                        | 02/11/2022                                                      | Ven                                                                                   |
| 2022                 | 07                        | 18/02/2022                                                      | Ven                                                                                   |

Un campo equivalente a `end_of_week` viene creato nell'oggetto grafico come misura utilizzando la funzione `makeweekdate()`. I campi `transaction_year` e `transaction_week` vengono passati come argomenti dell'anno e della settimana. Il valore 4 viene utilizzato per l'argomento giorno.

La funzione combina e converte questi valori in un campo data, restituendo i risultati nel formato della variabile di sistema `DateFormat`.

Anche la funzione `makeweekdate()` e i relativi argomenti sono disposti in una funzione `weekday()` per restituire un calcolo equivalente a quello del campo `week_day` del primo esempio. Come si può vedere nella tabella precedente, l'ultima colonna a destra mostra che queste date si verificano di venerdì.

## Esempio 4 - Scenario

Script di caricamento ed espressione del grafico

### Panoramica

In questo esempio, si crea un elenco di date contenente tutti i venerdì dell'anno 2022.

Aprire l'Editor di caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
SET FirstWeekDay=0;
SET BrokenWeeks=0;
SET ReferenceDay=4;

Calendar:
load
    *,
    weekday(date) as weekday
where year(date)=2022;
load
    makeweekdate(2022,recno()-2,4) as date
AutoGenerate 60;
```

### Risultati

Tabella dei risultati

| date             | giorno feriale |
|------------------|----------------|
| 01/07/2022       | Ven            |
| 01/14/2022       | Ven            |
| 01/21/2022       | Ven            |
| 01/28/2022       | Ven            |
| 02/04/2022       | Ven            |
| 02/11/2022       | Ven            |
| 18/02/2022       | Ven            |
| 02/25/2022       | Ven            |
| 03/04/2022       | Ven            |
| 03/11/2022       | Ven            |
| 03/18/2022       | Ven            |
| 03/25/2022       | Ven            |
| 04/01/2022       | Ven            |
| 04/08/2022       | Ven            |
| 04/15/2022       | Ven            |
| 04/22/2022       | Ven            |
| 04/29/2022       | Ven            |
| 05/06/2022       | Ven            |
| 05/13/2022       | Ven            |
| 05/20/2022       | Ven            |
| 05/27/2022       | Ven            |
| 06/03/2022       | Ven            |
| 06/10/2022       | Ven            |
| 06/17/2022       | Ven            |
| + altre 27 righe |                |

La funzione `makeweekdate()` trova ogni venerdì del 2022. L'utilizzo di un parametro di settimana pari a -2 garantisce che non vengano perse delle date. Infine, un caricamento precedente crea un campo `weekday` aggiuntivo per chiarezza, per mostrare che ogni valore `date` è un venerdì.

## minute

Questa funzione restituisce un numero intero che rappresenta il minuto in cui la frazione di **expression** viene interpretata come ora in base all'interpretazione numerica standard.

Sintassi:

```
minute(expression)
```

Tipo di dati restituiti: numero intero

### Casi di utilizzo

La funzione `minute()` è utile quando si desidera confrontare le aggregazioni per minuto. Ad esempio, è possibile utilizzare la funzione per visualizzare la distribuzione del conteggio delle attività per minuto.

È possibile creare queste dimensioni anche nello script di caricamento utilizzando la funzione che consente di creare un campo in una tabella Calendario principale. In alternativa, possono essere utilizzate direttamente in un grafico come dimensione calcolata.

Esempi di funzioni

| Esempio                            | Risultato                                  |
|------------------------------------|--------------------------------------------|
| <code>minute ( '09:14:36' )</code> | Restituisce 14.                            |
| <code>minute ( '0.5555' )</code>   | Restituisce 19 (poiché 0.5555 = 13:19:55). |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - Variabile (script)

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente transazioni per timestamp, caricato in una tabella chiamata `Transactions`.
- Viene utilizzata la variabile di sistema `Timestamp` predefinita (`M/D/YYYY h:mm:ss[.fff] TT`).
- La creazione di un campo, `minute`, per calcolare quando sono avvenute le transazioni.

### Script di caricamento

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

```
Transactions:
```

```
    Load
        *,
        minute(timestamp) as minute
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
id,timestamp,amount
9497,'2022-01-05 19:04:57',47.25,
9498,'2022-01-03 14:21:53',51.75,
9499,'2022-01-03 05:40:49',73.53,
9500,'2022-01-04 18:49:38',15.35,
9501,'2022-01-01 22:10:22',31.43,
9502,'2022-01-05 19:34:46',13.24,
9503,'2022-01-04 22:58:34',74.34,
9504,'2022-01-06 11:29:38',50.00,
9505,'2022-01-02 08:35:54',36.34,
9506,'2022-01-06 08:49:09',74.23
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `timestamp`
- `minute`

Tabella dei risultati

| data e ora          | minute |
|---------------------|--------|
| 2022-01-01 22:10:22 | 10     |
| 2022-01-02 08:35:54 | 35     |
| 2022-01-03 05:40:49 | 40     |
| 2022-01-03 14:21:53 | 21     |
| 2022-01-04 18:49:38 | 49     |

| data e ora          | minute |
|---------------------|--------|
| 2022-01-04 22:58:34 | 58     |
| 2022-01-05 19:04:57 | 4      |
| 2022-01-05 19:34:46 | 34     |
| 2022-01-06 08:49:09 | 49     |
| 2022-01-06 11:29:38 | 29     |

I valori nel campo `minute` sono creati usando la funzione `minute()` e trasferendo il valore `timestamp` come espressione nell'istruzione di caricamento precedente.

### Esempio 2 - Oggetto grafico (grafico)

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.
- Viene utilizzata la variabile di sistema `timestamp` predefinita (`M/D/YYYY h:mm:ss[.fff] TT`).

Tuttavia, in questo esempio, il set di dati invariato viene caricato nell'applicazione. I valori `minute` sono calcolati mediante una misura in un oggetto grafico.

#### Script di caricamento

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,timestamp,amount
```

```
9497,'2022-01-05 19:04:57',47.25,
```

```
9498,'2022-01-03 14:21:53',51.75,
```

```
9499,'2022-01-03 05:40:49',73.53,
```

```
9500,'2022-01-04 18:49:38',15.35,
```

```
9501,'2022-01-01 22:10:22',31.43,
```

```
9502,'2022-01-05 19:34:46',13.24,
```

```
9503,'2022-01-04 22:58:34',74.34,
```

```
9504,'2022-01-06 11:29:38',50.00,
```

```
9505,'2022-01-02 08:35:54',36.34,
```

```
9506,'2022-01-06 08:49:09',74.23
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: timestamp.

Creare la seguente misura:

```
=minute(timestamp)
```

Tabella dei risultati

| data e ora          | minute |
|---------------------|--------|
| 2022-01-01 22:10:22 | 10     |
| 2022-01-02 08:35:54 | 35     |
| 2022-01-03 05:40:49 | 40     |
| 2022-01-03 14:21:53 | 21     |
| 2022-01-04 18:49:38 | 49     |
| 2022-01-04 22:58:34 | 58     |
| 2022-01-05 19:04:57 | 4      |
| 2022-01-05 19:34:46 | 34     |
| 2022-01-06 08:49:09 | 49     |
| 2022-01-06 11:29:38 | 29     |

I valori per minute sono creati usando la funzione `minute()` e trasferendo il valore `timestamp` come espressione in una misura per l'oggetto grafico.

### Esempio 3 - Scenario

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati di timestamp, generato per rappresentare gli ingressi in un tornello.
- Informazioni con ogni timestamp e id corrispondente, che viene caricato in una tabella chiamata `Ticket_Barrier_Tracker`.
- Viene utilizzata la variabile di sistema `timestamp` predefinita (`M/D/YYYY h:mm:ss[.fff] TT`).

L'utente desidera un oggetto grafico che mostri, al minuto, il numero di ingressi al tornello.

### Script di caricamento

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';

tmpTimeStampCreator:
    load
        *
        where year(date)=2022;
load
    date(recno()+makedate(2021,12,31)) as date
AutoGenerate 1;

join load
    maketime(floor(rand()*24),floor(rand()*59),floor(rand()*59)) as time
autogenerate 10000;

Ticket_Barrier_Tracker:
load
    recno() as id,
    timestamp(date + time) as timestamp
resident tmpTimeStampCreator;

drop table tmpTimeStampCreator;
```

### Risultati

Procedere come indicato di seguito:

1. Caricare i dati e aprire un foglio. Creare una nuova tabella.
2. Creare una dimensione calcolata utilizzando la seguente espressione:  
=minute(timestamp)
3. Aggiungere la seguente misura di aggregazione per calcolare il conteggio totale degli ingressi:  
=count(id)
4. Impostare la **Formattazione numero** della misura su **Denaro**.

Tabella dei risultati

| minute(timestamp) | =count(id) |
|-------------------|------------|
| 0                 | 174        |
| 1                 | 171        |
| 2                 | 175        |
| 3                 | 165        |
| 4                 | 188        |
| 5                 | 176        |
| 6                 | 158        |
| 7                 | 187        |

| minute(timestamp) | =count(id) |
|-------------------|------------|
| 8                 | 178        |
| 9                 | 178        |
| 10                | 197        |
| 11                | 161        |
| 12                | 166        |
| 13                | 184        |
| 14                | 159        |
| 15                | 161        |
| 16                | 152        |
| 17                | 160        |
| 18                | 176        |
| 19                | 164        |
| 20                | 170        |
| 21                | 170        |
| 22                | 142        |
| 23                | 145        |
| 24                | 155        |
| + altre 35 righe  |            |

## month

Questa funzione restituisce un valore duale: il nome del mese come definito nella variabile di ambiente **MonthNames** e un numero intero compreso tra 1 e 12. Il numero del mese viene calcolato a partire dall'interpretazione della data dell'espressione in base all'interpretazione numerica standard.

La funzione restituisce il nome del mese nel formato del sistema `MonthName` variabile per una particolare data. È comunemente utilizzata per creare un campo dati come dimensione in un Calendario principale.

### Sintassi:

**month** (expression)

**Tipo di dati restituiti:** numero intero

### Esempi di funzioni

| Esempio             | Risultato                                 |
|---------------------|-------------------------------------------|
| month( 2012-10-12 ) | restituisce Oct                           |
| month( 35648 )      | restituisce Aug poiché 35648 = 1997-08-06 |

### Esempio 1 - set di dati DateFormat (script)

Script di caricamento e risultati

#### Panoramica

Aprire Editor caricamento dati, quindi aggiungere lo script di caricamento in basso in una nuova scheda.

Lo script di caricamento contiene:

- Un set di dati per le date denominato `master_calendar`. La variabile di sistema `DateFormat` è impostata su `GG/MM/AAAA`.
- Un caricamento precedente che crea un campo aggiuntivo, denominato `month_name`, mediante la funzione `month()`.
- Un campo aggiuntivo, denominato `long_date`, che utilizza la funzione `date()` per esprimere la data completa.

#### Script di caricamento

```
SET DateFormat='DD/MM/YYYY';
```

```
Master_Calendar:
```

```
Load
```

```
    date,  
    date(date,'dd-MMMM-YYYY') as long_date,  
    month(date) as month_name
```

```
Inline
```

```
[
```

```
date
```

```
03/01/2022
```

```
03/02/2022
```

```
03/03/2022
```

```
03/04/2022
```

```
03/05/2022
```

```
03/06/2022
```

```
03/07/2022
```

```
03/08/2022
```

```
03/09/2022
```

```
03/10/2022
```

```
03/11/2022
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- long\_date
- month\_name

Tabella dei risultati

| data       | long_date          | month_name |
|------------|--------------------|------------|
| 03/01/2022 | 03-January- 2022   | Jan        |
| 03/02/2022 | 03-February- 2022  | Feb        |
| 03/03/2022 | 03-March- 2022     | Mar        |
| 03/04/2022 | 03-April- 2022     | Apr        |
| 03/05/2022 | 03-May- 2022       | May        |
| 03/06/2022 | 03-June- 2022      | Jun        |
| 03/07/2022 | 03-July- 2022      | Jul        |
| 03/08/2022 | 03-August- 2022    | Aug        |
| 03/09/2022 | 03-September- 2022 | Sep        |
| 03/10/2022 | 03-October- 2022   | Oct        |
| 03/11/2022 | 03-November- 2022  | Nov        |

Il nome del mese viene valutato correttamente dalla funzione `month()` nello script.

### Esempio 2 - Date ANSI (script)

Script di caricamento e risultati

#### Panoramica

Aprire Editor caricamento dati, quindi aggiungere lo script di caricamento in basso in una nuova scheda.

Lo script di caricamento contiene:

- Un set di dati di date denominato `master_calendar`. Viene utilizzata la variabile di sistema `DateFormat GG/MM/AAAA`. Tuttavia, le date incluse nel set di dati sono nel formato data standard ANSI.
- Un caricamento precedente che crea un campo aggiuntivo, denominato `month_name`, mediante la funzione `month()`.

- Un campo aggiuntivo, denominato `long_date`, che utilizza la funzione `date()` per esprimere la data completa.

### Script di caricamento

```
SET DateFormat='DD/MM/YYYY';
Master_Calendar:
Load
    date,
    date(date,'dd-MMMM-YYYY') as long_date,
    month(date) as month_name

Inline
[
date
2022-01-11
2022-02-12
2022-03-13
2022-04-14
2022-05-15
2022-06-16
2022-07-17
2022-08-18
2022-09-19
2022-10-20
2022-11-21
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `date`
- `long_date`
- `month_name`

Tabella dei risultati

| data       | long_date      | month_name |
|------------|----------------|------------|
| 03/11/2022 | 11-March- 2022 | 11         |
| 03/12/2022 | 12-March- 2022 | 12         |
| 03/13/2022 | 13-March- 2022 | 13         |
| 03/14/2022 | 14-March- 2022 | 14         |
| 03/15/2022 | 15-March- 2022 | 15         |
| 03/16/2022 | 16-March- 2022 | 16         |
| 03/17/2022 | 17-March- 2022 | 17         |

| data       | long_date      | month_name |
|------------|----------------|------------|
| 03/18/2022 | 18-March- 2022 | 18         |
| 03/19/2022 | 19-March- 2022 | 19         |
| 03/20/2022 | 20-March- 2022 | 20         |
| 03/21/2022 | 21-March- 2022 | 21         |

Il nome del mese viene valutato correttamente dalla funzione `month()` nello script.

### Esempio 3 - Date non formattate (script)

Script di caricamento e risultati

#### Panoramica

Aprire Editor caricamento dati, quindi aggiungere lo script di caricamento in basso in una nuova scheda.

Lo script di caricamento contiene:

- Un set di dati di date denominato `Master_Calendar`. Viene utilizzata la variabile di sistema `DateFormat GG/MM/AAAA`.
- Un caricamento precedente che crea un campo aggiuntivo, denominato `month_name`, mediante la funzione `month()`.
- La data non formattata originale, denominata `unformatted_date`.
- Un campo aggiuntivo, denominato `long_date`, che utilizza la funzione `date()` per esprimere la data completa.

#### Script di caricamento

```
SET DateFormat='DD/MM/YYYY';
```

```
Master_Calendar:
```

```
Load
    unformatted_date,
    date(unformatted_date,'dd-MMMM-YYYY') as long_date,
    month(unformatted_date) as month_name
```

```
Inline
```

```
[
unformatted_date
44868
44898
44928
44958
44988
45018
45048
45078
45008
```

```
45038  
45068  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- unformatted\_date
- long\_date
- month\_name

Tabella dei risultati

| unformatted_date | long_date          | month_name |
|------------------|--------------------|------------|
| 44868            | 03-January- 2022   | Jan        |
| 44898            | 03-February- 2022  | Feb        |
| 44928            | 03-March- 2022     | Mar        |
| 44958            | 03-April- 2022     | Apr        |
| 44988            | 03-May- 2022       | May        |
| 45018            | 03-June- 2022      | Jun        |
| 45048            | 03-July- 2022      | Jul        |
| 45078            | 03-August- 2022    | Aug        |
| 45008            | 03-September- 2022 | Sep        |
| 45038            | 03-October- 2022   | Oct        |
| 45068            | 03-November- 2022  | Nov        |

Il nome del mese viene valutato correttamente dalla funzione `month()` nello script.

### Esempio 4 - Calcolo del mese di scadenza

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire Editor caricamento dati, quindi aggiungere lo script di caricamento in basso in una nuova scheda.

Lo script di caricamento contiene:

- Un set di dati degli ordini effettuati a marzo denominato `subscriptions`. La tabella contiene tre campi:

- id
- order\_date
- importo

### Script di caricamento

Subscriptions:

Load

```
id,  
order_date,  
amount
```

Inline

```
[  
id,order_date,amount  
1,03/01/2022,231.24  
2,03/02/2022,567.28  
3,03/03/2022,364.28  
4,03/04/2022,575.76  
5,03/05/2022,638.68  
6,03/06/2022,785.38  
7,03/07/2022,967.46  
8,03/08/2022,287.67  
9,03/09/2022,764.45  
10,03/10/2022,875.43  
11,03/11/2022,957.35  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: order\_date.

Per calcolare il mese in cui scadrà un ordine, creare questa misura: =month(order\_date+180).

Tabella dei risultati

| order_date | =month(order_date+180) |
|------------|------------------------|
| 03/01/2022 | Jul                    |
| 03/02/2022 | Aug                    |
| 03/03/2022 | Aug                    |
| 03/04/2022 | Sep                    |
| 03/05/2022 | Oct                    |
| 03/06/2022 | Nov                    |
| 03/07/2022 | Dec                    |
| 03/08/2022 | Jan                    |

|                   |                               |
|-------------------|-------------------------------|
| <b>order_date</b> | <b>=month(order_date+180)</b> |
| 03/09/2022        | Mar                           |
| 03/10/2022        | Apr                           |
| 03/11/2022        | May                           |

La funzione `month()` determina correttamente che un ordine effettuato l'11 marzo scadrebbe a luglio.

## monthend

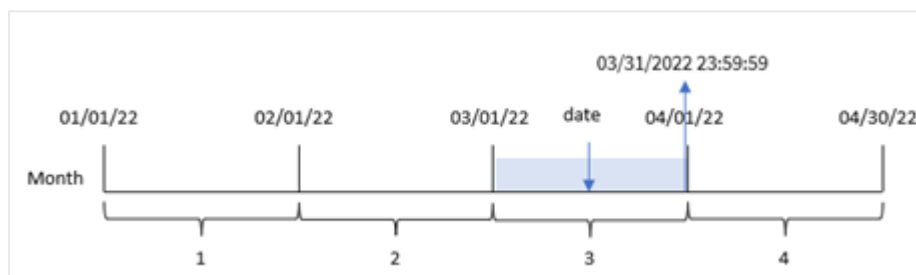
Questa funzione restituisce un valore corrispondente a un indicatore temporale recante l'ultimo millisecondo dell'ultimo giorno del mese contenente `date`. Il formato di output predefinito sarà il formato `DateFormat` impostato nello script.

### Sintassi:

**MonthEnd**(date[, period\_no])

In altre parole, la funzione `monthend()` determina in quale mese cade la data. Quindi, restituisce un timestamp, nel formato `data`, per l'ultimo millisecondo di quel mese.

*Schema della funzione monthend.*



### Casi di utilizzo

La funzione `monthend()` viene utilizzata come parte di un'espressione quando si desidera che il calcolo utilizzi la frazione del mese non ancora trascorsa. Ad esempio, se si vuole calcolare l'interesse totale non ancora maturato durante il mese.

**Tipo di dati restituiti:** duale

### Argomenti

| Argomento        | Descrizione                                                                                                                                                                                                                                 |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>date</b>      | La data o la data e ora da valutare.                                                                                                                                                                                                        |
| <b>period_no</b> | <b>period_no</b> è un numero intero che, se corrisponde a 0 o viene omissso, indica il mese contenente <b>date</b> . I valori negativi di <b>period_no</b> indicano i mesi precedenti, mentre i valori positivi indicano i mesi successivi. |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

Esempi di funzioni

| Esempio                                 | Risultato                        |
|-----------------------------------------|----------------------------------|
| <code>monthend('02/19/2012')</code>     | Restituisce 02/29/2012 23:59:59. |
| <code>monthend('02/19/2001', -1)</code> | Restituisce 01/31/2001 23:59:59. |

### Esempio 1 - Esempio di base

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2022 viene caricato in una tabella denominata 'Transactions'.
- Un campo data nel formato `DateFormat` della variabile di sistema (MM/DD/YYYY).
- Un'istruzione di caricamento precedente contenente:
  - La funzione `monthend()` impostata come campo, 'end\_of\_month'.
  - La funzione `timestamp` impostata come campo, 'end\_of\_month\_timestamp'.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
  Load  
    *,  
    monthend(date) as end_of_month,  
    timestamp(monthend(date)) as end_of_month_timestamp
```

```
;
Load
*
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- date
- end\_of\_month
- end\_of\_month\_timestamp

Tabella dei risultati

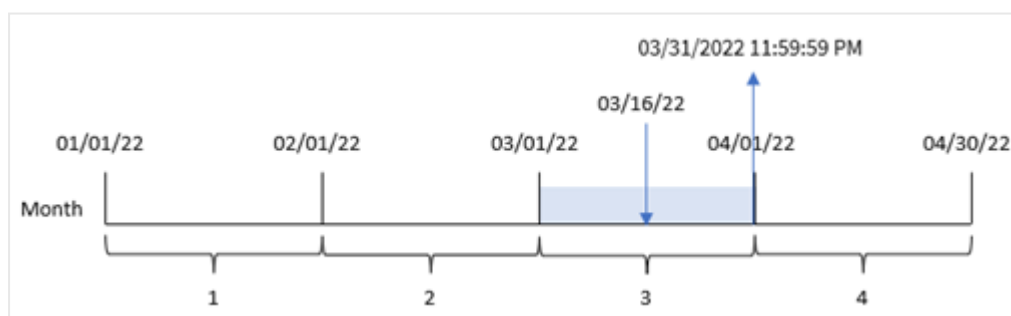
| id   | date      | end_of_month | end_of_month_timestamp |
|------|-----------|--------------|------------------------|
| 8188 | 1/7/2022  | 01/31/2022   | 1/31/2022 11:59:59 PM  |
| 8189 | 1/19/2022 | 01/31/2022   | 1/31/2022 11:59:59 PM  |
| 8190 | 2/5/2022  | 02/28/2022   | 2/28/2022 11:59:59 PM  |
| 8191 | 2/28/2022 | 02/28/2022   | 2/28/2022 11:59:59 PM  |
| 8192 | 3/16/2022 | 03/31/2022   | 3/31/2022 11:59:59 PM  |
| 8193 | 4/1/2022  | 04/30/2022   | 4/30/2022 11:59:59 PM  |
| 8194 | 5/7/2022  | 05/31/2022   | 5/31/2022 11:59:59 PM  |

| id   | date       | end_of_month | end_of_month_timestamp |
|------|------------|--------------|------------------------|
| 8195 | 5/16/2022  | 05/31/2022   | 5/31/2022 11:59:59 PM  |
| 8196 | 6/15/2022  | 06/30/2022   | 6/30/2022 11:59:59 PM  |
| 8197 | 6/26/2022  | 06/30/2022   | 6/30/2022 11:59:59 PM  |
| 8198 | 7/9/2022   | 07/31/2022   | 7/31/2022 11:59:59 PM  |
| 8199 | 7/22/2022  | 07/31/2022   | 7/31/2022 11:59:59 PM  |
| 8200 | 7/23/2022  | 07/31/2022   | 7/31/2022 11:59:59 PM  |
| 8201 | 7/27/2022  | 07/31/2022   | 7/31/2022 11:59:59 PM  |
| 8202 | 8/2/2022   | 08/31/2022   | 8/31/2022 11:59:59 PM  |
| 8203 | 8/8/2022   | 08/31/2022   | 8/31/2022 11:59:59 PM  |
| 8204 | 8/19/2022  | 08/31/2022   | 8/31/2022 11:59:59 PM  |
| 8205 | 9/26/2022  | 09/30/2022   | 9/30/2022 11:59:59 PM  |
| 8206 | 10/14/2022 | 10/31/2022   | 10/31/2022 11:59:59 PM |
| 8207 | 10/29/2022 | 10/31/2022   | 10/31/2022 11:59:59 PM |

Il campo 'end\_of\_month' viene creato nell'istruzione di caricamento precedente mediante l'uso della funzione `monthend()` e trasferendo il campo data come argomento della funzione.

La funzione `monthend()` identifica in quale mese cade il valore della data, restituendo un timestamp per l'ultimo millisecondo di quel mese.

*Schema della funzione `monthend` con marzo come mese selezionato.*



La transazione 8192 è avvenuta il 16 marzo. La funzione `monthend()` restituisce l'ultimo millisecondo di quel mese, ovvero il 31 marzo alle 23:59:59.

### Esempio 2 - period\_no

Script di caricamento e risultati

#### Panoramica

Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.

In questo esempio, si tratta di creare un campo, 'previous\_month\_end', che restituisca il timestamp della fine del mese precedente la transazione.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
```

```
    *,
```

```
    monthend(date,-1) as previous_month_end,
```

```
    timestamp(monthend(date,-1)) as previous_month_end_timestamp
```

```
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
```

```
8189,1/19/2022,37.23
```

```
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
```

```
8192,3/16/2022,53.80
```

```
8193,4/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/16/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
```

```
8200,7/23/2022,82.77
```

```
8201,7/27/2022,69.98
```

```
8202,8/2/2022,76.11
```

```
8203,8/8/2022,25.12
```

```
8204,8/19/2022,46.23
```

```
8205,9/26/2022,84.21
```

```
8206,10/14/2022,96.24
```

```
8207,10/29/2022,67.67
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

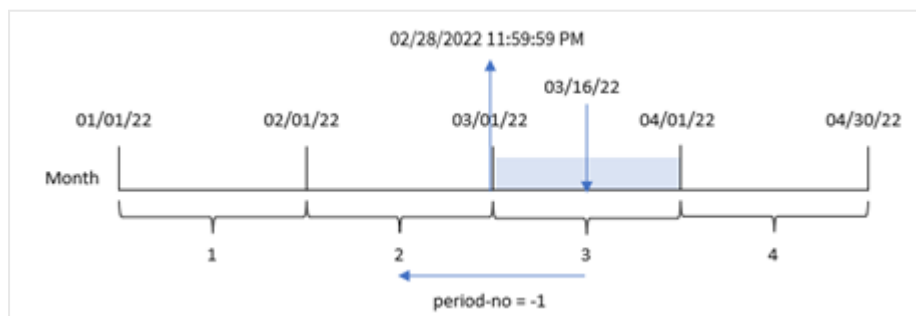
- id
- date
- previous\_month\_end
- previous\_month\_end\_timestamp

Tabella dei risultati

| id   | date       | previous_month_end | previous_month_end_timestamp |
|------|------------|--------------------|------------------------------|
| 8188 | 1/7/2022   | 12/31/2021         | 12/31/2021 11:59:59 PM       |
| 8189 | 1/19/2022  | 12/31/2021         | 12/31/2021 11:59:59 PM       |
| 8190 | 2/5/2022   | 01/31/2022         | 1/31/2022 11:59:59 PM        |
| 8191 | 2/28/2022  | 01/31/2022         | 1/31/2022 11:59:59 PM        |
| 8192 | 3/16/2022  | 02/28/2022         | 2/28/2022 11:59:59 PM        |
| 8193 | 4/1/2022   | 03/31/2022         | 3/31/2022 11:59:59 PM        |
| 8194 | 5/7/2022   | 04/30/2022         | 4/30/2022 11:59:59 PM        |
| 8195 | 5/16/2022  | 04/30/2022         | 4/30/2022 11:59:59 PM        |
| 8196 | 6/15/2022  | 05/31/2022         | 5/31/2022 11:59:59 PM        |
| 8197 | 6/26/2022  | 05/31/2022         | 5/31/2022 11:59:59 PM        |
| 8198 | 7/9/2022   | 06/30/2022         | 6/30/2022 11:59:59 PM        |
| 8199 | 7/22/2022  | 06/30/2022         | 6/30/2022 11:59:59 PM        |
| 8200 | 7/23/2022  | 06/30/2022         | 6/30/2022 11:59:59 PM        |
| 8201 | 7/27/2022  | 06/30/2022         | 6/30/2022 11:59:59 PM        |
| 8202 | 8/2/2022   | 07/31/2022         | 7/31/2022 11:59:59 PM        |
| 8203 | 8/8/2022   | 07/31/2022         | 7/31/2022 11:59:59 PM        |
| 8204 | 8/19/2022  | 07/31/2022         | 7/31/2022 11:59:59 PM        |
| 8205 | 9/26/2022  | 08/31/2022         | 8/31/2022 11:59:59 PM        |
| 8206 | 10/14/2022 | 09/30/2022         | 9/30/2022 11:59:59 PM        |
| 8207 | 10/29/2022 | 09/30/2022         | 9/30/2022 11:59:59 PM        |

La funzione `monthend()` identifica innanzitutto il mese in cui avvengono le transazioni, in quanto un valore `period_no` di -1 viene utilizzato come argomento di offset. Si sposta poi un mese prima e identifica il millisecondo finale di tale mese.

*Schema della funzione `monthend` con la variabile `period_no`.*



La transazione 8192 è avvenuta il 16 marzo. La funzione `monthend()` identifica che il mese precedente la transazione è avvenuta in febbraio. Quindi restituisce l'ultimo millisecondo di quel mese, il 28 febbraio alle 23:59:59.

### Esempio 3 - Esempio di grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.

In questo esempio, il set di dati è invariato e viene caricato nell'app. Per questa attività, è necessario creare un calcolo che restituisce data e ora per la fine del mese in cui sono avvenute le transazioni come misura in un grafico dell'app.

#### Script di caricamento

```
Transactions:
Load
*
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- id

Per calcolare la data di fine del mese in cui avviene una transazione, creare le seguenti misure:

- `=monthend(date)`
- `=timestamp(monthend(date))`

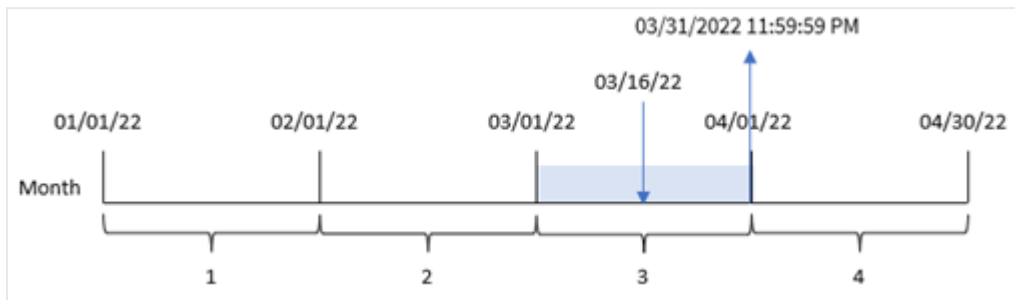
Tabella dei risultati

| id   | date       | =monthend(date) | =timestamp(monthend(date)) |
|------|------------|-----------------|----------------------------|
| 8188 | 10/14/2022 | 10/31/2022      | 10/31/2022 11:59:59 PM     |
| 8189 | 10/29/2022 | 10/31/2022      | 10/31/2022 11:59:59 PM     |
| 8190 | 9/26/2022  | 09/30/2022      | 9/30/2022 11:59:59 PM      |
| 8191 | 8/2/2022   | 08/31/2022      | 8/31/2022 11:59:59 PM      |
| 8192 | 8/8/2022   | 08/31/2022      | 8/31/2022 11:59:59 PM      |
| 8193 | 8/19/2022  | 08/31/2022      | 8/31/2022 11:59:59 PM      |
| 8194 | 7/9/2022   | 07/31/2022      | 7/31/2022 11:59:59 PM      |
| 8195 | 7/22/2022  | 07/31/2022      | 7/31/2022 11:59:59 PM      |
| 8196 | 7/23/2022  | 07/31/2022      | 7/31/2022 11:59:59 PM      |
| 8197 | 7/27/2022  | 07/31/2022      | 7/31/2022 11:59:59 PM      |
| 8198 | 6/15/2022  | 06/30/2022      | 6/30/2022 11:59:59 PM      |
| 8199 | 6/26/2022  | 06/30/2022      | 6/30/2022 11:59:59 PM      |
| 8200 | 5/7/2022   | 05/31/2022      | 5/31/2022 11:59:59 PM      |
| 8201 | 5/16/2022  | 05/31/2022      | 5/31/2022 11:59:59 PM      |
| 8202 | 4/1/2022   | 04/30/2022      | 4/30/2022 11:59:59 PM      |
| 8203 | 3/16/2022  | 03/31/2022      | 3/31/2022 11:59:59 PM      |
| 8204 | 2/5/2022   | 02/28/2022      | 2/28/2022 11:59:59 PM      |
| 8205 | 2/28/2022  | 02/28/2022      | 2/28/2022 11:59:59 PM      |
| 8206 | 1/7/2022   | 01/31/2022      | 1/31/2022 11:59:59 PM      |
| 8207 | 1/19/2022  | 01/31/2022      | 1/31/2022 11:59:59 PM      |

La misura 'end\_of\_month' viene creata nel grafico mediante l'utilizzo della funzione `monthend()` e trasferendo il campo data come argomento della funzione.

La funzione `monthend()` identifica in quale mese cade il valore della data, restituendo un timestamp per l'ultimo millisecondo di quel mese.

Schema della funzione `monthend` con la variabile `period_no`.



La transazione 8192 è avvenuta il 16 marzo. La funzione `monthend()` restituisce l'ultimo millisecondo di quel mese, ovvero il 31 marzo alle 23:59:59.

### Esempio 4 - Scenario

Script di caricamento e risultati

#### Panoramica

In questo esempio, un set di dati è caricato in una tabella denominata "Employee\_Expenses". La tabella contiene i seguenti campi:

- ID dipendenti
- Nomi dipendenti
- La media delle richieste di rimborso spese giornaliero di ciascun dipendente.

L'utente finale desidera un grafico che visualizzi, in base all'id e al nome del dipendente, la spesa stimata per il resto del mese.

#### Script di caricamento

```
Employee_Expenses:
Load
*
Inline
[
employee_id,employee_name,avg_daily_claim
182,Mark, $15
183,Deryck, $12.5
184,Dexter, $12.5
185,Sydney,$27
186,Agatha,$18
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- employee\_id
- employee\_name

Per calcolare gli interessi accumulati, creare questa misura:

```
=floor(monthend(today(1),0)-today(1))*avg_daily_claim
```



*Questa misura è dinamica e produrrà risultati diversi a seconda della data di caricamento dei dati.*

Impostare la misura **Formattazione numero** su **Denaro**.

Tabella dei risultati

| employee_id | employee_name | =floor(monthend(today(1),0)-today(1))*avg_daily_claim |
|-------------|---------------|-------------------------------------------------------|
| 182         | Contrassegno  | \$30.00                                               |
| 183         | Deryck        | \$25.00                                               |
| 184         | Dexter        | \$25.00                                               |
| 185         | Sydney        | \$54.00                                               |
| 186         | Agatha        | \$36.00                                               |

La funzione `monthend()` restituisce la data di fine del mese corrente utilizzando come unico argomento la data odierna. L'espressione restituisce il numero di giorni rimanenti nel mese sottraendo la data odierna dalla data di fine mese.

Questo valore viene quindi moltiplicato per la media delle richieste di rimborso spese giornaliere di ciascun dipendente per calcolare il valore stimato delle richieste che ogni dipendente dovrebbe presentare durante il periodo rimanente del mese.

### monthname

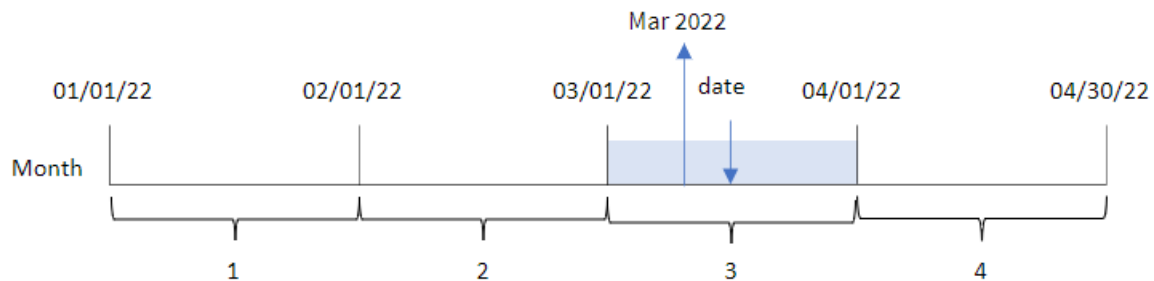
Questa funzione restituisce un valore di visualizzazione che mostra il mese (formattato in base alla variabile di script **MonthNames**) e l'anno il cui valore numerico sottostante corrisponde a un indicatore temporale recante il primo millisecondo del primo giorno del mese.

**Sintassi:**

```
MonthName (date[, period_no])
```

**Tipo di dati restituiti:** duale

*Schema della funzione monthname*



Argomenti

| Argomento        | Descrizione                                                                                                                                                                                                                                 |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>date</b>      | La data o la data e ora da valutare.                                                                                                                                                                                                        |
| <b>period_no</b> | <b>period_no</b> è un numero intero che, se corrisponde a 0 o viene ommesso, indica il mese contenente <b>date</b> . I valori negativi di <b>period_no</b> indicano i mesi precedenti, mentre i valori positivi indicano i mesi successivi. |

Esempi di funzioni

| Esempio                                  | Risultato            |
|------------------------------------------|----------------------|
| <code>monthname('10/19/2013')</code>     | Restituisce Oct 2013 |
| <code>monthname('10/19/2013', -1)</code> | Restituisce Sep 2013 |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - Esempio di base

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2022, caricato in una tabella denominata 'Transactions'.
- Il campo della data fornito nel formato della variabile di sistema `DateFormat` (MM/GG/AAAA).
- La creazione di un campo, `transaction_month`, che restituisce il mese in cui sono state effettuate le transazioni.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
SET MonthNames='Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec';
```

```
Transactions:
```

```
    Load
        *,
        monthname(date) as transaction_month
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
```

```
8189,1/19/2022,37.23
```

```
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
```

```
8192,3/16/2022,53.80
```

```
8193,4/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/16/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
```

```
8200,7/23/2022,82.77
```

```
8201,7/27/2022,69.98
```

```
8202,8/2/2022,76.11
```

```
8203,8/8/2022,25.12
```

```
8204,8/19/2022,46.23
```

```
8205,9/26/2022,84.21
```

```
8206,10/14/2022,96.24
```

```
8207,10/29/2022,67.67
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

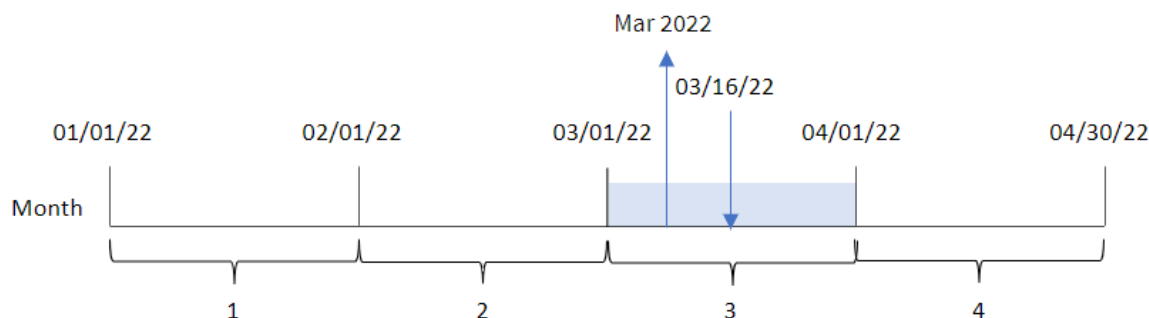
- date
- transaction\_month

Tabella dei risultati

| data       | mese_transazione |
|------------|------------------|
| 1/7/2022   | Gen 2022         |
| 1/19/2022  | Gen 2022         |
| 2/5/2022   | Feb 2022         |
| 2/28/2022  | Feb 2022         |
| 3/16/2022  | Mar 2022         |
| 4/1/2022   | Apr 2022         |
| 5/7/2022   | Mag 2022         |
| 5/16/2022  | Mag 2022         |
| 6/15/2022  | Giu 2022         |
| 6/26/2022  | Giu 2022         |
| 7/9/2022   | Lug 2022         |
| 7/22/2022  | Lug 2022         |
| 7/23/2022  | Lug 2022         |
| 7/27/2022  | Lug 2022         |
| 8/2/2022   | Ago 2022         |
| 8/8/2022   | Ago 2022         |
| 8/19/2022  | Ago 2022         |
| 9/26/2022  | Set 2022         |
| 10/14/2022 | Ott 2022         |
| 10/29/2022 | Ott 2022         |

Il campo transaction\_month viene creato nell'istruzione di caricamento precedente mediante l'uso della funzione monthname() e trasferendo il campo date come argomento della funzione.

*Schema della funzione monthname, esempio di base*



La funzione `monthname()` identifica la transazione 8192 avvenuta nel marzo 2022 e restituisce questo valore utilizzando la variabile di sistema `MonthNames`.

### Esempio 2 - period\_no

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati inline e lo stesso scenario del primo esempio.
- La creazione di un campo, `transaction_previous_month`, che restituisce data e ora per la fine del mese prima che fosse effettuata la transazione.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';  
SET MonthNames='Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec';
```

Transactions:

```
Load  
    *,  
    monthname(date,-1) as transaction_previous_month  
;
```

Load

\*

Inline

[

id,date,amount

8188,1/7/2022,17.17

8189,1/19/2022,37.23

8190,2/28/2022,88.27

8191,2/5/2022,57.42

8192,3/16/2022,53.80

8193,4/1/2022,82.06

8194,5/7/2022,40.39

```
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- transaction\_previous\_month

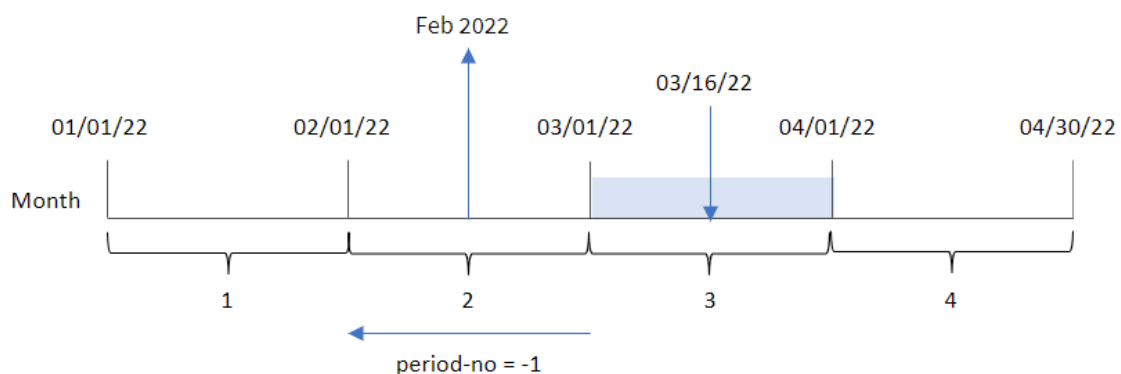
Tabella dei risultati

| data      | transazione_mese_precedente |
|-----------|-----------------------------|
| 1/7/2022  | Dic 2021                    |
| 1/19/2022 | Dic 2021                    |
| 2/5/2022  | Gen 2022                    |
| 2/28/2022 | Gen 2022                    |
| 3/16/2022 | Feb 2022                    |
| 4/1/2022  | Mar 2022                    |
| 5/7/2022  | Apr 2022                    |
| 5/16/2022 | Apr 2022                    |
| 6/15/2022 | Mag 2022                    |
| 6/26/2022 | Mag 2022                    |
| 7/9/2022  | Giu 2022                    |
| 7/22/2022 | Giu 2022                    |
| 7/23/2022 | Giu 2022                    |
| 7/27/2022 | Giu 2022                    |
| 8/2/2022  | Lug 2022                    |
| 8/8/2022  | Lug 2022                    |
| 8/19/2022 | Lug 2022                    |

| data       | transazione_mese_precedente |
|------------|-----------------------------|
| 9/26/2022  | Ago 2022                    |
| 10/14/2022 | Set 2022                    |
| 10/29/2022 | Set 2022                    |

In questo caso, poiché il valore `period_no` di -1 è stato utilizzato come argomento `offset` nella funzione `monthname()`, la funzione per prima cosa identifica il mese in cui avvengono le transazioni. Quindi, passa al mese anteriore e restituisce il nome del mese e l'anno.

*Schema della funzione `monthname`, esempio di `period_no`*



La transazione 8192 è avvenuta il 16 marzo. La funzione `monthname()` identifica che il mese precedente all'esecuzione della transazione è febbraio, quindi restituisce il mese nel formato della variabile di sistema `MonthNames`, insieme all'anno 2022.

### Esempio 3 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor di caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento utilizza lo stesso set di dati inline e lo stesso scenario del primo esempio. Tuttavia, in questo esempio, il set di dati invariato viene caricato nell'applicazione. Il calcolo che restituisce data e ora per la fine del mese in cui sono avvenute le transazioni viene creato come misura in un oggetto grafico dell'applicazione.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
SET MonthNames='Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec';
```

```
Transactions:
Load
```

\*

Inline

[

id,date,amount

8188,1/7/2022,17.17

8189,1/19/2022,37.23

8190,2/28/2022,88.27

8191,2/5/2022,57.42

8192,3/16/2022,53.80

8193,4/1/2022,82.06

8194,5/7/2022,40.39

8195,5/16/2022,87.21

8196,6/15/2022,95.93

8197,6/26/2022,45.89

8198,7/9/2022,36.23

8199,7/22/2022,25.66

8200,7/23/2022,82.77

8201,7/27/2022,69.98

8202,8/2/2022,76.11

8203,8/8/2022,25.12

8204,8/19/2022,46.23

8205,9/26/2022,84.21

8206,10/14/2022,96.24

8207,10/29/2022,67.67

];

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:date.

Creare la seguente misura:

=monthname(date)

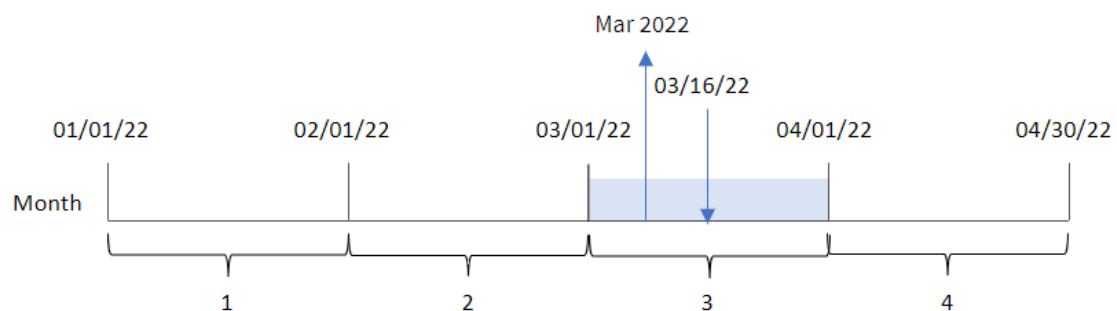
Tabella dei risultati

| data      | =monthname(date) |
|-----------|------------------|
| 1/7/2022  | Gen 2022         |
| 1/19/2022 | Gen 2022         |
| 2/5/2022  | Feb 2022         |
| 2/28/2022 | Feb 2022         |
| 3/16/2022 | Mar 2022         |
| 4/1/2022  | Apr 2022         |
| 5/7/2022  | Mag 2022         |
| 5/16/2022 | Mag 2022         |
| 6/15/2022 | Giu 2022         |

| data       | =monthname(date) |
|------------|------------------|
| 6/26/2022  | Giu 2022         |
| 7/9/2022   | Lug 2022         |
| 7/22/2022  | Lug 2022         |
| 7/23/2022  | Lug 2022         |
| 7/27/2022  | Lug 2022         |
| 8/2/2022   | Ago 2022         |
| 8/8/2022   | Ago 2022         |
| 8/19/2022  | Ago 2022         |
| 9/26/2022  | Set 2022         |
| 10/14/2022 | Ott 2022         |
| 10/29/2022 | Ott 2022         |

La misura `month_name` viene creata nell'oggetto grafico mediante l'utilizzo della funzione `monthname()` e trasferendo il campo `date` come argomento della funzione.

*Schema della funzione `monthname`, esempio di oggetto grafico*



La funzione `monthname()` identifica la transazione 8192 avvenuta nel marzo 2022 e restituisce questo valore utilizzando la variabile di sistema `MonthNames`.

### monthsend

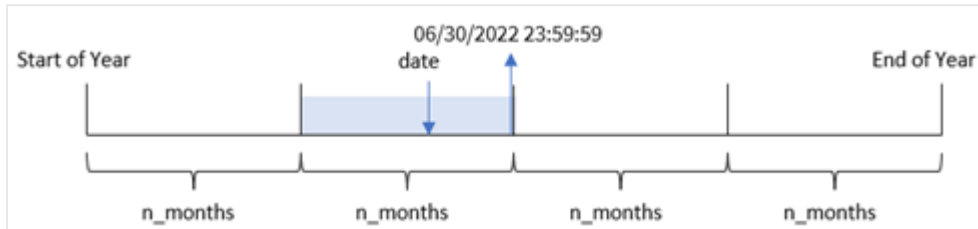
Questa funzione restituisce un valore corrispondente a un timestamp recante l'ultimo millisecondo del mese, del bimestre, del trimestre, del quadrimestre o del semestre contenente una data di base. È inoltre possibile individuare il timestamp per la fine di un periodo di tempo precedente o successivo. Il formato di output predefinito è il formato `DateFormat` impostato nello script.

**Sintassi:**

```
MonthsEnd(n_months, date[, period_no [, first_month_of_year]])
```

Tipo di dati restituiti: duale

Schema della funzione *monthsend*.



#### Argomenti

| Argomento                  | Descrizione                                                                                                                                                                                                                                                                                                                             |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>n_months</b>            | Il numero di mesi che definisce il periodo. Un numero intero o un'espressione la cui risoluzione è un numero intero corrispondente a: 1 (equivalente alla funzione <i>inmonth</i> ()), 2 (bimestre), 3 (equivalente alla funzione <i>inquarter</i> ()), 4 (quadrimestre) o 6 (semestre).                                                |
| <b>date</b>                | La data o la data e ora da valutare.                                                                                                                                                                                                                                                                                                    |
| <b>period_no</b>           | Il periodo può essere differito mediante <b>period_no</b> , un numero intero, o un'espressione la cui risoluzione è un numero intero, in cui il valore 0 indica il periodo che contiene <b>base_date</b> . I valori negativi di <b>period_no</b> indicano i periodi precedenti, mentre i valori positivi indicano i periodi successivi. |
| <b>first_month_of_year</b> | Se si intende utilizzare anni (fiscali) che non iniziano a gennaio, indicare un valore compreso tra 2 e 12 in <b>first_month_of_year</b> .                                                                                                                                                                                              |

La funzione *monthsend()* divide l'anno in segmenti in base all'argomento *n\_months* fornito. Quindi valuta in quale segmento rientra ogni data fornita e restituisce l'ultimo millisecondo, in formato data, di quel segmento. La funzione può restituire il timestamp finale dei segmenti precedenti o successivi e ridefinire il primo mese dell'anno.

I seguenti segmenti dell'anno sono disponibili nella funzione come argomenti *n\_month*.

#### Argomenti *n\_month*

| Periodo      | Numero di mesi |
|--------------|----------------|
| mese         | 1              |
| bimestre     | 2              |
| trimestre    | 3              |
| quattro mesi | 4              |
| semestre     | 6              |

### Casi di utilizzo

La funzione `monthsend()` viene utilizzata come parte di un'espressione quando l'utente desidera che il calcolo utilizzi la frazione di mese trascorsa fino a quel momento. L'utente ha la possibilità, tramite una variabile, di selezionare il periodo di sua scelta. Ad esempio, il `monthsend()` può fornire una variabile di input che consenta all'utente di calcolare il totale degli interessi non ancora maturati durante il mese, il trimestre o il semestre.

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

Esempi di funzioni

| Esempio                                       | Risultato                                                                   |
|-----------------------------------------------|-----------------------------------------------------------------------------|
| <code>monthsend(4, '07/19/2013')</code>       | Restituisce 08/31/2013.                                                     |
| <code>monthsend(4, '10/19/2013', -1)</code>   | Restituisce 08/31/2013.                                                     |
| <code>monthsend(4, '10/19/2013', 0, 2)</code> | Restituisce 01/31/2014.<br>Poiché l'inizio dell'anno corrisponde al mese 2. |

### Esempio 1 - Esempio di base

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2022 viene caricato in una tabella denominata 'Transactions'.
- Un campo data fornito nel formato (MM/DD/YYYY) della variabile di sistema `DateFormat`.
- Un'istruzione di caricamento precedente contenente:

- La funzione `monthsend` impostata come campo, `'bi_monthly_end'`. Questo raggruppa le transazioni in segmenti bimestrali.
- La funzione `timestamp` che restituisce il timestamp iniziale del segmento per ogni transazione.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
,  
monthsend(2,date) as bi_monthly_end,
```

```
timestamp(monthsend(2,date)) as bi_monthly_end_timestamp
```

```
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
```

```
8189,1/19/2022,37.23
```

```
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
```

```
8192,3/16/2022,53.80
```

```
8193,4/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/22/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
```

```
8200,7/23/2022,82.77
```

```
8201,7/27/2022,69.98
```

```
8202,8/2/2022,76.11
```

```
8203,8/8/2022,25.12
```

```
8204,8/19/2022,46.23
```

```
8205,9/26/2022,84.21
```

```
8206,10/14/2022,96.24
```

```
8207,10/29/2022,67.67
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

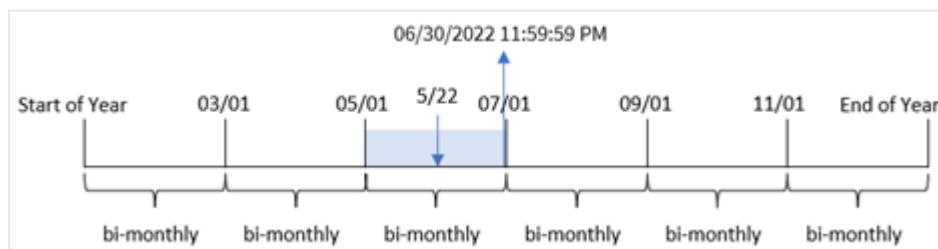
- `id`
- `date`
- `bi_monthly_end`
- `bi_monthly_end_timestamp`

Tabella dei risultati

| id   | date       | bi_monthly_end | bi_monthly_end_timestamp |
|------|------------|----------------|--------------------------|
| 8188 | 1/7/2022   | 02/28/2022     | 2/28/2022 11:59:59 PM    |
| 8189 | 1/19/2022  | 02/28/2022     | 2/28/2022 11:59:59 PM    |
| 8190 | 2/5/2022   | 02/28/2022     | 2/28/2022 11:59:59 PM    |
| 8191 | 2/28/2022  | 02/28/2022     | 2/28/2022 11:59:59 PM    |
| 8192 | 3/16/2022  | 04/30/2022     | 4/30/2022 11:59:59 PM    |
| 8193 | 4/1/2022   | 04/30/2022     | 4/30/2022 11:59:59 PM    |
| 8194 | 5/7/2022   | 06/30/2022     | 6/30/2022 11:59:59 PM    |
| 8195 | 5/22/2022  | 06/30/2022     | 6/30/2022 11:59:59 PM    |
| 8196 | 6/15/2022  | 06/30/2022     | 6/30/2022 11:59:59 PM    |
| 8197 | 6/26/2022  | 06/30/2022     | 6/30/2022 11:59:59 PM    |
| 8198 | 7/9/2022   | 08/31/2022     | 8/31/2022 11:59:59 PM    |
| 8199 | 7/22/2022  | 08/31/2022     | 8/31/2022 11:59:59 PM    |
| 8200 | 7/23/2022  | 08/31/2022     | 8/31/2022 11:59:59 PM    |
| 8201 | 7/27/2022  | 08/31/2022     | 8/31/2022 11:59:59 PM    |
| 8202 | 8/2/2022   | 08/31/2022     | 8/31/2022 11:59:59 PM    |
| 8203 | 8/8/2022   | 08/31/2022     | 8/31/2022 11:59:59 PM    |
| 8204 | 8/19/2022  | 08/31/2022     | 8/31/2022 11:59:59 PM    |
| 8205 | 9/26/2022  | 10/31/2022     | 10/31/2022 11:59:59 PM   |
| 8206 | 10/14/2022 | 10/31/2022     | 10/31/2022 11:59:59 PM   |
| 8207 | 10/29/2022 | 10/31/2022     | 10/31/2022 11:59:59 PM   |

Il campo 'bi\_monthly\_end' viene creato nell'istruzione LOAD precedente utilizzando la funzione `monthsend()`. Il primo argomento fornito è 2, dividendo l'anno in segmenti bimestrali. Il secondo argomento identifica il campo da valutare.

*Schema della funzione monthsend con segmenti bimestrali.*



La transazione 8195 avviene il 22 maggio. La funzione `monthsend()` divide inizialmente l'anno in segmenti bimestrali. La transazione 8195 rientra nel segmento tra il 1 maggio e il 30 giugno. Di conseguenza, la funzione restituisce l'ultimo millisecondo di questo segmento, il 30/06/2022 alle 23:59:59.

### Esempio 2 - period\_no

Script di caricamento e risultati

#### Panoramica

Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.

In questo esempio, il compito è quello di creare un campo, 'prev\_bi\_monthly\_end', che restituisca il primo millisecondo del segmento bimestrale prima che avvenga la transazione.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
```

```
    *,
```

```
    monthsend(2,date,-1) as prev_bi_monthly_end,
```

```
    timestamp(monthsend(2,date,-1)) as prev_bi_monthly_end_timestamp
```

```
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
```

```
8189,1/19/2022,37.23
```

```
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
```

```
8192,3/16/2022,53.80
```

```
8193,4/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/22/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
```

```
8200,7/23/2022,82.77
```

```
8201,7/27/2022,69.98
```

```
8202,8/2/2022,76.11
```

```
8203,8/8/2022,25.12
```

```
8204,8/19/2022,46.23
```

```
8205,9/26/2022,84.21
```

```
8206,10/14/2022,96.24
```

```
8207,10/29/2022,67.67
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

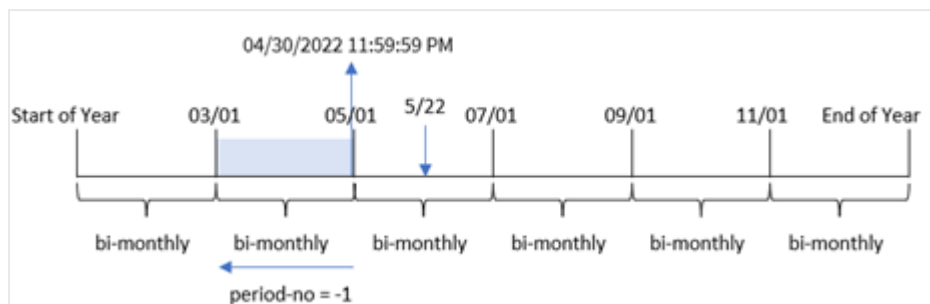
- id
- date
- prev\_bi\_monthly\_end
- prev\_bi\_monthly\_end\_timestamp

Tabella dei risultati

| id   | date       | prev_bi_monthly_end | prev_bi_monthly_end_timestamp |
|------|------------|---------------------|-------------------------------|
| 8188 | 1/7/2022   | 12/31/2021          | 12/31/2021 11:59:59 PM        |
| 8189 | 1/19/2022  | 12/31/2021          | 12/31/2021 11:59:59 PM        |
| 8190 | 2/5/2022   | 12/31/2021          | 12/31/2021 11:59:59 PM        |
| 8191 | 2/28/2022  | 12/31/2021          | 12/31/2021 11:59:59 PM        |
| 8192 | 3/16/2022  | 02/28/2022          | 2/28/2022 11:59:59 PM         |
| 8193 | 4/1/2022   | 02/28/2022          | 2/28/2022 11:59:59 PM         |
| 8194 | 5/7/2022   | 04/30/2022          | 4/30/2022 11:59:59 PM         |
| 8195 | 5/22/2022  | 04/30/2022          | 4/30/2022 11:59:59 PM         |
| 8196 | 6/15/2022  | 04/30/2022          | 4/30/2022 11:59:59 PM         |
| 8197 | 6/26/2022  | 04/30/2022          | 4/30/2022 11:59:59 PM         |
| 8198 | 7/9/2022   | 06/30/2022          | 6/30/2022 11:59:59 PM         |
| 8199 | 7/22/2022  | 06/30/2022          | 6/30/2022 11:59:59 PM         |
| 8200 | 7/23/2022  | 06/30/2022          | 6/30/2022 11:59:59 PM         |
| 8201 | 7/27/2022  | 06/30/2022          | 6/30/2022 11:59:59 PM         |
| 8202 | 8/2/2022   | 06/30/2022          | 6/30/2022 11:59:59 PM         |
| 8203 | 8/8/2022   | 06/30/2022          | 6/30/2022 11:59:59 PM         |
| 8204 | 8/19/2022  | 06/30/2022          | 6/30/2022 11:59:59 PM         |
| 8205 | 9/26/2022  | 08/31/2022          | 8/31/2022 11:59:59 PM         |
| 8206 | 10/14/2022 | 08/31/2022          | 8/31/2022 11:59:59 PM         |
| 8207 | 10/29/2022 | 08/31/2022          | 8/31/2022 11:59:59 PM         |

Utilizzando -1 come argomento `period_no` della funzione `monthsend()`, dopo aver inizialmente diviso un anno in segmenti bimestrali, la funzione restituisce l'ultimo millisecondo del segmento bimestrale precedente al momento in cui avviene una transazione.

Schema della funzione *monthsend* che restituisce il segmento bimestrale precedente.



La transazione 8195 è stata effettuata nel segmento tra maggio e giugno. Di conseguenza, il segmento bimestrale precedente era compreso tra il 1° marzo e il 30 aprile e quindi la funzione restituisce l'ultimo millisecondo di questo segmento, il 30/04/2022 alle 23:59:59.

### Esempio 3 - first\_month\_of\_year

Script di caricamento e risultati

#### Panoramica

Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.

In questo esempio, il criterio organizzativo prevede che aprile sia il primo mese dell'anno finanziario.

Creare un campo, *bi\_monthly\_end*, che raggruppa le transazioni in segmenti bimestrali e restituisce il timestamp dell'ultimo millisecondo di quel segmento per ogni transazione.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
monthsend(2,date,0,4) as bi_monthly_end,
```

```
timestamp(monthsend(2,date,0,4)) as bi_monthly_end_timestamp
```

```
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
```

```
8189,1/19/2022,37.23
```

```
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
```

```
8192,3/16/2022,53.80
```

```
8193,4/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/22/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- date
- bi\_monthly\_end
- bi\_monthly\_end\_timestamp

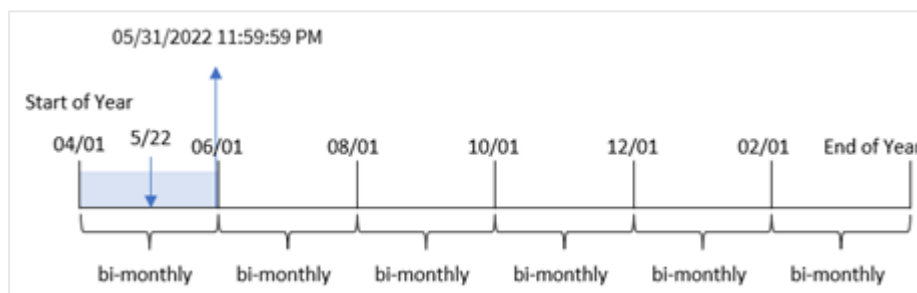
Tabella dei risultati

| id   | date      | bi_monthly_end | bi_monthly_end_timestamp |
|------|-----------|----------------|--------------------------|
| 8188 | 1/7/2022  | 01/31/2022     | 1/31/2022 11:59:59 PM    |
| 8189 | 1/19/2022 | 01/31/2022     | 1/31/2022 11:59:59 PM    |
| 8190 | 2/5/2022  | 03/31/2022     | 3/31/2022 11:59:59 PM    |
| 8191 | 2/28/2022 | 03/31/2022     | 3/31/2022 11:59:59 PM    |
| 8192 | 3/16/2022 | 03/31/2022     | 3/31/2022 11:59:59 PM    |
| 8193 | 4/1/2022  | 05/31/2022     | 5/31/2022 11:59:59 PM    |
| 8194 | 5/7/2022  | 05/31/2022     | 5/31/2022 11:59:59 PM    |
| 8195 | 5/22/2022 | 05/31/2022     | 5/31/2022 11:59:59 PM    |
| 8196 | 6/15/2022 | 07/31/2022     | 7/31/2022 11:59:59 PM    |
| 8197 | 6/26/2022 | 07/31/2022     | 7/31/2022 11:59:59 PM    |
| 8198 | 7/9/2022  | 07/31/2022     | 7/31/2022 11:59:59 PM    |
| 8199 | 7/22/2022 | 07/31/2022     | 7/31/2022 11:59:59 PM    |
| 8200 | 7/23/2022 | 07/31/2022     | 7/31/2022 11:59:59 PM    |
| 8201 | 7/27/2022 | 07/31/2022     | 7/31/2022 11:59:59 PM    |
| 8202 | 8/2/2022  | 09/30/2022     | 9/30/2022 11:59:59 PM    |
| 8203 | 8/8/2022  | 09/30/2022     | 9/30/2022 11:59:59 PM    |

| id   | date       | bi_monthly_end | bi_monthly_end_timestamp |
|------|------------|----------------|--------------------------|
| 8204 | 8/19/2022  | 09/30/2022     | 9/30/2022 11:59:59 PM    |
| 8205 | 9/26/2022  | 09/30/2022     | 9/30/2022 11:59:59 PM    |
| 8206 | 10/14/2022 | 11/30/2022     | 11/30/2022 11:59:59 PM   |
| 8207 | 10/29/2022 | 11/30/2022     | 11/30/2022 11:59:59 PM   |

Utilizzando 4 come argomento `first_month_of_year` nella funzione `monthsend()`, la funzione inizia l'anno il 1 aprile, quindi divide l'anno in segmenti bimestrali: Apr-Mag, Giu-Lug, Ago-Set, Ott-Nov, Dic-Gen, Feb-Mar.

*Schema della funzione `monthsend` con il primo mese dell'anno impostato come aprile*



La transazione 8195 è avvenuta il 22 maggio e rientra nel segmento tra il 1 aprile e il 31 maggio. Di conseguenza, la funzione restituisce l'ultimo millisecondo di questo segmento, il 31/05/2022 alle 23:59:59.

### Esempio 4 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio. Tuttavia, in questo esempio, il set di dati è invariato e viene caricato nell'app.

In questo esempio, si tratta di creare un calcolo che raggruppi le transazioni in segmenti bimestrali e restituisca il timestamp dell'ultimo millisecondo del segmento per ciascuna transazione come misura in un oggetto grafico di un'app.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,2/19/2022,37.23
```

```

8189, 3/7/2022, 17.17
8190, 3/30/2022, 88.27
8191, 4/5/2022, 57.42
8192, 4/16/2022, 53.80
8193, 5/1/2022, 82.06
8194, 5/7/2022, 40.39
8195, 5/22/2022, 87.21
8196, 6/15/2022, 95.93
8197, 6/26/2022, 45.89
8198, 7/9/2022, 36.23
8199, 7/22/2022, 25.66
8200, 7/23/2022, 82.77
8201, 7/27/2022, 69.98
8202, 8/2/2022, 76.11
8203, 8/8/2022, 25.12
8204, 8/19/2022, 46.23
8205, 9/26/2022, 84.21
8206, 10/14/2022, 96.24
8207, 10/29/2022, 67.67
];

```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

date

Per ottenere il timestamp dell'ultimo millisecondo del segmento bimestrale in cui è avvenuta la transazione, creare le seguenti misure:

- =monthsEnd(2,date)
- =timestamp(monthsend(2,date))

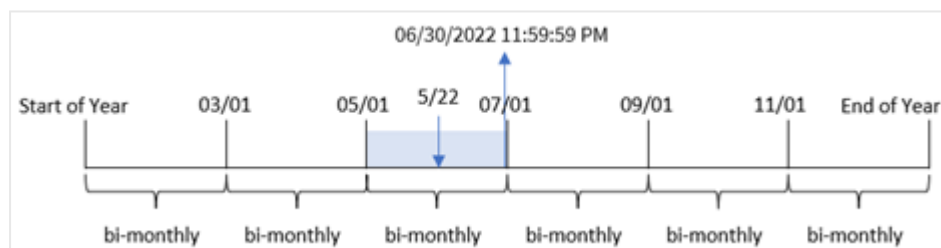
Tabella dei risultati

| id   | date      | =monthsend(2,date) | =timestamp(monthsend(2,date)) |
|------|-----------|--------------------|-------------------------------|
| 8188 | 1/7/2022  | 02/28/2022         | 2/28/2022 11:59:59 PM         |
| 8189 | 1/19/2022 | 02/28/2022         | 2/28/2022 11:59:59 PM         |
| 8190 | 2/5/2022  | 02/28/2022         | 2/28/2022 11:59:59 PM         |
| 8191 | 2/28/2022 | 02/28/2022         | 2/28/2022 11:59:59 PM         |
| 8192 | 3/16/2022 | 04/30/2022         | 4/30/2022 11:59:59 PM         |
| 8193 | 4/1/2022  | 04/30/2022         | 4/30/2022 11:59:59 PM         |
| 8194 | 5/7/2022  | 06/30/2022         | 6/30/2022 11:59:59 PM         |
| 8195 | 5/22/2022 | 06/30/2022         | 6/30/2022 11:59:59 PM         |
| 8196 | 6/15/2022 | 06/30/2022         | 6/30/2022 11:59:59 PM         |
| 8197 | 6/26/2022 | 06/30/2022         | 6/30/2022 11:59:59 PM         |

| id   | date       | =monthsend(2,date) | =timestamp(monthsend(2,date)) |
|------|------------|--------------------|-------------------------------|
| 8198 | 7/9/2022   | 08/31/2022         | 8/31/2022 11:59:59 PM         |
| 8199 | 7/22/2022  | 08/31/2022         | 8/31/2022 11:59:59 PM         |
| 8200 | 7/23/2022  | 08/31/2022         | 8/31/2022 11:59:59 PM         |
| 8201 | 7/27/2022  | 08/31/2022         | 8/31/2022 11:59:59 PM         |
| 8202 | 8/2/2022   | 08/31/2022         | 8/31/2022 11:59:59 PM         |
| 8203 | 8/8/2022   | 08/31/2022         | 8/31/2022 11:59:59 PM         |
| 8204 | 8/19/2022  | 08/31/2022         | 8/31/2022 11:59:59 PM         |
| 8205 | 9/26/2022  | 10/31/2022         | 10/31/2022 11:59:59 PM        |
| 8206 | 10/14/2022 | 10/31/2022         | 10/31/2022 11:59:59 PM        |
| 8207 | 10/29/2022 | 10/31/2022         | 10/31/2022 11:59:59 PM        |

Il campo 'bi\_monthly\_end' viene creato come misura nell'oggetto grafico mediante l'utilizzo della funzione monthsend(). Il primo argomento fornito è 2, che divide l'anno in segmenti bimestrali. Il secondo argomento identifica il campo da valutare.

*Schema della funzione monthsend con segmenti bimestrali.*



La transazione 8195 avviene il 22 maggio. La funzione monthsend() divide inizialmente l'anno in segmenti bimestrali. La transazione 8195 rientra nel segmento tra il 1 maggio e il 30 giugno. Di conseguenza, la funzione restituisce il primo millisecondo di questo segmento, il 30/06/2022 alle 23:59:59.

### Esempio 5 - Scenario

Script di caricamento e risultati

#### Panoramica

Aprire l'editor di caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

In questo esempio, un set di dati è caricato in una tabella denominata "Employee\_Expenses". La tabella contiene i seguenti campi:

- ID dipendenti
- Nomi dipendenti

- La media delle richieste di rimborso spese giornaliero di ciascun dipendente.

L'utente finale desidera un grafico che visualizzi, in base all'id e al nome del dipendente, la spesa stimata per il resto di un periodo di propria scelta. L'esercizio finanziario inizia a gennaio.

### Script di caricamento

```
SET vPeriod = 1;

Employee_Expenses:
Load
*
Inline
[
employee_id,employee_name,avg_daily_claim
182,Mark, $15
183,Deryck, $12.5
184,Dexter, $12.5
185,Sydney,$27
186,Agatha,$18
];
```

### Risultati

Caricare i dati e aprire un nuovo foglio.

All'inizio dello script di caricamento, viene creata una variabile `vPeriod`, legata al controllo dell'input variabile.

Procedere come segue:

1. Nel pannello delle risorse, fare clic su **Oggetti personalizzati**.
2. Selezionare **Qlik Dashboard bundle** e creare un oggetto **Input variabile**.
3. Immettere un titolo per l'oggetto grafico.
4. In **Variabile**, selezionare **vPeriod** come nome e impostare l'oggetto in modo che venga visualizzato come **Elenco a discesa**.
5. Sotto **Valori**, fare clic sui valori **Dinamici**. Inserire quanto segue:  
=`'1~month|2~bi-month|3~quarter|4~tertrial|6~half-year'`.

Creare una nuova tabella con tali campi come dimensioni:

- `employee_id`
- `employee_name`

Per calcolare gli interessi accumulati, creare questa misura:

```
=floor(monthsend($(vPeriod),today(1))-today(1))*avg_daily_claim
```



*Questa misura è dinamica e produrrà risultati diversi a seconda della data di caricamento dei dati.*

Impostare la misura **Formattazione numero** su **Denaro**.

Tabella dei risultati

| employee_id | employee_name | =floor(monthsend(\$(vPeriod),today(1))-today(1))*avg_daily_claim |
|-------------|---------------|------------------------------------------------------------------|
| 182         | Contrassegno  | \$1410.00                                                        |
| 183         | Deryck        | \$1175.00                                                        |
| 184         | Dexter        | \$1175.00                                                        |
| 185         | Sydney        | \$2538.00                                                        |
| 186         | Agatha        | \$1692.00                                                        |

La funzione `monthsend()` utilizza l'input dell'utente come primo argomento e la data odierna come secondo argomento. Restituisce la data di fine del periodo di tempo selezionato dall'utente. Quindi, l'espressione restituisce il numero di giorni che rimangono nel periodo di tempo selezionato, sottraendo la data odierna da questa data finale.

Questo valore viene quindi moltiplicato per la media delle richieste di rimborso spese giornaliere di ciascun dipendente per calcolare il valore stimato delle richieste che ogni dipendente dovrebbe presentare durante i giorni rimanenti di questo periodo.

## monthsname

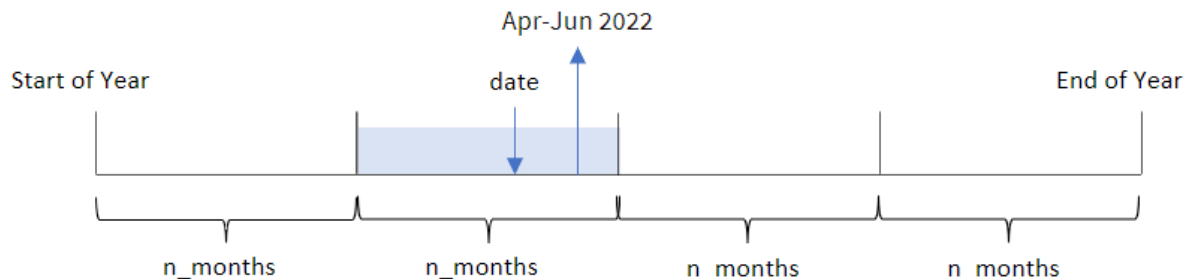
Questa funzione restituisce un valore di visualizzazione che rappresenta l'intervallo dei mesi del periodo (formattati in base alla variabile di script **MonthNames**) e l'anno. Il valore numerico sottostante corrisponde a un indicatore temporale recante il primo millisecondo del mese, del bimestre, del trimestre, del quadrimestre o del semestre contenente una data di base.

### Sintassi:

```
MonthsName(n_months, date[, period_no[, first_month_of_year]])
```

Tipo di dati restituiti: duale

Schema della funzione monthname



La funzione `monthname()` divide l'anno in segmenti in base all'argomento `n_months` fornito. Quindi, valuta il segmento a cui appartiene ogni valore `date` fornito e restituisce i nomi del mese di inizio e di fine di quel segmento, nonché l'anno. La funzione offre anche la possibilità di restituire questi limiti dai segmenti precedenti o successivi, oltre a ridefinire quale è il primo mese dell'anno.

I seguenti segmenti dell'anno sono disponibili nella funzione come argomenti di `n_month` :

Possibili argomenti di `n_month`

| Periodi      | Numero di mesi |
|--------------|----------------|
| mese         | 1              |
| bimestre     | 2              |
| trimestre    | 3              |
| quattro mesi | 4              |
| semestre     | 6              |

Argomenti

| Argomento                  | Descrizione                                                                                                                                                                                                                                                                                                                             |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>n_months</b>            | Il numero di mesi che definisce il periodo. Un numero intero o un'espressione la cui risoluzione è un numero intero corrispondente a: 1 (equivalente alla funzione <code>inmonth()</code> ), 2 (bimestre), 3 (equivalente alla funzione <code>inquarter()</code> ), 4 (quadrimestre) o 6 (semestre).                                    |
| <b>date</b>                | La data o la data e ora da valutare.                                                                                                                                                                                                                                                                                                    |
| <b>period_no</b>           | Il periodo può essere differito mediante <b>period_no</b> , un numero intero, o un'espressione la cui risoluzione è un numero intero, in cui il valore 0 indica il periodo che contiene <b>base_date</b> . I valori negativi di <b>period_no</b> indicano i periodi precedenti, mentre i valori positivi indicano i periodi successivi. |
| <b>first_month_of_year</b> | Se si intende utilizzare anni (fiscali) che non iniziano a gennaio, indicare un valore compreso tra 2 e 12 in <b>first_month_of_year</b> .                                                                                                                                                                                              |

### Casi di utilizzo

La funzione `monthsname()` è utile quando si desidera fornire all'utente la funzionalità per confrontare le aggregazioni in base a un periodo di sua scelta. Ad esempio, è possibile fornire una variabile di input per consentire all'utente di visualizzare le vendite totali dei prodotti per mese, trimestre o semestre.

Queste dimensioni possono essere create nello script di caricamento aggiungendo la funzione come campo in una tabella Calendario principale o, in alternativa, creando la dimensione direttamente in un grafico come dimensione calcolata.

Esempi di funzioni

| Esempio                                        | Risultato                                                                                                                                                         |
|------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>monthsname(4, '10/19/2013')</code>       | Restituisce 'Sep-Dec 2013'. In questo e in altri esempi, l'istruzione <b>SET Monthnames</b> è impostata su Jan;Feb;Mar e così via.                                |
| <code>monthsname(4, '10/19/2013', -1)</code>   | Restituisce 'May-Aug 2013'.                                                                                                                                       |
| <code>monthsname(4, '10/19/2013', 0, 2)</code> | Restituisce 'Oct-Jan 2014', poiché come mese di inizio anno è stato specificato 2. Quindi, il periodo di quattro mesi termina il primo mese dell'anno successivo. |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - Esempio di base

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2022, caricato in una tabella denominata 'Transactions'.
- Il campo della data fornito nel formato della variabile di sistema DateFormat (MM/GG/AAAA).
- La creazione di un campo, bi\_monthly\_range, che raggruppa le transazioni in segmenti bimestrali e restituisce i nomi limite di quel segmento per ogni transazione.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        monthsname(2,date) as bi_monthly_range
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
id,date,amount
8188,2/19/2022,37.23
8189,3/7/2022,17.17
8190,3/30/2022,88.27
8191,4/5/2022,57.42
8192,4/16/2022,53.80
8193,5/1/2022,82.06
8194,5/7/2022,40.39
8195,5/22/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

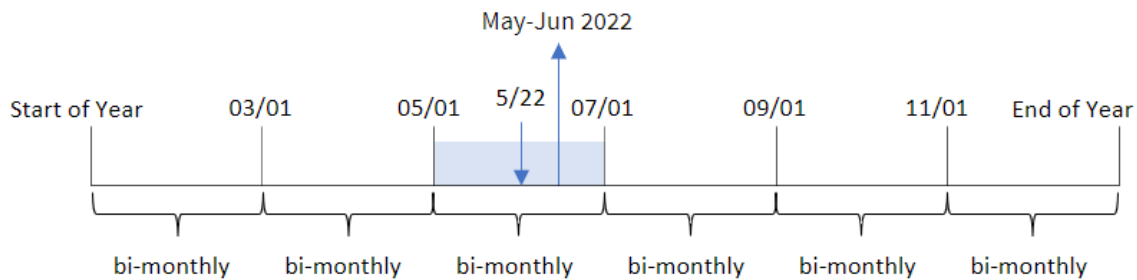
- date
- bi\_monthly\_range

Tabella dei risultati

| date       | intervallo bimestrale |
|------------|-----------------------|
| 2/19/2022  | Gen-Feb 2022          |
| 3/7/2022   | Mar-Apr 2022          |
| 3/30/2022  | Mar-Apr 2022          |
| 4/5/2022   | Mar-Apr 2022          |
| 4/16/2022  | Mar-Apr 2022          |
| 5/1/2022   | Mag-Giu 2022          |
| 5/7/2022   | Mag-Giu 2022          |
| 5/22/2022  | Mag-Giu 2022          |
| 6/15/2022  | Mag-Giu 2022          |
| 6/26/2022  | Mag-Giu 2022          |
| 7/9/2022   | Lug-Ago 2022          |
| 7/22/2022  | Lug-Ago 2022          |
| 7/23/2022  | Lug-Ago 2022          |
| 7/27/2022  | Lug-Ago 2022          |
| 8/2/2022   | Lug-Ago 2022          |
| 8/8/2022   | Lug-Ago 2022          |
| 8/19/2022  | Lug-Ago 2022          |
| 9/26/2022  | Set-Ott 2022          |
| 10/14/2022 | Set-Ott 2022          |
| 10/29/2022 | Set-Ott 2022          |

Il campo `bi_monthly_range` viene creato nell'istruzione caricamento precedente utilizzando la funzione `monthsname()`. Il primo argomento fornito è 2, dividendo l'anno in segmenti bimestrali. Il secondo argomento identifica il campo da valutare.

*Schema della funzione monthsname, esempio di base*



La transazione 8195 avviene il 22 maggio. La funzione `monthsname()` divide inizialmente l'anno in segmenti bimestrali. La transazione 8195 rientra nel segmento tra il 1 maggio e il 30 giugno. Pertanto, la funzione restituisce questi mesi nel formato della variabile di sistema `MonthNames`, nonché l'anno, maggio-giugno 2022.

### Esempio 2 - period\_no

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati inline e lo stesso scenario del primo esempio.
- La creazione di un campo, `prev_bi_monthly_range`, che raggruppa le transazioni in segmenti bimestrali e restituisce i nomi limite del segmento precedente per ogni transazione.

Aggiungere qui altro testo, se necessario, con elenchi, ecc.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        MonthsName(2,date,-1) as prev_bi_monthly_range
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,2/19/2022,37.23
```

```
8189,3/7/2022,17.17
```

```
8190,3/30/2022,88.27
```

```
8191,4/5/2022,57.42
```

```
8192,4/16/2022,53.80
8193,5/1/2022,82.06
8194,5/7/2022,40.39
8195,5/22/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- prev\_bi\_monthly\_range

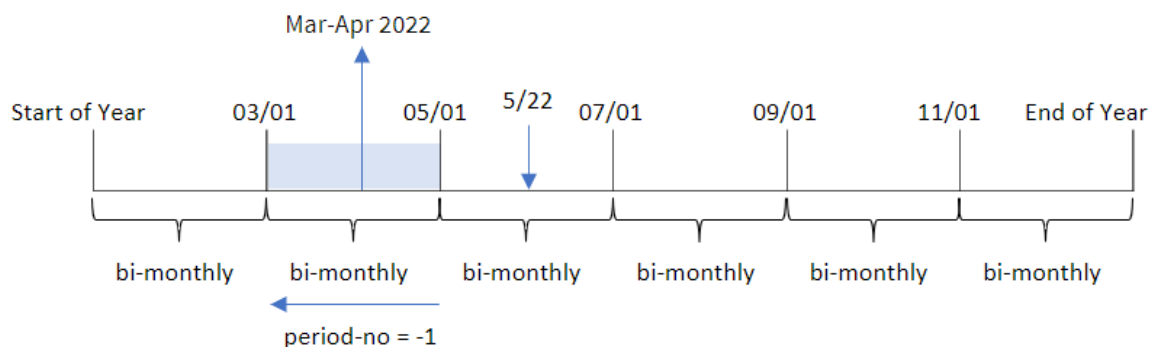
Tabella dei risultati

| date      | prev_bi_monthly_range |
|-----------|-----------------------|
| 2/19/2022 | Nov-Dic 2021          |
| 3/7/2022  | Gen-Feb 2022          |
| 3/30/2022 | Gen-Feb 2022          |
| 4/5/2022  | Gen-Feb 2022          |
| 4/16/2022 | Gen-Feb 2022          |
| 5/1/2022  | Mar-Apr 2022          |
| 5/7/2022  | Mar-Apr 2022          |
| 5/22/2022 | Mar-Apr 2022          |
| 6/15/2022 | Mar-Apr 2022          |
| 6/26/2022 | Mar-Apr 2022          |
| 7/9/2022  | Mag-Giu 2022          |
| 7/22/2022 | Mag-Giu 2022          |
| 7/23/2022 | Mag-Giu 2022          |
| 7/27/2022 | Mag-Giu 2022          |

| date       | prev_bi_monthly_range |
|------------|-----------------------|
| 8/2/2022   | Mag-Giu 2022          |
| 8/8/2022   | Mag-Giu 2022          |
| 8/19/2022  | Mag-Giu 2022          |
| 9/26/2022  | Lug-Ago 2022          |
| 10/14/2022 | Lug-Ago 2022          |
| 10/29/2022 | Lug-Ago 2022          |

In questo esempio, -1 viene utilizzato come argomento `period_no` nella funzione `monthsname()`. Dopo aver inizialmente diviso un anno in segmenti bimestrali, la funzione restituisce quindi i limiti del segmento precedente per il momento in cui avviene una transazione.

*Schema della funzione `monthsname`, esempio di `period_no`*



La transazione 8195 è stata effettuata nel segmento tra maggio e giugno. Pertanto, il segmento bimestrale precedente è compreso tra il 1 marzo e il 30 aprile, quindi la funzione restituisce Mar-Apr 2022.

### Esempio 3 - `first_month_of_year`

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati inline e lo stesso scenario del primo esempio.
- La creazione di un campo differente, `bi_monthly_range`, che raggruppa le transazioni in segmenti bimestrali e restituisce i limiti del segmento per ogni transazione.

Tuttavia, in questo esempio, dobbiamo anche impostare aprile come primo mese dell'anno finanziario.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';

Transactions:
    Load
        *,
        MonthsName(2,date,0,4) as bi_monthly_range
    ;
Load
*
Inline
[
id,date,amount
8188,2/19/2022,37.23
8189,3/7/2022,17.17
8190,3/30/2022,88.27
8191,4/5/2022,57.42
8192,4/16/2022,53.80
8193,5/1/2022,82.06
8194,5/7/2022,40.39
8195,5/22/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- bi\_monthly\_range

Tabella dei risultati

| date      | intervallo bimestrale |
|-----------|-----------------------|
| 2/19/2022 | Feb-Mar 2021          |
| 3/7/2022  | Feb-Mar 2021          |
| 3/30/2022 | Feb-Mar 2021          |
| 4/5/2022  | Apr-Mag 2022          |

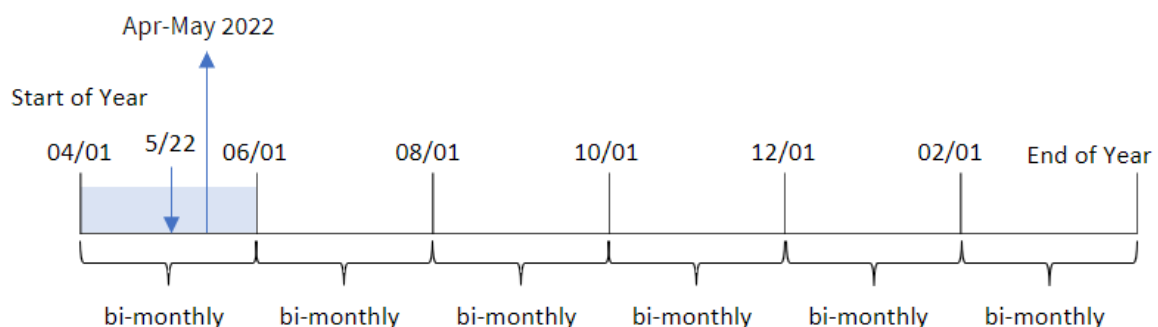
| date       | intervallo bimestrale |
|------------|-----------------------|
| 4/16/2022  | Apr-Mag 2022          |
| 5/1/2022   | Apr-Mag 2022          |
| 5/7/2022   | Apr-Mag 2022          |
| 5/22/2022  | Apr-Mag 2022          |
| 6/15/2022  | Giu-Lug 2022          |
| 6/26/2022  | Giu-Lug 2022          |
| 7/9/2022   | Giu-Lug 2022          |
| 7/22/2022  | Giu-Lug 2022          |
| 7/23/2022  | Giu-Lug 2022          |
| 7/27/2022  | Giu-Lug 2022          |
| 8/2/2022   | Ago-Set 2022          |
| 8/8/2022   | Ago-Set 2022          |
| 8/19/2022  | Ago-Set 2022          |
| 9/26/2022  | Ago-Set 2022          |
| 10/14/2022 | Ott-Nov 2022          |
| 10/29/2022 | Ott-Nov 2022          |

Utilizzando 4 come argomento `first_month_of_year` nella funzione `monthsname()`, la funzione inizia l'anno il 1 aprile, quindi divide l'anno in segmenti bimestrali: Apr-Mag, Giu-Lug, Ago-Set, Ott-Nov, Dic-Gen, Feb-Mar.

Testo del paragrafo per i Risultati.

La transazione 8195 è avvenuta il 22 maggio e rientra nel segmento tra il 1 aprile e il 31 maggio. Pertanto, la funzione restituisce Apr-Mag 2022.

*Diagramma della funzione `monthsname`, esempio di `first_month_of_year`*



### Esempio 4 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor di caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento utilizza lo stesso set di dati inline e lo stesso scenario del primo esempio. Tuttavia, in questo esempio, il set di dati invariato viene caricato nell'applicazione. Il calcolo che raggruppa le transazioni in segmenti bimestrali e restituisce i limiti del segmento per ogni transazione viene creato come una misura in un oggetto grafico dell'applicazione.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,2/19/2022,37.23
```

```
8189,3/7/2022,17.17
```

```
8190,3/30/2022,88.27
```

```
8191,4/5/2022,57.42
```

```
8192,4/16/2022,53.80
```

```
8193,5/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/22/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
```

```
8200,7/23/2022,82.77
```

```
8201,7/27/2022,69.98
```

```
8202,8/2/2022,76.11
```

```
8203,8/8/2022,25.12
```

```
8204,8/19/2022,46.23
```

```
8205,9/26/2022,84.21
```

```
8206,10/14/2022,96.24
```

```
8207,10/29/2022,67.67
```

```
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:date.

Creare la seguente misura:

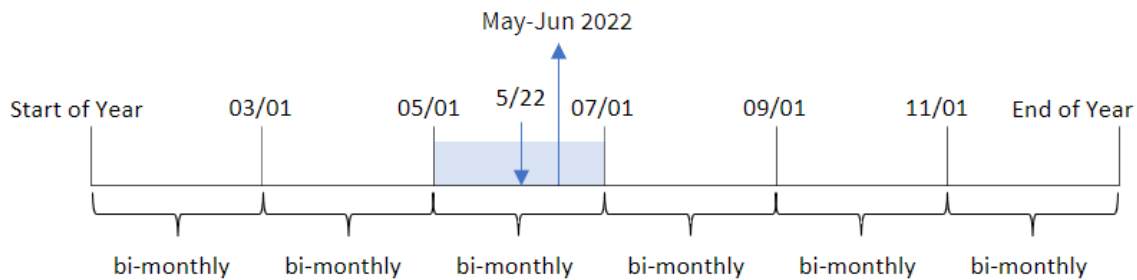
```
=monthsname(2,date)
```

Tabella dei risultati

| date       | =monthsname(2,date) |
|------------|---------------------|
| 2/19/2022  | Gen-Feb 2022        |
| 3/7/2022   | Mar-Apr 2022        |
| 3/30/2022  | Mar-Apr 2022        |
| 4/5/2022   | Mar-Apr 2022        |
| 4/16/2022  | Mar-Apr 2022        |
| 5/1/2022   | Mag-Giu 2022        |
| 5/7/2022   | Mag-Giu 2022        |
| 5/22/2022  | Mag-Giu 2022        |
| 6/15/2022  | Mag-Giu 2022        |
| 6/26/2022  | Mag-Giu 2022        |
| 7/9/2022   | Lug-Ago 2022        |
| 7/22/2022  | Lug-Ago 2022        |
| 7/23/2022  | Lug-Ago 2022        |
| 7/27/2022  | Lug-Ago 2022        |
| 8/2/2022   | Lug-Ago 2022        |
| 8/8/2022   | Lug-Ago 2022        |
| 8/19/2022  | Lug-Ago 2022        |
| 9/26/2022  | Set-Ott 2022        |
| 10/14/2022 | Set-Ott 2022        |
| 10/29/2022 | Set-Ott 2022        |

Il campo `bi_monthly_range` viene creato come misura nell'oggetto grafico mediante l'utilizzo della funzione `monthsname()`. Il primo argomento fornito è 2, dividendo l'anno in segmenti bimestrali. Il secondo argomento identifica il campo da valutare.

*Schema della funzione monthsname, esempio di oggetto grafico*



La transazione 8195 avviene il 22 maggio. La funzione `monthsname()` divide inizialmente l'anno in segmenti bimestrali. La transazione 8195 rientra nel segmento tra il 1 maggio e il 30 giugno. Pertanto, la funzione restituisce questi mesi nel formato della variabile di sistema `MonthNames`, nonché l'anno, maggio-giugno 2022.

### Esempio 5 - Scenario

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente le transazioni per il 2022, caricato in una tabella denominata `Transactions`.
- Il campo della data fornito nel formato della variabile di sistema `DateFormat` (MM/GG/AAAA).

L'utente finale vuole creare un oggetto grafico che visualizzi le vendite totali per un periodo selezionato. Questo può essere ottenuto anche quando la dimensione non è disponibile nel modello dati, utilizzando la funzione `monthsname()` come dimensione calcolata che viene modificata dinamicamente da un controllo di input variabile.

#### Script di caricamento

```
SET vPeriod = 1;  
SET DateFormat='MM/DD/YYYY';
```

`Transactions:`

`Load`

`*`

`Inline`

`[`

`id,date,amount`

`8188,'1/7/2022',17.17`

`8189,'1/19/2022',37.23`

`8190,'2/28/2022',88.27`

```
8191, '2/5/2022', 57.42
8192, '3/16/2022', 53.80
8193, '4/1/2022', 82.06
8194, '5/7/2022', 40.39
8195, '5/16/2022', 87.21
8196, '6/15/2022', 95.93
8197, '6/26/2022', 45.89
8198, '7/9/2022', 36.23
8199, '7/22/2022', 25.66
8200, '7/23/2022', 82.77
8201, '7/27/2022', 69.98
8202, '8/2/2022', 76.11
8203, '8/8/2022', 25.12
8204, '8/19/2022', 46.23
8205, '9/26/2022', 84.21
8206, '10/14/2022', 96.24
8207, '10/29/2022', 67.67
];
```

### Risultati

Caricare i dati e aprire un foglio.

All'inizio dello script di caricamento, viene creata una variabile (`vPeriod`) che sarà collegata al controllo di input della variabile. Quindi, configurare la variabile come oggetto personalizzato nel foglio.

### Procedere come indicato di seguito:

1. Nel pannello delle risorse, fare clic su **Oggetti personalizzati**.
2. Selezionare **Qlik Dashboard bundle** e creare un oggetto **Input variabile**.
3. Immettere un titolo per l'oggetto grafico.
4. In **Variabile**, selezionare **vPeriod** come Nome e impostare l'oggetto in modo che venga visualizzato come **elenco a discesa**.
5. In **Valori**, configurare l'oggetto in modo che utilizzi valori dinamici. Inserire quanto segue:  
`= '1~month|2~bi-month|3~quarter|4~tertia|6~half-year'`

Quindi, creare la tabella dei risultati.

### Procedere come indicato di seguito:

1. Creare una nuova tabella e aggiungere la seguente dimensione calcolata:  
`=monthsname($(vPeriod), date)`
2. Aggiungere questa misura per calcolare le vendite totali:  
`=sum(amount)`
3. Impostare la **Formattazione numero** della misura su **Denaro**. Fare clic su **✓ Termina modifica**. È ora possibile modificare i dati mostrati nella tabella regolando il segmento temporale nell'oggetto della variabile.

Ecco come apparirà la tabella dei risultati quando l'opzione `tertia` è selezionata:

Tabella dei risultati

| monthsname(\$(vPeriod),date) | sum(amount) |
|------------------------------|-------------|
| Gen-Apr 2022                 | 253.89      |
| Mag-ago 2022                 | 713.58      |
| Set-Dic 2022                 | 248.12      |

## monthsstart

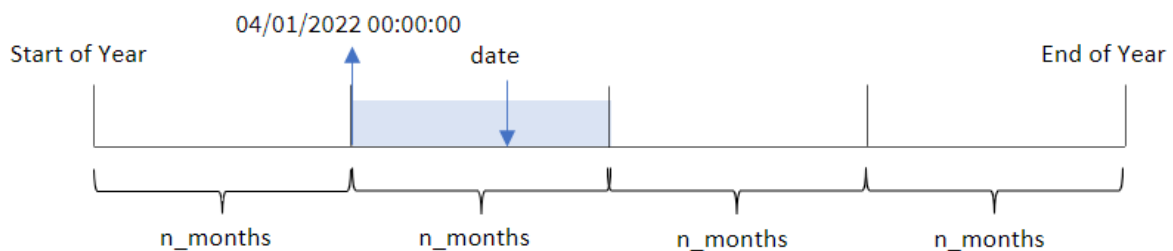
Questa funzione restituisce un valore corrispondente a un indicatore temporale recante il primo millisecondo del mese, del bimestre, del trimestre, del quadrimestre o del semestre contenente una data di base. È inoltre possibile individuare il timestamp per un periodo di tempo precedente o successivo. Il formato di output predefinito è **DateFormat**, impostato nello script.

### Sintassi:

```
MonthsStart(n_months, date[, period_no [, first_month_of_year]])
```

**Tipo di dati restituiti:** duale

Schema della funzione *monthsstart()*.



La funzione *monthsstart()* divide l'anno in segmenti in base all'argomento *n\_months* fornito. Quindi valuta in quale segmento rientra ogni data fornita e restituisce il primo millisecondo, in formato data, di quel segmento. La funzione offre anche la possibilità di restituire il timestamp iniziale dai segmenti precedenti o successivi, oltre a ridefinire qual è il primo mese dell'anno.

I seguenti segmenti dell'anno sono disponibili nella funzione come argomenti di *n\_month* :

Possibili argomenti di *n\_month*

| Periodi      | Numero di mesi |
|--------------|----------------|
| mese         | 1              |
| bimestre     | 2              |
| trimestre    | 3              |
| quattro mesi | 4              |
| semestre     | 6              |

## Argomenti

| Argomento                  | Descrizione                                                                                                                                                                                                                                                                                                                             |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>n_months</b>            | Il numero di mesi che definisce il periodo. Un numero intero o un'espressione la cui risoluzione è un numero intero corrispondente a: 1 (equivalente alla funzione <code>inmonth()</code> ), 2 (bimestre), 3 (equivalente alla funzione <code>inquarter()</code> ), 4 (quadrimestre) o 6 (semestre).                                    |
| <b>date</b>                | La data o la data e ora da valutare.                                                                                                                                                                                                                                                                                                    |
| <b>period_no</b>           | Il periodo può essere differito mediante <b>period_no</b> , un numero intero, o un'espressione la cui risoluzione è un numero intero, in cui il valore 0 indica il periodo che contiene <b>base_date</b> . I valori negativi di <b>period_no</b> indicano i periodi precedenti, mentre i valori positivi indicano i periodi successivi. |
| <b>first_month_of_year</b> | Se si intende utilizzare anni (fiscali) che non iniziano a gennaio, indicare un valore compreso tra 2 e 12 in <b>first_month_of_year</b> .                                                                                                                                                                                              |

## Casi di utilizzo

La funzione `monthsstart()` viene comunemente utilizzata come parte di un'espressione quando l'utente desidera che il calcolo utilizzi la frazione di un periodo non ancora trascorso. Questo potrebbe essere utilizzato, ad esempio, per fornire una variabile di input che consenta all'utente di calcolare il totale degli interessi accumulati fino a quel momento nel mese, nel trimestre o nel semestre.

## Esempi di funzioni

| Esempio                                         | Risultato                                                                |
|-------------------------------------------------|--------------------------------------------------------------------------|
| <code>monthsstart(4, '10/19/2013')</code>       | Restituisce 09/01/2013.                                                  |
| <code>monthsstart(4, '10/19/2013', -1)</code>   | Restituisce 05/01/2013.                                                  |
| <code>monthsstart(4, '10/19/2013', 0, 2)</code> | Restituisce 10/01/2013, poiché l'inizio dell'anno corrisponde al mese 2. |

## Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - Nessun argomento aggiuntivo

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2022, caricato in una tabella denominata 'Transactions'.
- Il campo della data fornito nel formato della variabile di sistema `DateFormat` (MM/GG/AAAA).
- La creazione di un campo, `bi_monthly_start`, che raggruppa le transazioni in segmenti bimestrali e restituisce il timestamp iniziale del segmento per ogni transazione.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        monthsstart(2,date) as bi_monthly_start,
        timestamp(monthsstart(2,date)) as bi_monthly_start_timestamp
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,2/19/2022,37.23
```

```
8189,3/7/2022,17.17
```

```
8190,3/30/2022,88.27
```

```
8191,4/5/2022,57.42
```

```
8192,4/16/2022,53.80
```

```
8193,5/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/22/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
```

```
8200,7/23/2022,82.77
```

```
8201,7/27/2022,69.98
```

```
8202,8/2/2022,76.11
```

```
8203,8/8/2022,25.12
```

```
8204,8/19/2022,46.23
```

```
8205,9/26/2022,84.21
```

```
8206,10/14/2022,96.24
```

```
8207,10/29/2022,67.67
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

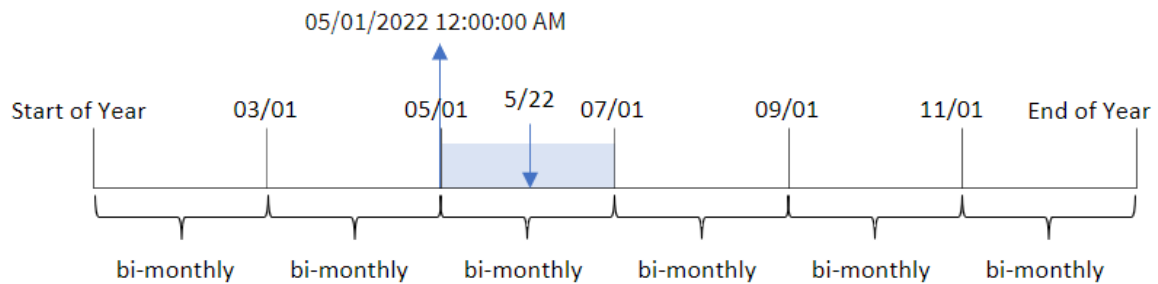
- date
- bi\_monthly\_start
- bi\_monthly\_start\_timestamp

Tabella dei risultati

| date       | bi_monthly_start | bi_monthly_start_timestamp |
|------------|------------------|----------------------------|
| 2/19/2022  | 01/01/2022       | 1/1/2022 12:00:00 AM       |
| 3/7/2022   | 03/01/2022       | 3/1/2022 12:00:00 AM       |
| 3/30/2022  | 03/01/2022       | 3/1/2022 12:00:00 AM       |
| 4/5/2022   | 03/01/2022       | 3/1/2022 12:00:00 AM       |
| 4/16/2022  | 03/01/2022       | 3/1/2022 12:00:00 AM       |
| 5/1/2022   | 05/01/2022       | 5/1/2022 12:00:00 AM       |
| 5/7/2022   | 01/05/2022       | 5/1/2022 12:00:00 AM       |
| 5/22/2022  | 05/01/2022       | 01/05/2022 00:00:00        |
| 6/15/2022  | 01/05/2022       | 01/05/2022 00:00:00        |
| 6/26/2022  | 01/05/2022       | 01/05/2022 00:00:00        |
| 7/9/2022   | 07/01/2022       | 1/7/2022 00:00:00          |
| 7/22/2022  | 07/01/2022       | 7/1/2022 12:00:00 AM       |
| 7/23/2022  | 07/01/2022       | 01/07/2022 00:00:00        |
| 7/27/2022  | 07/01/2022       | 1/7/2022 00:00:00          |
| 8/2/2022   | 07/01/2022       | 01/07/2022 00:00:00        |
| 8/8/2022   | 07/01/2022       | 01/07/2022 00:00:00        |
| 8/19/2022  | 07/01/2022       | 01/07/2022 00:00:00        |
| 9/26/2022  | 09/01/2022       | 1/9/2022 00:00:00          |
| 10/14/2022 | 01/09/2022       | 9/1/2022 12:00:00 AM       |
| 10/29/2022 | 01/09/2022       | 9/1/2022 12:00:00 AM       |

Il campo `bi_monthly_start` viene creato nell'istruzione caricamento precedente utilizzando la funzione `monthsstart()`. Il primo argomento fornito è 2, dividendo l'anno in segmenti bimestrali. Il secondo argomento identifica il campo da valutare.

Schema della funzione `monthsstart()`, esempio senza argomenti aggiuntivi



La transazione 8195 avviene il 22 maggio. La funzione `monthsstart()` divide inizialmente l'anno in segmenti bimestrali. La transazione 8195 rientra nel segmento tra il 1 maggio e il 30 giugno. Pertanto, la funzione restituisce il primo millisecondo di questo segmento, il 1° maggio 2022 alle ore 12:00:00 AM.

### Esempio 2 - period\_no

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.
- La creazione di un campo, 'prev\_bi\_monthly\_start', che restituisce il primo millisecondo del segmento bimestrale prima che avvenga la transazione.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        monthsstart(2,date,-1) as prev_bi_monthly_start,
        timestamp(monthsstart(2,date,-1)) as prev_bi_monthly_start_timestamp
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,2/19/2022,37.23
```

```
8189,3/7/2022,17.17
```

```
8190,3/30/2022,88.27
```

```
8191,4/5/2022,57.42
```

```
8192,4/16/2022,53.80
```

```
8193,5/1/2022,82.06
```

```
8194, 5/7/2022, 40.39
8195, 5/22/2022, 87.21
8196, 6/15/2022, 95.93
8197, 6/26/2022, 45.89
8198, 7/9/2022, 36.23
8199, 7/22/2022, 25.66
8200, 7/23/2022, 82.77
8201, 7/27/2022, 69.98
8202, 8/2/2022, 76.11
8203, 8/8/2022, 25.12
8204, 8/19/2022, 46.23
8205, 9/26/2022, 84.21
8206, 10/14/2022, 96.24
8207, 10/29/2022, 67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- prev\_bi\_monthly\_start
- prev\_bi\_monthly\_start\_timestamp

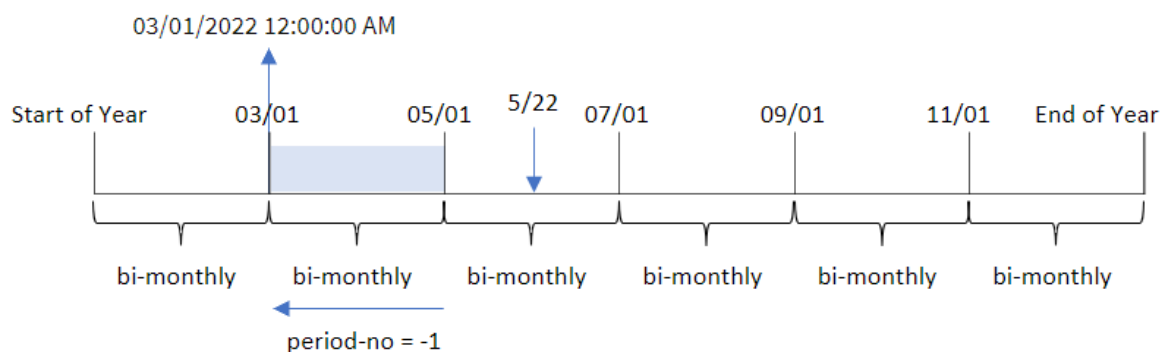
Tabella dei risultati

| date      | prev_bi_monthly_start | prev_bi_monthly_start_timestamp |
|-----------|-----------------------|---------------------------------|
| 2/19/2022 | 11/01/2021            | 11/1/2021 12:00:00 AM           |
| 3/7/2022  | 01/01/2022            | 1/1/2022 12:00:00 AM            |
| 3/30/2022 | 01/01/2022            | 1/1/2022 12:00:00 AM            |
| 4/5/2022  | 01/01/2022            | 1/1/2022 12:00:00 AM            |
| 4/16/2022 | 01/01/2022            | 1/1/2022 12:00:00 AM            |
| 5/1/2022  | 03/01/2022            | 1/3/2022 00:00:00               |
| 5/7/2022  | 03/01/2022            | 3/1/2022 12:00:00 AM            |
| 5/22/2022 | 03/01/2022            | 1/3/2022 12:00:00 AM            |
| 6/15/2022 | 03/01/2022            | 3/1/2022 12:00:00 AM            |
| 6/26/2022 | 03/01/2022            | 3/1/2022 12:00:00 AM            |
| 7/9/2022  | 05/01/2022            | 5/1/2022 12:00:00 AM            |
| 7/22/2022 | 05/01/2022            | 5/1/2022 12:00:00 AM            |
| 7/23/2022 | 05/01/2022            | 5/1/2022 12:00:00 AM            |
| 7/27/2022 | 05/01/2022            | 5/1/2022 12:00:00 AM            |
| 8/2/2022  | 05/01/2022            | 5/1/2022 12:00:00 AM            |

| date       | prev_bi_monthly_start | prev_bi_monthly_start_timestamp |
|------------|-----------------------|---------------------------------|
| 8/8/2022   | 05/01/2022            | 5/1/2022 12:00:00 AM            |
| 8/19/2022  | 05/01/2022            | 5/1/2022 12:00:00 AM            |
| 9/26/2022  | 07/01/2022            | 7/1/2022 12:00:00 AM            |
| 10/14/2022 | 07/01/2022            | 1/7/2022 00:00:00               |
| 10/29/2022 | 07/01/2022            | 7/1/2022 12:00:00 AM            |

Utilizzando -1 come argomento `period_no` della funzione `monthsstart()`, dopo aver inizialmente diviso un anno in segmenti bimestrali, la funzione restituisce il primo millisecondo del segmento bimestrale precedente al momento in cui avviene una transazione.

*Schema della funzione `monthsstart()`, esempio di `period_no`*



La transazione 8195 è stata effettuata nel segmento tra maggio e giugno. Pertanto, il segmento bimestrale precedente era compreso tra il 1° marzo e il 30 aprile, quindi la funzione restituisce il primo millisecondo di questo segmento, il 1° marzo 2022 alle 12:00:00.

### Esempio 3 - `first_month_of_year`

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.
- La creazione di un campo, `bi_monthly_start`, che raggruppa le transazioni in segmenti bimestrali e restituisce il timestamp iniziale del set per ogni transazione.

Tuttavia, in questo esempio, dobbiamo anche impostare aprile come primo mese dell'anno finanziario.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        monthsstart(2,date,0,4) as bi_monthly_start,
        timestamp(monthsstart(2,date,0,4)) as bi_monthly_start_timestamp
    ;

Load
*
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- bi\_monthly\_start
- bi\_monthly\_start\_timestamp

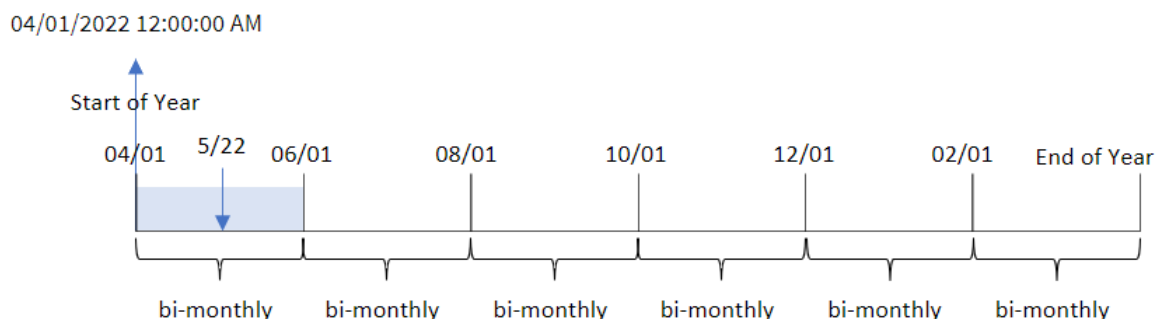
Tabella dei risultati

| date      | bi_monthly_start | bi_monthly_start_timestamp |
|-----------|------------------|----------------------------|
| 2/19/2022 | 02/01/2022       | 3/1/2022 12:00:00 AM       |
| 3/7/2022  | 02/01/2022       | 3/1/2022 12:00:00 AM       |

| date       | bi_monthly_start | bi_monthly_start_timestamp |
|------------|------------------|----------------------------|
| 3/30/2022  | 02/01/2022       | 3/1/2022 12:00:00 AM       |
| 4/5/2022   | 04/01/2022       | 4/1/2022 12:00:00 AM       |
| 4/16/2022  | 04/01/2022       | 4/1/2022 12:00:00 AM       |
| 5/1/2022   | 04/01/2022       | 01/04/2022 00:00:00        |
| 5/7/2022   | 04/01/2022       | 4/1/2022 12:00:00 AM       |
| 5/22/2022  | 04/01/2022       | 01/04/2022 00:00:00        |
| 6/15/2022  | 01/06/2022       | 6/1/2022 12:00:00 AM       |
| 6/26/2022  | 06/01/2022       | 01/06/2022 00:00:00        |
| 7/9/2022   | 01/06/2022       | 01/06/2022 00:00:00        |
| 7/22/2022  | 01/06/2022       | 01/06/2022 00:00:00        |
| 7/23/2022  | 01/06/2022       | 01/06/2022 00:00:00        |
| 7/27/2022  | 01/06/2022       | 01/06/2022 00:00:00        |
| 8/2/2022   | 01/08/2022       | 8/1/2022 12:00:00 AM       |
| 8/8/2022   | 08/01/2022       | 8/1/2022 12:00:00 AM       |
| 8/19/2022  | 08/01/2022       | 1/8/2022 00:00:00          |
| 9/26/2022  | 01/08/2022       | 01/08/2022 00:00:00        |
| 10/14/2022 | 01/10/2022       | 10/1/2022 12:00:00 AM      |
| 10/29/2022 | 10/01/2022       | 10/1/2022 12:00:00 AM      |

Utilizzando 4 come argomento `first_month_of_year` nella funzione `monthsstart()`, la funzione inizia l'anno il 1 aprile, quindi divide l'anno in segmenti bimestrali: Apr-Mag, Giu-Lug, Ago-Set, Ott-Nov, Dic-Gen, Feb-Mar.

Schema della funzione `monthsstart()`, esempio di `first_month_of_year`



La transazione 8195 è avvenuta il 22 maggio e rientra nel segmento tra il 1 aprile e il 31 maggio. Pertanto, la funzione restituisce il primo millisecondo di questo segmento, il 1° aprile 2022 alle ore 12:00:00 AM.

### Esempio 4 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor di caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento contiene lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, in questo esempio, il set di dati invariato viene caricato nell'applicazione. Il calcolo che raggruppa le transazioni in segmenti bimestrali e restituisce il timestamp iniziale del set per ogni transazione viene creato come misura in un oggetto grafico dell'applicazione.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,2/19/2022,37.23
```

```
8189,3/7/2022,17.17
```

```
8190,3/30/2022,88.27
```

```
8191,4/5/2022,57.42
```

```
8192,4/16/2022,53.80
```

```
8193,5/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/22/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
```

```
8200,7/23/2022,82.77
```

```
8201,7/27/2022,69.98
```

```
8202,8/2/2022,76.11
```

```
8203,8/8/2022,25.12
```

```
8204,8/19/2022,46.23
```

```
8205,9/26/2022,84.21
```

```
8206,10/14/2022,96.24
```

```
8207,10/29/2022,67.67
```

```
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: date.

Creare le seguenti misure:

=monthsstart(2,date)

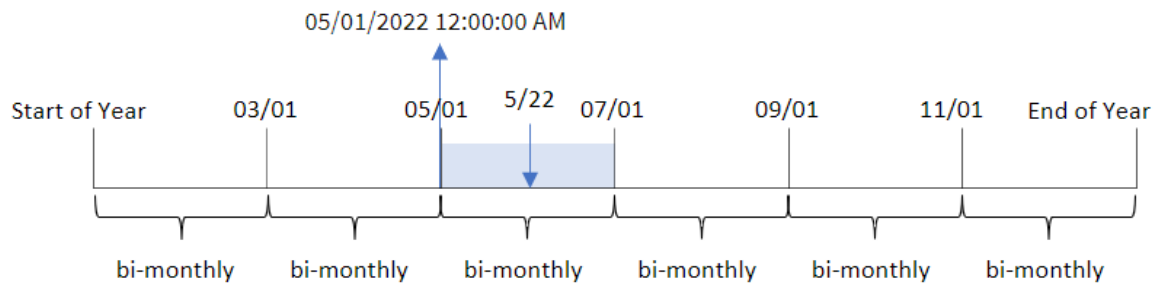
=timestamp(monthsstart(2,date))

Questi calcoli recuperano il timestamp iniziale del segmento bimestrale in cui ha avuto luogo ciascuna transazione.

Tabella dei risultati

| date       | =monthsstart(2,date) | =timestamp(monthsstart(2,date)) |
|------------|----------------------|---------------------------------|
| 9/26/2022  | 09/01/2022           | 1/9/2022 00:00:00               |
| 10/14/2022 | 01/09/2022           | 9/1/2022 12:00:00 AM            |
| 10/29/2022 | 01/09/2022           | 9/1/2022 12:00:00 AM            |
| 7/9/2022   | 07/01/2022           | 1/7/2022 00:00:00               |
| 7/22/2022  | 07/01/2022           | 7/1/2022 12:00:00 AM            |
| 7/23/2022  | 07/01/2022           | 01/07/2022 00:00:00             |
| 7/27/2022  | 07/01/2022           | 1/7/2022 00:00:00               |
| 8/2/2022   | 07/01/2022           | 01/07/2022 00:00:00             |
| 8/8/2022   | 07/01/2022           | 01/07/2022 00:00:00             |
| 8/19/2022  | 07/01/2022           | 7/1/2022 12:00:00 AM            |
| 5/1/2022   | 05/01/2022           | 5/1/2022 12:00:00 AM            |
| 5/7/2022   | 01/05/2022           | 5/1/2022 12:00:00 AM            |
| 5/22/2022  | 05/01/2022           | 01/05/2022 00:00:00             |
| 6/15/2022  | 01/05/2022           | 01/05/2022 00:00:00             |
| 6/26/2022  | 01/05/2022           | 5/1/2022 12:00:00 AM            |
| 3/7/2022   | 03/01/2022           | 3/1/2022 12:00:00 AM            |
| 3/30/2022  | 03/01/2022           | 3/1/2022 12:00:00 AM            |
| 4/5/2022   | 03/01/2022           | 3/1/2022 12:00:00 AM            |
| 4/16/2022  | 03/01/2022           | 3/1/2022 12:00:00 AM            |
| 2/19/2022  | 01/01/2022           | 11/1/2021 12:00:00 AM           |

*Schema della funzione monthsstart(), esempio di oggetto grafico*



La transazione 8195 è avvenuta il 22 maggio. La funzione monthsstart() divide inizialmente l'anno in segmenti bimestrali. La transazione 8195 rientra nel segmento tra il 1 maggio e il 30 giugno. Pertanto, la funzione restituisce il primo millisecondo di questo segmento, in data 01/05/2022 alle ore 12:00:00 AM.

### Esempio 5 - Scenario

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di saldi di prestiti, che viene caricato in una tabella chiamata Loans.
- Dati costituiti dagli ID dei prestiti, dal saldo all'inizio del mese e dal tasso di interesse semplice annuo applicato a ciascun prestito.

L'utente finale desidera un oggetto grafico che visualizzi, in base all'ID del prestito, gli interessi correnti maturati su ciascun prestito per un periodo a scelta. L'esercizio finanziario inizia a gennaio.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Loans:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
loan_id,start_balance,rate
```

```
8188,$10000.00,0.024
```

```
8189,$15000.00,0.057
```

```
8190,$17500.00,0.024
```

```
8191,$21000.00,0.034
```

```
8192,$90000.00,0.084
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio.

All'inizio dello script di caricamento, viene creata una variabile (`vPeriod`) che sarà collegata al controllo di input della variabile. Quindi, configurare la variabile come oggetto personalizzato nel foglio.

**Procedere come indicato di seguito:**

1. Nel pannello delle risorse, fare clic su **Oggetti personalizzati**.
2. Selezionare **Includi dashboard Qlik** e creare un oggetto **Input variabile**.
3. Immettere un titolo per l'oggetto grafico.
4. In **Variabile**, selezionare **vPeriod** come Nome e impostare l'oggetto in modo che venga visualizzato come **elenco a discesa**.
5. In **Valori**, configurare l'oggetto in modo che utilizzi valori dinamici. Inserire quanto segue:  
`= '1~month|2~bi-month|3~quarter|4~tertial|6~half-year'`

Quindi, creare la tabella dei risultati.

**Procedere come indicato di seguito:**

1. Creare una nuova tabella. Aggiungere i seguenti campi come dimensioni:
  - `employee_id`
  - `employee_name`
2. Creare una misura per calcolare gli interessi accumulati:  
`=start_balance*(rate*(today(1)-monthsstart($(vPeriod),today(1)))/365)`
3. Impostare la **Formattazione numero** della misura su **Denaro**. Fare clic su **✓ Termina modifica**. È ora possibile modificare i dati mostrati nella tabella regolando il segmento temporale nell'oggetto della variabile.

Ecco come apparirà la tabella dei risultati quando sarà selezionata l'opzione del periodo `month`:

Tabella dei risultati

| loan_id | start_balance | =start_balance*(rate*(today(1)-monthsstart(\$(vPeriod),today(1)))/365) |
|---------|---------------|------------------------------------------------------------------------|
| 8188    | \$10000.00    | \$7.95                                                                 |
| 8189    | \$15000.00    | \$67.93                                                                |
| 8190    | \$17500.00    | \$33.37                                                                |
| 8191    | \$21000.00    | \$56.73                                                                |
| 8192    | \$90000.00    | \$600.66                                                               |

La funzione `monthsstart()`, utilizzando l'input dell'utente come primo argomento e la data odierna come secondo argomento, restituisce la data di inizio del periodo scelto dall'utente. Sottraendo tale risultato dalla data corrente, l'espressione restituisce il numero di giorni trascorsi fino a ora in questo periodo.

Questo valore viene quindi moltiplicato per il tasso di interesse e diviso per 365, per restituire il tasso di interesse effettivo incorso per questo periodo. Il risultato viene poi moltiplicato per il saldo iniziale del prestito per ottenere gli interessi maturati fino a ora in questo periodo.

## monthstart

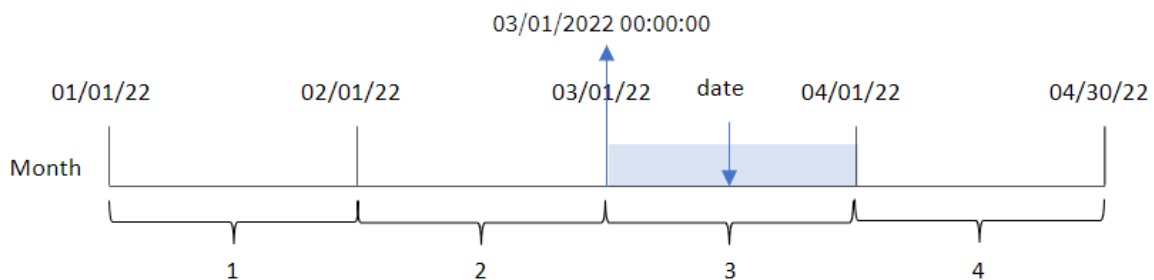
Questa funzione restituisce un valore corrispondente a un indicatore temporale recante il primo millisecondo del primo giorno del mese contenente **date**. Il formato di output predefinito sarà il formato **DateFormat** impostato nello script.

### Sintassi:

```
MonthStart(date[, period_no])
```

Tipo di dati restituiti: duale

Schema della funzione monthstart().



La funzione monthstart() determina in quale mese cade la data. Restituisce quindi un timestamp, in formato data, per il primo millisecondo di quel mese.

### Argomenti

| Argomento        | Descrizione                                                                                                                                                                                                                                 |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>date</b>      | La data o la data e ora da valutare.                                                                                                                                                                                                        |
| <b>period_no</b> | <b>period_no</b> è un numero intero che, se corrisponde a 0 o viene omissso, indica il mese contenente <b>date</b> . I valori negativi di <b>period_no</b> indicano i mesi precedenti, mentre i valori positivi indicano i mesi successivi. |

## Casi di utilizzo

La funzione monthstart() viene comunemente utilizzata come parte di un'espressione quando l'utente desidera che il calcolo utilizzi la frazione del mese trascorsa finora. Ad esempio, può essere utilizzato per calcolare gli interessi accumulati in un mese fino a una certa data.

### Esempi di funzioni

| Esempio                      | Risultato               |
|------------------------------|-------------------------|
| monthstart('10/19/2001')     | Restituisce 10/01/2001. |
| monthstart('10/19/2001', -1) | Restituisce 09/01/2001. |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - Nessun argomento aggiuntivo

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2022, caricato in una tabella denominata 'Transactions'.
- Il campo della data fornito nel formato della variabile di sistema `DateFormat` (MM/GG/AAAA).
- La creazione di un campo, `start_of_month`, che restituisce un timestamp per l'inizio del mese in cui sono avvenute le transazioni.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        monthstart(date) as start_of_month,
        timestamp(monthstart(date)) as start_of_month_timestamp
    ;
Load
*
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- start\_of\_month
- start\_of\_month\_timestamp

Tabella dei risultati

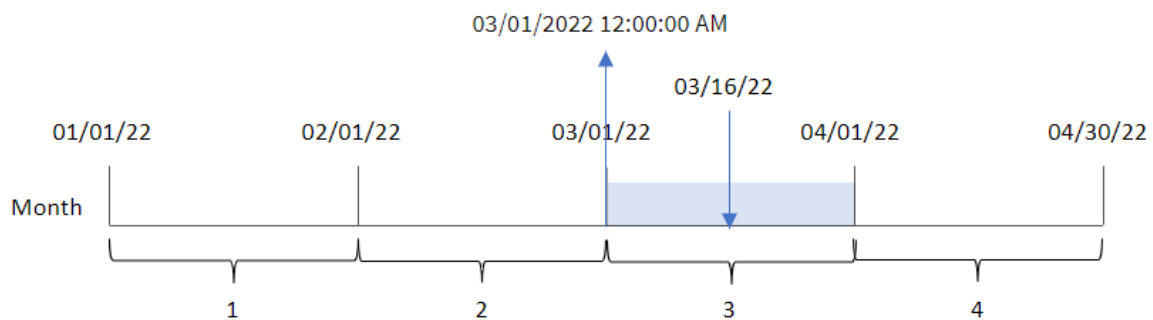
| date      | inizio_mese | timestamp_inizio_mese  |
|-----------|-------------|------------------------|
| 1/7/2022  | 01/01/2022  | 1/1/2022 00:00:00      |
| 1/19/2022 | 01/01/2022  | 01/01/2022 12:00:00 AM |
| 2/5/2022  | 01/02/2022  | 01/02/2022 00:00:00    |
| 2/28/2022 | 01/02/2022  | 01/02/2022 00:00:00    |
| 3/16/2022 | 03/01/2022  | 01/03/2022 00:00:00    |
| 4/1/2022  | 04/01/2022  | 01/04/2022 00:00:00    |
| 5/7/2022  | 01/05/2022  | 01/05/2022 00:00:00    |
| 5/16/2022 | 01/05/2022  | 01/05/2022 00:00:00    |
| 6/15/2022 | 01/06/2022  | 6/1/2022 12:00:00 AM   |
| 6/26/2022 | 07/01/2022  | 1/6/2022 00:00:00      |
| 7/9/2022  | 07/01/2022  | 1/7/2022 00:00:00      |
| 7/22/2022 | 07/01/2022  | 7/1/2022 12:00:00 AM   |
| 7/23/2022 | 07/01/2022  | 01/07/2022 00:00:00    |

| date       | inizio_mese | timestamp_inizio_mese |
|------------|-------------|-----------------------|
| 7/27/2022  | 07/01/2022  | 1/7/2022 00:00:00     |
| 8/2/2022   | 01/08/2022  | 8/1/2022 12:00:00 AM  |
| 8/8/2022   | 08/01/2022  | 8/1/2022 12:00:00 AM  |
| 8/19/2022  | 08/01/2022  | 1/8/2022 00:00:00     |
| 9/26/2022  | 09/01/2022  | 1/9/2022 00:00:00     |
| 10/14/2022 | 01/10/2022  | 10/1/2022 12:00:00 AM |
| 10/29/2022 | 10/01/2022  | 10/1/2022 12:00:00 AM |

Il campo 'start\_of\_month' viene creato nell'istruzione di caricamento precedente mediante l'uso della funzione `monthstart()` e trasferendo il campo data come argomento della funzione.

La funzione `monthstart()` identifica in quale mese cade il valore della data, restituendo un timestamp per il primo millisecondo di quel mese.

*Schema della funzione `monthstart()`, esempio senza argomenti aggiuntivi*



La transazione 8192 è avvenuta il 16 marzo. La funzione `monthstart()` restituisce il primo millisecondo di quel mese, ovvero il 1° marzo alle ore 12:00:00 AM.

### Esempio 2 - `period_no`

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.
- La creazione di un campo, `previous_month_start`, che restituisce il timestamp dell'inizio del mese precedente la transazione.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        monthstart(date,-1) as previous_month_start,
        timestamp(monthstart(date,-1)) as previous_month_start_timestamp
    ;

Load
*
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- previous\_month\_start
- previous\_month\_start\_timestamp

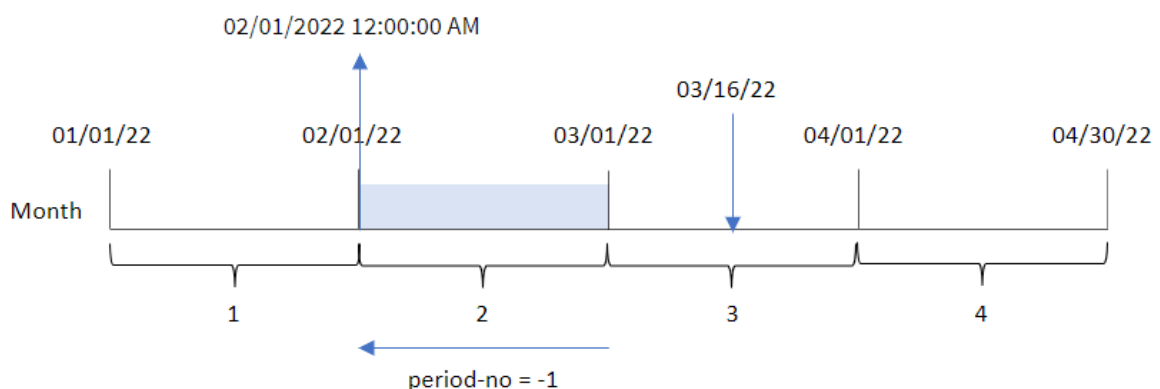
Tabella dei risultati

| date      | previous_month_start | timestamp_inizio_mese_precedente |
|-----------|----------------------|----------------------------------|
| 1/7/2022  | 01/12/2021           | 01/12/2021 00:00:00              |
| 1/19/2022 | 01/12/2021           | 01/12/2021 00:00:00              |

| date       | previous_month_start | timestamp_inizio_mese_precedente |
|------------|----------------------|----------------------------------|
| 2/5/2022   | 01/01/2022           | 01/01/2022 00:00:00              |
| 2/28/2022  | 01/01/2022           | 01/01/2022 00:00:00              |
| 3/16/2022  | 01/02/2022           | 1/2/2022 00:00:00                |
| 4/1/2022   | 03/01/2022           | 1/3/2022 00:00:00                |
| 5/7/2022   | 04/01/2022           | 01/04/2022 00:00:00              |
| 5/16/2022  | 04/01/2022           | 01/04/2022 00:00:00              |
| 6/15/2022  | 01/05/2022           | 01/05/2022 00:00:00              |
| 6/26/2022  | 01/05/2022           | 01/05/2022 00:00:00              |
| 7/9/2022   | 01/06/2022           | 01/06/2022 00:00:00              |
| 7/22/2022  | 01/06/2022           | 01/06/2022 00:00:00              |
| 7/23/2022  | 01/06/2022           | 01/06/2022 00:00:00              |
| 7/27/2022  | 01/06/2022           | 01/06/2022 00:00:00              |
| 8/2/2022   | 07/01/2022           | 01/07/2022 00:00:00              |
| 8/8/2022   | 07/01/2022           | 01/07/2022 00:00:00              |
| 8/19/2022  | 07/01/2022           | 01/07/2022 00:00:00              |
| 9/26/2022  | 01/08/2022           | 01/08/2022 00:00:00              |
| 10/14/2022 | 01/09/2022           | 9/1/2022 12:00:00 AM             |
| 10/29/2022 | 01/09/2022           | 01/09/2022 00:00:00              |

In questo caso, poiché il valore `period_no` di -1 è stato utilizzato come argomento `offset` nella funzione `monthstart()`, la funzione per prima cosa identifica il mese in cui avvengono le transazioni. Si sposta poi un mese prima e identifica il millisecondo iniziale di tale mese.

*Schema della funzione `monthstart()`, esempio di `period_no`*



La transazione 8192 è avvenuta il 16 marzo. La funzione `monthstart()` identifica che il mese precedente la transazione è avvenuta in febbraio. Quindi restituisce il primo millisecondo di quel mese, il 1° febbraio alle 12:00:00.

### Esempio 3 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor di caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento contiene lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, in questo esempio, il set di dati invariato viene caricato nell'applicazione. Il calcolo che restituisce un timestamp per l'inizio del mese in cui sono avvenute le transazioni viene creato come misura in un oggetto grafico dell'applicazione.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
```

```
8189,1/19/2022,37.23
```

```
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
```

```
8192,3/16/2022,53.80
```

```
8193,4/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/16/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
```

```
8200,7/23/2022,82.77
```

```
8201,7/27/2022,69.98
```

```
8202,8/2/2022,76.11
```

```
8203,8/8/2022,25.12
```

```
8204,8/19/2022,46.23
```

```
8205,9/26/2022,84.21
```

```
8206,10/14/2022,96.24
```

```
8207,10/29/2022,67.67
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: date.

Per calcolare la data di inizio del mese in cui avviene una transazione, creare le seguenti misure:

- `=monthstart(date)`
- `=timestamp(monthstart(date))`

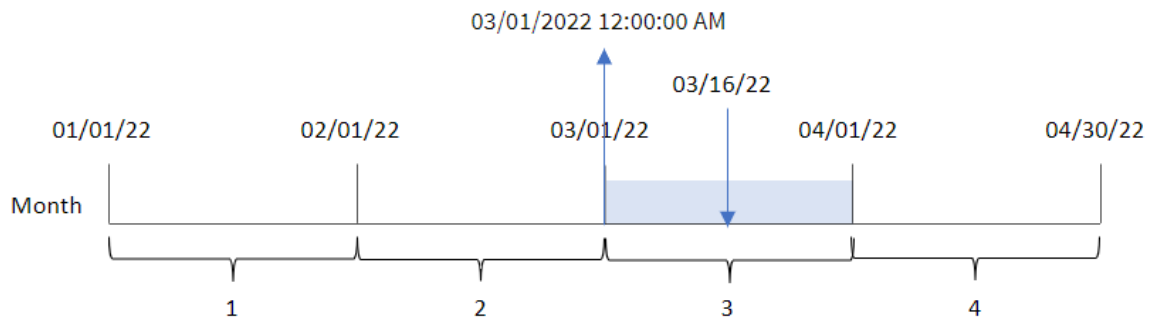
Tabella dei risultati

| date       | =monthstart(date) | =timestamp(monthstart(date)) |
|------------|-------------------|------------------------------|
| 10/14/2022 | 01/10/2022        | 10/1/2022 12:00:00 AM        |
| 10/29/2022 | 10/01/2022        | 01/10/2022 12:00:00 AM       |
| 9/26/2022  | 09/01/2022        | 01/09/2022 00:00:00          |
| 8/2/2022   | 01/08/2022        | 8/1/2022 12:00:00 AM         |
| 8/8/2022   | 08/01/2022        | 8/1/2022 12:00:00 AM         |
| 8/19/2022  | 08/01/2022        | 1/8/2022 00:00:00            |
| 7/9/2022   | 07/01/2022        | 1/7/2022 00:00:00            |
| 7/22/2022  | 07/01/2022        | 7/1/2022 12:00:00 AM         |
| 7/23/2022  | 07/01/2022        | 01/07/2022 00:00:00          |
| 7/27/2022  | 07/01/2022        | 01/07/2022 00:00:00          |
| 6/15/2022  | 01/06/2022        | 6/1/2022 12:00:00 AM         |
| 6/26/2022  | 06/01/2022        | 01/06/2022 00:00:00          |
| 5/7/2022   | 01/05/2022        | 01/05/2022 00:00:00          |
| 5/16/2022  | 01/05/2022        | 1/5/2022 00:00:00            |
| 4/1/2022   | 04/01/2022        | 1/4/2022 00:00:00            |
| 3/16/2022  | 03/01/2022        | 1/3/2022 00:00:00            |
| 2/5/2022   | 01/02/2022        | 01/02/2022 00:00:00          |
| 2/28/2022  | 01/02/2022        | 01/02/2022 00:00:00          |
| 1/7/2022   | 01/01/2022        | 1/1/2022 00:00:00            |
| 1/19/2022  | 01/01/2022        | 01/01/2022 00:00:00          |

La misura `start_of_month` viene creata nell'oggetto grafico mediante l'utilizzo della funzione `monthstart()` e trasferendo il campo data come argomento della funzione.

La funzione `monthstart()` identifica in quale mese cade il valore della data, restituendo un timestamp per il primo millisecondo di quel mese.

*Schema della funzione monthstart(), esempio di oggetto grafico*



La transazione 8192 è avvenuta il 16 marzo. La funzione monthstart() identifica che la transazione ha avuto luogo nel mese di marzo e restituisce il primo millisecondo di quel mese, ovvero il 1° marzo alle 12:00:00 AM.

### Esempio 4 - Scenario

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di saldi di prestiti, che viene caricato in una tabella chiamata Loans.
- Dati costituiti dagli ID dei prestiti, dal saldo all'inizio del mese e dal tasso di interesse semplice annuo applicato a ciascun prestito.

L'utente finale desidera un oggetto grafico che mostri, in base all'ID del prestito, gli interessi correnti che sono stati maturati per ciascun prestito nel mese a oggi.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Loans:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
loan_id,start_balance,rate
```

```
8188,$10000.00,0.024
```

```
8189,$15000.00,0.057
```

```
8190,$17500.00,0.024
```

```
8191,$21000.00,0.034
```

```
8192,$90000.00,0.084
```

```
];
```

### Risultati

Procedere come indicato di seguito:

1. Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:
  - loan\_id
  - start\_balance
2. Quindi, creare una misura per calcolare gli interessi accumulati:  
$$=start\_balance*(rate*(today(1)-monthstart(today(1)))/365)$$
3. Impostare la **Formattazione numero** della misura su **Denaro**.

Tabella dei risultati

| loan_id | saldo iniziale | =start_balance*(rate*(today(1)-yearstart(today(1)))/365) |
|---------|----------------|----------------------------------------------------------|
| 8188    | \$10000.00     | \$16.44                                                  |
| 8189    | \$15000.00     | \$58.56                                                  |
| 8190    | \$17500.00     | \$28.77                                                  |
| 8191    | \$21000.00     | \$48.90                                                  |
| 8192    | \$90000.00     | \$517.81                                                 |

Utilizzando la data odierna come unico argomento, la funzione `monthstart()` restituisce la data di inizio del mese corrente. Sottraendo tale risultato dalla data corrente, l'espressione restituisce il numero di giorni trascorsi fino ad ora questo mese.

Questo valore viene quindi moltiplicato per il tasso di interesse e diviso per 365, per restituire il tasso di interesse effettivo incorso per questo periodo. Il risultato viene poi moltiplicato per il saldo iniziale del prestito per ottenere gli interessi maturati fino a ora in questo mese.

### networkdays

La funzione **networkdays** restituisce il numero di giorni lavorativi (dal lunedì al venerdì) compresi tra e inclusi in **start\_date** e **end\_date**, tenendo in considerazione qualsiasi eventuale valore di festività **holiday** nel calendario.

Sintassi:

```
networkdays (start_date, end_date [, holiday])
```

**Tipo di dati restituiti:** numero intero

*Schema del calendario che visualizza l'intervallo di date restituito dalla funzione networkdays*

| Sun | Mon | Tue | Wed              | Thu | Fri            | Sat |
|-----|-----|-----|------------------|-----|----------------|-----|
|     | 1   | 2   | 3                | 4   | 5              | 6   |
| 7   | 8   | 9   | 10<br>start_date | 11  | 12             | 13  |
| 14  | 15  | 16  | 17               | 18  | 19             | 20  |
| 21  | 22  | 23  | 24               | 25  | 26<br>end_date | 27  |
| 28  | 29  | 30  | 31               |     |                |     |

La funzione networkdays presenta le seguenti limitazioni:

- Non esiste un metodo per modificare i giorni lavorativi. In altre parole, non c'è modo di modificare la funzione per le regioni o le situazioni che comportano un lavoro diverso dal lunedì al venerdì.
- Il parametro holiday deve essere una costante stringa. Le espressioni non sono accettate.

### Argomenti

| Argomento         | Descrizione                                                                                                                                                                                                                                                         |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>start_date</b> | La data di inizio da valutare.                                                                                                                                                                                                                                      |
| <b>end_date</b>   | La data di fine da valutare.                                                                                                                                                                                                                                        |
| <b>holiday</b>    | Periodi di vacanza da escludere dai giorni lavorativi. Una vacanza è definita come una stringa di data costante. È possibile specificare più periodi di vacanza, separati da virgole.<br><br><b>Esempio:</b> '12/25/2013', '12/26/2013', '12/31/2013', '01/01/2014' |

### Casi di utilizzo

La funzione networkdays() viene comunemente utilizzata come parte di un'espressione quando l'utente desidera che il calcolo utilizzi il numero di giorni lavorativi che intercorrono tra due date. Ad esempio, se un utente desidera calcolare il salario totale che verrà percepito da un dipendente con contratto PAYE (pay-as-you-earn).

### Esempi di funzioni

| Esempio                                                                                                       | Risultato                                                                                                    |
|---------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|
| <code>networkdays ('12/19/2013', '01/07/2014')</code>                                                         | Restituisce 14. In questo esempio, le festività non vengono prese in considerazione.                         |
| <code>networkdays ('12/19/2013', '01/07/2014', '12/25/2013', '12/26/2013')</code>                             | Restituisce 12. In questo esempio, vengono presi in considerazione i giorni festivi 12/25/2013 e 12/26/2013. |
| <code>networkdays ('12/19/2013', '01/07/2014', '12/25/2013', '12/26/2013', '12/31/2013', '01/01/2014')</code> | Restituisce 10. In questo esempio, vengono presi in considerazione due periodi di festività.                 |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - Esempio di base

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente gli ID dei progetti, le date di inizio e di fine. Queste informazioni vengono caricate in una tabella denominata `Projects`.
- Il campo della data fornito nel formato della variabile di sistema `DateFormat` (MM/GG/AAAA).
- La creazione di un campo aggiuntivo, `net_work_days`, per calcolare il numero di giorni lavorativi coinvolti in ogni progetto.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Projects:
    Load
        *,
        networkdays(start_date,end_date) as net_work_days
    ;

Load
id,
start_date,
end_date
Inline
[
id,start_date,end_date
1,01/01/2022,01/18/2022
2,02/10/2022,02/17/2022
3,05/17/2022,07/05/2022
4,06/01/2022,06/12/2022
5,08/10/2022,08/26/2022
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- start\_date
- end\_date
- net\_work\_days

Tabella dei risultati

| id | data_inizio | data_fine  | giorni_lavorativi_netti |
|----|-------------|------------|-------------------------|
| 1  | 01/01/2022  | 18/01/2022 | 12                      |
| 2  | 10/02/2022  | 17/02/2022 | 6                       |
| 3  | 05/17/2022  | 05/07/2022 | 36                      |
| 4  | 01/06/2022  | 12/06/2022 | 8                       |
| 5  | 10/08/2022  | 26/08/2022 | 13                      |

Poiché non ci sono festività programmate (che sarebbero state presenti nel terzo argomento della funzione `networkdays()`), la funzione sottrae il valore `start_date` da `end_date`, così come tutti i fine settimana, per calcolare il numero di giorni lavorativi tra le due date.

*Schema di calendario che evidenzia i giorni di lavoro per il progetto 5 (senza giorni festivi)*

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     | 1   | 2   | 3   | 4   | 5   | 6   |
| 7   | 8   | 9   | 10  | 11  | 12  | 13  |
| 14  | 15  | 16  | 17  | 18  | 19  | 20  |
| 21  | 22  | 23  | 24  | 25  | 26  | 27  |
| 28  | 29  | 30  | 31  |     |     |     |

Il calendario qui sopra illustra visivamente il progetto con un valore `id` di 5. Il progetto 5 inizia mercoledì 10 agosto 2022 e termina il 26 agosto 2022. Ignorando tutti i sabati e le domeniche, sono presenti 13 giorni lavorativi tra queste due date.

### Esempio 2 - Singolo giorno festivo

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati e lo stesso scenario ricavati dal precedente esempio.
- Il campo della data fornito nel formato della variabile di sistema `DateFormat` (MM/GG/AAAA).
- La creazione di un campo aggiuntivo, `net_work_days`, per calcolare il numero di giorni lavorativi coinvolti in ogni progetto.

In questo esempio, il 19 agosto 2022 è previsto un giorno di vacanza.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

Projects:

```
Load
    *,
    networkdays(start_date,end_date,'08/19/2022') as net_work_days
;

Load
id,
start_date,
end_date
Inline
[
id,start_date,end_date
1,01/01/2022,01/18/2022
2,02/10/2022,02/17/2022
3,05/17/2022,07/05/2022
4,06/01/2022,06/12/2022
5,08/10/2022,08/26/2022
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- start\_date
- end\_date
- net\_work\_days

Tabella dei risultati

| id | data_inizio | data_fine  | giorni_lavorativi_netti |
|----|-------------|------------|-------------------------|
| 1  | 01/01/2022  | 18/01/2022 | 12                      |
| 2  | 10/02/2022  | 17/02/2022 | 6                       |
| 3  | 05/17/2022  | 05/07/2022 | 36                      |
| 4  | 01/06/2022  | 12/06/2022 | 8                       |
| 5  | 10/08/2022  | 26/08/2022 | 12                      |

La singola vacanza programmata viene inserita come terzo argomento della funzione `networkdays()`.

*Schema di calendario che evidenzia i giorni di lavoro per il progetto 5 (con un giorno festivo)*

| Sun | Mon | Tue | Wed | Thu | Fri           | Sat |
|-----|-----|-----|-----|-----|---------------|-----|
|     | 1   | 2   | 3   | 4   | 5             | 6   |
| 7   | 8   | 9   | 10  | 11  | 12            | 13  |
| 14  | 15  | 16  | 17  | 18  | 19<br>Holiday | 20  |
| 21  | 22  | 23  | 24  | 25  | 26            | 27  |
| 28  | 29  | 30  | 31  |     |               |     |

Il calendario qui sopra illustra visivamente il progetto 5, dimostrando questo adattamento per includere il giorno di vacanza. Questa vacanza si verifica durante il progetto 5, venerdì 19 agosto 2022. Di conseguenza, il valore `net_work_days` totale del progetto 5 diminuisce di un giorno, passando da 13 a 12 giorni.

### Esempio 3 - Vacanze multiple

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati e lo stesso scenario ricavato dal primo esempio.
- Il campo della data fornito nel formato della variabile di sistema `DateFormat (MM/GG/AAAA)`.
- La creazione di un campo aggiuntivo, `net_work_days`, per calcolare il numero di giorni lavorativi coinvolti in ogni progetto.

Tuttavia, in questo esempio sono previsti quattro giorni di vacanza, dal 18 al 21 agosto 2022.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';

Projects:
    Load
        *,
        networkdays(start_date,end_date,'08/18/2022','08/19/2022','08/20/2022','08/21/2022')
    as net_work_days
    ;

Load
id,
start_date,
end_date
Inline
[
id,start_date,end_date
1,01/01/2022,01/18/2022
2,02/10/2022,02/17/2022
3,05/17/2022,07/05/2022
4,06/01/2022,06/12/2022
5,08/10/2022,08/26/2022
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- start\_date
- end\_date
- net\_work\_days

Tabella dei risultati

| id | data_inizio | data_fine  | giorni_lavorativi_netti |
|----|-------------|------------|-------------------------|
| 1  | 01/01/2022  | 18/01/2022 | 12                      |
| 2  | 10/02/2022  | 17/02/2022 | 6                       |
| 3  | 05/17/2022  | 05/07/2022 | 36                      |
| 4  | 01/06/2022  | 12/06/2022 | 8                       |
| 5  | 10/08/2022  | 26/08/2022 | 11                      |

I quattro giorni di vacanza programmati vengono inseriti come elenco separato da virgole, a partire dal terzo argomento della funzione `networkdays()`.

*Schema di calendario che evidenzia i giorni di lavoro per il progetto 5 (con più giorni di vacanza)*

| Sun | Mon | Tue | Wed | Thu           | Fri           | Sat |
|-----|-----|-----|-----|---------------|---------------|-----|
|     | 1   | 2   | 3   | 4             | 5             | 6   |
| 7   | 8   | 9   | 10  | 11            | 12            | 13  |
| 14  | 15  | 16  | 17  | 18<br>Holiday | 19<br>Holiday | 20  |
| 21  | 22  | 23  | 24  | 25            | 26            | 27  |
| 28  | 29  | 30  | 31  |               |               |     |

Il calendario qui sopra illustra visivamente il progetto 5, dimostrando questo adattamento per includere i giorni di vacanza. Questo periodo di ferie programmate si verifica durante il progetto 5, con due giorni di giovedì e venerdì. Di conseguenza, il valore totale `net_work_days` del progetto 5 diminuisce da 13 a 11 giorni.

### Esempio 4 - Singolo giorno di vacanza

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati e lo stesso scenario ricavato dal primo esempio.
- Il campo della data fornito nel formato della variabile di sistema `DateFormat` (MM/GG/AAAA).

Il 19 agosto 2022 è previsto un giorno di vacanza.

Tuttavia, in questo esempio, il set di dati invariato viene caricato nell'applicazione. Il campo `net_work_days` viene calcolato come misura in un oggetto grafico.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Projects:
```

```
Load
```

```
id,
```

```
start_date,
```

```
end_date
```

```
Inline
```

```
[
```

```
id,start_date,end_date
```

```
1,01/01/2022,01/18/2022
```

```
2,02/10/2022,02/17/2022
```

```
3,05/17/2022,07/05/2022
```

```
4,06/01/2022,06/12/2022
```

```
5,08/10/2022,08/26/2022
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `id`
- `start_date`
- `end_date`

Creare la seguente misura:

```
= networkdays(start_date,end_date,'08/19/2022')
```

Tabella dei risultati

| id | data_inizio | data_fine  | giorni_lavorativi_netti |
|----|-------------|------------|-------------------------|
| 1  | 01/01/2022  | 18/01/2022 | 12                      |
| 2  | 10/02/2022  | 17/02/2022 | 6                       |
| 3  | 05/17/2022  | 05/07/2022 | 36                      |
| 4  | 01/06/2022  | 12/06/2022 | 8                       |
| 5  | 10/08/2022  | 26/08/2022 | 12                      |

La singola vacanza programmata viene inserita come terzo argomento della funzione `networkdays()`.

## 8 Funzioni per script e grafici

*Schema del calendario che mostra i giorni di lavoro netti con una singola festività (oggetto grafico)*

| Sun | Mon | Tue | Wed | Thu | Fri           | Sat |
|-----|-----|-----|-----|-----|---------------|-----|
|     | 1   | 2   | 3   | 4   | 5             | 6   |
| 7   | 8   | 9   | 10  | 11  | 12            | 13  |
| 14  | 15  | 16  | 17  | 18  | 19<br>Holiday | 20  |
| 21  | 22  | 23  | 24  | 25  | 26            | 27  |
| 28  | 29  | 30  | 31  |     |               |     |

Il calendario qui sopra illustra visivamente il progetto 5, dimostrando questo adattamento per includere il giorno di vacanza. Questa vacanza si verifica durante il progetto 5, venerdì 19 agosto 2022. Di conseguenza, il valore `net_work_days` totale del progetto 5 diminuisce di un giorno, passando da 13 a 12 giorni.

### now

Questa funzione restituisce un indicatore temporale dell'ora attuale. La funzione restituisce i valori nel formato della variabile di sistema **TimeStamp**. Il valore predefinito di `timer_mode` è 1.

#### Sintassi:

```
now([ timer_mode])
```

#### Tipo di dati restituiti: duale

La funzione `now()` può essere utilizzata sia nello script di caricamento che negli oggetti del grafico.

## Argomenti

| Argomento  | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| timer_mode | <p>Può presentare i valori seguenti:</p> <ul style="list-style-type: none"> <li>0 (ora al momento dell'ultimo caricamento di dati completato)</li> <li>1 (ora al momento della chiamata della funzione)</li> <li>2 (ora di apertura dell'app)</li> </ul> <div>  <p>Se si utilizza la funzione in uno script di caricamento dei dati, <b>timer_mode=0</b> restituirà l'ora del completamento dell'ultimo caricamento dei dati, mentre <b>timer_mode=1</b> restituirà l'ora della chiamata di funzione nel caricamento dei dati attuale.</p> </div> |



La funzione `now()` ha un impatto a prestazioni elevate, che è possibile causare problemi di scorrimento se è utilizzata nelle espressioni delle tabelle. Quando il suo utilizzo non è strettamente necessario, si consiglia di preferire la funzione `today()`. Se l'utilizzo di `now()` è richiesto in un layout, si consiglia di utilizzare le impostazioni `now(0)` o `now(2)` non predefinite quando possibile, poiché non richiedono calcoli costanti.

## Casi di utilizzo

La funzione `now()` viene comunemente utilizzata come componente di un'espressione. Ad esempio, può essere utilizzato per calcolare il tempo rimanente nel ciclo di vita di un prodotto. La funzione `now()` verrebbe utilizzata al posto della funzione `today()` quando l'espressione richiede l'uso di una frazione di giorno.

La tabella seguente fornisce una spiegazione del risultato restituito dalla funzione `now()`, in presenza di diversi valori dell'argomento `timer_mode`:

## Esempi di funzioni

| valore timer_mode | Risultato se utilizzato nello script di caricamento                                                                                                                             | Risultato se utilizzato nell'oggetto grafico                                                                                   |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| 0                 | Restituisce un timestamp, nel formato della variabile di sistema <code>timestamp</code> , dell'ultimo ricaricamento dei dati riuscito prima dell'ultimo ricaricamento dei dati. | Restituisce un timestamp, nel formato della variabile di sistema <code>timestamp</code> , per l'ultimo ricaricamento dei dati. |
| 1                 | Restituisce un timestamp, nel formato della variabile di sistema <code>timestamp</code> , per l'ultimo ricaricamento dei dati.                                                  | Restituisce un timestamp, nel formato della variabile di sistema <code>timestamp</code> , della chiamata alla funzione.        |

| valore timer_mode | Risultato se utilizzato nello script di caricamento                                                                                                                                                                    | Risultato se utilizzato nell'oggetto grafico                                                                                                                                                                                                                                |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2                 | Restituisce un timestamp, nel formato della variabile di sistema <code>timestamp</code> , per l'inizio della sessione dell'utente nell'applicazione. Non verrà aggiornato a meno che l'utente non ricarichi lo script. | Restituisce il timestamp, nel formato della variabile di sistema <code>timestamp</code> , per l'inizio della sessione dell'utente nell'applicazione. Questo valore verrà aggiornato all'inizio di una nuova sessione o quando i dati dell'applicazione verranno ricaricati. |

## Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

## Esempio 1 - Generazione di oggetti tramite script di caricamento

Script di caricamento e risultati

### Panoramica

Aprire l'Editor di caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Questo esempio crea tre variabili utilizzando la funzione `now()`. Ogni variabile utilizza una delle opzioni `timer_mode` per dimostrarne l'effetto.

Affinché le variabili dimostrino il loro scopo, e lo script e poi, dopo un breve periodo di tempo, ricaricare lo script una seconda volta. In questo modo le variabili `now(0)` e `now(1)` mostreranno valori diversi, dimostrando così correttamente il loro scopo.

### Script di caricamento

```
LET vPreviousDataLoad = now(0);
LET vCurrentDataLoad = now(1);
LET vApplicationOpened = now(2);
```

### Risultati

Una volta che i dati vengono caricati una seconda volta, creare tre caselle di testo seguendo le indicazioni riportate di seguito.

Per prima cosa, creare una casella di testo per i dati che sono stati caricati precedentemente.

#### Procedere come indicato di seguito:

1. Utilizzando l'oggetto grafico **Testo e immagine**, creare una casella di testo.
2. Aggiungere la seguente misura all'oggetto:  
`=vPreviousDataLoad`
3. In **Aspetto**, selezionare **Show titles** e aggiungere il titolo 'Ora di caricamento precedente' all'oggetto.

Quindi, creare una casella di testo per i dati che si stanno caricando.

#### Procedere come indicato di seguito:

1. Utilizzando l'oggetto grafico **Testo e immagine**, creare una casella di testo.
2. Aggiungere la seguente misura all'oggetto:  
`=vCurrentDataLoad`
3. In **Aspetto**, selezionare **Show titles** e aggiungere il titolo 'Ora di caricamento attuale' all'oggetto.

Creare una casella di testo finale da mostrare quando è stata avviata la sessione dell'utente nell'applicazione.

#### Procedere come indicato di seguito:

1. Utilizzando l'oggetto grafico **Testo e immagine**, creare una casella di testo.
2. Aggiungere la seguente misura all'oggetto:  
`=vApplicationOpened`
3. In **Aspetto**, selezionare **Show titles** e aggiungere il titolo 'Inizio sessione utente' all'oggetto.

*Variabili dello script di caricamento now()*

| Previous Reload Time | Current Reload Time  | User Session Began   |
|----------------------|----------------------|----------------------|
| 6/22/2022 8:54:03 AM | 6/22/2022 9:02:08 AM | 6/22/2022 8:40:40 AM |

L'immagine qui sopra mostra dei valori di esempio per ciascuna delle variabili create. Ad esempio, i valori potrebbero essere i seguenti:

- Ora di ricaricamento precedente: 22/06/2022 8:54:03 AM
- Ora di ricaricamento corrente: 22/06/2022 9:02:08 AM
- Inizio della sessione utente: 22/06/2022 8:40:40 AM

### Esempio 2 - Generazione di oggetti senza script di caricamento

Script di caricamento ed espressione del grafico

#### Panoramica

In questo esempio, si creeranno tre oggetti grafico utilizzando la funzione `now()`, senza caricare alcuna variabile o dato nell'applicazione. Ogni oggetto del grafico utilizza una delle opzioni `timer_mode` per dimostrarne l'effetto.

Per questo esempio non esiste uno script di caricamento.

#### Procedere come indicato di seguito:

1. Aprire l'Editor caricamento dati.
2. Senza modificare lo script di caricamento esistente, fare clic su **Carica dati**.
3. Dopo un breve periodo di tempo, caricare lo script una seconda volta.

#### Risultati

Una volta che i dati vengono caricati una seconda volta, creare tre caselle di testo.

Per prima cosa, creare una casella di testo per l'ultimo ricaricamento di dati.

#### Procedere come indicato di seguito:

1. Utilizzando l'oggetto grafico **Testo e immagine**, creare una casella di testo.
2. Aggiungere la seguente misura:  
`=now(0)`
3. Nella sezione **Aspetto**, selezionare **Mostra titoli** e aggiungere il titolo 'Ricaricamento ultimi dati' all'oggetto.

Quindi, creare una casella di testo per mostrare l'ora corrente.

#### Procedere come indicato di seguito:

1. Utilizzando l'oggetto grafico **Testo e immagine**, creare una casella di testo.
2. Aggiungere la misura seguente:  
`=now(1)`
3. In **Aspetto**, selezionare **Mostra titoli** e aggiungere il titolo 'Ora attuale' all'oggetto.

Creare una casella di testo finale da mostrare quando è stata avviata la sessione dell'utente nell'applicazione.

Procedere come indicato di seguito:

1. Utilizzando l'oggetto grafico **Testo e immagine**, creare una casella di testo.
2. Aggiungere la misura seguente:  
`=now(2)`
3. In **Aspetto**, selezionare **Mostra titoli** e aggiungere il titolo 'Sessione utente iniziata' all'oggetto.

Esempi oggetto grafico `now()`

|                                                   |                                             |                                                   |
|---------------------------------------------------|---------------------------------------------|---------------------------------------------------|
| <b>Latest Data Reload</b><br>6/22/2022 9:02:08 AM | <b>Current Time</b><br>6/22/2022 9:25:16 AM | <b>User Session Began</b><br>6/22/2022 8:40:40 AM |
|---------------------------------------------------|---------------------------------------------|---------------------------------------------------|

L'immagine qui sopra mostra dei valori di esempio per ciascuno degli oggetti creati. Ad esempio, i valori potrebbero essere i seguenti:

- Dati ultimo ricaricamento: 22/6/2022 9:02:08 AM
- Ora corrente: 22/6/2022 9:25:16 AM
- Inizio della sessione utente: 22/06/2022 8:40:40 AM

L'oggetto grafico 'Ultimo caricamento dati' utilizza un valore `timer_mode` pari a 0. Restituisce il timestamp dell'ultima volta in cui i dati sono stati ricaricati correttamente.

L'oggetto grafico 'Ora corrente' utilizza un valore `timer_mode` di 1. Questo restituisce l'ora corrente secondo l'orologio di sistema. Se il foglio o l'oggetto viene ricaricato, questo valore verrà aggiornato.

L'oggetto grafico 'Sessione utente iniziata' utilizza un valore `timer_mode` di 2. Restituisce il timestamp di quando l'applicazione è stata aperta e la sessione dell'utente è iniziata.

### Esempio 3 - Scenario

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati costituito dall'inventario di un'operazione di estrazione di criptovalute, che viene caricato in una tabella denominata `Inventory`.
- Dati con i seguenti campi: `id`, `purchase_date` e `wph` (watt all'ora).

L'utente vorrebbe una tabella che visualizzi, per `id`, il costo totale che ogni impianto di mining ha sostenuto nel mese fino a quel momento, in termini di consumo energetico.

Questo valore deve essere aggiornato ogni volta che l'oggetto grafico viene aggiornato. Il costo attuale dell'elettricità è di 0,0678 dollari per kWh.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Inventory:
Load
*
Inline
[
id,purchase_date,wph
8188,1/7/2022,1123
8189,1/19/2022,1432
8190,2/28/2022,1227
8191,2/5/2022,1322
8192,3/16/2022,1273
8193,4/1/2022,1123
8194,5/7/2022,1342
8195,5/16/2022,2342
8196,6/15/2022,1231
8197,6/26/2022,1231
8198,7/9/2022,1123
8199,7/22/2022,1212
8200,7/23/2022,1223
8201,7/27/2022,1232
8202,8/2/2022,1232
8203,8/8/2022,1211
8204,8/19/2022,1243
8205,9/26/2022,1322
8206,10/14/2022,1133
8207,10/29/2022,1231
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: id.

Creare la seguente misura:

```
=(now(1)-monthstart(now(1)))*24*wph/1000*0.0678
```

Se l'oggetto grafico viene aggiornato il 22/06/2022 alle 10:39:05, i risultati sono i seguenti:

Tabella dei risultati

| id   | =(now(1)-monthstart(now(1)))*24*wph/1000*0.0678 |
|------|-------------------------------------------------|
| 8188 | \$39.18                                         |
| 8189 | \$49.97                                         |

| id   | $=(\text{now}(1)-\text{monthstart}(\text{now}(1)))*24*\text{wph}/1000*0.0678$ |
|------|-------------------------------------------------------------------------------|
| 8190 | \$42.81                                                                       |
| 8191 | \$46.13                                                                       |
| 8192 | \$44.42                                                                       |
| 8193 | \$39.18                                                                       |
| 8194 | \$46.83                                                                       |
| 8195 | \$81.72                                                                       |
| 8196 | \$42.95                                                                       |
| 8197 | \$42.95                                                                       |
| 8198 | \$39.18                                                                       |
| 8199 | \$42.29                                                                       |
| 8200 | \$42.67                                                                       |
| 8201 | \$42.99                                                                       |
| 8202 | \$42.99                                                                       |
| 8203 | \$42.25                                                                       |
| 8204 | \$43.37                                                                       |
| 8205 | \$46.13                                                                       |
| 8206 | \$39.53                                                                       |

L'utente desidera che i risultati dell'oggetto vengano aggiornati ogni volta che si aggiorna l'oggetto. Pertanto, l'argomento `timer_mode` viene fornito per le istanze della funzione `now()` nell'espressione. Il timestamp dell'inizio del mese, identificato dalla funzione `now()` come argomento timestamp nella funzione `monthstart()`, viene sottratto dall'ora corrente identificata dalla funzione `now()`. Fornisce il tempo totale trascorso finora nel mese, in giorni.

Questo valore viene moltiplicato per 24 (il numero di ore in un giorno) e quindi per il valore del campo `wph`.

Per convertire i watt all'ora in kilowatt all'ora, il risultato viene diviso per 1000 prima di essere moltiplicato per la tariffa kWh fornita.

## quarterend

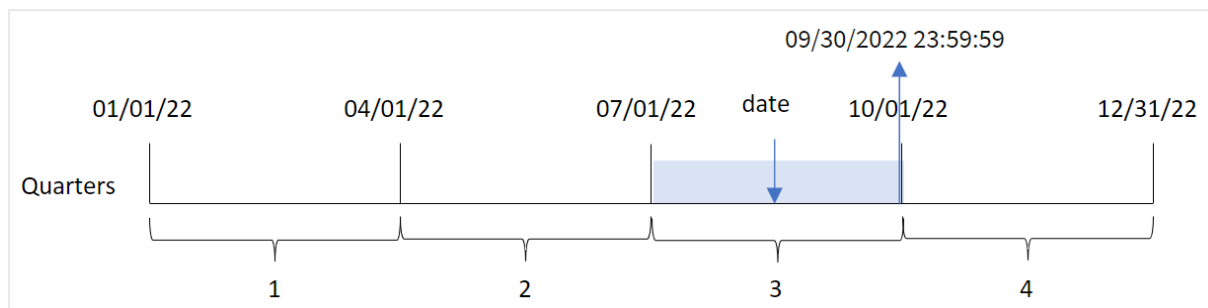
Questa funzione restituisce un valore corrispondente a un indicatore temporale recante l'ultimo millisecondo del trimestre contenente **date**. Il formato di output predefinito sarà il formato **DateFormat** impostato nello script.

### Sintassi:

```
QuarterEnd(date[, period_no[, first_month_of_year]])
```

Tipo di dati restituiti: duale

Schema della funzione `quarterend()`



In altre parole, la funzione `quarterend()` determina in quale trimestre ricadrà la data. Quindi restituisce un timestamp, in formato data, per l'ultimo millisecondo dell'ultimo mese di quel trimestre. Il primo mese dell'anno è, per impostazione predefinita, gennaio. Tuttavia, è possibile modificare il mese da impostare come primo utilizzando l'argomento `first_month_of_year` nella funzione `quarterend()`.



La funzione `quarterend()` non considera la variabile di sistema `FirstMonthOfYear`. L'anno inizia il 1 gennaio a condizione che l'argomento `first_month_of_year` non venga utilizzato per modificarlo.

### Casi di utilizzo

La funzione `quarterend()` viene comunemente utilizzata come parte di un'espressione quando si desidera che il calcolo utilizzi la frazione del trimestre non ancora trascorsa. Ad esempio, se si vuole calcolare l'interesse totale non ancora maturato durante il trimestre.

#### Argomenti

| Argomento                  | Descrizione                                                                                                                                                                                                                             |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>date</b>                | La data o la data e ora da valutare.                                                                                                                                                                                                    |
| <b>period_no</b>           | <b>period_no</b> è un numero intero, in cui il valore 0 indica il trimestre che contiene <b>date</b> . I valori negativi di <b>period_no</b> indicano i trimestri precedenti, mentre i valori positivi indicano i trimestri successivi. |
| <b>first_month_of_year</b> | Se si intende utilizzare anni (fiscali) che non iniziano a gennaio, indicare un valore compreso tra 2 e 12 in <b>first_month_of_year</b> .                                                                                              |

È possibile utilizzare i seguenti valori per impostare il primo mese dell'anno nell'argomento `first_month_of_year`:

valori first\_month\_of\_year

| Mese      | Valore |
|-----------|--------|
| Febbraio  | 2      |
| Marzo     | 3      |
| Aprile    | 4      |
| Maggio    | 5      |
| Giugno    | 6      |
| Luglio    | 7      |
| Agosto    | 8      |
| Settembre | 9      |
| Ottobre   | 10     |
| Novembre  | 11     |
| Dicembre  | 12     |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

#### Esempi di funzioni

| Esempio                                     | Risultato                        |
|---------------------------------------------|----------------------------------|
| <code>quarterend('10/29/2005')</code>       | Restituisce 12/31/2005 23:59:59. |
| <code>quarterend('10/29/2005', -1)</code>   | Restituisce 09/30/2005 23:59:59. |
| <code>quarterend('10/29/2005', 0, 3)</code> | Restituisce 11/30/2005 23:59:59. |

### Esempio 1 - Esempio di base

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2022, caricato in una tabella denominata 'Transactions'.
- Un'istruzione di caricamento precedente che contiene i seguenti elementi:
  - La funzione `quarterend()` impostata come campo `'end_of_quarter'` restituisce un timestamp per la fine del trimestre in cui sono avvenute le transazioni.
  - La funzione `timestamp()` impostata come campo `'end_of_quarter_timestamp'` restituisce il timestamp esatto della fine del trimestre selezionato.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        quarterend(date) as end_of_quarter,
        timestamp(quarterend(date)) as end_of_quarter_timestamp
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
```

```
8189,1/19/2022,37.23
```

```
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
```

```
8192,3/16/2022,53.80
```

```
8193,4/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/16/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
```

```
8200,7/23/2022,82.77
```

```
8201,7/27/2022,69.98
```

```
8202,8/2/2022,76.11
```

```
8203,8/8/2022,25.12
```

```
8204,8/19/2022,46.23
```

```
8205,9/26/2022,84.21
```

```
8206,10/14/2022,96.24
```

```
8207,10/29/2022,67.67  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- date
- end\_of\_quarter
- end\_of\_quarter\_timestamp

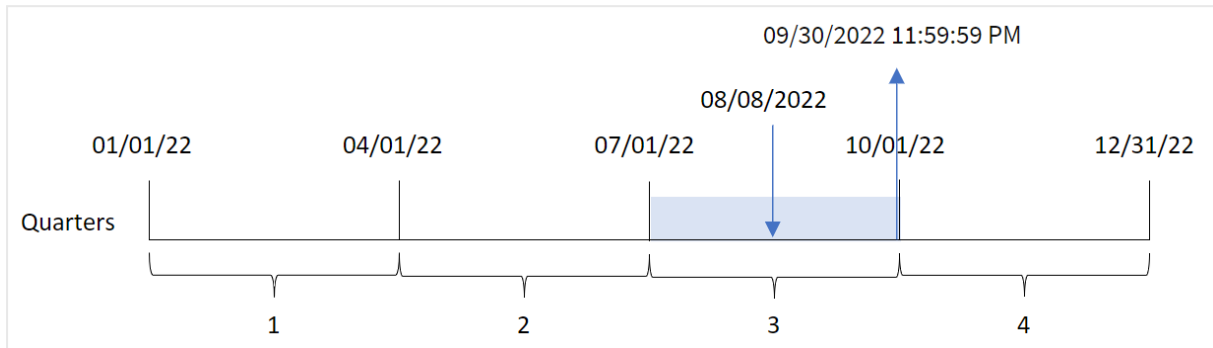
Tabella dei risultati

| id   | date       | end_of_quarter | end_of_quarter_timestamp |
|------|------------|----------------|--------------------------|
| 8188 | 1/7/2022   | 03/31/2022     | 3/31/2022 11:59:59 PM    |
| 8189 | 1/19/2022  | 03/31/2022     | 3/31/2022 11:59:59 PM    |
| 8190 | 2/5/2022   | 03/31/2022     | 3/31/2022 11:59:59 PM    |
| 8191 | 2/28/2022  | 03/31/2022     | 3/31/2022 11:59:59 PM    |
| 8192 | 3/16/2022  | 03/31/2022     | 3/31/2022 11:59:59 PM    |
| 8193 | 4/1/2022   | 06/30/2022     | 6/30/2022 11:59:59 PM    |
| 8194 | 5/7/2022   | 06/30/2022     | 6/30/2022 11:59:59 PM    |
| 8195 | 5/16/2022  | 06/30/2022     | 6/30/2022 11:59:59 PM    |
| 8196 | 6/15/2022  | 06/30/2022     | 6/30/2022 11:59:59 PM    |
| 8197 | 6/26/2022  | 06/30/2022     | 6/30/2022 11:59:59 PM    |
| 8198 | 7/9/2022   | 09/30/2022     | 9/30/2022 11:59:59 PM    |
| 8199 | 7/22/2022  | 09/30/2022     | 9/30/2022 11:59:59 PM    |
| 8200 | 7/23/2022  | 09/30/2022     | 9/30/2022 11:59:59 PM    |
| 8201 | 7/27/2022  | 09/30/2022     | 9/30/2022 11:59:59 PM    |
| 8202 | 8/2/2022   | 09/30/2022     | 9/30/2022 11:59:59 PM    |
| 8203 | 8/8/2022   | 09/30/2022     | 9/30/2022 11:59:59 PM    |
| 8204 | 8/19/2022  | 09/30/2022     | 9/30/2022 11:59:59 PM    |
| 8205 | 9/26/2022  | 09/30/2022     | 9/30/2022 11:59:59 PM    |
| 8206 | 10/14/2022 | 12/31/2022     | 12/31/2022 11:59:59 PM   |
| 8207 | 10/29/2022 | 12/31/2022     | 12/31/2022 11:59:59 PM   |

Il campo 'end\_of\_quarter' viene creato nell'istruzione di caricamento precedente mediante l'uso della funzione `quarterend()` e trasferendo il campo `data` come argomento della funzione.

La funzione `quarterend()` inizialmente identifica in quale trimestre ricada il valore della data e quindi restituisce un timestamp per l'ultimo millisecondo di quel trimestre.

*Schema della funzione `quarterend()` con identificazione della fine del trimestre della transazione 8203*



La transazione 8203 è avvenuta l'8 agosto. La funzione `quarterend()` identifica che la transazione è avvenuta nel terzo trimestre e restituisce l'ultimo millisecondo di quel trimestre, ovvero il 30 settembre alle 23:59:59.

### Esempio 2 - `period_no`

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2022, caricato in una tabella denominata 'Transactions'.
- Un'istruzione `LOAD` precedente che contiene i seguenti elementi:
  - La funzione `quarterend()` impostata come campo `'previous_quarter_end'` restituisce un timestamp per la fine del trimestre precedente la transazione.
  - La funzione `timestamp()` impostata come campo `'previous_end_of_quarter_timestamp'` restituisce il timestamp esatto della fine del trimestre precedente la transazione.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        quarterend(date, -1) as previous_quarter_end,
        timestamp(quarterend(date, -1)) as previous_quarter_end_timestamp
    ;
Load
*
Inline
```

```
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- date
- previous\_quarter\_end
- previous\_quarter\_end\_timestamp

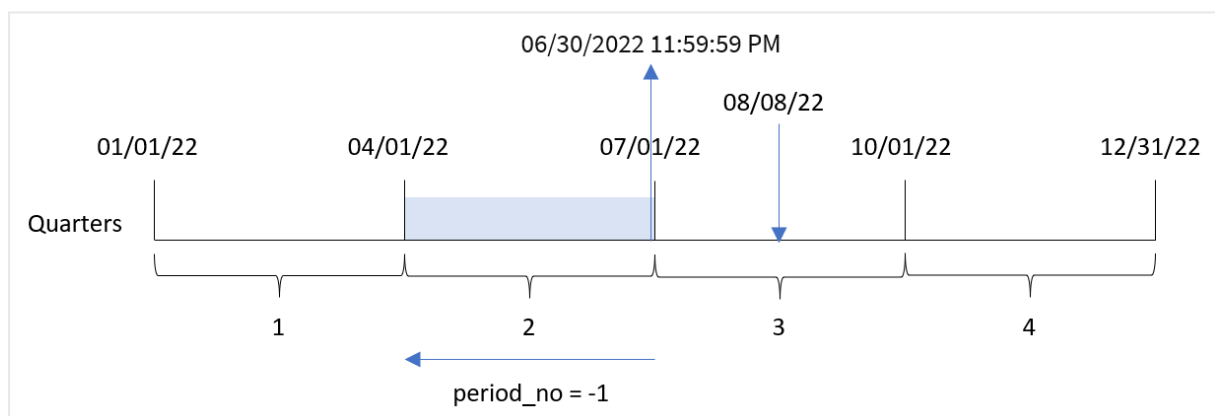
Tabella dei risultati

| id   | date      | previous_quarter_end | previous_quarter_end_timestamp |
|------|-----------|----------------------|--------------------------------|
| 8188 | 1/7/2022  | 12/31/2021           | 12/31/2021 11:59:59 PM         |
| 8189 | 1/19/2022 | 12/31/2021           | 12/31/2021 11:59:59 PM         |
| 8190 | 2/5/2022  | 12/31/2021           | 12/31/2021 11:59:59 PM         |
| 8191 | 2/28/2022 | 12/31/2021           | 12/31/2021 11:59:59 PM         |
| 8192 | 3/16/2022 | 12/31/2021           | 12/31/2021 11:59:59 PM         |
| 8193 | 4/1/2022  | 03/31/2022           | 3/31/2022 11:59:59 PM          |
| 8194 | 5/7/2022  | 03/31/2022           | 3/31/2022 11:59:59 PM          |
| 8195 | 5/16/2022 | 03/31/2022           | 3/31/2022 11:59:59 PM          |
| 8196 | 6/15/2022 | 03/31/2022           | 3/31/2022 11:59:59 PM          |

| id   | date       | previous_quarter_end | previous_quarter_end_timestamp |
|------|------------|----------------------|--------------------------------|
| 8197 | 6/26/2022  | 03/31/2022           | 3/31/2022 11:59:59 PM          |
| 8198 | 7/9/2022   | 06/30/2022           | 6/30/2022 11:59:59 PM          |
| 8199 | 7/22/2022  | 06/30/2022           | 6/30/2022 11:59:59 PM          |
| 8200 | 7/23/2022  | 06/30/2022           | 6/30/2022 11:59:59 PM          |
| 8201 | 7/27/2022  | 06/30/2022           | 6/30/2022 11:59:59 PM          |
| 8202 | 8/2/2022   | 06/30/2022           | 6/30/2022 11:59:59 PM          |
| 8203 | 8/8/2022   | 06/30/2022           | 6/30/2022 11:59:59 PM          |
| 8204 | 8/19/2022  | 06/30/2022           | 6/30/2022 11:59:59 PM          |
| 8205 | 9/26/2022  | 06/30/2022           | 6/30/2022 11:59:59 PM          |
| 8206 | 10/14/2022 | 09/30/2022           | 9/30/2022 11:59:59 PM          |
| 8207 | 10/29/2022 | 09/30/2022           | 9/30/2022 11:59:59 PM          |

Poiché il valore `period_no` di `-1` viene utilizzato come argomento `offset` nella funzione `quarterend()`, la funzione per prima cosa identifica il trimestre in cui avvengono le transazioni. Si sposta poi un trimestre prima e identifica il millisecondo finale di tale trimestre.

*Schema della funzione `quarterend()` con un valore `period_no` di `-1`*



La transazione 8203 è avvenuta l'8 agosto. La funzione `quarterend()` identifica che il trimestre precedente la transazione era compreso tra il 1° aprile e il 30 giugno. La funzione restituisce quindi il millisecondo finale di quel trimestre, il 30 giugno alle 23:59:59.

### Esempio 3 - `first_month_of_year`

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2022, caricato in una tabella denominata 'Transactions'.
- Un'istruzione LOAD precedente che contiene i seguenti elementi:
  - La funzione `quarterend()` impostata come campo `'end_of_quarter'` restituisce un timestamp per la fine del trimestre in cui sono avvenute le transazioni.
  - La funzione `timestamp()` impostata come campo `'end_of_quarter_timestamp'` restituisce il timestamp esatto della fine del trimestre selezionato.

Tuttavia, in questo esempio, la politica aziendale prevede che l'esercizio finanziario inizi il 1° marzo.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        quarterend(date, 0, 3) as end_of_quarter,
        timestamp(quarterend(date, 0, 3)) as end_of_quarter_timestamp
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
```

```
8189,1/19/2022,37.23
```

```
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
```

```
8192,3/16/2022,53.80
```

```
8193,4/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/16/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
```

```
8200,7/23/2022,82.77
```

```
8201,7/27/2022,69.98
```

```
8202,8/2/2022,76.11
```

```
8203,8/8/2022,25.12
```

```
8204,8/19/2022,46.23
```

```
8205,9/26/2022,84.21
```

```
8206,10/14/2022,96.24
```

```
8207,10/29/2022,67.67
```

```
];
```

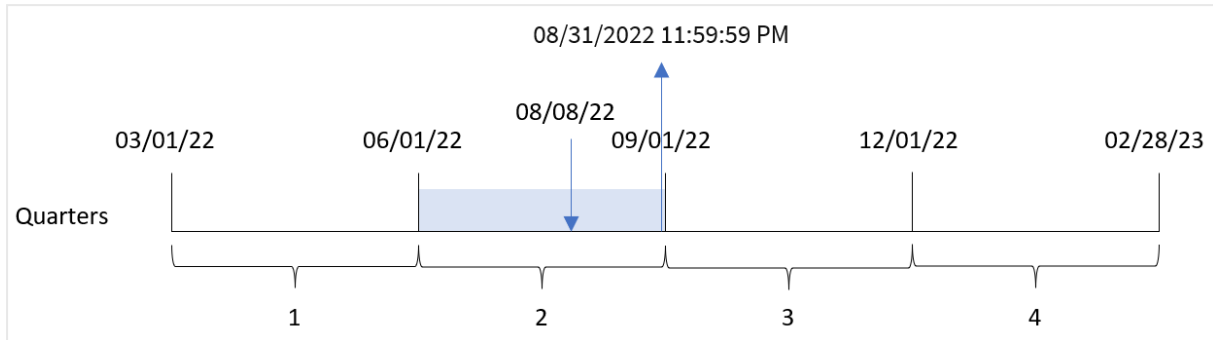
## Risultati

Tabella dei risultati

| id   | date       | end_of_quarter | end_of_quarter_timestamp |
|------|------------|----------------|--------------------------|
| 8188 | 1/7/2022   | 02/28/2022     | 2/28/2022 11:59:59 PM    |
| 8189 | 1/19/2022  | 02/28/2022     | 2/28/2022 11:59:59 PM    |
| 8190 | 2/5/2022   | 02/28/2022     | 2/28/2022 11:59:59 PM    |
| 8191 | 2/28/2022  | 02/28/2022     | 2/28/2022 11:59:59 PM    |
| 8192 | 3/16/2022  | 05/31/2022     | 5/31/2022 11:59:59 PM    |
| 8193 | 4/1/2022   | 05/31/2022     | 5/31/2022 11:59:59 PM    |
| 8194 | 5/7/2022   | 05/31/2022     | 5/31/2022 11:59:59 PM    |
| 8195 | 5/16/2022  | 05/31/2022     | 5/31/2022 11:59:59 PM    |
| 8196 | 6/15/2022  | 08/31/2022     | 8/31/2022 11:59:59 PM    |
| 8197 | 6/26/2022  | 08/31/2022     | 8/31/2022 11:59:59 PM    |
| 8198 | 7/9/2022   | 08/31/2022     | 8/31/2022 11:59:59 PM    |
| 8199 | 7/22/2022  | 08/31/2022     | 8/31/2022 11:59:59 PM    |
| 8200 | 7/23/2022  | 08/31/2022     | 8/31/2022 11:59:59 PM    |
| 8201 | 7/27/2022  | 08/31/2022     | 8/31/2022 11:59:59 PM    |
| 8202 | 8/2/2022   | 08/31/2022     | 8/31/2022 11:59:59 PM    |
| 8203 | 8/8/2022   | 08/31/2022     | 8/31/2022 11:59:59 PM    |
| 8204 | 8/19/2022  | 08/31/2022     | 8/31/2022 11:59:59 PM    |
| 8205 | 9/26/2022  | 11/30/2022     | 11/30/2022 11:59:59 PM   |
| 8206 | 10/14/2022 | 11/30/2022     | 11/30/2022 11:59:59 PM   |
| 8207 | 10/29/2022 | 11/30/2022     | 11/30/2022 11:59:59 PM   |

Poiché nella funzione `quarterend()` viene utilizzato l'argomento `first_month_of_year` di 3, l'inizio dell'anno si sposta dal 1° gennaio al 1° marzo.

Lo schema della funzione `quarterend()` con marzo impostato come primo mese dell'anno



La transazione 8203 è avvenuta l'8 agosto. Poiché l'inizio dell'anno è il 1° marzo, i trimestri dell'anno sono compresi tra marzo-maggio, giugno-agosto, settembre-novembre e dicembre-febbraio.

La funzione `quarterend()` identifica che la transazione ha avuto luogo nel trimestre compreso tra l'inizio di giugno e agosto e restituisce l'ultimo millisecondo di quel trimestre, ovvero il 31 agosto alle 23:59:59.

### Esempio 4 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, in questo esempio, il set di dati è invariato e viene caricato nell'applicazione. Il calcolo che restituisce un timestamp per la fine del trimestre in cui sono avvenute le transazioni viene creato come misura in un grafico nell'app.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
```

```
8189,1/19/2022,37.23
```

```
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
```

```
8192,3/16/2022,53.80
```

```
8193,4/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/16/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
```

```
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- date

Per calcolare la data di fine del trimestre in cui avviene una transazione, creare le seguenti misure:

- =quarterend(date)
- =timestamp(quarterend(date))

Tabella dei risultati

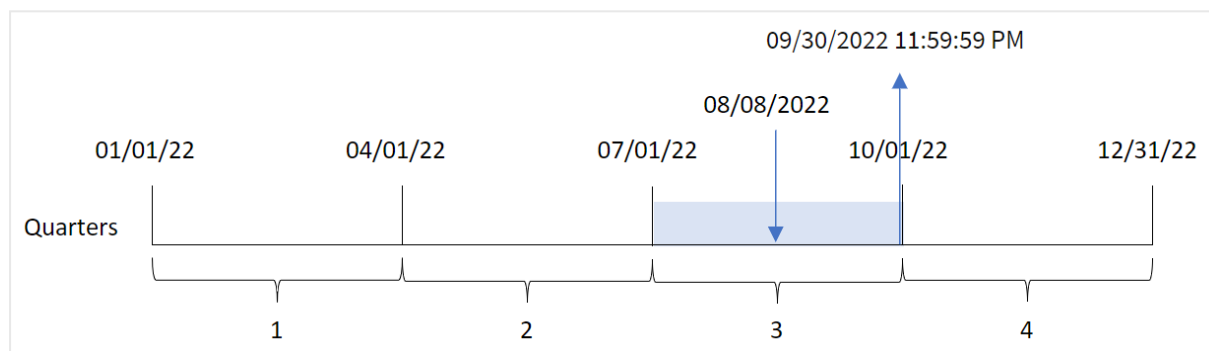
| id   | date      | =quarterend(date) | =timestamp(quarterend(date)) |
|------|-----------|-------------------|------------------------------|
| 8188 | 1/7/2022  | 03/31/2022        | 3/31/2022 11:59:59 PM        |
| 8189 | 1/19/2022 | 03/31/2022        | 3/31/2022 11:59:59 PM        |
| 8190 | 2/5/2022  | 03/31/2022        | 3/31/2022 11:59:59 PM        |
| 8191 | 2/28/2022 | 03/31/2022        | 3/31/2022 11:59:59 PM        |
| 8192 | 3/16/2022 | 03/31/2022        | 3/31/2022 11:59:59 PM        |
| 8193 | 4/1/2022  | 06/30/2022        | 6/30/2022 11:59:59 PM        |
| 8194 | 5/7/2022  | 06/30/2022        | 6/30/2022 11:59:59 PM        |
| 8195 | 5/16/2022 | 06/30/2022        | 6/30/2022 11:59:59 PM        |
| 8196 | 6/15/2022 | 06/30/2022        | 6/30/2022 11:59:59 PM        |
| 8197 | 6/26/2022 | 06/30/2022        | 6/30/2022 11:59:59 PM        |
| 8198 | 7/9/2022  | 09/30/2022        | 9/30/2022 11:59:59 PM        |
| 8199 | 7/22/2022 | 09/30/2022        | 9/30/2022 11:59:59 PM        |
| 8200 | 7/23/2022 | 09/30/2022        | 9/30/2022 11:59:59 PM        |
| 8201 | 7/27/2022 | 09/30/2022        | 9/30/2022 11:59:59 PM        |
| 8202 | 8/2/2022  | 09/30/2022        | 9/30/2022 11:59:59 PM        |
| 8203 | 8/8/2022  | 09/30/2022        | 9/30/2022 11:59:59 PM        |

| id   | date       | =quarterend(date) | =timestamp(quarterend(date)) |
|------|------------|-------------------|------------------------------|
| 8204 | 8/19/2022  | 09/30/2022        | 9/30/2022 11:59:59 PM        |
| 8205 | 9/26/2022  | 09/30/2022        | 9/30/2022 11:59:59 PM        |
| 8206 | 10/14/2022 | 12/31/2022        | 12/31/2022 11:59:59 PM       |
| 8207 | 10/29/2022 | 12/31/2022        | 12/31/2022 11:59:59 PM       |

Il campo 'end\_of\_quarter' viene creato nell'istruzione di caricamento precedente mediante l'uso della funzione `quarterend()` e trasferendo il campo data come argomento della funzione.

La funzione `quarterend()` inizialmente identifica in quale trimestre ricada il valore della data e quindi restituisce un timestamp per l'ultimo millisecondo di quel trimestre.

*Schema della funzione `quarterend()` con identificazione della fine del trimestre della transazione 8203*



La transazione 8203 è avvenuta l'8 agosto. La funzione `quarterend()` identifica che la transazione è avvenuta nel terzo trimestre e restituisce l'ultimo millisecondo di quel trimestre, ovvero il 30 settembre alle 23:59:59.

### Esempio 5 - Scenario

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati caricato in una tabella denominata 'Employee\_Expenses'. La tabella contiene i seguenti campi:
  - ID dipendenti
  - Nomi dipendenti
  - La media delle richieste di rimborso spese giornaliero di ciascun dipendente.

L'utente finale desidera un oggetto grafico che visualizzi, in base all'ID dipendente e al nome del dipendente, le richieste di rimborso spese stimate ancora da sostenere per il resto del trimestre. L'esercizio finanziario inizia a gennaio.

### Script di caricamento

```
Employee_Expenses:
Load
*
Inline
[
employee_id,employee_name,avg_daily_claim
182,Mark, $15
183,Deryck, $12.5
184,Dexter, $12.5
185,Sydney,$27
186,Agatha,$18
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- employee\_id
- employee\_name

Per calcolare gli interessi maturati, creare la seguente misura:

- $=(\text{quarterend}(\text{today}(1))-\text{today}(1))*\text{avg\_daily\_claim}$

Impostare la misura **Formattazione numero** su **Denaro**.

Tabella dei risultati

| employee_id | employee_name | $=(\text{quarterend}(\text{today}(1))-\text{today}(1))*\text{avg\_daily\_claim}$ |
|-------------|---------------|----------------------------------------------------------------------------------|
| 182         | Contrassegno  | \$480.00                                                                         |
| 183         | Deryck        | \$400.00                                                                         |
| 184         | Dexter        | \$400.00                                                                         |
| 185         | Sydney        | \$864.00                                                                         |
| 186         | Agatha        | \$576.00                                                                         |

La funzione `quarterend()` utilizza come unico argomento la data odierna e restituisce la data finale del mese corrente. Quindi, sottrae la data odierna dalla data di fine anno e l'espressione restituisce il numero di giorni rimanenti nel mese.

Questo valore viene quindi moltiplicato per la media delle richieste di rimborso spese giornaliere di ciascun dipendente per calcolare il valore stimato delle richieste che ogni dipendente dovrebbe presentare durante il periodo rimanente del trimestre.

## quartername

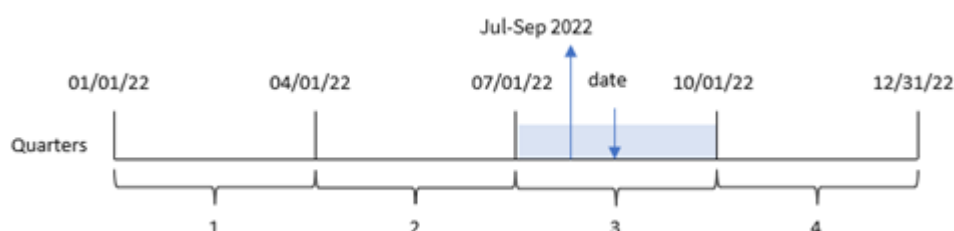
Questa funzione restituisce un valore di visualizzazione che mostra i mesi del trimestre (formattati in base alla variabile di script **MonthNames**) e l'anno con valore numerico sottostante corrispondente a un indicatore temporale recante il primo millisecondo del primo giorno del trimestre.

### Sintassi:

```
QuarterName(date[, period_no[, first_month_of_year]])
```

Tipo di dati restituiti: duale

Schema della funzione `quartername()`.



In altre parole, la funzione `quartername()` determina in quale trimestre ricadrà la data. Restituisce quindi un valore che mostra i mesi iniziali e finali di questo trimestre e dell'anno. Il valore numerico sottostante di questo risultato è il primo millisecondo del trimestre.

### Argomenti

| Argomento                  | Descrizione                                                                                                                                                                                                                             |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>date</b>                | La data o la data e ora da valutare.                                                                                                                                                                                                    |
| <b>period_no</b>           | <b>period_no</b> è un numero intero, in cui il valore 0 indica il trimestre che contiene <b>date</b> . I valori negativi di <b>period_no</b> indicano i trimestri precedenti, mentre i valori positivi indicano i trimestri successivi. |
| <b>first_month_of_year</b> | Se si intende utilizzare anni (fiscali) che non iniziano a gennaio, indicare un valore compreso tra 2 e 12 in <b>first_month_of_year</b> .                                                                                              |

## Casi di utilizzo

La funzione `quartername()` è utile quando si desidera confrontare le aggregazioni per trimestre. Ad esempio, può essere usata se si desidera visualizzare le vendite totali dei prodotti in base al trimestre.

Questa funzione può essere utilizzata nello script di caricamento per creare un campo in una tabella del Calendario principale. In alternativa, può essere utilizzata direttamente in un grafico come dimensione calcolata.

In questi esempi viene utilizzato il formato della data (MM/GG/AAAA). Il formato della data viene specificato nell'istruzione `SET DateFormat` nella parte superiore dello script di caricamento dei dati. Modificare il formato negli esempi in base alle proprie necessità.

Esempi di funzioni

| Esempio                                      | Risultato                 |
|----------------------------------------------|---------------------------|
| <code>quartername('10/29/2013')</code>       | Restituisce Oct-Dec 2013. |
| <code>quartername('10/29/2013', -1)</code>   | Restituisce Jul-Sep 2013. |
| <code>quartername('10/29/2013', 0, 3)</code> | Restituisce Sep-Nov 2013. |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - data senza alcun argomento aggiuntivo

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2022, caricato in una tabella denominata 'Transactions'.
- Il campo della data fornito nel formato della variabile di sistema `DateFormat` (MM/GG/AAAA).
- La creazione di un campo, `transaction_quarter`, che restituisce il trimestre in cui sono avvenute le transazioni.

Aggiungere qui altro testo, se necessario, con elenchi, ecc.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
SET MonthNames='Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec';
```

```
Transactions:
    Load
        *,
        quartername(date) as transaction_quarter
    ;
Load
*
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- transaction\_quarter

Tabella dei risultati

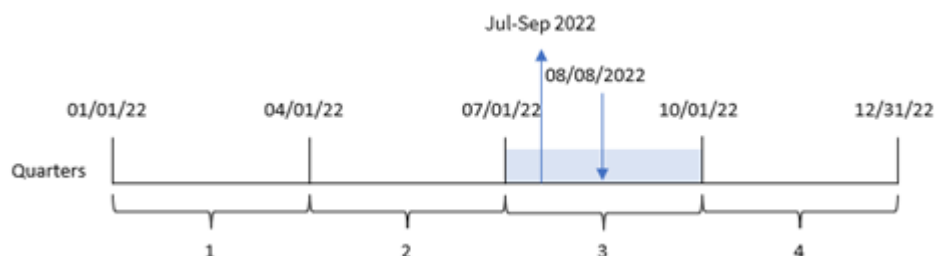
| date      | trimestre della transazione |
|-----------|-----------------------------|
| 1/7/2022  | gen-mar 2022                |
| 1/19/2022 | gen-mar 2022                |
| 2/5/2022  | gen-mar 2022                |

| date       | trimestre della transazione |
|------------|-----------------------------|
| 2/28/2022  | gen-mar 2022                |
| 3/16/2022  | gen-mar 2022                |
| 4/1/2022   | apr-giu 2022                |
| 5/7/2022   | apr-giu 2022                |
| 5/16/2022  | apr-giu 2022                |
| 6/15/2022  | apr-giu 2022                |
| 6/26/2022  | apr-giu 2022                |
| 7/9/2022   | lug-set 2022                |
| 7/22/2022  | lug-set 2022                |
| 7/23/2022  | lug-set 2022                |
| 7/27/2022  | lug-set 2022                |
| 8/2/2022   | lug-set 2022                |
| 8/8/2022   | lug-set 2022                |
| 8/19/2022  | lug-set 2022                |
| 9/26/2022  | lug-set 2022                |
| 10/14/2022 | ott-dic 2022                |
| 10/29/2022 | ott-dic 2022                |

Il campo 'transaction\_quarter' viene creato nell'istruzione di caricamento precedente mediante l'uso della funzione `quartername()` e trasferendo il campo data come argomento della funzione.

La funzione `quartername()` identifica inizialmente il trimestre in cui rientra il valore della data. Restituisce quindi un valore che mostra i mesi iniziali e finali di questo trimestre e dell'anno.

*Schema della funzione `quartername()`, esempio senza argomenti aggiuntivi*



La transazione 8203 è avvenuta l'8 agosto 2022. La funzione `quartername()` identifica che la transazione è avvenuta nel terzo trimestre e quindi restituisce luglio-settembre 2022. I mesi vengono visualizzati nello stesso formato della variabile di sistema `MonthNames`.

### Esempio 2 - data con argomento period\_no

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.
- La creazione di un campo, `previous_quarter`, che restituisce il trimestre precedente in cui sono state effettuate le transazioni.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';  
SET MonthNames='Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec';
```

Transactions:

```
    Load  
        *,  
        quartername(date,-1) as previous_quarter  
    ;  
Load  
*  
Inline  
[  
id,date,amount  
8188,1/7/2022,17.17  
8189,1/19/2022,37.23  
8190,2/28/2022,88.27  
8191,2/5/2022,57.42  
8192,3/16/2022,53.80  
8193,4/1/2022,82.06  
8194,5/7/2022,40.39  
8195,5/16/2022,87.21  
8196,6/15/2022,95.93  
8197,6/26/2022,45.89  
8198,7/9/2022,36.23  
8199,7/22/2022,25.66  
8200,7/23/2022,82.77  
8201,7/27/2022,69.98  
8202,8/2/2022,76.11  
8203,8/8/2022,25.12  
8204,8/19/2022,46.23  
8205,9/26/2022,84.21  
8206,10/14/2022,96.24  
8207,10/29/2022,67.67  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

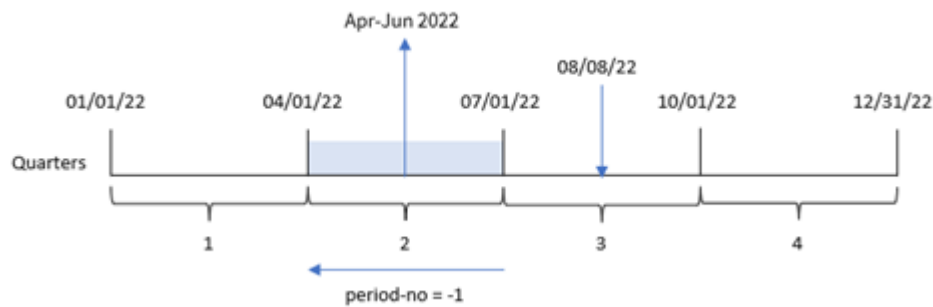
- date
- previous\_quarter

Tabella dei risultati

| date       | trimestre precedente |
|------------|----------------------|
| 1/7/2022   | ott-dic 2021         |
| 1/19/2022  | ott-dic 2021         |
| 2/5/2022   | ott-dic 2021         |
| 2/28/2022  | ott-dic 2021         |
| 3/16/2022  | ott-dic 2021         |
| 4/1/2022   | gen-mar 2022         |
| 5/7/2022   | gen-mar 2022         |
| 5/16/2022  | gen-mar 2022         |
| 6/15/2022  | gen-mar 2022         |
| 6/26/2022  | gen-mar 2022         |
| 7/9/2022   | apr-giu 2022         |
| 7/22/2022  | apr-giu 2022         |
| 7/23/2022  | apr-giu 2022         |
| 7/27/2022  | apr-giu 2022         |
| 8/2/2022   | apr-giu 2022         |
| 8/8/2022   | apr-giu 2022         |
| 8/19/2022  | apr-giu 2022         |
| 9/26/2022  | apr-giu 2022         |
| 10/14/2022 | lug-set 2022         |
| 10/29/2022 | lug-set 2022         |

In questo caso, poiché il valore `period_no` di -1 è stato utilizzato come argomento `offset` nella funzione `quartername()`, la funzione per prima cosa identifica le transazioni avvenute nel terzo trimestre. Passa quindi al trimestre precedente e restituisce un valore che mostra i mesi iniziali e finali di questo trimestre e dell'anno.

Schema della funzione `quartername()`, esempio di `period_no`



La transazione 8203 è avvenuta l'8 agosto. La funzione `quartername()` identifica che il trimestre precedente la transazione era compreso tra il 1° aprile e il 30 giugno. Pertanto, restituisce Apr-Giu 2022.

### Esempio 3 - data con argomento `first_week_day`

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor di caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento contiene lo stesso set di dati e lo stesso scenario del primo esempio. Tuttavia, in questo esempio, dobbiamo impostare il 1° marzo come inizio dell'anno finanziario.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';  
SET MonthNames='Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec';
```

Transactions:

```
Load  
    *,  
    quartername(date,0,3) as transaction_quarter  
;
```

Load

\*

Inline

[

id,date,amount

8188,1/7/2022,17.17

8189,1/19/2022,37.23

8190,2/28/2022,88.27

8191,2/5/2022,57.42

8192,3/16/2022,53.80

8193,4/1/2022,82.06

8194,5/7/2022,40.39

8195,5/16/2022,87.21

8196,6/15/2022,95.93

8197,6/26/2022,45.89

```
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- transaction\_quarter

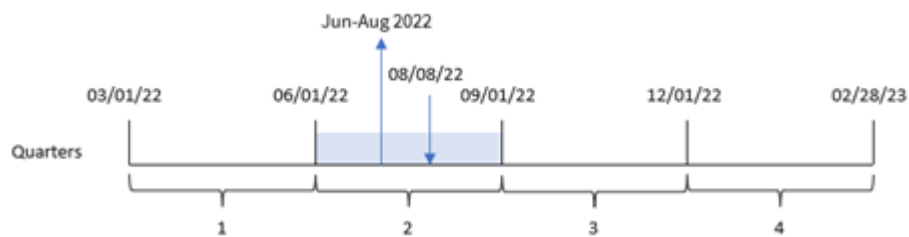
Tabella dei risultati

| date      | trimestre della transazione |
|-----------|-----------------------------|
| 1/7/2022  | Dic-Feb 2021                |
| 1/19/2022 | Dic-Feb 2021                |
| 2/5/2022  | Dic-Feb 2021                |
| 2/28/2022 | Dic-Feb 2021                |
| 3/16/2022 | Mar-Mag 2022                |
| 4/1/2022  | Mar-Mag 2022                |
| 5/7/2022  | Mar-Mag 2022                |
| 5/16/2022 | Mar-Mag 2022                |
| 6/15/2022 | Giu-Ago 2022                |
| 6/26/2022 | Giu-Ago 2022                |
| 7/9/2022  | Giu-Ago 2022                |
| 7/22/2022 | Giu-Ago 2022                |
| 7/23/2022 | Giu-Ago 2022                |
| 7/27/2022 | Giu-Ago 2022                |
| 8/2/2022  | Giu-Ago 2022                |
| 8/8/2022  | Giu-Ago 2022                |
| 8/19/2022 | Giu-Ago 2022                |
| 9/26/2022 | Set-Nov 2022                |

| date       | trimestre della transazione |
|------------|-----------------------------|
| 10/14/2022 | Set-Nov 2022                |
| 10/29/2022 | Set-Nov 2022                |

In questo caso, poiché nella funzione `quartername()` viene utilizzato l'argomento `first_month_of_year` di 3, l'inizio dell'anno si sposta dal 1° gennaio al 1° marzo. Pertanto, i trimestri dell'anno sono separati in marzo-maggio, giugno-agosto, settembre-novembre e dicembre-febbraio.

*Schema della funzione `quartername()`, esempio `first_week_day`*



La transazione 8203 è avvenuta l'8 agosto. La funzione `quartername()` identifica che la transazione è avvenuta nel secondo trimestre, tra l'inizio di giugno e la fine di agosto. Pertanto, restituisce `Giu-Ago 2022`.

### Esempio 4 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento contiene lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, in questo esempio, il set di dati invariato viene caricato nell'applicazione. Il calcolo che restituisce un timestamp per la fine del trimestre in cui sono avvenute le transazioni viene creato come misura in un oggetto grafico dell'applicazione.

#### Script di caricamento

```
Transactions:
Load
*
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
```

```
8194, 5/7/2022, 40.39
8195, 5/16/2022, 87.21
8196, 6/15/2022, 95.93
8197, 6/26/2022, 45.89
8198, 7/9/2022, 36.23
8199, 7/22/2022, 25.66
8200, 7/23/2022, 82.77
8201, 7/27/2022, 69.98
8202, 8/2/2022, 76.11
8203, 8/8/2022, 25.12
8204, 8/19/2022, 46.23
8205, 9/26/022, 84.21
8206, 10/14/2022, 96.24
8207, 10/29/2022, 67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: date.

Creare la seguente misura:

```
=quartername(date)
```

Tabella dei risultati

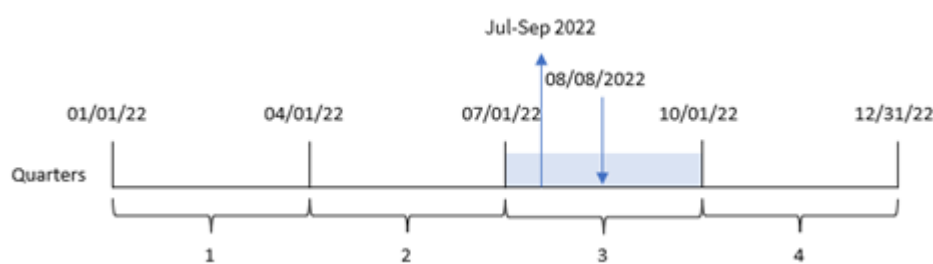
| date      | =nome_trimestre(data) |
|-----------|-----------------------|
| 1/7/2022  | gen-mar 2022          |
| 1/19/2022 | gen-mar 2022          |
| 2/5/2022  | gen-mar 2022          |
| 2/28/2022 | gen-mar 2022          |
| 3/16/2022 | gen-mar 2022          |
| 4/1/2022  | apr-giu 2022          |
| 5/7/2022  | apr-giu 2022          |
| 5/16/2022 | apr-giu 2022          |
| 6/15/2022 | apr-giu 2022          |
| 6/26/2022 | apr-giu 2022          |
| 7/9/2022  | lug-set 2022          |
| 7/22/2022 | lug-set 2022          |
| 7/23/2022 | lug-set 2022          |
| 7/27/2022 | lug-set 2022          |
| 8/2/2022  | lug-set 2022          |

| date       | =nome_trimestre(data) |
|------------|-----------------------|
| 8/8/2022   | lug-set 2022          |
| 8/19/2022  | lug-set 2022          |
| 9/26/2022  | lug-set 2022          |
| 10/14/2022 | ott-dic 2022          |
| 10/29/2022 | ott-dic 2022          |

La misura `transaction_quarter` viene creata nell'oggetto grafico mediante l'utilizzo della funzione `quartername()` e trasferendo il campo `date` come argomento della funzione.

La funzione `quartername()` identifica inizialmente il trimestre in cui rientra il valore della data. Restituisce quindi un valore che mostra i mesi iniziali e finali di questo trimestre e dell'anno.

*Schema della funzione `quartername()`, esempio di oggetto grafico*



La transazione 8203 è avvenuta l'8 agosto 2022. La funzione `quartername()` identifica che la transazione è avvenuta nel terzo trimestre e quindi restituisce luglio-settembre 2022. I mesi vengono visualizzati nello stesso formato della variabile di sistema `MonthNames`.

## Esempio 5 - Scenario

Script di caricamento ed espressione del grafico

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2022, caricato in una tabella denominata `Transactions`.
- Il campo della data fornito nel formato della variabile di sistema `DateFormat` (MM/GG/AAAA).

L'utente finale desidera un oggetto grafico che presenti le vendite totali per trimestre per le transazioni. Questo può essere ottenuto anche quando la dimensione non è disponibile nel modello dati, utilizzando la funzione `quartername()` come dimensione calcolata nel grafico.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'1/7/2022',17.17
```

```
8189,'1/19/2022',37.23
```

```
8190,'2/28/2022',88.27
```

```
8191,'2/5/2022',57.42
```

```
8192,'3/16/2022',53.80
```

```
8193,'4/1/2022',82.06
```

```
8194,'5/7/2022',40.39
```

```
8195,'5/16/2022',87.21
```

```
8196,'6/15/2022',95.93
```

```
8197,'6/26/2022',45.89
```

```
8198,'7/9/2022',36.23
```

```
8199,'7/22/2022',25.66
```

```
8200,'7/23/2022',82.77
```

```
8201,'7/27/2022',69.98
```

```
8202,'8/2/2022',76.11
```

```
8203,'8/8/2022',25.12
```

```
8204,'8/19/2022',46.23
```

```
8205,'9/26/2022',84.21
```

```
8206,'10/14/2022',96.24
```

```
8207,'10/29/2022',67.67
```

```
];
```

### Risultati

Procedere come indicato di seguito:

1. Caricare i dati e aprire un foglio. Creare una nuova tabella.
2. Creare una dimensione calcolata utilizzando la seguente espressione:  
=quartername(date)
3. Quindi, calcolare le vendite totali utilizzando la seguente misura di aggregazione:  
=sum(amount)
4. Impostare la **Formattazione numero** della misura su **Denaro**.

Tabella dei risultati

| =nome_trimestre(data) | sum(amount) |
|-----------------------|-------------|
| lug-set 2022          | USD 446.31  |
| apr-giu 2022          | 351.48 \$   |
| gen-mar 2022          | 253.89 \$   |
| ott-dic 2022          | 163.91 \$   |

## quarterstart

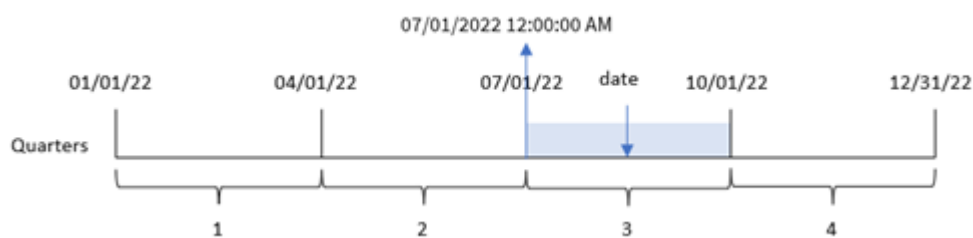
Questa funzione restituisce un valore corrispondente a un indicatore temporale recante il primo millisecondo del trimestre contenente **date**. Il formato di output predefinito sarà il formato **DateFormat** impostato nello script.

### Sintassi:

```
QuarterStart(date[, period_no[, first_month_of_year]])
```

Tipo di dati restituiti: duale

Schema della funzione `quarterstart()`.



In altre parole, la funzione `quarterstart()` determina in quale trimestre ricadrà `date`. Quindi restituisce un timestamp, in formato data, per l'ultimo millisecondo del primo mese di quel trimestre.

### Argomenti

| Argomento                  | Descrizione                                                                                                                                                                                                                             |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>date</b>                | La data o la data e ora da valutare.                                                                                                                                                                                                    |
| <b>period_no</b>           | <b>period_no</b> è un numero intero, in cui il valore 0 indica il trimestre che contiene <b>date</b> . I valori negativi di <b>period_no</b> indicano i trimestri precedenti, mentre i valori positivi indicano i trimestri successivi. |
| <b>first_month_of_year</b> | Se si intende utilizzare anni (fiscali) che non iniziano a gennaio, indicare un valore compreso tra 2 e 12 in <b>first_month_of_year</b> .                                                                                              |

## Casi di utilizzo

La funzione `quarterstart()` viene comunemente utilizzata come parte di un'espressione quando l'utente desidera che il calcolo utilizzi la frazione del trimestre trascorsa finora. Ad esempio, potrebbe essere utilizzata se un utente desidera calcolare gli interessi accumulati in un trimestre fino a oggi.

### Esempi di funzioni

| Esempio                                       | Risultato               |
|-----------------------------------------------|-------------------------|
| <code>quarterstart('10/29/2005')</code>       | Restituisce 10/01/2005. |
| <code>quarterstart('10/29/2005', -1 )</code>  | Restituisce 07/01/2005. |
| <code>quarterstart('10/29/2005', 0, 3)</code> | Restituisce 09/01/2005. |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - Nessun argomento aggiuntivo

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2022, caricato in una tabella denominata 'Transactions'.
- Il campo della data fornito nel formato della variabile di sistema `DateFormat` (MM/GG/AAAA).
- La creazione di un campo, `start_of_quarter`, che restituisce un timestamp per l'inizio del trimestre in cui sono avvenute le transazioni.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        quarterstart(date) as start_of_quarter,
        timestamp(quarterstart(date)) as start_of_quarter_timestamp
    ;
Load
*
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- start\_of\_quarter
- start\_of\_quarter\_timestamp

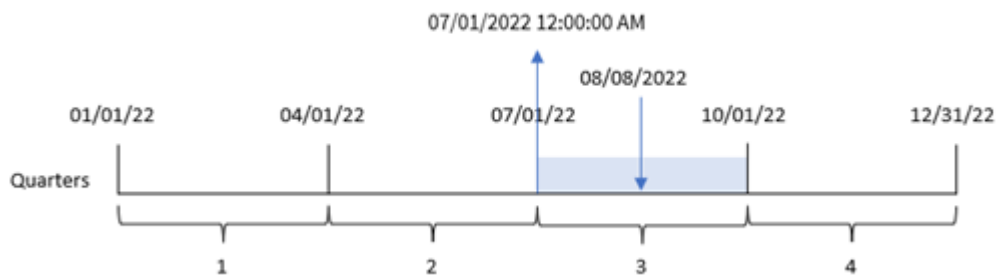
Tabella dei risultati

| date      | inizio_del_trimestre | inizio_del_trimestre_timestamp |
|-----------|----------------------|--------------------------------|
| 1/7/2022  | 01/01/2022           | 1/1/2022 00:00:00              |
| 1/19/2022 | 01/01/2022           | 01/01/2022 12:00:00 AM         |
| 2/5/2022  | 01/01/2022           | 01/01/2022 00:00:00            |
| 2/28/2022 | 01/01/2022           | 01/01/2022 00:00:00            |
| 3/16/2022 | 01/01/2022           | 1/1/2022 00:00:00              |
| 4/1/2022  | 04/01/2022           | 01/04/2021 00:00:00            |
| 5/7/2022  | 04/01/2022           | 01/04/2021 00:00:00            |
| 5/16/2022 | 04/01/2022           | 01/04/2021 00:00:00            |
| 6/15/2022 | 04/01/2022           | 1/4/2021 00:00:00              |
| 6/26/2022 | 04/01/2022           | 1/4/2021 00:00:00              |
| 7/9/2022  | 07/01/2022           | 1/7/2021 00:00:00              |
| 7/22/2022 | 07/01/2022           | 1/7/2021 00:00:00              |
| 7/23/2022 | 07/01/2022           | 1/7/2021 00:00:00              |

| date       | inizio_del_trimestre | inizio_del_trimestre_timestamp |
|------------|----------------------|--------------------------------|
| 7/27/2022  | 07/01/2022           | 1/7/2021 00:00:00              |
| 8/2/2022   | 07/01/2022           | 1/7/2021 00:00:00              |
| 8/8/2022   | 07/01/2022           | 1/7/2021 00:00:00              |
| 8/19/2022  | 07/01/2022           | 1/7/2021 00:00:00              |
| 9/26/2022  | 07/01/2022           | 1/7/2021 00:00:00              |
| 10/14/2022 | 01/10/2022           | 10/1/2022 12:00:00 AM          |
| 10/29/2022 | 10/01/2022           | 10/1/2022 12:00:00 AM          |

Il campo 'start\_of\_quarter' viene creato nell'istruzione di caricamento precedente mediante l'uso della funzione `quarterstart()` e trasferendo il campo data come argomento della funzione. La funzione `quarterstart()` identifica inizialmente il trimestre in cui rientra il valore della data. Quindi restituisce un timestamp per il primo millisecondo di quel trimestre.

*Schema della funzione `quarterstart()`, esempio senza argomenti aggiuntivi*



La transazione 8203 è avvenuta l'8 agosto. La funzione `quarterstart()` identifica che la transazione è avvenuta nel terzo trimestre e restituisce il primo millisecondo del trimestre, ovvero il 1° luglio alle 12:00:00.

## Esempio 2 - period\_no

Script di caricamento e risultati

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.
- La creazione di un campo, `previous_quarter_start`, che restituisce il timestamp per l'inizio del trimestre prima che fosse effettuata la transazione.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        quarterstart(date,-1) as previous_quarter_start,
        timestamp(quarterstart(date,-1)) as previous_quarter_start_timestamp
    ;

Load
*
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- previous\_quarter\_start
- previous\_quarter\_start\_timestamp

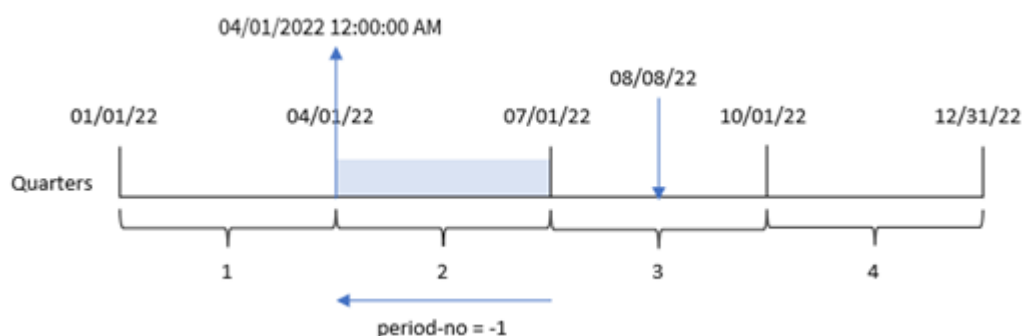
Tabella dei risultati

| date      | inizio_trimestre_precedente | timestamp_inizio_trimestre_precedente |
|-----------|-----------------------------|---------------------------------------|
| 1/7/2022  | 01/10/2021                  | 01/10/2021 00:00:00                   |
| 1/19/2022 | 01/10/2021                  | 01/10/2021 00:00:00                   |

| date       | inizio_trimestre_precedente | timestamp_inizio_trimestre_precedente |
|------------|-----------------------------|---------------------------------------|
| 2/5/2022   | 01/10/2021                  | 01/10/2021 00:00:00                   |
| 2/28/2022  | 01/10/2021                  | 01/10/2021 00:00:00                   |
| 3/16/2022  | 01/10/2021                  | 01/10/2021 00:00:00                   |
| 4/1/2022   | 01/01/2022                  | 01/01/2022 00:00:00                   |
| 5/7/2022   | 01/01/2022                  | 01/01/2022 00:00:00                   |
| 5/16/2022  | 01/01/2022                  | 01/01/2022 00:00:00                   |
| 6/15/2022  | 01/01/2022                  | 1/1/2022 00:00:00                     |
| 6/26/2022  | 01/01/2022                  | 1/1/2022 00:00:00                     |
| 7/9/2022   | 04/01/2022                  | 1/4/2021 00:00:00                     |
| 7/22/2022  | 04/01/2022                  | 01/04/2021 00:00:00                   |
| 7/23/2022  | 04/01/2022                  | 01/04/2021 00:00:00                   |
| 7/27/2022  | 04/01/2022                  | 01/04/2021 00:00:00                   |
| 8/2/2022   | 04/01/2022                  | 01/04/2021 00:00:00                   |
| 8/8/2022   | 04/01/2022                  | 01/04/2021 00:00:00                   |
| 8/19/2022  | 04/01/2022                  | 01/04/2021 00:00:00                   |
| 9/26/2022  | 04/01/2022                  | 1/4/2021 00:00:00                     |
| 10/14/2022 | 07/01/2022                  | 1/7/2022 00:00:00                     |
| 10/29/2022 | 07/01/2022                  | 1/7/2022 00:00:00                     |

In questo caso, poiché il valore `period_no` di -1 è stato utilizzato come argomento `offset` nella funzione `quarterstart()`, la funzione per prima cosa identifica il trimestre in cui avvengono le transazioni. Si sposta poi un trimestre prima e identifica il primo millisecondo di tale trimestre.

Schema della funzione `quarterstart()`, esempio di `period_no`



La transazione 8203 è avvenuta l'8 agosto. La funzione `quarterstart()` identifica che il trimestre precedente la transazione era compreso tra il 1° aprile e il 30 giugno. Quindi restituisce il primo millisecondo di quel trimestre, il 1° aprile alle 12:00:00.

### Esempio 3 - first\_month\_of\_year

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor di caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento contiene lo stesso set di dati e lo stesso scenario del primo esempio. Tuttavia, in questo esempio, dobbiamo impostare il 1° marzo come inizio dell'anno finanziario.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*,
```

```
quarterstart(date,0,3) as start_of_quarter,
```

```
timestamp(quarterstart(date,0,3)) as start_of_quarter_timestamp
```

```
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
```

```
8189,1/19/2022,37.23
```

```
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
```

```
8192,3/16/2022,53.80
```

```
8193,4/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/16/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
```

```
8200,7/23/2022,82.77
```

```
8201,7/27/2022,69.98
```

```
8202,8/2/2022,76.11
```

```
8203,8/8/2022,25.12
```

```
8204,8/19/2022,46.23
```

```
8205,9/26/2022,84.21
```

```
8206,10/14/2022,96.24
```

```
8207,10/29/2022,67.67
```

```
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

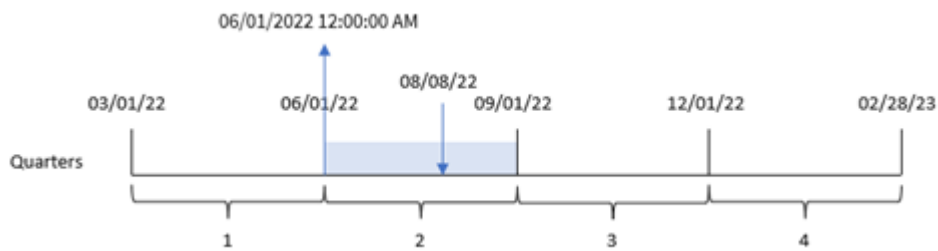
- date
- start\_of\_quarter
- start\_of\_quarter\_timestamp

Tabella dei risultati

| date       | inizio_del_trimestre | inizio_del_trimestre_timestamp |
|------------|----------------------|--------------------------------|
| 1/7/2022   | 01/12/2021           | 01/12/2021 00:00:00            |
| 1/19/2022  | 01/12/2021           | 01/12/2021 00:00:00            |
| 2/5/2022   | 01/12/2021           | 01/12/2021 00:00:00            |
| 2/28/2022  | 01/12/2021           | 01/12/2021 00:00:00            |
| 3/16/2022  | 03/01/2022           | 01/03/2022 00:00:00            |
| 4/1/2022   | 03/01/2022           | 1/3/2022 00:00:00              |
| 5/7/2022   | 03/01/2022           | 01/03/2022 00:00:00            |
| 5/16/2022  | 03/01/2022           | 1/3/2022 12:00:00 AM           |
| 6/15/2022  | 01/06/2022           | 6/1/2022 12:00:00 AM           |
| 6/26/2022  | 06/01/2022           | 01/06/2022 00:00:00            |
| 7/9/2022   | 01/06/2022           | 01/06/2022 00:00:00            |
| 7/22/2022  | 01/06/2022           | 01/06/2022 00:00:00            |
| 7/23/2022  | 01/06/2022           | 01/06/2022 00:00:00            |
| 7/27/2022  | 01/06/2022           | 01/06/2022 00:00:00            |
| 8/2/2022   | 01/06/2022           | 01/06/2022 00:00:00            |
| 8/8/2022   | 01/06/2022           | 01/06/2022 00:00:00            |
| 8/19/2022  | 01/06/2022           | 01/06/2022 00:00:00            |
| 9/26/2022  | 09/01/2022           | 1/9/2022 00:00:00              |
| 10/14/2022 | 01/09/2022           | 9/1/2022 12:00:00 AM           |
| 10/29/2022 | 01/09/2022           | 9/1/2022 12:00:00 AM           |

In questa istanza, poiché l'argomento `first_month_of_year` di 3 viene utilizzato nella funzione `quarterstart()`, l'inizio dell'anno si sposta dal 1° gennaio al 1° marzo.

Schema della funzione `quarterstart()`, esempio di `first_month_of_year`



La transazione 8203 è avvenuta l'8 agosto. Poiché l'inizio dell'anno è il 1° marzo, i trimestri dell'anno sono compresi tra marzo-maggio, giugno-agosto, settembre-novembre e dicembre-febbraio. La funzione `quarterstart()` identifica che la transazione ha avuto luogo nel trimestre compreso tra l'inizio di giugno e agosto e restituisce il primo millisecondo di tale trimestre, ovvero il 1° giugno alle 12:00:00.

### Esempio 4 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento contiene lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, in questo esempio, il set di dati invariato viene caricato nell'applicazione. Il calcolo che restituisce un timestamp per la fine del trimestre in cui sono avvenute le transazioni viene creato come misura in un oggetto grafico dell'applicazione.

#### Script di caricamento

```
Transactions:
Load
*
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
```

```
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: date.

Aggiungere le seguenti misure:

- =quarterstart(date)
- =timestamp(quarterstart(date))

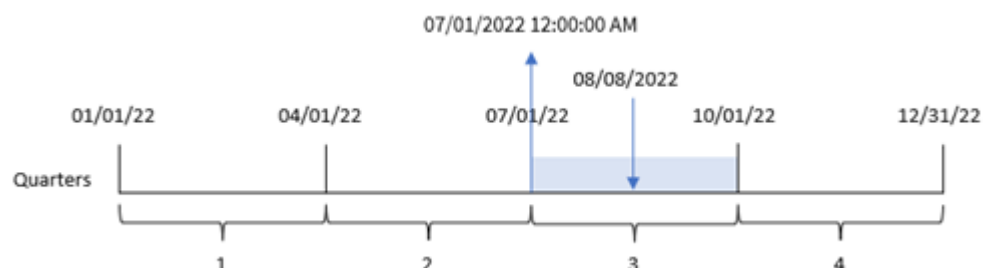
Tabella dei risultati

| date       | =quarterstart(date) | =timestamp(quarterstart(date)) |
|------------|---------------------|--------------------------------|
| 10/14/2022 | 01/10/2022          | 10/1/2022 12:00:00 AM          |
| 10/29/2022 | 10/01/2022          | 01/10/2022 12:00:00 AM         |
| 7/9/2022   | 07/01/2022          | 1/7/2022 00:00:00              |
| 7/22/2022  | 07/01/2022          | 7/1/2022 12:00:00 AM           |
| 7/23/2022  | 07/01/2022          | 01/07/2022 00:00:00            |
| 7/27/2022  | 07/01/2022          | 1/7/2022 00:00:00              |
| 8/2/2022   | 07/01/2022          | 01/07/2022 00:00:00            |
| 8/8/2022   | 07/01/2022          | 01/07/2022 00:00:00            |
| 8/19/2022  | 07/01/2022          | 01/07/2022 00:00:00            |
| 9/26/2022  | 07/01/2022          | 1/7/2022 00:00:00              |
| 4/1/2022   | 04/01/2022          | 01/04/2022 00:00:00            |
| 5/7/2022   | 04/01/2022          | 01/04/2022 00:00:00            |
| 5/16/2022  | 04/01/2022          | 01/04/2022 00:00:00            |
| 6/15/2022  | 04/01/2022          | 01/04/2022 00:00:00            |
| 6/26/2022  | 04/01/2022          | 01/04/2022 00:00:00            |
| 1/7/2022   | 01/01/2022          | 1/1/2022 00:00:00              |
| 1/19/2022  | 01/01/2022          | 01/01/2022 12:00:00 AM         |
| 2/5/2022   | 01/01/2022          | 01/01/2022 00:00:00            |
| 2/28/2022  | 01/01/2022          | 01/01/2022 00:00:00            |
| 3/16/2022  | 01/01/2022          | 1/1/2022 12:00:00 AM           |

La misura `start_of_quarter` viene creata nell'oggetto grafico mediante l'utilizzo della funzione `quarterstart()` e trasferendo il campo `date` come argomento della funzione.

La funzione `quarterstart()` identifica il trimestre in cui cade il valore della data, restituendo un timestamp per il primo millisecondo di quel trimestre.

*Schema della funzione `quarterstart()`, esempio di oggetto grafico*



La transazione 8203 è avvenuta l'8 agosto. La funzione `quarterstart()` identifica che la transazione è avvenuta nel terzo trimestre e restituisce il primo millisecondo del trimestre. Il valore restituito è il 1° luglio alle 12:00:00.

### Esempio 5 - Scenario

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di saldi di prestiti, che viene caricato in una tabella chiamata `Loans`.
- Dati costituiti dagli ID dei prestiti, dal saldo all'inizio del trimestre e dal tasso di interesse semplice annuo applicato a ciascun prestito.

L'utente finale desidera un oggetto grafico che mostri, in base all'ID del prestito, gli interessi correnti che sono stati maturati per ciascun prestito nel trimestre in corso.

#### Script di caricamento

```
Loans:
Load
*
Inline
[
loan_id,start_balance,rate
8188,$10000.00,0.024
8189,$15000.00,0.057
8190,$17500.00,0.024
```

```
8191,$21000.00,0.034  
8192,$90000.00,0.084  
];
```

### Risultati

Procedere come indicato di seguito:

1. Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:
  - loan\_id
  - start\_balance
2. Quindi, creare questa misura per calcolare l'interesse accumulato:  
`=start_balance*(rate*(today(1)-quarterstart(today(1)))/365)`
3. Impostare la **Formattazione numero** della misura su **Denaro**.

Tabella dei risultati

| loan_id | start_balance | =start_balance*(rate*(today(1)-quarterstart(today(1)))/365) |
|---------|---------------|-------------------------------------------------------------|
| 8188    | \$10000.00    | \$15.07                                                     |
| 8189    | \$15000.00    | \$128.84                                                    |
| 8190    | \$17500.00    | \$63.29                                                     |
| 8191    | \$21000.00    | \$107.59                                                    |
| 8192    | \$90000.00    | \$1139.18                                                   |

Utilizzando la data odierna come unico argomento, la funzione `quarterstart()` restituisce la data di inizio dell'anno corrente. Sottraendo tale risultato dalla data corrente, l'espressione restituisce il numero di giorni trascorsi fino ad ora questo trimestre.

Questo valore viene quindi moltiplicato per il tasso di interesse e diviso per 365, per restituire il tasso di interesse effettivo incorso per questo periodo. Il risultato viene poi moltiplicato per il saldo iniziale del prestito per ottenere gli interessi maturati fino a questo momento del trimestre.

### second

Questa funzione restituisce un numero intero che rappresenta il secondo in cui la frazione di **expression** viene interpretata come ora in base all'interpretazione numerica standard.

Sintassi:

```
second (expression)
```

**Tipo di dati restituiti:** numero intero

### Casi di utilizzo

La funzione `second()` è utile quando si desidera confrontare le aggregazioni per secondo. Ad esempio, la funzione può essere utilizzata se si desidera visualizzare la distribuzione del conteggio delle attività per secondo.

Queste dimensioni possono essere sia create nello script di caricamento, utilizzando la funzione per creare un campo in una tabella del Calendario principale, sia utilizzate direttamente in un grafico come dimensione calcolata.

Esempi di funzioni

| Esempio                           | Risultato                                 |
|-----------------------------------|-------------------------------------------|
| <code>second( '09:14:36' )</code> | restituisce 36                            |
| <code>second( '0.5555' )</code>   | restituisce 55 (poiché 0.5555 = 13:19:55) |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - Variabile

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente transazioni per timestamp, caricato in una tabella chiamata `Transactions`.
- Viene utilizzata la variabile di sistema `Timestamp` predefinita (`M/D/YYYY h:mm:ss[.fff] TT`).
- La creazione di un campo, `second`, per calcolare quando sono avvenuti gli acquisti.

### Script di caricamento

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

```
Transactions:
```

```
    Load
        *,
        second(date) as second
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
id,date,amount
9497,'01/05/2022 7:04:57 PM',47.25
9498,'01/03/2022 2:21:53 PM',51.75
9499,'01/03/2022 5:40:49 AM',73.53
9500,'01/04/2022 6:49:38 PM',15.35
9501,'01/01/2022 10:10:22 PM',31.43
9502,'01/05/2022 7:34:46 PM',13.24
9503,'01/06/2022 10:58:34 PM',74.34
9504,'01/06/2022 11:29:38 AM',50.00
9505,'01/02/2022 8:35:54 AM',36.34
9506,'01/06/2022 8:49:09 AM',74.23
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- second

Tabella dei risultati

| date                   | secondo |
|------------------------|---------|
| 01/01/2022 10:10:22 PM | 22      |
| 01/02/2022 8:35:54 AM  | 54      |
| 01/03/2022 5:40:49 AM  | 49      |
| 01/03/2022 2:21:53 PM  | 53      |
| 01/04/2022 6:49:38 PM  | 38      |
| 01/05/2022 7:04:57 PM  | 57      |
| 01/05/2022 7:34:46 PM  | 46      |
| 01/06/2022 8:49:09 AM  | 9       |
| 01/06/2022 11:29:38 AM | 38      |
| 01/06/2022 10:58:34 PM | 34      |

I valori nel campo `second ora` sono creati usando la funzione `second()` e trasferendo la data come espressione nell'istruzione di caricamento precedente.

### Esempio 2 - Oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor di caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento contiene lo stesso set di dati e lo stesso scenario del primo esempio. Tuttavia, in questo esempio, il set di dati invariato viene caricato nell'applicazione. I valori `second` sono calcolati mediante una misura in un oggetto grafico.

#### Script di caricamento

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
9497,'01/05/2022 7:04:57 PM',47.25
```

```
9498,'01/03/2022 2:21:53 PM',51.75
```

```
9499,'01/03/2022 5:40:49 AM',73.53
```

```
9500,'01/04/2022 6:49:38 PM',15.35
```

```
9501,'01/01/2022 10:10:22 PM',31.43
```

```
9502,'01/05/2022 7:34:46 PM',13.24
```

```
9503,'01/06/2022 10:58:34 PM',74.34
```

```
9504,'01/06/2022 11:29:38 AM',50.00
```

```
9505,'01/02/2022 8:35:54 AM',36.34
```

```
9506,'01/06/2022 8:49:09 AM',74.23
```

```
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: `date`.

Creare la seguente misura:

```
=second(date)
```

Tabella dei risultati

| date                   | =second(date) |
|------------------------|---------------|
| 01/01/2022 10:10:22 PM | 22            |

| date                   | =second(date) |
|------------------------|---------------|
| 01/02/2022 8:35:54 AM  | 54            |
| 01/03/2022 5:40:49 AM  | 49            |
| 01/03/2022 2:21:53 PM  | 53            |
| 01/04/2022 6:49:38 PM  | 38            |
| 01/05/2022 7:04:57 PM  | 57            |
| 01/05/2022 7:34:46 PM  | 46            |
| 01/06/2022 8:49:09 AM  | 9             |
| 01/06/2022 11:29:38 AM | 38            |
| 01/06/2022 10:58:34 PM | 34            |

I valori per 'second' sono creati usando la funzione `second()` e trasferendo la data come espressione in una misura per l'oggetto grafico.

### Esempio 3 - Scenario

Script di caricamento ed espressioni del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un insieme di timestamp, generato per rappresentare il traffico verso il sito web di vendita dei biglietti di un particolare festival. Questi timestamp e un id corrispondente vengono caricati in una tabella chiamata `web_Traffic`.
- Viene utilizzata la variabile di sistema `Timestamp M/D/YYYY h:mm:ss[.fff] TT`.

In questo scenario, erano presenti 10.000 biglietti, messi in vendita alle 9:00 del 20 maggio 2021. Un minuto dopo, i biglietti erano esauriti.

L'utente desidera un oggetto grafico che mostri, al secondo, il numero di visite al sito web.

#### Script di caricamento

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

```
tmpTimeStampCreator:
```

```
load
```

```
    makedate(2022,05,20) as date
```

```
AutoGenerate 1;
```

```
join load
```

```
    maketime(9+floor(rand()*2),0,floor(rand()*59)) as time
```

```
autogenerate 10000;
```

```
Web_Traffic:
```

```
load
```

```
    recno() as id,
```

```
    timestamp(date + time) as timestamp
```

```
resident tmpTimeStampCreator;
```

```
drop table tmpTimeStampCreator;
```

### Risultati

**Procedere come indicato di seguito:**

1. Caricare i dati e aprire un foglio. Creare una nuova tabella.
2. Quindi, creare una dimensione calcolata utilizzando la seguente espressione:  
=second(timestamp)
3. Creare una misura di aggregazione per calcolare il conteggio totale delle voci:  
=count(id)

La tabella dei risultati sarà simile alla tabella seguente, ma con valori diversi per la misura di aggregazione:

Tabella dei risultati

| second(timestamp) | =count(id) |
|-------------------|------------|
| 0                 | 150        |
| 1                 | 184        |
| 2                 | 163        |
| 3                 | 178        |
| 4                 | 179        |
| 5                 | 158        |
| 6                 | 177        |
| 7                 | 169        |
| 8                 | 149        |
| 9                 | 186        |
| 10                | 169        |
| 11                | 179        |
| 12                | 186        |
| 13                | 182        |
| 14                | 180        |
| 15                | 153        |

|                          |                   |
|--------------------------|-------------------|
| <b>second(timestamp)</b> | <b>=count(id)</b> |
| 16                       | 191               |
| 17                       | 203               |
| 18                       | 158               |
| 19                       | 159               |
| 20                       | 163               |
| +39 ulteriori righe      |                   |

## setdateyear

Questa funzione utilizza come input un **timestamp** e un **year** e aggiorna il **timestamp** con l'**year** specificato nell'input.

Sintassi:

```
setdateyear (timestamp, year)
```

Tipo di dati restituiti: duale

Argomenti:

Argomenti

| Argomento        | Descrizione                                                            |
|------------------|------------------------------------------------------------------------|
| <b>timestamp</b> | Un indicatore temporale standard di Qlik Sense (spesso solo una data). |
| <b>year</b>      | Un anno a quattro cifre.                                               |

Esempi e risultati:

In questi esempi viene utilizzato il formato della data **DD/MM/YYYY**. Il formato della data viene specificato nell'istruzione **SET DateFormat** nella parte superiore dello script di caricamento dei dati. Modificare il formato negli esempi in base alle proprie necessità.

Esempi di script

| Esempio                                           | Risultato                                                                                                                                                                                                                                                                                                   |
|---------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| setdateyear<br>( '29/10/2005' ,<br>2013)          | Restituisce '29/10/2013'                                                                                                                                                                                                                                                                                    |
| setdateyear<br>( '29/10/2005<br>04:26:14' , 2013) | Restituisce '29/10/2013 04:26:14'<br>Per visualizzare la sezione relativa alla data e all'ora dell'indicatore temporale in una visualizzazione, è necessario impostare la formattazione dei numeri su Data e scegliere un valore per Formattazione che consenta di mostrare i valori della data e dell'ora. |

### Esempio:

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

SetYear:

Load \*,

SetDateYear(testdates, 2013) as NewYear

Inline [

testdates

1/11/2012

10/12/2012

1/5/2013

2/1/2013

19/5/2013

15/9/2013

11/12/2013

2/3/2014

14/5/2014

13/6/2014

7/7/2014

4/8/2014

];

La tabella risultante contiene le date originali e una colonna in cui l'anno è stato impostato su 2013.

Tabella dei risultati

| testdates  | NewYear    |
|------------|------------|
| 1/11/2012  | 1/11/2013  |
| 10/12/2012 | 10/12/2013 |
| 2/1/2012   | 2/1/2013   |

| testdates  | NewYear    |
|------------|------------|
| 1/5/2013   | 1/5/2013   |
| 19/5/2013  | 19/5/2013  |
| 15/9/2013  | 15/9/2013  |
| 11/12/2013 | 11/12/2013 |
| 2/3/2014   | 2/3/2013   |
| 14/5/2014  | 14/5/2013  |
| 13/6/2014  | 13/6/2013  |
| 7/7/2014   | 7/7/2013   |
| 4/8/2014   | 4/8/2013   |

## setdateyearmonth

Questa funzione utilizza come input un **timestamp**, un **month** e un **year** e aggiorna il **timestamp** con l'**year** e il **month** specificati nell'input. .

Sintassi:

```
SetDateYearMonth (timestamp, year, month)
```

Tipo di dati restituiti: duale

Argomenti:

Argomenti

| Argomento        | Descrizione                                                            |
|------------------|------------------------------------------------------------------------|
| <b>timestamp</b> | Un indicatore temporale standard di Qlik Sense (spesso solo una data). |
| <b>year</b>      | Un anno a quattro cifre.                                               |
| <b>month</b>     | Un mese a una o due cifre.                                             |

Esempi e risultati:

In questi esempi viene utilizzato il formato della data **DD/MM/YYYY**. Il formato della data viene specificato nell'istruzione **SET DateFormat** nella parte superiore dello script di caricamento dei dati. Modificare il formato negli esempi in base alle proprie necessità.

Esempi di script

| Esempio                                          | Risultato                |
|--------------------------------------------------|--------------------------|
| setdateyearmonth<br>( '29/10/2005' ,<br>2013, 3) | Restituisce '29/03/2013' |

| Esempio                                                        | Risultato                                                                                                                                                                                                                                                                                                   |
|----------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>setdateyearmonth ('29/10/2005 04:26:14', 2013, 3)</code> | Restituisce '29/03/2013 04:26:14'<br>Per visualizzare la sezione relativa alla data e all'ora dell'indicatore temporale in una visualizzazione, è necessario impostare la formattazione dei numeri su Data e scegliere un valore per Formattazione che consenta di mostrare i valori della data e dell'ora. |

### Esempio:

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

SetYearMonth:

Load \*,

SetDateYearMonth(testdates, 2013,3) as NewYearMonth

Inline [

testdates

1/11/2012

10/12/2012

2/1/2013

19/5/2013

15/9/2013

11/12/2013

14/5/2014

13/6/2014

7/7/2014

4/8/2014

];

La tabella risultante contiene le date originali e una colonna in cui l'anno è stato impostato su 2013.

Tabella dei risultati

| testdates  | NewYearMonth |
|------------|--------------|
| 1/11/2012  | 1/3/2013     |
| 10/12/2012 | 10/3/2013    |
| 2/1/2012   | 2/3/2013     |
| 19/5/2013  | 19/3/2013    |
| 15/9/2013  | 15/3/2013    |
| 11/12/2013 | 11/3/2013    |
| 14/5/2014  | 14/3/2013    |
| 13/6/2014  | 13/3/2013    |
| 7/7/2014   | 7/3/2013     |
| 4/8/2014   | 4/3/2013     |

### timezone

Questa funzione restituisce il fuso orario impostato sul computer dove è in esecuzione il motore Qlik.

#### Sintassi:

```
TimeZone ( )
```

**Tipo di dati restituiti:** duale

#### Esempio:

```
timezone( )
```

Se si desidera visualizzare un fuso orario differente per una misura della propria app, è possibile usare la funzione `localtime()` in una misura.

### today

Questa funzione restituisce la data attuale. La funzione restituisce i valori nel formato della variabile di sistema `DateFormat`.

#### Sintassi:

```
today ([ timer_mode])
```

**Tipo di dati restituiti:** duale

La funzione `today()` può essere utilizzata sia nello script di caricamento che negli oggetti del grafico.

Il valore predefinito `timer_mode` è 1.

### Argomenti

| Argomento  | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| timer_mode | <p>Può presentare i valori seguenti:</p> <ul style="list-style-type: none"> <li>0 (giorno dell'ultimo caricamento di dati completato)</li> <li>1 (giorno della chiamata della funzione)</li> <li>2 (giorno di apertura dell'app)</li> </ul> <div>  <p><i>Se si utilizza la funzione in uno script di caricamento, <b>timer_mode=0</b> restituirà il giorno di completamento dell'ultimo caricamento dei dati, mentre <b>timer_mode=1</b> restituirà il giorno del caricamento dei dati attuale.</i></p> </div> |

### Esempi di funzioni

| valore timer_mode | Risultato se utilizzato nello script di caricamento                                                                                                                                                   | Risultato se utilizzato nell'oggetto grafico                                                                                                                                                                                                              |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0                 | Restituisce una data, nel formato della variabile di sistema DateFormat, dell'ultimo ricaricamento dei dati riuscito prima dell'ultimo ricaricamento dei dati.                                        | Restituisce una data, nel formato della variabile di sistema DateFormat, per l'ultimo ricaricamento dei dati.                                                                                                                                             |
| 1                 | Restituisce una data, nel formato della variabile di sistema DateFormat, per l'ultimo ricaricamento dei dati.                                                                                         | Restituisce una data, nel formato della variabile di sistema DateFormat, della chiamata alla funzione.                                                                                                                                                    |
| 2                 | Restituisce una data, nel formato della variabile di sistema DateFormat, per l'inizio della sessione dell'utente nell'applicazione. Non verrà aggiornato a meno che l'utente non ricarichi lo script. | Restituisce la data, nel formato della variabile di sistema DateFormat, per l'inizio della sessione dell'utente nell'applicazione. Questo valore verrà aggiornato all'inizio di una nuova sessione o quando i dati dell'applicazione verranno ricaricati. |

### Casi di utilizzo

La funzione `today()` viene comunemente utilizzata come componente di un'espressione. Ad esempio, può essere utilizzata per calcolare gli interessi accumulati in un mese fino alla data corrente.

La tabella seguente fornisce una spiegazione del risultato restituito dalla funzione `today()`, in presenza di diversi valori dell'argomento `timer_mode`:

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema,

a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - Generazione di oggetti tramite script di caricamento

Script di caricamento e risultati

#### Panoramica

L'esempio seguente crea tre variabili utilizzando la funzione `today()`. Ogni variabile utilizza una delle opzioni `timer_mode` per dimostrarne l'effetto.

Affinché le variabili dimostrino il loro scopo, ricaricare lo script e poi, dopo 24 ore, ricaricare lo script una seconda volta. In questo modo le variabili `today(0)` e `today(1)` mostreranno valori diversi, dimostrando così correttamente il loro scopo.

#### Script di caricamento

```
LET vPreviousDataLoad = today(0);  
LET vCurrentDataLoad = today(1);  
LET vApplicationOpened = today(2);
```

#### Risultati

Una volta che i dati vengono caricati una seconda volta, creare tre caselle di testo seguendo le indicazioni riportate di seguito.

Per prima cosa, creare una casella di testo per i dati che sono stati caricati precedentemente.

**Procedere come indicato di seguito:**

1. Utilizzando l'oggetto grafico **Testo e immagine**, creare una casella di testo.
2. Aggiungere la seguente misura all'oggetto:  
`=vPreviousDataLoad`
3. In **Aspetto**, selezionare **Show titles** e aggiungere il titolo 'Ora di caricamento precedente' all'oggetto.

Quindi, creare una casella di testo per i dati che si stanno caricando.

**Procedere come indicato di seguito:**

1. Utilizzando l'oggetto grafico **Testo e immagine**, creare una casella di testo.
2. Aggiungere la seguente misura all'oggetto:  
`=vCurrentDataLoad`
3. In **Aspetto**, selezionare **Show titles** e aggiungere il titolo 'Ora di caricamento attuale' all'oggetto.

Creare una casella di testo finale da mostrare quando è stata avviata la sessione dell'utente nell'applicazione.

**Procedere come indicato di seguito:**

1. Utilizzando l'oggetto grafico **Testo e immagine**, creare una casella di testo.
2. Aggiungere la seguente misura all'oggetto:  
`=vApplicationOpened`
3. In **Aspetto**, selezionare **Show titles** e aggiungere il titolo 'Inizio sessione utente' all'oggetto.

*Schema delle variabili create con la funzione `today()` nello script di caricamento*

| Previous Reload Time | Current Reload Time | User Session Began |
|----------------------|---------------------|--------------------|
| 06/22/2022           | 06/23/2022          | 06/23/2022         |

L'immagine qui sopra mostra dei valori di esempio per ciascuna delle variabili create. Ad esempio, i valori potrebbero essere i seguenti:

- Ora di ricaricamento precedente: 22/06/2022
- Ora di ricaricamento attuale: 23/06/2022
- Inizio della sessione utente: 23/06/2022

### Esempio 2 - Generazione di oggetti senza script di caricamento

Script di caricamento ed espressione del grafico

#### Panoramica

L'esempio seguente crea tre oggetti grafici utilizzando la funzione `today()`. Ogni oggetto del grafico utilizza una delle opzioni `timer_mode` per dimostrarne l'effetto.

Per questo esempio non esiste uno script di caricamento.

#### Risultati

Una volta che i dati vengono caricati una seconda volta, creare tre caselle di testo.

Per prima cosa, creare una casella di testo per l'ultimo ricaricamento di dati.

**Procedere come indicato di seguito:**

1. Utilizzando l'oggetto grafico **Testo e immagine**, creare una casella di testo.
2. Aggiungere la seguente misura.  
`=today(0)`
3. Nella sezione **Aspetto**, selezionare **Mostra titoli** e aggiungere il titolo 'Ricaricamento ultimi dati' all'oggetto.

Quindi, creare una casella di testo per mostrare l'ora corrente.

**Procedere come indicato di seguito:**

1. Utilizzando l'oggetto grafico **Testo e immagine**, creare una casella di testo.
2. Aggiungere la misura seguente:  
`=today(1)`
3. In **Aspetto**, selezionare **Mostra titoli** e aggiungere il titolo 'Ora attuale' all'oggetto.

Creare una casella di testo finale da mostrare quando è stata avviata la sessione dell'utente nell'applicazione.

**Procedere come indicato di seguito:**

1. Utilizzando l'oggetto grafico **Testo e immagine**, creare una casella di testo.
2. Aggiungere la misura seguente:  
`=today(2)`
3. In **Aspetto**, selezionare **Mostra titoli** e aggiungere il titolo 'Sessione utente iniziata' all'oggetto.

*Schema degli oggetti creato usando la funzione `today()` senza script di caricamento*

|                                         |                                   |                                         |
|-----------------------------------------|-----------------------------------|-----------------------------------------|
| <b>Latest Data Reload</b><br>06/23/2022 | <b>Current Time</b><br>06/23/2022 | <b>User Session Began</b><br>06/23/2022 |
|-----------------------------------------|-----------------------------------|-----------------------------------------|

L'immagine qui sopra mostra dei valori di esempio per ciascuno degli oggetti creati. Ad esempio, i valori potrebbero essere i seguenti:

- Data ultimo ricaricamento: 23/06/2022
- Orario corrente: 23/06/2022
- Inizio della sessione utente: 23/06/2022

L'oggetto grafico 'Ultimo caricamento dati' utilizza un valore `timer_mode` pari a 0. Restituisce il timestamp dell'ultima volta in cui i dati sono stati ricaricati correttamente.

L'oggetto grafico 'Ora corrente' utilizza un valore `timer_mode` di 1. Questo restituisce l'ora corrente secondo l'orologio di sistema. Se il foglio o l'oggetto viene ricaricato, questo valore verrà aggiornato.

L'oggetto grafico 'Sessione utente iniziata' utilizza un valore `timer_mode` di 2. Restituisce il timestamp di quando l'applicazione è stata aperta e la sessione dell'utente è iniziata.

### Esempio 3 - Scenario

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di saldi di prestiti, caricato in una tabella denominata `Loans`.
- Dati della tabella con campi per ID prestito, saldo all'inizio del mese e tasso di interesse semplice annuo applicato a ciascun prestito.

L'utente finale desidera un oggetto grafico che mostri, in base all'ID del prestito, gli interessi correnti che sono stati maturati per ciascun prestito nel mese a oggi. Anche se l'applicazione viene ricaricata solo una volta alla settimana, l'utente vorrebbe che i risultati venissero aggiornati ogni volta che l'oggetto o l'applicazione vengono aggiornati.

#### Script di caricamento

```
Loans:
Load
*
Inline
[
loan_id,start_balance,rate
8188,$10000.00,0.024
8189,$15000.00,0.057
8190,$17500.00,0.024
8191,$21000.00,0.034
8192,$90000.00,0.084
];
```

#### Risultati

Procedere come indicato di seguito:

1. Caricare i dati e aprire un foglio. Creare una nuova tabella.
2. Aggiungere i seguenti campi come dimensioni:
  - `loan_id`
  - `start_balance`
3. Quindi, creare una misura per calcolare gli interessi accumulati:

`=start_balance*(rate*(today(1)-monthstart(today(1)))/365)`

4. Impostare la **Formattazione numero** della misura su **Denaro**.

Tabella dei risultati

| loan_id | saldo iniziale | <code>=start_balance*(rate*(today(1)-yearstart(today(1)))/365)</code> |
|---------|----------------|-----------------------------------------------------------------------|
| 8188    | \$10000.00     | \$16.44                                                               |
| 8189    | \$15000.00     | \$58.56                                                               |
| 8190    | \$17500.00     | \$28.77                                                               |
| 8191    | \$21000.00     | \$48.90                                                               |
| 8192    | \$90000.00     | \$517.81                                                              |

Utilizzando la funzione `today()` per restituire la data odierna come unico argomento, la funzione `monthstart()` restituisce la data di inizio del mese corrente. Sottraendo tale risultato dalla data corrente, di nuovo usando la funzione `today()`, l'espressione restituisce il numero di giorni trascorsi fino a ora questo mese.

Questo valore viene quindi moltiplicato per il tasso di interesse e diviso per 365, per restituire il tasso di interesse effettivo incorso per questo periodo. Il risultato viene poi moltiplicato per il saldo iniziale del prestito per ottenere gli interessi maturati fino a ora in questo mese.

Poiché il valore 1 è usato come argomento `timer_mode` nelle funzioni `today()` all'interno dell'espressione, ogni volta che l'oggetto grafico viene aggiornato (aprendo l'applicazione, aggiornando la pagina, passando da un foglio all'altro, ecc.), la data restituita sarà quella corrente e i risultati saranno aggiornati di conseguenza.

## UTC

Restituisce il Coordinated Universal Time attuale.

**Sintassi:**

```
UTC ( )
```

**Tipo di dati restituiti:** duale

**Esempio:**

```
utc ( )
```

## week

Questa funzione restituisce un numero intero che rappresenta il numero della settimana corrispondente alla data inserita.

**Sintassi:**

```
week (timestamp [, first_week_day [, broken_weeks [, reference_day]])
```

Tipo di dati restituiti: numero intero

#### Argomenti

| Argomento             | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>timestamp</b>      | La data o la data e ora da valutare.                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>first_week_day</b> | <p>Specifica il giorno di inizio della settimana. Se omesso, viene utilizzato il valore della variabile <b>FirstWeekDay</b>.</p> <p>I valori possibili per <b>first_week_day</b> sono 0 per lunedì, 1 per martedì, 2 per mercoledì, 3 per giovedì, 4 per venerdì, 5 per sabato e 6 per domenica.</p> <p>Per maggiori informazioni sulle variabili di sistema, vedere <a href="#">FirstWeekDay (page 242)</a>.</p>                                                   |
| <b>broken_weeks</b>   | Se non si specificab <b>broken_weeks</b> , il valore della variabile <b>BrokenWeeks</b> verrà utilizzato per definire se le settimane sono interrotte o meno.                                                                                                                                                                                                                                                                                                       |
| <b>reference_day</b>  | Se non si specificare <b>reference_day</b> , il valore della variabile <b>ReferenceDay</b> verrà utilizzato per definire quale giorno di gennaio impostare come giorno di riferimento per definire la settimana 1. Per impostazione predefinita, le funzioni Qlik Sense utilizzano 4 come giorno di riferimento. Questo significa che la settimana 1 deve contenere il 4 gennaio, vale a dire che la settimana 1 deve sempre contenere 4 quattro giorni di gennaio. |

La funzione `week()` determina la settimana in cui cade la data e restituisce il numero della settimana.

In Qlik Sense, le impostazioni regionali vengono recuperate alla creazione dell'app e le impostazioni corrispondenti vengono memorizzate nello script come variabili d'ambiente. Queste vengono utilizzate per determinare il numero della settimana.

Ciò significa che la maggior parte degli sviluppatori di applicazioni europei ottiene le seguenti variabili d'ambiente, corrispondenti alla definizione ISO 8601:

```
Set FirstWeekDay =0;    // Monday as first week day
Set BrokenWeeks =0;    // Use unbroken weeks
Set ReferenceDay =4;    // Jan 4th is always in week 1
```

Uno sviluppatore di app nordamericano ottiene spesso le seguenti variabili d'ambiente:

```
Set FirstWeekDay =6;    // Sunday as first week day
Set BrokenWeeks =1;    // Use broken weeks
Set ReferenceDay =1;    // Jan 1st is always in week 1
```

Il primo giorno della settimana è determinato dalla variabile di sistema `FirstWeekDay`. È possibile anche modificare il primo giorno della settimana utilizzando l'argomento `first_week_day` nella funzione `week()`.

Se l'applicazione utilizza settimane interrotte, il conteggio del numero di settimane inizia il 1° gennaio e termina il giorno precedente alla variabile di sistema `FirstWeekDay`, indipendentemente dal numero di giorni trascorsi.

Se l'applicazione utilizza settimane ininterrotte, la settimana 1 può iniziare nell'anno precedente o nei primi giorni di gennaio. Dipende da come si utilizzano le variabili di ambiente `FirstWeekDay` e `ReferenceDay`.

### Casi di utilizzo

La funzione `The week()` è utile per confrontare le aggregazioni per settimane. Ad esempio, potrebbe essere usata se si desidera visualizzare le vendite totali dei prodotti in base alla settimana. La funzione `week()` viene preferita a `weekname()` quando l'utente desidera che il calcolo non utilizzi necessariamente le variabili di sistema `BrokenWeeks`, `FirstWeekDay` o `ReferenceDay` dell'applicazione.

Ad esempio, può essere usata se si desidera visualizzare le vendite totali dei prodotti in base alla settimana.

Se l'applicazione utilizza settimane ininterrotte, la settimana 1 può contenere date di dicembre dell'anno precedente o escludere date di gennaio dell'anno in corso. Se l'applicazione utilizza settimane interrotte, la settimana 1 può contenere meno di sette giorni.

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

Gli esempi riportati di seguito presuppongono

```
Set DateFormat= 'MM/DD/YYYY';  
Set FirstWeekDay=0;  
Set BrokenWeeks=0;  
Set ReferenceDay=4;
```

Esempi di funzioni

| Esempio                            | Risultato                                                         |
|------------------------------------|-------------------------------------------------------------------|
| <code>week('12/28/2021')</code>    | Restituisce 52.                                                   |
| <code>week(44614)</code>           | Restituisce 8, poiché questo è il numero di serie del 22/02/2022. |
| <code>week('01/03/2021')</code>    | Restituisce 53.                                                   |
| <code>week('01/03/2021', 6)</code> | Restituisce 1.                                                    |

### Esempio 1 - Variabili di sistema predefinite

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per l'ultima settimana del 2021 e le prime due settimane del 2022 caricato in una tabella denominata `Transactions`.
- Il campo della data fornito nel formato della variabile di sistema `DateFormat` (MM/GG/AAAA).
- La creazione di un campo, `week_number`, che restituisce l'anno e il numero di settimana in cui sono avvenute le transazioni.
- La creazione di un campo chiamato `week_day`, che mostra il valore del giorno della settimana per ogni data di transazione.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
SET FirstWeekDay=6;
SET BrokenWeeks=1;
SET ReferenceDay=0;

Transactions:
    Load
        *,
        weekDay(date) as week_day,
        week(date) as week_number
    ;

Load
*
Inline
[
id,date,amount
8183,12/27/2021,58.27
8184,12/28/2021,67.42
8185,12/29/2021,23.80
8186,12/30/2021,82.06
8187,12/31/2021,40.56
8188,01/01/2022,37.23
8189,01/02/2022,17.17
8190,01/03/2022,88.27
8191,01/04/2022,57.42
8192,01/05/2022,53.80
8193,01/06/2022,82.06
8194,01/07/2022,40.56
8195,01/08/2022,53.67
8196,01/09/2022,26.63
8197,01/10/2022,72.48
8198,01/11/2022,18.37
```

```
8199,01/12/2022,45.26
8200,01/13/2022,58.23
8201,01/14/2022,18.52
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- date
- week\_day
- week\_number

Tabella dei risultati

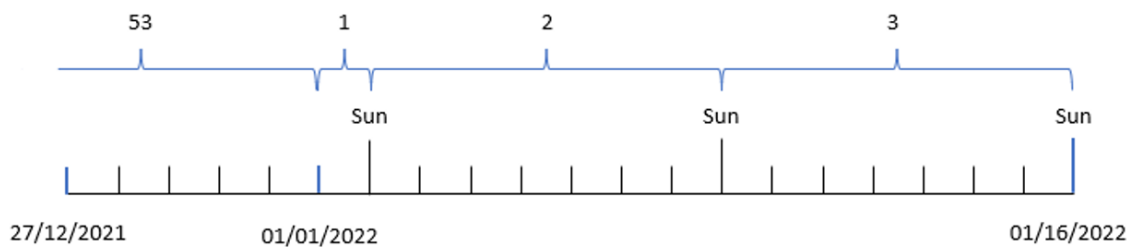
| id   | date       | week_day | numero_settimana |
|------|------------|----------|------------------|
| 8183 | 12/27/2021 | Mon      | 53               |
| 8184 | 28/12/2021 | Tue      | 53               |
| 8185 | 29/12/2021 | Wed      | 53               |
| 8186 | 30/12/2021 | Thu      | 53               |
| 8187 | 12/31/2021 | Ven      | 53               |
| 8188 | 01/01/2022 | Sat      | 1                |
| 8189 | 01/02/2022 | Sun      | 2                |
| 8190 | 01/03/2022 | Mon      | 2                |
| 8191 | 01/04/2022 | Tue      | 2                |
| 8192 | 01/05/2022 | Wed      | 2                |
| 8193 | 01/06/2022 | Thu      | 2                |
| 8194 | 01/07/2022 | Ven      | 2                |
| 8195 | 08/01/2022 | Sat      | 2                |
| 8196 | 09/01/2022 | Sun      | 3                |
| 8197 | 10/01/2022 | Mon      | 3                |
| 8198 | 01/11/2022 | Tue      | 3                |
| 8199 | 12/01/2022 | Wed      | 3                |
| 8200 | 01/13/2022 | Thu      | 3                |
| 8201 | 01/14/2022 | Ven      | 3                |

Il campo week\_number viene creato nell'istruzione di caricamento precedente mediante l'uso della funzione week() e trasferendo il campo date come argomento della funzione.

Non vengono passati altri parametri alla funzione e quindi sono attive le seguenti variabili predefinite che influenzano la funzione `week()`:

- `BrokenWeeks`: il conteggio delle settimane inizia il 1 gennaio
- `FirstWeekDay`: il primo giorno della settimana è la domenica

*Schema della funzione `week()`, utilizzando le variabili di sistema predefinite*



Poiché l'applicazione utilizza la variabile di sistema predefinita `BrokenWeeks`, la settimana 1 inizia il 1° gennaio, un sabato.

A causa della variabile di sistema `FirstWeekDay` predefinita, le settimane iniziano di domenica. La prima domenica dopo il 1° gennaio è il 2 gennaio, quando inizia la seconda settimana.

### Esempio 2 - `first_week_day`

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- La creazione di un campo, `week_number`, che restituisce l'anno e il numero di settimana in cui sono avvenute le transazioni.
- La creazione di un campo chiamato `week_day`, che mostra il valore del giorno della settimana per ogni data di transazione.

In questo esempio, vogliamo impostare l'inizio della settimana lavorativa al martedì.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';  
SET FirstWeekDay=6;  
SET BrokenWeeks=1;  
SET ReferenceDay=0;
```

Transactions:

```
Load  
    * ,
```

```
        weekDay(date) as week_day,
        week(date,1) as week_number
    ;
Load
*
Inline
[
id,date,amount
8183,12/27/2022,58.27
8184,12/28/2022,67.42
8185,12/29/2022,23.80
8186,12/30/2022,82.06
8187,12/31/2021,40.56
8188,01/01/2022,37.23
8189,01/02/2022,17.17
8190,01/03/2022,88.27
8191,01/04/2022,57.42
8192,01/05/2022,53.80
8193,01/06/2022,82.06
8194,01/07/2022,40.56
8195,01/08/2022,53.67
8196,01/09/2022,26.63
8197,01/10/2022,72.48
8198,01/11/2022,18.37
8199,01/12/2022,45.26
8200,01/13/2022,58.23
8201,01/14/2022,18.52
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- date
- week\_day
- week\_number

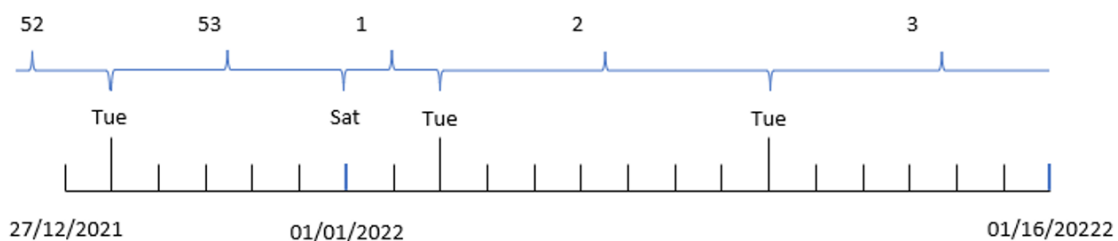
Tabella dei risultati

| id   | date       | week_day | numero_settimana |
|------|------------|----------|------------------|
| 8183 | 12/27/2021 | Mon      | 52               |
| 8184 | 28/12/2021 | Tue      | 53               |
| 8185 | 29/12/2021 | Wed      | 53               |
| 8186 | 30/12/2021 | Thu      | 53               |
| 8187 | 12/31/2021 | Ven      | 53               |
| 8188 | 01/01/2022 | Sat      | 1                |

| id   | date       | week_day | numero_settimana |
|------|------------|----------|------------------|
| 8189 | 01/02/2022 | Sun      | 1                |
| 8190 | 01/03/2022 | Mon      | 1                |
| 8191 | 01/04/2022 | Tue      | 2                |
| 8192 | 01/05/2022 | Wed      | 2                |
| 8193 | 01/06/2022 | Thu      | 2                |
| 8194 | 01/07/2022 | Ven      | 2                |
| 8195 | 08/01/2022 | Sat      | 2                |
| 8196 | 09/01/2022 | Sun      | 2                |
| 8197 | 10/01/2022 | Mon      | 2                |
| 8198 | 01/11/2022 | Tue      | 3                |
| 8199 | 12/01/2022 | Wed      | 3                |
| 8200 | 01/13/2022 | Thu      | 3                |
| 8201 | 01/14/2022 | Ven      | 3                |

L'applicazione utilizza ancora le settimane parziali. Tuttavia, l'argomento `first_week_day` è stato impostato a 1 nella funzione `week()`. In questo modo si imposta il primo giorno della settimana come martedì.

*Schema della funzione `week()`, esempio `first_week_day`*



L'applicazione utilizza la variabile di sistema predefinita `Brokenweeks`, la settimana 1 inizia il 1° gennaio, un sabato.

L'argomento `first_week_day` della funzione `week()` imposta il primo giorno della settimana a un martedì. Pertanto, la settimana 53 inizia il 28 dicembre 2021.

Tuttavia, poiché la funzione utilizza ancora le settimane parziali, la settimana 1 sarà di soli due giorni, poiché il primo martedì dopo il 1° gennaio è il 3 gennaio.

### Esempio 3 - unbroken\_weeks

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor di caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento contiene lo stesso set di dati e lo stesso scenario del primo esempio.

In questo esempio, utilizziamo settimane intere.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
SET FirstWeekDay=6;
SET BrokenWeeks=1;
SET ReferenceDay=0;

Transactions:
    Load
        *,
        weekDay(date) as week_day,
        week(date,6,0) as week_number
    ;
Load
*
Inline
[
id,date,amount
8183,12/27/2022,58.27
8184,12/28/2022,67.42
8185,12/29/2022,23.80
8186,12/30/2022,82.06
8187,12/31/2021,40.56
8188,01/01/2022,37.23
8189,01/02/2022,17.17
8190,01/03/2022,88.27
8191,01/04/2022,57.42
8192,01/05/2022,53.80
8193,01/06/2022,82.06
8194,01/07/2022,40.56
8195,01/08/2022,53.67
8196,01/09/2022,26.63
8197,01/10/2022,72.48
8198,01/11/2022,18.37
8199,01/12/2022,45.26
8200,01/13/2022,58.23
8201,01/14/2022,18.52
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- date
- week\_day
- week\_number

*Schema della funzione week(), esempio di oggetto grafico*

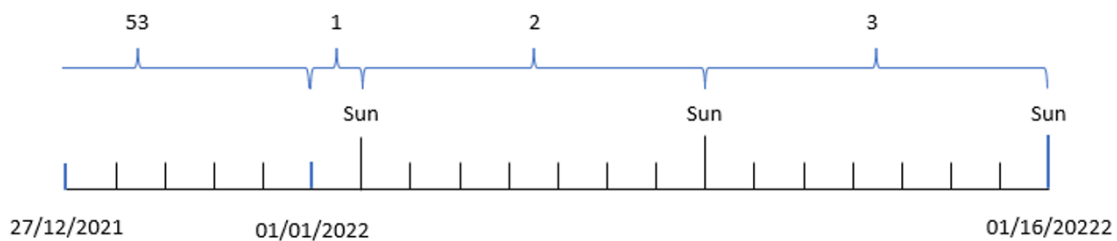


Tabella dei risultati

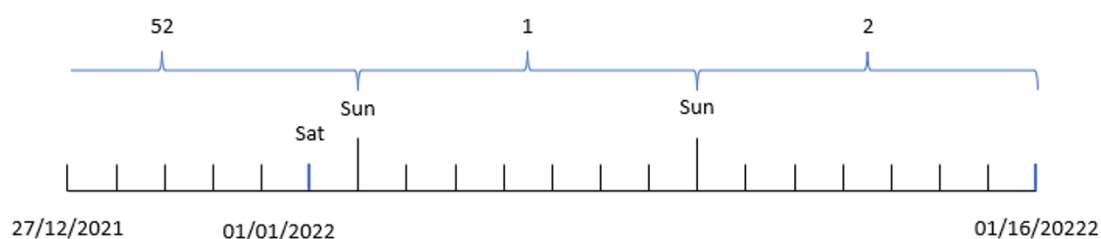
| id   | date       | week_day | numero_settimana |
|------|------------|----------|------------------|
| 8183 | 12/27/2021 | Mon      | 52               |
| 8184 | 28/12/2021 | Tue      | 52               |
| 8185 | 29/12/2021 | Wed      | 52               |
| 8186 | 30/12/2021 | Thu      | 52               |
| 8187 | 12/31/2021 | Ven      | 52               |
| 8188 | 01/01/2022 | Sat      | 52               |
| 8189 | 01/02/2022 | Sun      | 1                |
| 8190 | 01/03/2022 | Mon      | 1                |
| 8191 | 01/04/2022 | Tue      | 1                |
| 8192 | 01/05/2022 | Wed      | 1                |
| 8193 | 01/06/2022 | Thu      | 1                |
| 8194 | 01/07/2022 | Ven      | 1                |
| 8195 | 08/01/2022 | Sat      | 1                |
| 8196 | 09/01/2022 | Sun      | 2                |
| 8197 | 10/01/2022 | Mon      | 2                |
| 8198 | 01/11/2022 | Tue      | 2                |

| id   | date       | week_day | numero_settimana |
|------|------------|----------|------------------|
| 8199 | 12/01/2022 | Wed      | 2                |
| 8200 | 01/13/2022 | Thu      | 2                |
| 8201 | 01/14/2022 | Ven      | 2                |

Il parametro `first_week_date` è impostato su 1, rendendo il martedì il primo giorno della settimana. Il parametro `broken_weeks` viene impostato a 0, forzando la funzione a utilizzare settimane intere. Infine, il terzo parametro imposta il `reference_day` su 2.

Il parametro `first_week_date` è impostato su 6, rendendo la domenica il primo giorno della settimana. Il parametro `broken_weeks` è impostato su 0, obbligando la funzione a utilizzare settimane intere.

*Schema della funzione `week()`, esempio di utilizzo di settimane intere*



Utilizzando le settimane intere, la settimana 1 non inizia necessariamente il 1° gennaio, ma deve avere un minimo di quattro giorni. Pertanto, nel set di dati, la settimana 52 si conclude sabato 1° gennaio 2022. La settimana 1 inizia con la variabile di sistema `Firstweekday`, domenica 2 gennaio. Questa settimana si concluderà il sabato successivo, l'8 gennaio.

### Esempio 4 - `reference_day`

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati e lo stesso scenario del terzo esempio.
- La creazione di un campo, `week_number`, che restituisce l'anno e il numero di settimana in cui sono avvenute le transazioni.
- La creazione di un campo chiamato `week_day`, che mostra il valore del giorno della settimana per ogni data di transazione.

Inoltre, devono essere soddisfatte le seguenti condizioni:

- La settimana lavorativa inizia di martedì.
- L'azienda utilizza settimane intere.
- Il valore `reference_day` è 2. In altre parole, il numero minimo di giorni a gennaio nella settimana 1 sarà 2.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
SET FirstWeekDay=6;
SET BrokenWeeks=1;
SET ReferenceDay=0;
```

Transactions:

```
Load
    *,
    weekDay(date) as week_day,
    week(date,1,0,2) as week_number
;
```

Load

\*

Inline

```
[
id,date,amount
8183,12/27/2022,58.27
8184,12/28/2022,67.42
8185,12/29/2022,23.80
8186,12/30/2022,82.06
8187,12/31/2021,40.56
8188,01/01/2022,37.23
8189,01/02/2022,17.17
8190,01/03/2022,88.27
8191,01/04/2022,57.42
8192,01/05/2022,53.80
8193,01/06/2022,82.06
8194,01/07/2022,40.56
8195,01/08/2022,53.67
8196,01/09/2022,26.63
8197,01/10/2022,72.48
8198,01/11/2022,18.37
8199,01/12/2022,45.26
8200,01/13/2022,58.23
8201,01/14/2022,18.52
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `id`
- `date`

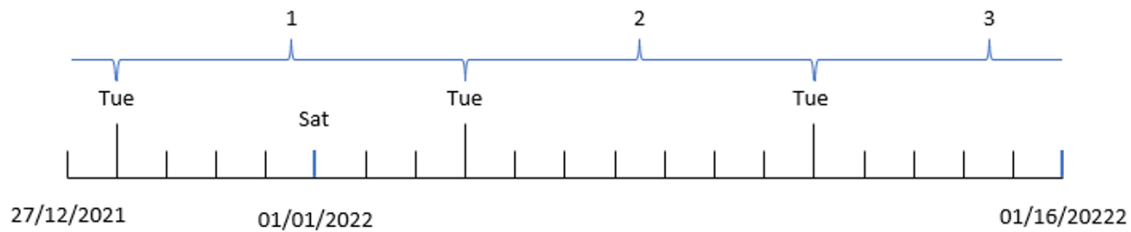
- week\_day
- week\_number

Tabella dei risultati

| id   | date       | week_day | numero_settimana |
|------|------------|----------|------------------|
| 8183 | 12/27/2021 | Mon      | 52               |
| 8184 | 28/12/2021 | Tue      | 1                |
| 8185 | 29/12/2021 | Wed      | 1                |
| 8186 | 30/12/2021 | Thu      | 1                |
| 8187 | 12/31/2021 | Ven      | 1                |
| 8188 | 01/01/2022 | Sat      | 1                |
| 8189 | 01/02/2022 | Sun      | 1                |
| 8190 | 01/03/2022 | Mon      | 1                |
| 8191 | 01/04/2022 | Tue      | 2                |
| 8192 | 01/05/2022 | Wed      | 2                |
| 8193 | 01/06/2022 | Thu      | 2                |
| 8194 | 01/07/2022 | Ven      | 2                |
| 8195 | 08/01/2022 | Sat      | 2                |
| 8196 | 09/01/2022 | Sun      | 2                |
| 8197 | 10/01/2022 | Mon      | 2                |
| 8198 | 01/11/2022 | Tue      | 3                |
| 8199 | 12/01/2022 | Wed      | 3                |
| 8200 | 01/13/2022 | Thu      | 3                |
| 8201 | 01/14/2022 | Ven      | 3                |

Il parametro `first_week_date` è impostato su 1, rendendo il martedì il primo giorno della settimana. Il parametro `broken_weeks` è impostato su 0, obbligando la funzione a utilizzare settimane intere. Infine, il terzo parametro imposta il parametro `reference_day` a 2.

Schema della funzione `week()`, esempio `reference_day`



Con la funzione che utilizza settimane intere e un valore `reference_day` di 2 come parametro, la settimana 1 deve includere solo due giorni di gennaio. Poiché il primo giorno feriale è il martedì, la settimana 1 inizia il 28 dicembre 2021 e si conclude lunedì 3 gennaio 2022.

### Esempio 5 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento contiene lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, in questo esempio, il set di dati invariato viene caricato nell'applicazione. Il calcolo che restituisce il numero della settimana viene creato come misura in un oggetto grafico.

#### Script di caricamento

Transactions:

Load

\*

Inline

[

id,date,amount

8183,12/27/2022,58.27

8184,12/28/2022,67.42

8185,12/29/2022,23.80

8186,12/30/2022,82.06

8187,12/31/2021,40.56

8188,01/01/2022,37.23

8189,01/02/2022,17.17

8190,01/03/2022,88.27

8191,01/04/2022,57.42

8192,01/05/2022,53.80

8193,01/06/2022,82.06

8194,01/07/2022,40.56

8195,01/08/2022,53.67

8196,01/09/2022,26.63

8197,01/10/2022,72.48

```
8198,01/11/2022,18.37
8199,01/12/2022,45.26
8200,01/13/2022,58.23
8201,01/14/2022,18.52
];
```

### Risultati

#### Procedere come indicato di seguito:

1. Caricare i dati e aprire un foglio. Creare una nuova tabella.
2. Aggiungere i seguenti campi come dimensioni:
  - id
  - date
3. Quindi, creare la seguente misura:  
=week (date)
4. Creare una misura , week\_day che mostri il valore del giorno della settimana per ogni data di transazione:  
=weekday(date)

Tabella dei risultati

| id   | date       | =settimana(data) | =Giorno feriale(data) |
|------|------------|------------------|-----------------------|
| 8183 | 12/27/2021 | 53               | Mon                   |
| 8184 | 28/12/2021 | 53               | Tue                   |
| 8185 | 29/12/2021 | 53               | Wed                   |
| 8186 | 30/12/2021 | 53               | Thu                   |
| 8187 | 12/31/2021 | 53               | Ven                   |
| 8188 | 01/01/2022 | 1                | Sat                   |
| 8189 | 01/02/2022 | 2                | Sun                   |
| 8190 | 01/03/2022 | 2                | Mon                   |
| 8191 | 01/04/2022 | 2                | Tue                   |
| 8192 | 01/05/2022 | 2                | Wed                   |
| 8193 | 01/06/2022 | 2                | Thu                   |
| 8194 | 01/07/2022 | 2                | Ven                   |
| 8195 | 08/01/2022 | 2                | Sat                   |
| 8196 | 09/01/2022 | 3                | Sun                   |
| 8197 | 10/01/2022 | 3                | Mon                   |
| 8198 | 01/11/2022 | 3                | Tue                   |

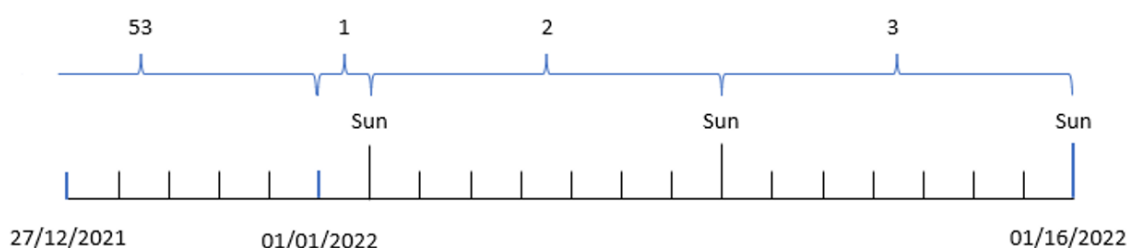
| id   | date       | =settimana(data) | =Giorno feriale(data) |
|------|------------|------------------|-----------------------|
| 8199 | 01/12/2022 | 3                | Wed                   |
| 8200 | 01/13/2022 | 3                | Thu                   |
| 8201 | 01/14/2022 | 3                | Ven                   |

Il campo `week_number` viene creato nell'istruzione `LOAD` precedente mediante l'uso della funzione `week()` e trasferendo il campo `date` come argomento della funzione.

Non vengono passati altri parametri alla funzione e quindi sono attive le seguenti variabili predefinite che influenzano la funzione `week()`:

- `BrokenWeeks`: il conteggio delle settimane inizia il 1 gennaio
- `FirstWeekDay`: il primo giorno della settimana è la domenica

*Schema della funzione `week()`, esempio di oggetto grafico*



Poiché l'applicazione utilizza la variabile di sistema predefinita `BrokenWeeks`, la settimana 1 inizia il 1° gennaio, un sabato.

A causa della variabile di sistema `FirstWeekDay` predefinita, le settimane iniziano di domenica. La prima domenica dopo il 1° gennaio è il 2 gennaio, quando inizia la seconda settimana.

### Esempio 6 - Scenario

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per l'ultima settimana del 2019 e le prime due settimane del 2020 caricato in una tabella denominata `Transactions`.
- Il campo della data fornito nel formato della variabile di sistema `DateFormat` (MM/GG/AAAA).

L'applicazione utilizza principalmente le settimane parziali nel suo dashboard. Tuttavia, l'utente finale desidera un oggetto grafico che presenti il totale delle vendite per settimana utilizzando settimane intere. Il giorno di riferimento dovrebbe essere il 2 gennaio, con le settimane che iniziano di martedì. Questo può essere ottenuto anche quando la dimensione non è disponibile nel modello dati, utilizzando la funzione `week()` come dimensione calcolata nel grafico.

### Script di caricamento

```
SET BrokenWeeks=1;  
SET ReferenceDay=0;  
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8183,12/27/2019,58.27
```

```
8184,12/28/2019,67.42
```

```
8185,12/29/2019,23.80
```

```
8186,12/30/2019,82.06
```

```
8187,12/31/2019,40.56
```

```
8188,01/01/2020,37.23
```

```
8189,01/02/2020,17.17
```

```
8190,01/03/2020,88.27
```

```
8191,01/04/2020,57.42
```

```
8192,01/05/2020,53.80
```

```
8193,01/06/2020,82.06
```

```
8194,01/07/2020,40.56
```

```
8195,01/08/2020,53.67
```

```
8196,01/09/2020,26.63
```

```
8197,01/10/2020,72.48
```

```
8198,01/11/2020,18.37
```

```
8199,01/12/2020,45.26
```

```
8200,01/13/2020,58.23
```

```
8201,01/14/2020,18.52
```

```
];
```

### Risultati

**Procedere come indicato di seguito:**

1. Caricare i dati e aprire un foglio. Creare una nuova tabella.
2. Creare la seguente dimensione calcolata:  
`=week(date)`
3. Quindi, creare la seguente misura di aggregazione:  
`=sum(amount)`
4. Impostare la **Formattazione numero** della misura su **Denaro**.

5. Selezionare il menu **Ordinamento** e, per la dimensione calcolata, rimuovere l'ordinamento personalizzato.
6. Deselezionare le opzioni **Ordina per numero** e **Ordina alfabeticamente**.

Tabella dei risultati

| settimana(data) | sum(amount) |
|-----------------|-------------|
| 52              | \$125.69    |
| 53              | \$146.42    |
| 1               | \$200.09    |
| 2               | \$347.57    |
| 3               | \$122.01    |

### weekday

Questa funzione restituisce un valore duale con:

- Il nome di un giorno come definito nella variabile di ambiente **DayNames**.
- Un numero intero compreso tra 0 e 6 che corrisponde al giorno nominale della settimana (0-6).

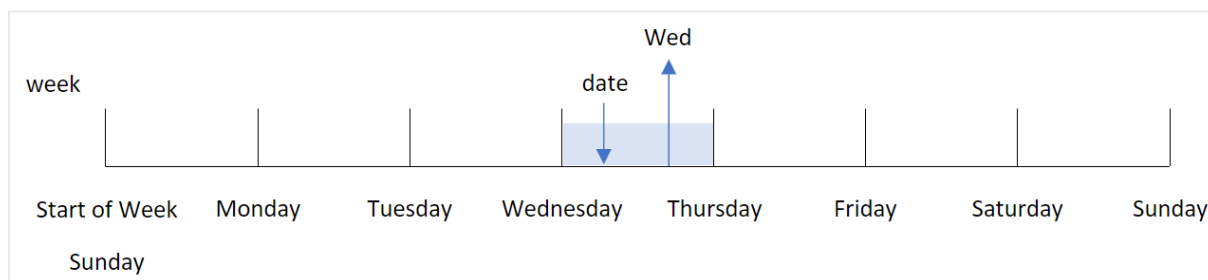
**Sintassi:**

```
weekday (date [, first_week_day=0])
```

**Tipo di dati restituiti:** duale

La funzione `weekday()` determina il giorno della settimana in cui si verifica una data. Restituisce quindi un valore stringa che rappresenta quel giorno.

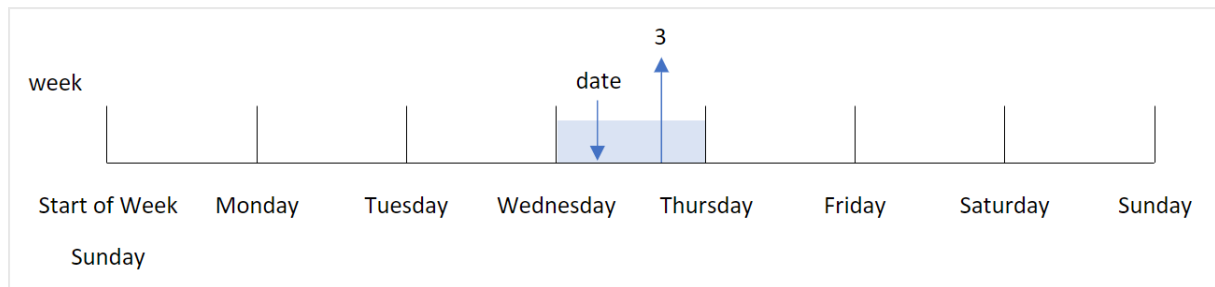
*Schema della funzione `weekday()` che restituisce il nome del giorno in cui cade una data*



Il risultato restituisce il valore numerico corrispondente a quel giorno della settimana (0-6), in base al giorno di inizio della settimana. Ad esempio, se il primo giorno della settimana è impostato su domenica, un mercoledì restituirà un valore numerico di 3. Il giorno di inizio è determinato dalla variabile di sistema `FirstWeekDay` o dal parametro della funzione `first_week_day`.

È possibile utilizzare il valore di questo numero come parte di un'espressione aritmetica. Ad esempio, moltiplicarlo per 1 per restituire il valore stesso.

Schema della funzione `weekday()` con il valore numerico del giorno visualizzato al posto del nome del giorno



### Casi di utilizzo

La funzione `weekday()` è utile quando si desidera confrontare le aggregazioni per giorno della settimana. Ad esempio, se si desidera confrontare le vendite medie dei prodotti per giorno della settimana.

Queste dimensioni possono essere create nello script di caricamento utilizzando la funzione per creare un campo in una tabella del **Calendario principale**, oppure essere create direttamente in un grafico come misura calcolata.

#### Argomenti correlati

| Argomenti                               | Interazione                                      |
|-----------------------------------------|--------------------------------------------------|
| <a href="#">FirstWeekDay (page 242)</a> | Definisce il giorno di inizio di ogni settimana. |

#### Argomenti

| Argomento             | Descrizione                                                                                                                                                            |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>date</b>           | La data o la data e ora da valutare.                                                                                                                                   |
| <b>first_week_day</b> | Specifica il giorno di inizio della settimana. Se omissso, viene utilizzato il valore della variabile <b>FirstWeekDay</b> .<br><a href="#">FirstWeekDay (page 242)</a> |

È possibile utilizzare i seguenti valori per impostare il giorno in cui inizia la settimana nell'argomento `first_week_day`:

#### valori first\_week\_day

| Giorno    | Valore |
|-----------|--------|
| Lunedì    | 0      |
| Martedì   | 1      |
| Mercoledì | 2      |
| Giovedì   | 3      |

| Giorno   | Valore |
|----------|--------|
| Venerdì  | 4      |
| Sabato   | 5      |
| Domenica | 6      |

## Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.



Salvo ove indicato diversamente, *FirstWeekDay* è impostato su 0 in questi esempi.

### Esempi di funzioni

| Esempio                                                                       | Risultato                                                                                        |
|-------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|
| <code>weekday('10/12/1971')</code>                                            | Restituisce 'Tue' e 1.                                                                           |
| <code>weekday('10/12/1971', 6)</code>                                         | Restituisce 'Tue' e 2.<br><br>In questo esempio, domenica (6) è il primo giorno della settimana. |
| <code>SET FirstWeekDay=6;</code><br>...<br><code>weekday('10/12/1971')</code> | Restituisce 'Tue' e 2.                                                                           |

## Esempio 1 - Stringa Weekday

Script di caricamento e risultati

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2022, caricato in una tabella denominata 'Transactions'.
- La variabile di sistema FirstWeekDay che è impostata su 6 (domenica).
- La variabile DayNames che è impostata per utilizzare i nomi dei giorni predefiniti.
- Un caricamento precedente che contiene la funzione weekday(), impostata come campo 'week\_day', e che restituisce il giorno della settimana in cui sono avvenute le transazioni.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
SET DayNames='Mon;Tue;Wed;Thu;Fri;Sat;Sun';
SET FirstWeekDay=6;

Transactions:
    Load
        *,
        WeekDay(date) as week_day
    ;
Load
*
Inline
[
id,date,amount
8188,01/01/2022,37.23
8189,01/02/2022,17.17
8190,01/03/2022,88.27
8191,01/04/2022,57.42
8192,01/05/2022,53.80
8193,01/06/2022,82.06
8194,01/07/2022,40.39
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- date
- week\_day

Tabella dei risultati

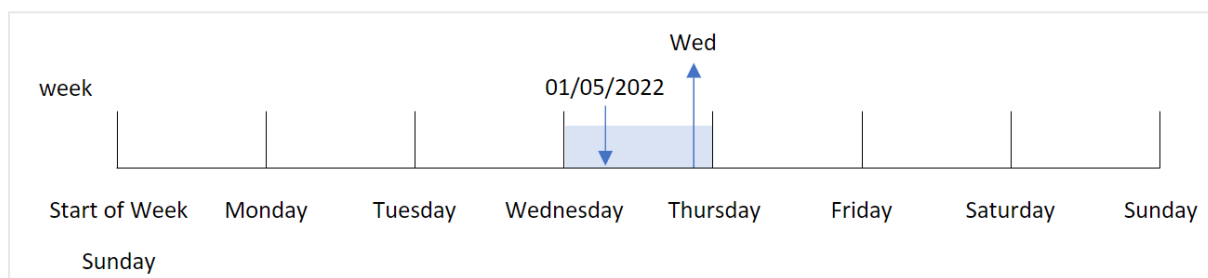
| id   | date       | week_day |
|------|------------|----------|
| 8188 | 01/01/2022 | Sat      |
| 8189 | 01/02/2022 | Sun      |
| 8190 | 01/03/2022 | Mon      |
| 8191 | 01/04/2022 | Tue      |
| 8192 | 01/05/2022 | Wed      |

| id   | date       | week_day |
|------|------------|----------|
| 8193 | 01/06/2022 | Thu      |
| 8194 | 01/07/2022 | Ven      |

Il campo 'week\_day' viene creato nell'istruzione di caricamento precedente mediante l'uso della funzione `weekday()` e trasferendo il campo data come argomento della funzione.

La funzione `weekday()` restituisce il valore della stringa del giorno della settimana, ovvero il nome del giorno della settimana impostato dalla variabile di sistema `DayNames`.

*Schema della funzione `weekday()` che restituisce mercoledì come giorno della settimana per la transazione 8192*



La transazione 8192 è avvenuta il 5 gennaio. La variabile di sistema `FirstWeekDay` imposta il primo giorno della settimana come domenica. La transazione della funzione `weekday()` ha avuto luogo un mercoledì e restituisce questo valore, nella forma abbreviata della variabile di sistema `DayNames`, nel campo `week_day`.

I valori del campo 'week\_day' sono allineati a destra nella colonna perché per il campo esiste un doppio risultato di numero e testo (mercoledì, 3). Per convertire il valore del campo nel suo equivalente numerico, il campo può essere avvolto all'interno della funzione `num()`. Ad esempio, nella transazione 8192, il valore del mercoledì verrà convertito nel numero 3.

### Esempio 2 - first\_week\_day

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2022, caricato in una tabella denominata 'Transactions'.
- La variabile di sistema `FirstWeekDay` che è impostata su 6 (domenica).
- La variabile `DayNames` che è impostata per utilizzare i nomi dei giorni predefiniti.
- Un caricamento precedente che contiene la funzione `weekday()`, impostata come campo 'week\_day', e che restituisce il giorno della settimana in cui sono avvenute le transazioni.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
SET DayNames='Mon;Tue;Wed;Thu;Fri;Sat;Sun';
SET FirstWeekDay=6;
```

```
Transactions:
    Load
        *,
        weekDay(date,1) as week_day
    ;
Load
*
Inline
[
id,date,amount
8188,01/01/2022,37.23
8189,01/02/2022,17.17
8190,01/03/2022,88.27
8191,01/04/2022,57.42
8192,01/05/2022,53.80
8193,01/06/2022,82.06
8194,01/07/2022,40.39
];
```

### Risultati

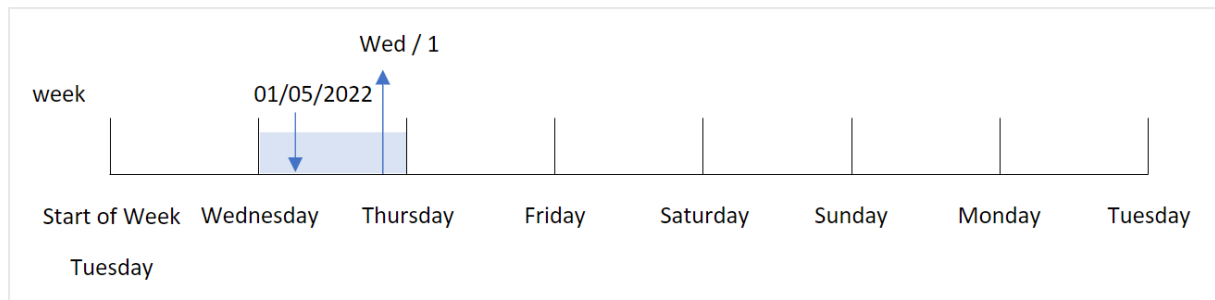
Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- date
- week\_day

Tabella dei risultati

| id   | date       | week_day |
|------|------------|----------|
| 8188 | 01/01/2022 | Sat      |
| 8189 | 01/02/2022 | Sun      |
| 8190 | 01/03/2022 | Mon      |
| 8191 | 01/04/2022 | Tue      |
| 8192 | 01/05/2022 | Wed      |
| 8193 | 01/06/2022 | Thu      |
| 8194 | 01/07/2022 | Ven      |

*Schema della funzione weekday() che mostra che mercoledì ha il valore numerico duale di 1*



Poiché l'argomento `first_week_day` è impostato su 1 nella funzione `weekday()`, il primo giorno della settimana è martedì. Pertanto, tutte le transazioni che avvengono di martedì avranno un valore numerico doppio di 0.

La transazione 8192 è avvenuta il 5 gennaio. La funzione `weekday()` identifica che si tratta di un mercoledì e quindi l'espressione restituisce il valore numerico doppio di 1.

### Esempio 3 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2022, caricato in una tabella denominata 'Transactions'.
- La variabile di sistema `FirstWeekDay` che è impostata su 6 (domenica).
- La variabile `DayNames` che è impostata per utilizzare i nomi dei giorni predefiniti.

Tuttavia, in questo esempio, il set di dati è invariato e viene caricato nell'applicazione. Il calcolo che identifica il valore del giorno della settimana viene creato come misura in un grafico dell'app.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
SET DayNames='Mon;Tue;Wed;Thu;Fri;Sat;Sun';
SET FirstWeekDay=6;
```

Transactions:

Load

\*

Inline

[

id,date,amount

8188,01/01/2022,37.23

8189,01/02/2022,17.17

8190,01/03/2022,88.27

```
8191,01/04/2022,57.42
8192,01/05/2022,53.80
8193,01/06/2022,82.06
8194,01/07/2022,40.39
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- date

Per calcolare il valore dei giorni feriali, creare la seguente misura:

- =weekday(date)

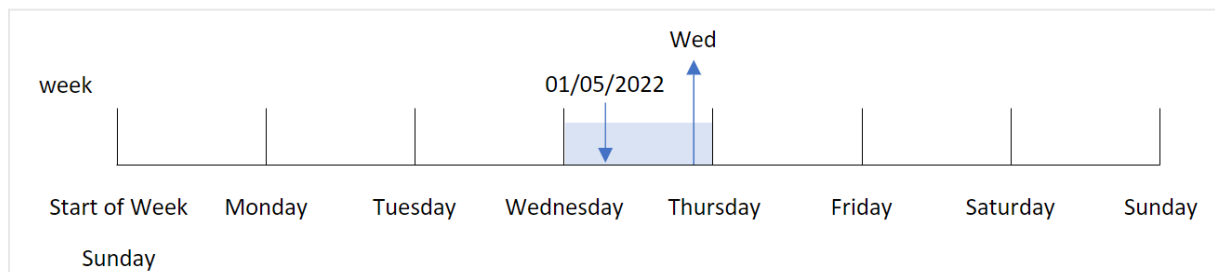
Tabella dei risultati

| id   | date       | =weekday(date) |
|------|------------|----------------|
| 8188 | 01/01/2022 | Sat            |
| 8189 | 01/02/2022 | Sun            |
| 8190 | 01/03/2022 | Mon            |
| 8191 | 01/04/2022 | Tue            |
| 8192 | 01/05/2022 | Wed            |
| 8193 | 01/06/2022 | Thu            |
| 8194 | 01/07/2022 | Ven            |

Il campo '=weekday(date)' viene creato nel grafico mediante l'utilizzo della funzione `weekday()` e trasferendo il campo data come argomento della funzione.

La funzione `weekday()` restituisce il valore della stringa del giorno della settimana, ovvero il nome del giorno della settimana impostato dalla variabile di sistema `DayNames`.

*Schema della funzione `weekday()` che restituisce mercoledì come giorno della settimana per la transazione 8192*



La transazione 8192 è avvenuta il 5 gennaio. La variabile di sistema `FirstWeekDay` imposta il primo giorno della settimana come domenica. La transazione della funzione `weekday()` ha avuto luogo un mercoledì e restituisce questo valore, nella forma abbreviata della variabile di sistema `DayNames`, nel campo `=weekday(date)`.

### Esempio 4 - Scenario

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2022, caricato in una tabella denominata 'Transactions'.
- La variabile di sistema `FirstWeekDay` che è impostata su 6 (domenica).
- La variabile `DayNames` che è impostata per utilizzare i nomi dei giorni predefiniti.

L'utente finale desidera un grafico che presenti le vendite medie per giorno della settimana per le transazioni.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';  
SET DayNames='Mon;Tue;Wed;Thu;Fri;Sat;Sun';  
SET FirstWeekDay=6;
```

Transactions:

```
LOAD  
  RecNo() AS id,  
  MakeDate(2022, 1, Ceil(Rand() * 31)) as date,  
  Rand() * 1000 AS amount
```

```
Autogenerate(1000);
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `=weekday(date)`
- `=avg(amount)`

Impostare la **Formattazione numero** della misura su **Denaro**.

Tabella dei risultati

| <b>weekday(date)</b> | <b>Avg(amount)</b> |
|----------------------|--------------------|
| Sun                  | \$536.96           |

| weekday(date) | Avg(amount) |
|---------------|-------------|
| Mon           | \$500.80    |
| Tue           | \$515.63    |
| Wed           | \$509.21    |
| Thu           | \$482.70    |
| Ven           | \$441.33    |
| Sat           | \$505.22    |

## weekend

Questa funzione restituisce un valore corrispondente a un timestamp recante l'ultimo millisecondo dell'ultimo giorno della settimana di calendario contenente **date**. Il formato di output predefinito sarà il formato **DateFormat** impostato nello script.

### Sintassi:

```
WeekEnd(timestamp [, period_no [, first_week_day ]])
```

### Tipo di dati restituiti: duale

In altre parole, la funzione `weekend()` determina in quale settimana cade la data. Quindi, restituisce un timestamp, nel formato data, per l'ultimo millisecondo di quella settimana. Il primo giorno della settimana è determinato dalla variabile ambientale `FirstWeekDay`. Tuttavia, questo può essere sostituito dall'argomento `first_week_day` della funzione `weekend()`.

### Argomenti

| Argomento             | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>timestamp</b>      | La data o la data e ora da valutare.                                                                                                                                                                                                                                                                                                                                                                              |
| <b>period_no</b>      | <b>shift</b> è un numero intero, in cui il valore 0 indica la settimana che contiene <b>date</b> . I valori negativi di <b>shift</b> indicano le settimane precedenti, mentre i valori positivi indicano le settimane successive.                                                                                                                                                                                 |
| <b>first_week_day</b> | <p>Specifica il giorno di inizio della settimana. Se omesso, viene utilizzato il valore della variabile <b>FirstWeekDay</b>.</p> <p>I valori possibili per <b>first_week_day</b> sono 0 per lunedì, 1 per martedì, 2 per mercoledì, 3 per giovedì, 4 per venerdì, 5 per sabato e 6 per domenica.</p> <p>Per maggiori informazioni sulle variabili di sistema, vedere <a href="#">FirstWeekDay (page 242)</a>.</p> |

## Casi di utilizzo

La funzione `weekend()` viene comunemente utilizzata come parte di un'espressione quando l'utente desidera che il calcolo utilizzi i giorni rimanenti della settimana per la data specificata. Ad esempio, potrebbe essere utilizzata se un utente desidera calcolare il totale degli interessi non ancora maturati

durante la settimana.

Gli esempi seguenti presuppongono:

```
SET FirstWeekDay=0;
```

| Esempio                                  | Risultato                         |
|------------------------------------------|-----------------------------------|
| <code>weekend('01/10/2013')</code>       | Restituisce 01/12/2013 23:59:59.  |
| <code>weekend('01/10/2013', -1)</code>   | Restituisce 01/05/2013 23:59:59.. |
| <code>weekend('01/10/2013', 0, 1)</code> | Restituisce 01/14/2013 23:59:59.  |

## Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempi:

Se si desiderano le impostazioni ISO per le settimane e i numeri di settimana, assicurarsi di inserire nello script quanto segue:

```
Set DateFormat = 'YYYY-MM-DD';
Set FirstWeekDay = 0; // Monday as first week day
Set BrokenWeeks = 0; //(use unbroken weeks)
Set ReferenceDay = 4; // Jan 4th is always in week 1
```

Se si desiderano le impostazioni USA, assicurarsi che nello script sia presente quanto segue:

```
Set DateFormat = 'M/D/YYYY';
Set FirstWeekDay = 6; // Sunday as first week day
Set BrokenWeeks = 1; //(use broken weeks)
Set ReferenceDay = 1; // Jan 1st is always in week 1
```

Gli esempi sopra riportati danno come risultato quanto segue dalla funzione `weekend()`:

Esempio di funzione Weekend

| Data            | Fine settimana ISO | Fine settimana USA |
|-----------------|--------------------|--------------------|
| Sab 26 dic 2020 | 2020-12-27         | 12/26/2020         |

| Data            | Fine settimana ISO | Fine settimana USA |
|-----------------|--------------------|--------------------|
| Dom 27 dic 2020 | 2020-12-27         | 1/2/2021           |
| Lun 28 dic 2020 | 2021-01-03         | 1/2/2021           |
| Mar 29 dic 2020 | 2021-01-03         | 1/2/2021           |
| Mer 30 dic 2020 | 2021-01-03         | 1/2/2021           |
| Gio 31 dic 2020 | 2021-01-03         | 1/2/2021           |
| Ven 1° gen 2021 | 2021-01-03         | 1/2/2021           |
| Sab 2 gen 2021  | 2021-01-03         | 1/2/2021           |
| Dom 3 gen 2021  | 2021-01-03         | 1/9/2021           |
| Lun 4 gen 2021  | 2021-01-10         | 1/9/2021           |
| Mar 5 gen 2021  | 2021-01-10         | 1/9/2021           |



*Gli ultimi giorni della settimana corrispondono alla domenica nella colonna ISO e al sabato nella colonna USA.*

### Esempio 1 - Esempio di base

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2022, caricato in una tabella denominata 'Transactions'.
- Il campo della data fornito nel formato della variabile di sistema DateFormat (MM/GG/AAAA).
- La creazione di un campo, end\_of\_week, che restituisce un timestamp per la fine della settimana in cui sono avvenute le transazioni.

#### Script di caricamento

```
SET FirstWeekDay=6;
```

```
Transactions:
```

```
  Load
    *,
    weekend(date) as end_of_week,
    timestamp(weekend(date)) as end_of_week_timestamp
  ;
```

```
Load
*
```

```
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- end\_of\_week
- end\_of\_week\_timestamp

Tabella dei risultati

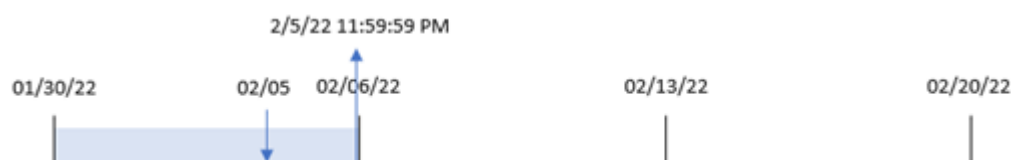
| date      | end_of_week | end_of_week_timestamp |
|-----------|-------------|-----------------------|
| 1/7/2022  | 01/08/2022  | 1/8/2022 11:59:59 PM  |
| 1/19/2022 | 01/22/2022  | 1/22/2022 11:59:59 PM |
| 2/5/2022  | 02/05/2022  | 2/5/2022 11:59:59 PM  |
| 2/28/2022 | 03/05/2022  | 3/5/2022 11:59:59 PM  |
| 3/16/2022 | 03/19/2022  | 3/19/2022 11:59:59 PM |
| 4/1/2022  | 04/02/2022  | 4/2/2022 11:59:59 PM  |
| 5/7/2022  | 05/07/2022  | 5/7/2022 11:59:59 PM  |
| 5/16/2022 | 05/21/2022  | 5/21/2022 11:59:59 PM |
| 6/15/2022 | 06/18/2022  | 6/18/2022 11:59:59 PM |

| date       | end_of_week | end_of_week_timestamp  |
|------------|-------------|------------------------|
| 6/26/2022  | 07/02/2022  | 7/2/2022 11:59:59 PM   |
| 7/9/2022   | 07/09/2022  | 7/9/2022 11:59:59 PM   |
| 7/22/2022  | 07/23/2022  | 7/23/2022 11:59:59 PM  |
| 7/23/2022  | 07/23/2022  | 7/23/2022 11:59:59 PM  |
| 7/27/2022  | 07/30/2022  | 7/30/2022 11:59:59 PM  |
| 8/2/2022   | 08/06/2022  | 8/6/2022 11:59:59 PM   |
| 8/8/2022   | 08/13/2022  | 8/13/2022 11:59:59 PM  |
| 8/19/2022  | 08/20/2022  | 8/20/2022 11:59:59 PM  |
| 9/26/2022  | 10/01/2022  | 10/1/2022 11:59:59 PM  |
| 10/14/2022 | 10/15/2022  | 10/15/2022 11:59:59 PM |
| 10/29/2022 | 10/29/2022  | 10/29/2022 11:59:59 PM |

Il campo 'end\_of\_week' viene creato nell'istruzione di caricamento precedente mediante l'uso della funzione `weekend()` e trasferendo il campo data come argomento della funzione.

La funzione `weekend()` identifica in quale settimana rientra il valore della data e restituisce un timestamp per il primo millisecondo di quella settimana.

*Schema della funzione `weekend()`, esempio di base*



La transazione 8191 è avvenuta il 5 febbraio. La variabile di sistema `FirstWeekDay` imposta il primo giorno della settimana come domenica. La funzione `weekend()` identifica che il primo sabato dopo il 5 febbraio - e quindi la fine della settimana - era il 5 febbraio. Pertanto, il valore `end_of_week` per questa transazione restituisce l'ultimo millisecondo di quel giorno, ovvero il 5 febbraio alle 23:59:59.

### Esempio 2 - period\_no

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.
- La creazione di un campo, `previous_week_end`, che restituisce il timestamp dell'inizio della settimana precedente la transazione.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*,  
weekend(date,-1) as previous_week_end,  
timestamp(weekend(date,-1)) as previous_week_end_timestamp  
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
```

```
8189,1/19/2022,37.23
```

```
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
```

```
8192,3/16/2022,53.80
```

```
8193,4/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/16/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
```

```
8200,7/23/2022,82.77
```

```
8201,7/27/2022,69.98
```

```
8202,8/2/2022,76.11
```

```
8203,8/8/2022,25.12
```

```
8204,8/19/2022,46.23
```

```
8205,9/26/2022,84.21
```

```
8206,10/14/2022,96.24
```

```
8207,10/29/2022,67.67
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

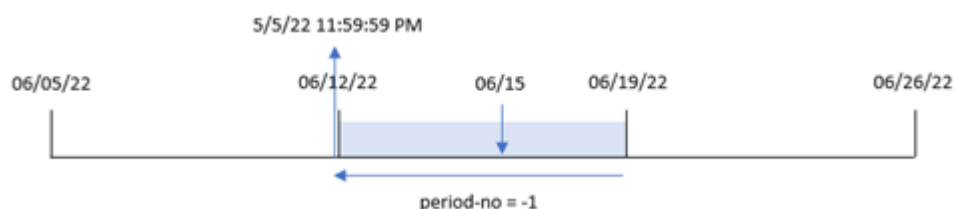
- `date`
- `previous_week_end`
- `previous_week_end_timestamp`

Tabella dei risultati

| date       | end_of_week | end_of_week_timestamp  |
|------------|-------------|------------------------|
| 1/7/2022   | 01/01/2022  | 1/1/2022 11:59:59 PM   |
| 1/19/2022  | 01/15/2022  | 1/15/2022 11:59:59 PM  |
| 2/5/2022   | 01/29/2022  | 1/29/2022 11:59:59 PM  |
| 2/28/2022  | 02/26/2022  | 2/26/2022 11:59:59 PM  |
| 3/16/2022  | 03/12/2022  | 3/12/2022 11:59:59 PM  |
| 4/1/2022   | 03/26/2022  | 3/26/2022 11:59:59 PM  |
| 5/7/2022   | 04/30/2022  | 4/30/2022 11:59:59 PM  |
| 5/16/2022  | 05/14/2022  | 5/14/2022 11:59:59 PM  |
| 6/15/2022  | 06/11/2022  | 6/11/2022 11:59:59 PM  |
| 6/26/2022  | 06/25/2022  | 6/25/2022 11:59:59 PM  |
| 7/9/2022   | 07/02/2022  | 7/2/2022 11:59:59 PM   |
| 7/22/2022  | 07/16/2022  | 7/16/2022 11:59:59 PM  |
| 7/23/2022  | 07/16/2022  | 7/16/2022 11:59:59 PM  |
| 7/27/2022  | 07/23/2022  | 7/23/2022 11:59:59 PM  |
| 8/2/2022   | 07/30/2022  | 7/30/2022 11:59:59 PM  |
| 8/8/2022   | 08/06/2022  | 8/6/2022 11:59:59 PM   |
| 8/19/2022  | 08/13/2022  | 8/13/2022 11:59:59 PM  |
| 9/26/2022  | 09/24/2022  | 9/24/2022 11:59:59 PM  |
| 10/14/2022 | 10/08/2022  | 10/8/2022 11:59:59 PM  |
| 10/29/2022 | 10/22/2022  | 10/22/2022 11:59:59 PM |

In questo caso, poiché il valore `period_no` di -1 è stato utilizzato come argomento `offset` nella funzione `weekend()`, la funzione per prima cosa identifica la settimana in cui sono avvenute le transazioni. Quindi, cerca la settimana precedente e identifica l'ultimo millisecondo di quella settimana.

*Schema della funzione `weekend()`, esempio di `period_no`*



La transazione 8196 è avvenuta il 15 giugno. La funzione `weekend()` identifica che la settimana è iniziata il 12 giugno. Pertanto, la settimana precedente termina l'11 giugno alle 11:59:59 PM; ciò rappresenta il valore restituito per il campo `previous_week_end`.

### Esempio 3 - first\_week\_day

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor di caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento contiene lo stesso set di dati e lo stesso scenario del primo esempio. Tuttavia, in questo esempio, dobbiamo impostare il martedì come primo giorno della settimana lavorativa.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
```

```
        *
```

```
        weekend(date,0,1) as end_of_week,
```

```
        timestamp(weekend(date,0,1)) as end_of_week_timestamp,
```

```
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
```

```
8189,1/19/2022,37.23
```

```
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
```

```
8192,3/16/2022,53.80
```

```
8193,4/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/16/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
```

```
8200,7/23/2022,82.77
```

```
8201,7/27/2022,69.98
```

```
8202,8/2/2022,76.11
```

```
8203,8/8/2022,25.12
```

```
8204,8/19/2022,46.23
```

```
8205,9/26/2022,84.21
```

```
8206,10/14/2022,96.24
```

```
8207,10/29/2022,67.67
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- end\_of\_week
- end\_of\_week\_timestamp

Tabella dei risultati

| date       | end_of_week | end_of_week_timestamp  |
|------------|-------------|------------------------|
| 1/7/2022   | 01/10/2022  | 1/10/2022 11:59:59 PM  |
| 1/19/2022  | 01/24/2022  | 1/24/2022 11:59:59 PM  |
| 2/5/2022   | 02/07/2022  | 2/7/2022 11:59:59 PM   |
| 2/28/2022  | 02/28/2022  | 2/28/2022 11:59:59 PM  |
| 3/16/2022  | 03/21/2022  | 3/21/2022 11:59:59 PM  |
| 4/1/2022   | 04/04/2022  | 4/4/2022 11:59:59 PM   |
| 5/7/2022   | 05/09/2022  | 5/9/2022 11:59:59 PM   |
| 5/16/2022  | 05/16/2022  | 5/16/2022 11:59:59 PM  |
| 6/15/2022  | 06/20/2022  | 6/20/2022 11:59:59 PM  |
| 6/26/2022  | 06/27/2022  | 6/27/2022 11:59:59 PM  |
| 7/9/2022   | 07/11/2022  | 7/11/2022 11:59:59 PM  |
| 7/22/2022  | 07/25/2022  | 7/25/2022 11:59:59 PM  |
| 7/23/2022  | 07/25/2022  | 7/25/2022 11:59:59 PM  |
| 7/27/2022  | 08/01/2022  | 8/1/2022 11:59:59 PM   |
| 8/2/2022   | 08/08/2022  | 8/8/2022 11:59:59 PM   |
| 8/8/2022   | 08/08/2022  | 8/8/2022 11:59:59 PM   |
| 8/19/2022  | 08/22/2022  | 8/22/2022 11:59:59 PM  |
| 9/26/2022  | 09/26/2022  | 9/26/2022 11:59:59 PM  |
| 10/14/2022 | 10/17/2022  | 10/17/2022 11:59:59 PM |
| 10/29/2022 | 10/31/2022  | 10/31/2022 11:59:59 PM |

In questa istanza, dato che l'argomento `first_week_date` di 1 è utilizzato nella funzione `weekend()`, imposta il primo giorno della settimana a martedì.

Schema della funzione `weekend()`, esempio `first_week_day`



La transazione 8191 è avvenuta il 5 febbraio. La funzione `weekend()` identifica che il primo lunedì successivo a questa data - e quindi la fine della settimana e il valore restituito - era il 6 febbraio alle 23:59:59.

### Esempio 4 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento contiene lo stesso set di dati e lo stesso scenario del primo esempio. Tuttavia, in questo esempio, il set di dati invariato viene caricato nell'applicazione. Il calcolo che restituisce un timestamp per la fine della settimana in cui sono avvenute le transazioni viene creato come misura in un oggetto grafico dell'applicazione.

#### Script di caricamento

```
Transactions:
Load
*
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
```

```
8206,10/14/2022,96.24  
8207,10/29/2022,67.67  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: date.

Per calcolare l'inizio della settimana in cui avviene una transazione, aggiungere le seguenti misure:

- =weekend(date)
- =timestamp(weekend(date))

Tabella dei risultati

| date       | =weekend(date) | =timestamp(weekend(date)) |
|------------|----------------|---------------------------|
| 1/7/2022   | 01/08/2022     | 1/8/2022 11:59:59 PM      |
| 1/19/2022  | 01/22/2022     | 1/22/2022 11:59:59 PM     |
| 2/5/2022   | 02/05/2022     | 2/5/2022 11:59:59 PM      |
| 2/28/2022  | 03/05/2022     | 3/5/2022 11:59:59 PM      |
| 3/16/2022  | 03/19/2022     | 3/19/2022 11:59:59 PM     |
| 4/1/2022   | 04/02/2022     | 4/2/2022 11:59:59 PM      |
| 5/7/2022   | 05/07/2022     | 5/7/2022 11:59:59 PM      |
| 5/16/2022  | 05/21/2022     | 5/21/2022 11:59:59 PM     |
| 6/15/2022  | 06/18/2022     | 6/18/2022 11:59:59 PM     |
| 6/26/2022  | 07/02/2022     | 7/2/2022 11:59:59 PM      |
| 7/9/2022   | 07/09/2022     | 7/9/2022 11:59:59 PM      |
| 7/22/2022  | 07/23/2022     | 7/23/2022 11:59:59 PM     |
| 7/23/2022  | 07/23/2022     | 7/23/2022 11:59:59 PM     |
| 7/27/2022  | 07/30/2022     | 7/30/2022 11:59:59 PM     |
| 8/2/2022   | 08/06/2022     | 8/6/2022 11:59:59 PM      |
| 8/8/2022   | 08/13/2022     | 8/13/2022 11:59:59 PM     |
| 8/19/2022  | 08/20/2022     | 8/20/2022 11:59:59 PM     |
| 9/26/2022  | 10/01/2022     | 10/1/2022 11:59:59 PM     |
| 10/14/2022 | 10/15/2022     | 10/15/2022 11:59:59 PM    |
| 10/29/2022 | 10/29/2022     | 10/29/2022 11:59:59 PM    |

La misura `end_of_week` viene creata nell'oggetto grafico mediante l'utilizzo della funzione `weekend()` e trasferendo il campo `data` come argomento della funzione. La funzione `weekend()` identifica in quale settimana cade il valore della data, restituendo un timestamp per l'ultimo millisecondo di quella settimana.

*Schema della funzione `weekend()`, esempio di oggetto grafico*



La transazione 8191 è avvenuta il 5 febbraio. La variabile di sistema `FirstWeekDay` imposta il primo giorno della settimana come domenica. La funzione `weekend()` identifica che il primo sabato dopo il 5 febbraio - e quindi la fine della settimana - era il 5 febbraio. Pertanto, il valore `end_of_week` per questa transazione restituisce l'ultimo millisecondo di quel giorno, ovvero il 5 febbraio alle 23:59:59.

### Esempio 5 - Scenario

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella chiamata `Employee_Expenses`.
- I dati consistono negli ID dei dipendenti, nei nomi dei dipendenti e nelle richieste di rimborso delle spese medie giornaliere di ciascun dipendente.

L'utente finale desidera un oggetto grafico che visualizzi, in base all'ID dipendente e al nome del dipendente, le richieste di rimborso spese stimate ancora da sostenere per il resto della settimana.

#### Script di caricamento

```
Employee_Expenses:
Load
*
Inline
[
employee_id,employee_name,avg_daily_claim
182,Mark, $15
183,Deryck, $12.5
184,Dexter, $12.5
185,Sydney,$27
186,Agatha,$18
];
```

## Risultati

### Procedere come indicato di seguito:

1. Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:
  - employee\_id
  - employee\_name
2. Quindi, creare una misura per calcolare gli interessi accumulati:  
$$=(\text{weekend}(\text{today}(1))-\text{today}(1))*\text{avg\_daily\_claim}$$
3. Impostare la **Formattazione numero** della misura su **Denaro**.

Tabella dei risultati

| employee_id | employee_name | =(weekend(today(1))-today(1))*avg_daily_claim |
|-------------|---------------|-----------------------------------------------|
| 182         | Contrassegno  | \$90.00                                       |
| 183         | Deryck        | \$75.00                                       |
| 184         | Dexter        | \$75.00                                       |
| 185         | Sydney        | \$162.00                                      |
| 186         | Agatha        | \$108.00                                      |

La funzione `weekend()`, utilizzando come unico argomento la data odierna, restituisce la data finale della settimana corrente. Quindi, sottraendo la data odierna dalla data di fine settimana, l'espressione restituisce il numero di giorni rimanenti di questa settimana.

Questo valore viene quindi moltiplicato per la media delle richieste di rimborso spese giornaliere di ciascun dipendente per calcolare il valore stimato delle richieste che ogni dipendente dovrebbe presentare durante il periodo rimanente della settimana.

## weekname

Questa funzione restituisce un valore che mostra l'anno e il numero della settimana con un valore numerico sottostante corrispondente a un indicatore temporale recante il primo millisecondo del primo giorno della settimana contenente **date**.

### Sintassi:

```
WeekName (date[, period_no [, first_week_day [, broken_weeks [, reference_ day]]]])
```

La funzione `weekname()` determina la settimana in cui cade la data e restituisce il numero della settimana e l'anno per quella settimana. Il primo giorno della settimana è determinato dalla variabile di sistema `FirstWeekDay`. Tuttavia, è possibile anche modificare il primo giorno della settimana utilizzando l'argomento `first_week_day` nella funzione `weekname()`.

In Qlik Sense, le impostazioni regionali vengono recuperate alla creazione dell'app e le impostazioni corrispondenti vengono memorizzate nello script come variabili d'ambiente.

Uno sviluppatore di app nordamericano ottiene spesso `set BrokenWeeks=1`; nello script, corrispondente a settimane interrotte. Uno sviluppatore di app europeo ottiene spesso `set BrokenWeeks=0`; nello script, corrispondente a settimane intere.

Se l'applicazione utilizza settimane interrotte, il conteggio del numero di settimane inizia il 1° gennaio e termina il giorno precedente alla variabile di sistema `FirstWeekDay`, indipendentemente dal numero di giorni trascorsi.

Tuttavia, se l'applicazione utilizza settimane ininterrotte, la settimana 1 può iniziare nell'anno precedente o nei primi giorni di gennaio. Dipende da come si utilizzano le variabili di sistema `ReferenceDay` e `FirstWeekDay`.

Esempio di funzione Weekname

| Data            | Nome settimana ISO | Nome settimana US |
|-----------------|--------------------|-------------------|
| Sab 26 dic 2020 | 2020/52            | 2020/52           |
| Dom 27 dic 2020 | 2020/52            | 2020/53           |
| Lun 28 dic 2020 | 2020/53            | 2020/53           |
| Mar 29 dic 2020 | 2020/53            | 2020/53           |
| Mer 30 dic 2020 | 2020/53            | 2020/53           |
| Gio 31 dic 2020 | 2020/53            | 2020/53           |
| Ven 1° gen 2021 | 2020/53            | 2021/01           |
| Sab 2 gen 2021  | 2020/53            | 2021/01           |
| Dom 3 gen 2021  | 2020/53            | 2021/02           |
| Lun 4 gen 2021  | 2021/01            | 2021/02           |
| Mar 5 gen 2021  | 2021/01            | 2021/02           |

### Casi di utilizzo

La funzione `weekname()` è utile per confrontare le aggregazioni per settimane.

Ad esempio, può essere usata se si desidera visualizzare le vendite totali dei prodotti in base alla settimana. Per mantenere la coerenza con la variabile di ambiente `BrokenWeeks` nell'applicazione, utilizzare `weekname()` anziché `1unarweekname()`. Se l'applicazione utilizza settimane ininterrotte, la settimana 1 può contenere date di dicembre dell'anno precedente o escludere date di gennaio dell'anno in corso. Se l'applicazione utilizza settimane interrotte, la settimana 1 può contenere meno di sette giorni.

**Tipo di dati restituiti:** duale

Argomenti

| Argomento        | Descrizione                          |
|------------------|--------------------------------------|
| <b>timestamp</b> | La data o la data e ora da valutare. |

| Argomento             | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>period_no</b>      | <b>shift</b> è un numero intero, in cui il valore 0 indica la settimana che contiene <b>date</b> . I valori negativi di <b>shift</b> indicano le settimane precedenti, mentre i valori positivi indicano le settimane successive.                                                                                                                                                                                                                                 |
| <b>first_week_day</b> | Specifica il giorno di inizio della settimana. Se omesso, viene utilizzato il valore della variabile <b>FirstWeekDay</b> .<br><br>I valori possibili per <b>first_week_day</b> sono 0 per lunedì, 1 per martedì, 2 per mercoledì, 3 per giovedì, 4 per venerdì, 5 per sabato e 6 per domenica.<br><br>Per maggiori informazioni sulle variabili di sistema, vedere <a href="#">FirstWeekDay (page 242)</a> .                                                      |
| <b>broken_weeks</b>   | Se non si specifica <b>broken_weeks</b> , il valore della variabile <b>BrokenWeeks</b> verrà utilizzato per definire se le settimane sono interrotte o meno.                                                                                                                                                                                                                                                                                                      |
| <b>reference_day</b>  | Se non si specifica <b>reference_day</b> , il valore della variabile <b>ReferenceDay</b> verrà utilizzato per definire quale giorno di gennaio impostare come giorno di riferimento per definire la settimana 1. Per impostazione predefinita, le funzioni Qlik Sense utilizzano 4 come giorno di riferimento. Questo significa che la settimana 1 deve contenere il 4 gennaio, vale a dire che la settimana 1 deve sempre contenere 4 quattro giorni di gennaio. |

## Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

Gli esempi riportati di seguito presuppongono:

```
Set FirstWeekDay=0;
Set BrokenWeeks=0;
Set ReferenceDay=4;
```

Esempi di funzioni

| Esempio                                   | Risultato            |
|-------------------------------------------|----------------------|
| <code>weekname('01/12/2013')</code>       | Restituisce 2013/02. |
| <code>weekname('01/12/2013', -1)</code>   | Restituisce 2013/01. |
| <code>weekname('01/12/2013', 0, 1)</code> | Restituisce 2013/02. |

### Esempio 1 - Nessun argomento aggiuntivo

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per l'ultima settimana del 2021 e le prime due settimane del 2022 caricato in una tabella denominata 'Transactions'.
- La variabile di sistema `DateFormat` che è impostata sul formato `MM/DD/YYYY`.
- La variabile di sistema `BrokenWeeks` che è impostata su 1.
- La variabile di sistema `FirstWeekDay` che è impostata su 6.
- Un'istruzione di caricamento precedente che contiene i seguenti elementi:
  - La funzione `weekday()` che è impostata come campo, `'week_number'`, che restituisce il numero dell'anno e della settimana in cui sono state effettuate le transazioni.
  - La funzione `weekname()` che è impostata come campo denominato `'week_day'`, per mostrare il valore del giorno della settimana di ogni data di transazione.

#### Script di caricamento

```
SET BrokenWeeks=1;
SET DateFormat='MM/DD/YYYY';
SET FirstWeekDay=6;

Transactions:
    Load
        *,
        weekday(date) as week_day,
        weekname(date) as week_number
    ;
Load
*
Inline
[
id,date,amount
8183,12/27/2021,58.27
8184,12/28/2021,67.42
8185,12/29/2021,23.80
8186,12/30/2021,82.06
8187,12/31/2021,40.56
8188,01/01/2022,37.23
8189,01/02/2022,17.17
8190,01/03/2022,88.27
8191,01/04/2022,57.42
8192,01/05/2022,53.80
8193,01/06/2022,82.06
8194,01/07/2022,40.56
8195,01/08/2022,53.67
```

```
8196,01/09/2022,26.63
8197,01/10/2022,72.48
8198,01/11/2022,18.37
8199,01/12/2022,45.26
8200,01/13/2022,58.23
8201,01/14/2022,18.52
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- date
- week\_day
- week\_number

Tabella dei risultati

| id   | date       | week_day | numero_settimana |
|------|------------|----------|------------------|
| 8183 | 12/27/2021 | Mon      | 2021/53          |
| 8184 | 28/12/2021 | Tue      | 2021/53          |
| 8185 | 29/12/2021 | Wed      | 2021/53          |
| 8186 | 30/12/2021 | Thu      | 2021/53          |
| 8187 | 12/31/2021 | Ven      | 2021/53          |
| 8188 | 01/01/2022 | Sab      | 2022/01          |
| 8189 | 01/02/2022 | Sun      | 2022/02          |
| 8190 | 01/03/2022 | Mon      | 2022/02          |
| 8191 | 01/04/2022 | Tue      | 2022/02          |
| 8192 | 01/05/2022 | Wed      | 2022/02          |
| 8193 | 01/06/2022 | Thu      | 2022/02          |
| 8194 | 01/07/2022 | Ven      | 2022/02          |
| 8195 | 08/01/2022 | Sat      | 2022/02          |
| 8196 | 09/01/2022 | Sun      | 2022/03          |
| 8197 | 10/01/2022 | Mon      | 2022/03          |
| 8198 | 01/11/2022 | Tue      | 2022/03          |
| 8199 | 12/01/2022 | Wed      | 2022/03          |
| 8200 | 01/13/2022 | Thu      | 2022/03          |
| 8201 | 01/14/2022 | Ven      | 2022/03          |

Il campo 'week\_number' viene creato nell'istruzione di caricamento precedente mediante l'uso della funzione `weekname()` e trasferendo il campo `data` come argomento della funzione.

La funzione `weekname()` identifica inizialmente la settimana in cui rientra il valore della data e restituisce il conteggio del numero della settimana e l'anno in cui si verifica la transazione.

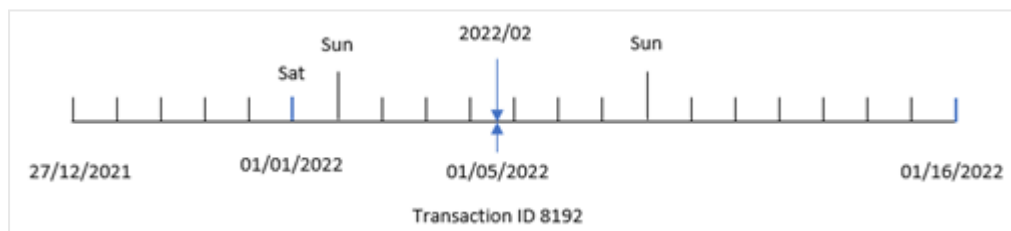
La variabile di sistema `imposta` la domenica come primo giorno della settimana. `FirstWeekDay` La variabile di sistema `Brokenweeks` imposta l'applicazione in modo che utilizzi le settimane interrotte, quindi la settimana 1 inizierà il 1 gennaio.

*Schema della funzione `weekname()` con le variabili predefinite.*



La settimana 1 inizia il 1 gennaio, che è un sabato, pertanto le transazioni che si verificano in questa data restituiscono il valore 2022/01 (l'anno e il numero della settimana).

*Diagramma della funzione `weekname()` che identifica il numero della settimana per la transazione 8192.*



Poiché l'applicazione utilizza settimane interrotte e il primo giorno della settimana è la domenica, le transazioni che si verificano dal 2 all'8 gennaio restituiscono il valore 2022/02 (settimana numero 2 nel 2022). Un esempio potrebbe essere la transazione 8192 avvenuta il 5 gennaio che restituisce il valore 2022/02 per il campo 'week\_number'.

### Esempio 2 - period\_no

Script di caricamento e risultati

#### Panoramica

Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, in questo esempio, l'attività consiste nel creare un campo, 'previous\_week\_number', che restituisce l'anno e il numero della settimana precedente alla data in cui sono state eseguite le transazioni.

Aprire Editor caricamento dati, quindi aggiungere il seguente script di caricamento in una nuova sezione.

### Script di caricamento

```
SET BrokenWeeks=1;
SET FirstWeekDay=6;

Transactions:
    Load
        *,
        weekname(date,-1) as previous_week_number
    ;
Load
*
Inline
[
id,date,amount
8183,12/27/2021,58.27
8184,12/28/2021,67.42
8185,12/29/2021,23.80
8186,12/30/2021,82.06
8187,12/31/2021,40.56
8188,01/01/2022,37.23
8189,01/02/2022,17.17
8190,01/03/2022,88.27
8191,01/04/2022,57.42
8192,01/05/2022,53.80
8193,01/06/2022,82.06
8194,01/07/2022,40.56
8195,01/08/2022,53.67
8196,01/09/2022,26.63
8197,01/10/2022,72.48
8198,01/11/2022,18.37
8199,01/12/2022,45.26
8200,01/13/2022,58.23
8201,01/14/2022,18.52
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- date
- week\_day
- week\_number

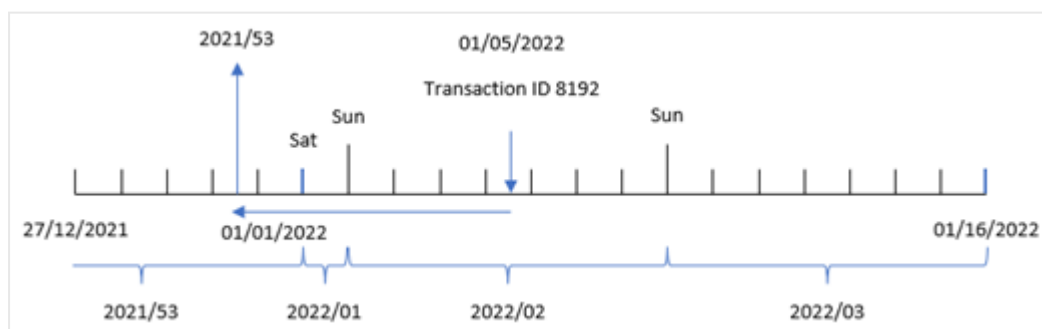
Tabella dei risultati

| id   | date       | week_day | numero_settimana |
|------|------------|----------|------------------|
| 8183 | 12/27/2021 | Mon      | 2021/52          |
| 8184 | 28/12/2021 | Tue      | 2021/52          |

| id   | date       | week_day | numero_settimana |
|------|------------|----------|------------------|
| 8185 | 29/12/2021 | Wed      | 2021/52          |
| 8186 | 30/12/2021 | Thu      | 2021/52          |
| 8187 | 12/31/2021 | Ven      | 2021/52          |
| 8188 | 01/01/2022 | Sat      | 2021/52          |
| 8189 | 01/02/2022 | Sun      | 2021/53          |
| 8190 | 01/03/2022 | Mon      | 2021/53          |
| 8191 | 01/04/2022 | Tue      | 2021/53          |
| 8192 | 01/05/2022 | Wed      | 2021/53          |
| 8193 | 01/06/2022 | Thu      | 2021/53          |
| 8194 | 01/07/2022 | Ven      | 2021/53          |
| 8195 | 08/01/2022 | Sat      | 2022/01          |
| 8196 | 09/01/2022 | Sun      | 2022/02          |
| 8197 | 10/01/2022 | Mon      | 2022/02          |
| 8198 | 01/11/2022 | Tue      | 2022/02          |
| 8199 | 12/01/2022 | Wed      | 2022/02          |
| 8200 | 01/13/2022 | Thu      | 2022/02          |
| 8201 | 01/14/2022 | Ven      | 2022/02          |

Poiché il valore `period_no` di `-1` è stato utilizzato come argomento `offset` nella funzione `weekname()`, la funzione per prima cosa identifica la settimana in cui avvengono le transazioni. Quindi, cerca la settimana precedente e identifica il primo millisecondo di quella settimana.

*Schema della funzione `weekname()` con un offset `period_no` di `-1`.*



La transazione 8192 è avvenuta il 5 gennaio 2022. La funzione `weekname()` cerca una settimana prima, il 30 dicembre 2021, e restituisce il numero della settimana e l'anno per quella data - 2021/53.

### Esempio 3 - first\_week\_day

Script di caricamento e risultati

#### Panoramica

Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, in questo esempio, la politica aziendale prevede che la settimana lavorativa inizi il martedì.

Aprire Editor caricamento dati, quindi aggiungere il seguente script di caricamento in una nuova sezione.

#### Script di caricamento

```
SET BrokenWeeks=1;
SET DateFormat='MM/DD/YYYY';

Transactions:
    Load
        *,
        weekday(date) as week_day,
        weekname(date,0,1) as week_number
    ;
Load
*
Inline
[
id,date,amount
8183,12/27/2021,58.27
8184,12/28/2021,67.42
8185,12/29/2021,23.80
8186,12/30/2021,82.06
8187,12/31/2021,40.56
8188,01/01/2022,37.23
8189,01/02/2022,17.17
8190,01/03/2022,88.27
8191,01/04/2022,57.42
8192,01/05/2022,53.80
8193,01/06/2022,82.06
8194,01/07/2022,40.56
8195,01/08/2022,53.67
8196,01/09/2022,26.63
8197,01/10/2022,72.48
8198,01/11/2022,18.37
8199,01/12/2022,45.26
8200,01/13/2022,58.23
8201,01/14/2022,18.52
];
```

#### Risultati

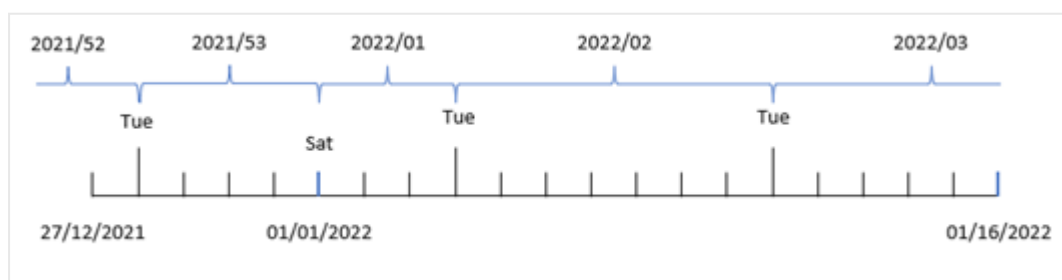
Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- date
- week\_day
- week\_number

Tabella dei risultati

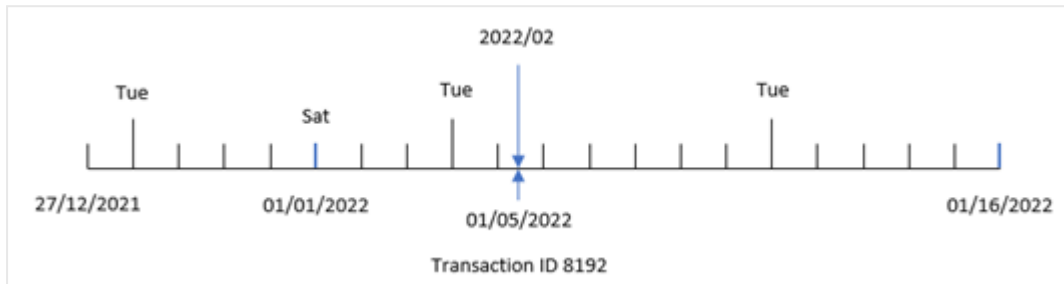
| id   | date       | week_day | numero_settimana |
|------|------------|----------|------------------|
| 8183 | 12/27/2021 | Mon      | 2021/52          |
| 8184 | 28/12/2021 | Tue      | 2021/53          |
| 8185 | 29/12/2021 | Wed      | 2021/53          |
| 8186 | 30/12/2021 | Thu      | 2021/53          |
| 8187 | 12/31/2021 | Ven      | 2021/53          |
| 8188 | 01/01/2022 | Sab      | 2022/01          |
| 8189 | 01/02/2022 | Sun      | 2022/01          |
| 8190 | 01/03/2022 | Mon      | 2022/01          |
| 8191 | 01/04/2022 | Tue      | 2022/02          |
| 8192 | 01/05/2022 | Wed      | 2022/02          |
| 8193 | 01/06/2022 | Thu      | 2022/02          |
| 8194 | 01/07/2022 | Ven      | 2022/02          |
| 8195 | 08/01/2022 | Sat      | 2022/02          |
| 8196 | 09/01/2022 | Sun      | 2022/02          |
| 8197 | 10/01/2022 | Mon      | 2022/02          |
| 8198 | 01/11/2022 | Tue      | 2022/03          |
| 8199 | 12/01/2022 | Wed      | 2022/03          |
| 8200 | 01/13/2022 | Thu      | 2022/03          |
| 8201 | 01/14/2022 | Ven      | 2022/03          |

Schema della funzione `weekname()` con martedì come primo giorno della settimana.



Poiché l'argomento `first_week_date` di 1 viene utilizzato nella funzione `weekname()`, il martedì viene utilizzato come primo giorno della settimana. La funzione determina quindi che la settimana 53 del 2021 è iniziata martedì 28 dicembre; e, a causa dell'utilizzo da parte dell'applicazione di settimane interrotte, la settimana 1 inizia il 1 gennaio 2022 e termina l'ultimo millisecondo di lunedì 3 gennaio 2022.

*Diagramma che mostra il numero della settimana della transazione 8192 con martedì come primo giorno della settimana.*



La transazione 8192 è avvenuta il 5 gennaio 2022. Pertanto, utilizzando il parametro `first_week_day` di martedì, la funzione `weekname()` restituisce il valore 2022/02 per il campo 'week\_number'.

### Esempio 4 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, in questo esempio, il set di dati è invariato e viene caricato nell'applicazione. Il calcolo che restituisce il numero dell'anno per la fine settimana in cui sono avvenute le transazioni viene creato come misura in un oggetto grafico dell'applicazione.

#### Script di caricamento

```
SET BrokenWeeks=1;
Transactions:
Load
*
Inline
[
id,date,amount
8183,12/27/2021,58.27
8184,12/28/2021,67.42
8185,12/29/2021,23.80
8186,12/30/2021,82.06
8187,12/31/2021,40.56
8188,01/01/2022,37.23
8189,01/02/2022,17.17
8190,01/03/2022,88.27
8191,01/04/2022,57.42
8192,01/05/2022,53.80
8193,01/06/2022,82.06
```

```
8194,01/07/2022,40.56
8195,01/08/2022,53.67
8196,01/09/2022,26.63
8197,01/10/2022,72.48
8198,01/11/2022,18.37
8199,01/12/2022,45.26
8200,01/13/2022,58.23
8201,01/14/2022,18.52
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- date
- =weekday (date)

Per calcolare l'inizio della settimana in cui è avvenuta una transazione, creare la seguente misura:

=weekname(date)

Tabella dei risultati

| id   | date       | =weekday(date) | =weekname(date) |
|------|------------|----------------|-----------------|
| 8183 | 12/27/2021 | Mon            | 2021/53         |
| 8184 | 28/12/2021 | Tue            | 2021/53         |
| 8185 | 29/12/2021 | Wed            | 2021/53         |
| 8186 | 30/12/2021 | Thu            | 2021/53         |
| 8187 | 12/31/2021 | Ven            | 2021/53         |
| 8188 | 01/01/2022 | Sab            | 2022/01         |
| 8189 | 01/02/2022 | Sun            | 2022/02         |
| 8190 | 01/03/2022 | Mon            | 2022/02         |
| 8191 | 01/04/2022 | Tue            | 2022/02         |
| 8192 | 01/05/2022 | Wed            | 2022/02         |
| 8193 | 01/06/2022 | Thu            | 2022/02         |
| 8194 | 01/07/2022 | Ven            | 2022/02         |
| 8195 | 08/01/2022 | Sat            | 2022/02         |
| 8196 | 09/01/2022 | Sun            | 2022/03         |
| 8197 | 10/01/2022 | Mon            | 2022/03         |
| 8198 | 01/11/2022 | Tue            | 2022/03         |

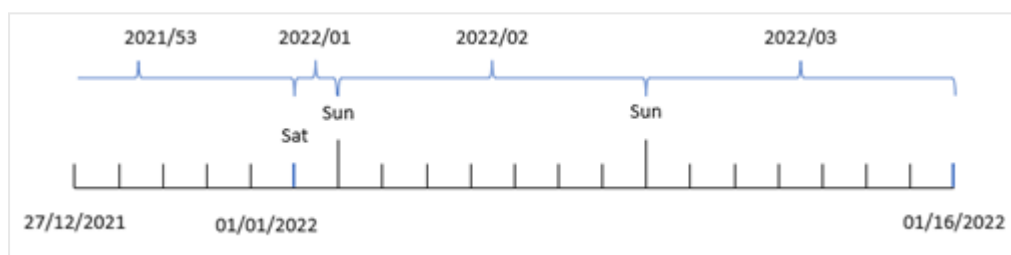
| id   | date       | =weekday(date) | =weekname(date) |
|------|------------|----------------|-----------------|
| 8199 | 12/01/2022 | Wed            | 2022/03         |
| 8200 | 01/13/2022 | Thu            | 2022/03         |
| 8201 | 01/14/2022 | Ven            | 2022/03         |

Il campo `week_number` viene creato come misura nell'oggetto grafico mediante l'utilizzo della funzione `weekname()` e trasferendo il campo `date` come argomento della funzione.

La funzione `weekname()` identifica inizialmente la settimana in cui rientra il valore della data e restituisce il conteggio del numero della settimana e l'anno in cui si è verificata la transazione.

La variabile di sistema `imposta la domenica come primo giorno della settimana`. `FirstWeekDay` La variabile di sistema `Brokenweeks` imposta l'applicazione in modo che utilizzi le settimane interrotte, quindi la settimana 1 inizia il 1 gennaio.

*Diagramma che mostra il numero della settimana con domenica come primo giorno della settimana.*



*Diagramma che mostra che la transazione 8192 ha avuto luogo nella settimana numero due.*



Poiché l'applicazione utilizza settimane interrotte e il primo giorno della settimana è la domenica, le transazioni che si verificano dal 2 all'8 gennaio restituiscono il valore 2022/02, ossia la settimana numero 2 del 2022. Si noti che la transazione 8192 è avvenuta il 5 gennaio e restituisce il valore 2022/02 per il campo `'week_number'`.

### Esempio 5 - Scenario

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- un set di dati contenente un insieme di transazioni per l'ultima settimana del 2019 e le prime due settimane del 2020 caricato in una tabella denominata 'Transactions'.
- La variabile di sistema BrokenWeeks che è impostata su 0.
- La variabile di sistema ReferenceDay che è impostata su 2.
- La variabile di sistema DateFormat che è impostata sul formato MM/DD/YYYY.

### Script di caricamento

```
SET BrokenWeeks=0;  
SET ReferenceDay=2;  
SET DateFormat='MM/DD/YYYY';
```

Transactions:

Load

\*

Inline

[

id,date,amount

8183,12/27/2019,58.27

8184,12/28/2019,67.42

8185,12/29/2019,23.80

8186,12/30/2019,82.06

8187,12/31/2019,40.56

8188,01/01/2020,37.23

8189,01/02/2020,17.17

8190,01/03/2020,88.27

8191,01/04/2020,57.42

8192,01/05/2020,53.80

8193,01/06/2020,82.06

8194,01/07/2020,40.56

8195,01/08/2020,53.67

8196,01/09/2020,26.63

8197,01/10/2020,72.48

8198,01/11/2020,18.37

8199,01/12/2020,45.26

8200,01/13/2020,58.23

8201,01/14/2020,18.52

];

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella.

Creare una dimensione calcolata utilizzando la seguente espressione:

```
=weekname(date)
```

Per calcolare le vendite totali, creare la seguente misura di aggregazione:

```
=sum(amount)
```

Impostare la **Formattazione numero** della misura su **Denaro**.

Tabella dei risultati

| NomeSettimana(data) | sum(amount) |
|---------------------|-------------|
| 2019/52             | \$125.69    |
| 2020/01             | \$346.51    |
| 2020/02             | \$347.57    |
| 2020/03             | \$122.01    |

Per dimostrare i risultati dell'utilizzo della funzione weekname() in questo scenario, aggiungere il campo seguente come dimensione:

date

Tabella dei risultati con campo data

| weekname(date) | date       | sum(amount) |
|----------------|------------|-------------|
| 2019/52        | 12/27/2019 | \$58.27     |
| 2019/52        | 12/28/2019 | \$67.42     |
| 2020/01        | 12/29/2019 | \$23.80     |
| 2020/01        | 12/30/2019 | \$82.06     |
| 2020/01        | 12/31/2019 | \$40.56     |
| 2020/01        | 01/01/2020 | \$37.23     |
| 2020/01        | 01/02/2020 | \$17.17     |
| 2020/01        | 01/03/2020 | \$88.27     |
| 2020/01        | 01/04/2020 | \$57.42     |
| 2020/02        | 01/05/2020 | \$53.80     |
| 2020/02        | 01/06/2020 | \$82.06     |
| 2020/02        | 01/07/2020 | \$40.56     |
| 2020/02        | 01/08/2020 | \$53.67     |
| 2020/02        | 01/09/2020 | \$26.63     |
| 2020/02        | 01/10/2020 | \$72.48     |
| 2020/02        | 01/11/2020 | \$18.37     |
| 2020/03        | 01/12/2020 | \$45.26     |
| 2020/03        | 01/13/2020 | \$58.23     |
| 2020/03        | 01/14/2020 | \$18.52     |

Poiché l'applicazione utilizza settimane ininterrotte e la settimana 1 richiede un minimo di due giorni a gennaio a causa della variabile di sistema ReferenceDay, la settimana 1 del 2020 include le transazioni del 29 dicembre 2019.

## weekstart

Questa funzione restituisce un valore corrispondente a un indicatore temporale recante il primo millisecondo del primo giorno della settimana di calendario contenente **date**. Il formato di output predefinito è il formato **DateFormat** impostato nello script.

### Sintassi:

```
WeekStart(timestamp [, period_no [, first_week_day ]])
```

### Tipo di dati restituiti: duale

In altre parole, la funzione `weekstart()` determina in quale settimana cade la data. Quindi, restituisce un timestamp, nel formato data, per il primo millisecondo di quella settimana. Il primo giorno della settimana è determinato dalla variabile ambientale `FirstWeekDay`. Tuttavia, questo può essere sostituito dall'argomento `first_week_day` della funzione `weekstart()`.

### Argomenti

| Argomento             | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>timestamp</b>      | La data o la data e ora da valutare.                                                                                                                                                                                                                                                                                                                                                                          |
| <b>period_no</b>      | <b>shift</b> è un numero intero, in cui il valore 0 indica la settimana che contiene <b>date</b> . I valori negativi di <b>shift</b> indicano le settimane precedenti, mentre i valori positivi indicano le settimane successive.                                                                                                                                                                             |
| <b>first_week_day</b> | Specifica il giorno di inizio della settimana. Se omissso, viene utilizzato il valore della variabile <b>FirstWeekDay</b> .<br><br>I valori possibili per <b>first_week_day</b> sono 0 per lunedì, 1 per martedì, 2 per mercoledì, 3 per giovedì, 4 per venerdì, 5 per sabato e 6 per domenica.<br><br>Per maggiori informazioni sulle variabili di sistema, vedere <a href="#">FirstWeekDay (page 242)</a> . |

## Casi di utilizzo

La funzione `weekstart()` viene comunemente utilizzata come parte di un'espressione quando l'utente desidera che il calcolo utilizzi la frazione della settimana trascorsa finora. Ad esempio, potrebbe essere utilizzato se un utente desidera calcolare il totale dei salari guadagnati dai dipendenti nella settimana in corso.

Gli esempi seguenti presuppongono:

```
SET FirstWeekDay=0;
```

### Esempi di funzioni

| Esempio                                    | Risultato               |
|--------------------------------------------|-------------------------|
| <code>weekstart('01/12/2013')</code>       | Restituisce 01/07/2013. |
| <code>weekstart('01/12/2013', -1 )</code>  | Restituisce 11/31/2012. |
| <code>weekstart('01/12/2013', 0, 1)</code> | Restituisce 01/08/2013. |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempi:

Se si desiderano le impostazioni ISO per le settimane e i numeri di settimana, assicurarsi di inserire nello script quanto segue:

```
Set DateFormat = 'YYYY-MM-DD';
Set FirstWeekDay = 0; // Monday as first week day
Set BrokenWeeks = 0; // (use unbroken weeks)
Set ReferenceDay = 4; // Jan 4th is always in week 1
```

Se si desiderano le impostazioni USA, assicurarsi che nello script sia presente quanto segue:

```
Set DateFormat = 'M/D/YYYY';
Set FirstWeekDay = 6; // Sunday as first week day
Set BrokenWeeks = 1; // (use broken weeks)
Set ReferenceDay = 1; // Jan 1st is always in week 1
```

Gli esempi sopra riportati danno come risultato quanto segue dalla funzione `weekstart()`:

Esempio di funzione Weekstart

| Data            | Inizio settimana ISO | Inizio settimana US |
|-----------------|----------------------|---------------------|
| Sab 26 dic 2020 | 2020-12-21           | 12/20/2020          |
| Dom 27 dic 2020 | 2020-12-21           | 27/12/2020          |
| Lun 28 dic 2020 | 2020-12-28           | 27/12/2020          |
| Mar 29 dic 2020 | 2020-12-28           | 27/12/2020          |
| Mer 30 dic 2020 | 28/12/2020           | 27/12/2020          |
| Gio 31 dic 2020 | 28-12-2020           | 27/12/2020          |
| Ven 1° gen 2021 | 28-12-2020           | 27/12/2020          |
| Sab 2 gen 2021  | 28/12/2020           | 27/12/2020          |
| Dom 3 gen 2021  | 28/12/2020           | 03/01/2021          |

| Data           | Inizio settimana ISO | Inizio settimana US |
|----------------|----------------------|---------------------|
| Lun 4 gen 2021 | 2021-01-04           | 03/01/2021          |
| Mar 5 gen 2021 | 04/01/2021           | 03/01/2021          |



*Le settimane iniziano il lunedì nella colonna ISO e la domenica nella colonna USA.*

### Esempio 1 - Nessun argomento aggiuntivo

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per il 2022, caricato in una tabella denominata 'Transactions'.
- Il campo della data fornito nel formato della variabile di sistema `DateFormat` (MM/GG/AAAA).
- La creazione di un campo, `start_of_week`, che restituisce un timestamp per l'inizio della settimana in cui sono avvenute le transazioni.

#### Script di caricamento

```
SET FirstWeekDay=6;
```

```
Transactions:
```

```
    Load
        *,
        weekstart(date) as start_of_week,
        timestamp(weekstart(date)) as start_of_week_timestamp
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
```

```
8189,1/19/2022,37.23
```

```
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
```

```
8192,3/16/2022,53.80
```

```
8193,4/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/16/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- start\_of\_week
- start\_of\_week\_timestamp

Tabella dei risultati

| date      | inizio_settimana | inizio_settimana_timestamp |
|-----------|------------------|----------------------------|
| 1/7/2022  | 01/02/2022       | 02/01/2022 00:00:00        |
| 1/19/2022 | 16/01/2022       | 16/01/2022 00:00:00        |
| 2/5/2022  | 30/01/2022       | 30/01/2022 12:00:00 AM     |
| 2/28/2022 | 27/02/2022       | 27/02/2022 00:00:00        |
| 3/16/2022 | 03/13/2022       | 13/03/2022 00:00:00        |
| 4/1/2022  | 27/03/2022       | 27/03/2022 00:00:00        |
| 5/7/2022  | 01/05/2022       | 01/05/2022 00:00:00        |
| 5/16/2022 | 15/05/2022       | 15/05/2022 00:00:00        |
| 6/15/2022 | 12/06/2022       | 12/06/2022 00:00:00        |
| 6/26/2022 | 26/06/2022       | 26/06/2022 00:00:00        |
| 7/9/2022  | 03/07/2022       | 03/07/2022 00:00:00        |
| 7/22/2022 | 17/07/2022       | 17/07/2022 00:00:00        |
| 7/23/2022 | 17/07/2022       | 17/07/2022 00:00:00        |
| 7/27/2022 | 24/07/2022       | 24/07/2022 00:00:00        |
| 8/2/2022  | 07/31/2022       | 31/07/2022 00:00:00        |
| 8/8/2022  | 07/08/2022       | 7/8/2022 12:00:00 AM       |
| 8/19/2022 | 14/08/2022       | 14/08/2022 00:00:00        |
| 9/26/2022 | 25/09/2022       | 25/09/2022 00:00:00        |

| date       | inizio_settimana | inizio_settimana_timestamp |
|------------|------------------|----------------------------|
| 10/14/2022 | 09/10/2022       | 09/10/2022 12:00:00 AM     |
| 10/29/2022 | 23/10/2022       | 23/10/2022 00:00:00        |

Il campo 'start\_of\_week' viene creato nell'istruzione di caricamento precedente mediante l'uso della funzione `weekstart()` e trasferendo il campo data come argomento della funzione.

La funzione `weekstart()` identifica inizialmente in quale settimana cade il valore della data, restituendo un timestamp per il primo millisecondo di quella settimana.

*Schema della funzione `weekstart()`, esempio senza argomenti aggiuntivi*



La transazione 8191 è avvenuta il 5 febbraio. La variabile di sistema `Firstweekday` imposta il primo giorno della settimana come domenica. La funzione `weekstart()` identifica che la prima domenica prima del 5 febbraio - e quindi l'inizio della settimana - è stata il 30 gennaio. Pertanto, il valore `start_of_week` per questa transazione restituisce il primo millisecondo di quel giorno, che è il 30 gennaio alle 12:00:00 AM.

### Esempio 2 - period\_no

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.
- La creazione di un campo, `previous_week_start`, che restituisce il timestamp per l'inizio del trimestre prima che fosse effettuata la transazione.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        weekstart(date,-1) as previous_week_start,
        timestamp(weekstart(date,-1)) as previous_week_start_timestamp
    ;
Load
    *
```

```
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- previous\_week\_start
- previous\_week\_start\_timestamp

Tabella dei risultati

| date      | inizio settimana precedente | timestamp_inizio_settimana_precedente |
|-----------|-----------------------------|---------------------------------------|
| 1/7/2022  | 12/26/2021                  | 12/26/2021 12:00:00 AM                |
| 1/19/2022 | 01/09/2022                  | 09/01/2022 00:00:00                   |
| 2/5/2022  | 01/23/2022                  | 23/1/2022 00:00:00                    |
| 2/28/2022 | 02/20/2022                  | 20/2/2022 00:00:00                    |
| 3/16/2022 | 03/06/2022                  | 6/3/2022 00:00:00                     |
| 4/1/2022  | 03/20/2022                  | 20/03/2022 00:00:00                   |
| 5/7/2022  | 04/24/2022                  | 24/04/2022 00:00:00                   |
| 5/16/2022 | 05/08/2022                  | 08/05/2022 00:00:00                   |
| 6/15/2022 | 06/05/2022                  | 5/6/2022 00:00:00                     |

| date       | inizio settimana precedente | timestamp_inizio_settimana_precedente |
|------------|-----------------------------|---------------------------------------|
| 6/26/2022  | 06/19/2022                  | 19/6/2022 00:00:00                    |
| 7/9/2022   | 26/06/2022                  | 26/06/2022                            |
| 7/22/2022  | 10/07/2022                  | 10/07/2022                            |
| 7/23/2022  | 10/07/2022                  | 10/07/2022 00:00:00                   |
| 7/27/2022  | 17/07/2022                  | 17/07/2022 00:00:00                   |
| 8/2/2022   | 24/07/2022                  | 24/07/2022 00:00:00                   |
| 8/8/2022   | 07/31/2022                  | 31/07/2022 00:00:00                   |
| 8/19/2022  | 07/08/2022                  | 07/08/2022 12:00:00 AM                |
| 9/26/2022  | 09/18/2022                  | 18/09/2022 12:00:00 AM                |
| 10/14/2022 | 10/02/2022                  | 02/10/2022 00:00:00                   |
| 10/29/2022 | 10/16/2022                  | 16/10/2022 00:00:00                   |

In questo caso, poiché il valore `period_no` di -1 è stato utilizzato come argomento `offset` nella funzione `weekstart()`, la funzione per prima cosa identifica la settimana in cui avvengono le transazioni. Quindi, cerca la settimana precedente e identifica il primo millisecondo di quella settimana.

*Schema della funzione `weekstart()`, esempio di `period_no`*



La transazione 8196 è avvenuta il 15 giugno. La funzione `weekstart()` identifica che la settimana è iniziata il 12 giugno. Pertanto, la settimana precedente è iniziata il 5 giugno alle 12:00:00; questo è il valore che viene restituito per il campo `previous_week_start`.

### Esempio 3 - `first_week_day`

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor di caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento contiene lo stesso set di dati e lo stesso scenario del primo esempio. Tuttavia, in questo esempio, dobbiamo impostare il martedì come primo giorno della settimana lavorativa.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        weekstart(date,0,1) as start_of_week,
        timestamp(weekstart(date,0,1)) as start_of_week_timestamp
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
```

```
8189,1/19/2022,37.23
```

```
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
```

```
8192,3/16/2022,53.80
```

```
8193,4/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/16/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
```

```
8200,7/23/2022,82.77
```

```
8201,7/27/2022,69.98
```

```
8202,8/2/2022,76.11
```

```
8203,8/8/2022,25.12
```

```
8204,8/19/2022,46.23
```

```
8205,9/26/2022,84.21
```

```
8206,10/14/2022,96.24
```

```
8207,10/29/2022,67.67
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- start\_of\_week
- start\_of\_week\_timestamp

Tabella dei risultati

| date      | inizio_settimana | inizio_settimana_timestamp |
|-----------|------------------|----------------------------|
| 1/7/2022  | 01/04/2022       | 04/01/2022 00:00:00        |
| 1/19/2022 | 01/18/2022       | 18/01/2022 00:00:00        |

| date       | inizio_settimana | inizio_settimana_timestamp |
|------------|------------------|----------------------------|
| 2/5/2022   | 01/02/2022       | 01/02/2022 00:00:00        |
| 2/28/2022  | 02/22/2022       | 22/02/2022 00:00:00        |
| 3/16/2022  | 03/15/2022       | 15/03/2022 00:00:00        |
| 4/1/2022   | 03/29/2022       | 29/3/2022 00:00:00         |
| 5/7/2022   | 05/03/2022       | 3/5/2022 00:00:00          |
| 5/16/2022  | 05/10/2022       | 10/5/2022 00:00:00         |
| 6/15/2022  | 06/14/2022       | 14/06/2022 00:00:00        |
| 6/26/2022  | 06/21/2022       | 21/06/2022 00:00:00        |
| 7/9/2022   | 07/05/2022       | 05/07/2022 00:00:00        |
| 7/22/2022  | 19/07/2022       | 19/07/2022 00:00:00        |
| 7/23/2022  | 19/07/2022       | 19/07/2022 00:00:00        |
| 7/27/2022  | 07/26/2022       | 26/7/2022 00:00:00         |
| 8/2/2022   | 08/02/2022       | 2/8/2022 00:00:00          |
| 8/8/2022   | 08/02/2022       | 2/8/2022 00:00:00          |
| 8/19/2022  | 08/16/2022       | 16/08/2022 00:00:00        |
| 9/26/2022  | 09/20/2022       | 20/09/2022 00:00:00        |
| 10/14/2022 | 10/11/2022       | 11/10/2022 00:00:00        |
| 10/29/2022 | 10/25/2022       | 10/25/2022 12:00:00 AM     |

In questa istanza, dato che l'argomento `first_week_date` di 1 è utilizzato nella funzione `weekstart()`, imposta il primo giorno della settimana a martedì.

*Schema della funzione `weekstart()`, esempio `first_week_day`*



La transazione 8191 è avvenuta il 5 febbraio. La funzione `weekstart()` identifica che il primo martedì precedente a questa data - e quindi l'inizio della settimana e il valore restituito - è stato il 1° febbraio alle 12:00:00.

### Esempio 4 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento contiene lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, in questo esempio, il set di dati invariato viene caricato nell'applicazione. Il calcolo che restituisce un timestamp per l'inizio della settimana in cui sono avvenute le transazioni viene creato come misura in un oggetto grafico dell'applicazione.

#### Script di caricamento

Transactions:

```
Load
*
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: date.

Per calcolare l'inizio della settimana in cui avviene una transazione, sommare le seguenti misure:

- =weekstart(date)
- =timestamp(weekstart(date))

Tabella dei risultati

| date       | inizio_settimana | inizio_settimana_timestamp |
|------------|------------------|----------------------------|
| 1/7/2022   | 01/02/2022       | 02/01/2022 00:00:00        |
| 1/19/2022  | 16/01/2022       | 16/01/2022 00:00:00        |
| 2/5/2022   | 30/01/2022       | 30/01/2022 12:00:00 AM     |
| 2/28/2022  | 27/02/2022       | 27/02/2022 00:00:00        |
| 3/16/2022  | 03/13/2022       | 13/03/2022 00:00:00        |
| 4/1/2022   | 27/03/2022       | 27/03/2022 00:00:00        |
| 5/7/2022   | 01/05/2022       | 01/05/2022 00:00:00        |
| 5/16/2022  | 15/05/2022       | 15/05/2022 00:00:00        |
| 6/15/2022  | 12/06/2022       | 12/06/2022 00:00:00        |
| 6/26/2022  | 26/06/2022       | 26/06/2022 00:00:00        |
| 7/9/2022   | 03/07/2022       | 03/07/2022 00:00:00        |
| 7/22/2022  | 17/07/2022       | 17/07/2022 00:00:00        |
| 7/23/2022  | 17/07/2022       | 17/07/2022 00:00:00        |
| 7/27/2022  | 24/07/2022       | 24/07/2022 00:00:00        |
| 8/2/2022   | 07/31/2022       | 31/07/2022 00:00:00        |
| 8/8/2022   | 07/08/2022       | 7/8/2022 12:00:00 AM       |
| 8/19/2022  | 14/08/2022       | 14/08/2022 00:00:00        |
| 9/26/2022  | 25/09/2022       | 25/09/2022 00:00:00        |
| 10/14/2022 | 09/10/2022       | 09/10/2022 12:00:00 AM     |
| 10/29/2022 | 23/10/2022       | 23/10/2022 00:00:00        |

La misura start\_of\_week viene creata nell'oggetto grafico mediante l'utilizzo della funzione weekstart() e trasferendo il campo date come argomento della funzione.

La funzione weekstart() identifica inizialmente in quale settimana cade il valore della data, restituendo un timestamp per il primo millisecondo di quella settimana.

*Schema della funzione weekstart(), esempio di oggetto grafico*



La transazione 8191 è avvenuta il 5 febbraio. La variabile di sistema `FirstweekDay` imposta il primo giorno della settimana come domenica. La funzione `weekstart()` identifica che la prima domenica prima del 5 febbraio - e quindi l'inizio della settimana - era il 30 gennaio. Pertanto, il valore `start_of_week` per questa transazione restituisce il primo millisecondo di quel giorno, ovvero il 30 gennaio alle 12:00:00.

### Esempio 5 - Scenario

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella chiamata `Payroll`.
- Dati costituiti da ID dei dipendenti, nomi dei dipendenti e salario giornaliero percepito da ciascun dipendente.

I dipendenti iniziano a lavorare il lunedì e lavorano sei giorni alla settimana. La variabile di sistema `FirstweekDay` non deve essere modificata.

L'utente finale desidera un oggetto grafico che visualizzi, in base all'ID e al nome del dipendente, i salari percepiti nella settimana fino a quel momento.

#### Script di caricamento

```
Payroll:
Load
*
Inline
[
employee_id,employee_name,day_rate
182,Mark, $150
183,Deryck, $125
184,Dexter, $125
185,Sydney,$270
186,Agatha,$128
];
```

## Risultati

Procedere come indicato di seguito:

1. Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:
  - employee\_id
  - employee\_name
2. Quindi, creare una misura per calcolare i salari guadagnati nella settimana in corso:  
`=if(today(1)-weekstart(today(1),0,0)<7,(today(1)-weekstart(today(1),0,0))*day_rate,day_rate*6)`
3. Impostare la **Formattazione numero** della misura su **Denaro**.

Tabella dei risultati

| employee_id | employee_name | =if(today(1)-weekstart(today(1),0,0)<7,(today(1)-weekstart(today(1),0,0))*day_rate,day_rate*6) |
|-------------|---------------|------------------------------------------------------------------------------------------------|
| 182         | Contrassegno  | \$600.00                                                                                       |
| 183         | Deryck        | 500.00 USD                                                                                     |
| 184         | Dexter        | 500.00 \$                                                                                      |
| 185         | Sydney        | \$1080.00                                                                                      |
| 186         | Agatha        | \$512.00                                                                                       |

La funzione `weekstart()`, utilizzando la data di oggi come primo argomento e 0 come terzo argomento, imposta il lunedì come primo giorno della settimana e restituisce la data di inizio della settimana corrente. Sottraendo tale risultato dalla data corrente, l'espressione restituisce il numero di giorni trascorsi fino ad ora questa settimana.

La condizione valuta quindi se ci sono stati più di sei giorni nella settimana. In caso di risposta affermativa, il valore `day_rate` del dipendente viene moltiplicato per 6. Altrimenti, il valore `day_rate` viene moltiplicato per il numero di giorni che si sono verificati finora nella settimana.

## weekyear

Questa funzione restituisce l'anno a cui appartiene il numero della settimana in base alle variabili di ambiente. I numeri della settimana rientrano in un intervallo approssimativo compreso tra 1 e 52.

Sintassi:

```
weekyear (timestamp [, first_week_day [, broken_weeks [, reference_day]])
```

Tipo di dati restituiti: numero intero

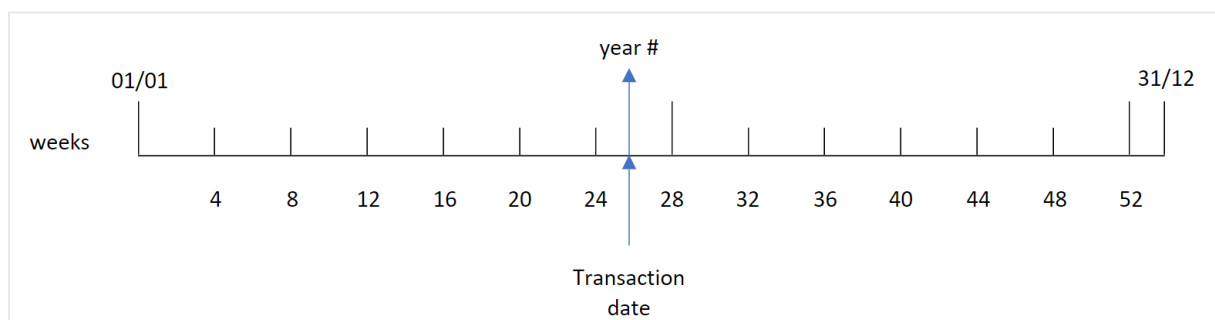
#### Argomenti

| Argomento             | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>timestamp</b>      | La data o la data e ora da valutare.                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>first_week_day</b> | <p>Specifica il giorno di inizio della settimana. Se omesso, viene utilizzato il valore della variabile <b>FirstWeekDay</b>.</p> <p>I valori possibili per <b>first_week_day</b> sono 0 per lunedì, 1 per martedì, 2 per mercoledì, 3 per giovedì, 4 per venerdì, 5 per sabato e 6 per domenica.</p> <p>Per maggiori informazioni sulle variabili di sistema, vedere <a href="#">FirstWeekDay (page 242)</a>.</p>                                                   |
| <b>broken_weeks</b>   | Se non si specificab <b>roken_weeks</b> , il valore della variabile <b>BrokenWeeks</b> verrà utilizzato per definire se le settimane sono interrotte o meno.                                                                                                                                                                                                                                                                                                        |
| <b>reference_day</b>  | Se non si specificare <b>reference_day</b> , il valore della variabile <b>ReferenceDay</b> verrà utilizzato per definire quale giorno di gennaio impostare come giorno di riferimento per definire la settimana 1. Per impostazione predefinita, le funzioni Qlik Sense utilizzano 4 come giorno di riferimento. Questo significa che la settimana 1 deve contenere il 4 gennaio, vale a dire che la settimana 1 deve sempre contenere 4 quattro giorni di gennaio. |

La funzione `weekyear()` determina in quale settimana dell'anno cade una data. Quindi restituisce l'anno corrispondente a quel numero di settimana.

Se `BrokenWeeks` è impostato su 0 (false), `weekyear()` restituirà lo stesso come `year()`.

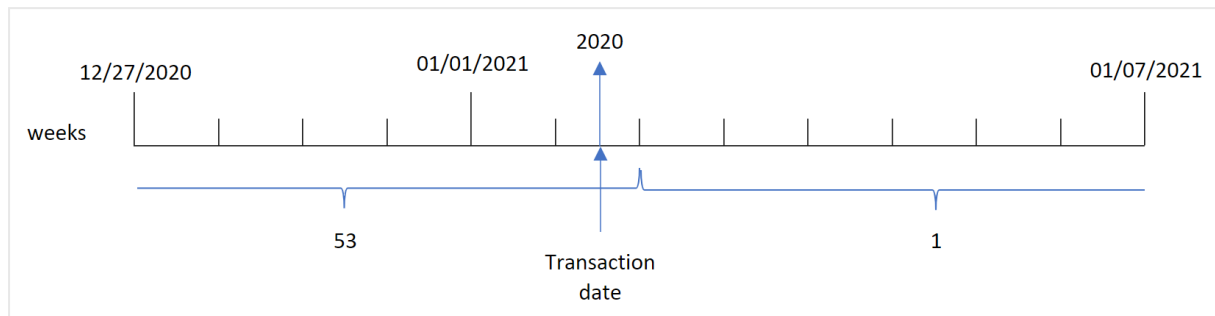
*Schema dell'intervallo della funzione `weekyear()`*



Tuttavia, se la variabile di sistema `brokenweeks` è impostata per l'utilizzo di settimane intere, la settimana 1 deve contenere solo un certo numero di giorni di gennaio, in base al valore specificato nella variabile di sistema `ReferenceDay`.

Ad esempio, se si utilizza un valore `referenceDay` pari a 4, la settimana 1 deve includere almeno quattro giorni di gennaio. È possibile che la settimana 1 includa date del dicembre dell'anno precedente o che il numero della settimana finale di un anno includa date del gennaio dell'anno successivo. In situazioni come questa, la funzione `weekyear()` restituirà un valore diverso per la funzione `year()`.

Schema dell'intervallo della funzione `weekyear()` quando si utilizzano le settimane intere



### Casi di utilizzo

La funzione `weekyear()` è utile quando si desidera confrontare le aggregazioni per anni. Ad esempio, può essere usata se si desidera visualizzare le vendite totali dei prodotti in base all'anno. La funzione `weekyear()` viene scelta rispetto a `year()` quando l'utente desidera mantenere la coerenza con la variabile di sistema `BrokenWeeks` nell'app.

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempi di funzioni

| Esempio                                      | Risultato                                                                  |
|----------------------------------------------|----------------------------------------------------------------------------|
| <code>weekyear('12/30/1996', 0, 0, 4)</code> | Restituisce 1997, poiché la settimana 1 del 1997 inizia il 30/12/1996      |
| <code>weekyear('01/02/1997', 0, 0, 4)</code> | Restituisce 1997                                                           |
| <code>weekyear('12/28/1997', 0, 0, 4)</code> | Restituisce 1997                                                           |
| <code>weekyear('12/30/1997', 0, 0, 4)</code> | Restituisce il 1998, perché la settimana 1 del 1998 inizia il 29/12/1997   |
| <code>weekyear('01/02/1999', 0, 0, 4)</code> | Restituisce il 1998, perché la settimana 53 del 1998 termina il 03/01/1999 |

### Argomenti correlati

| Argomento                        | Interazione                                                                                                                                          |
|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">week (page 1167)</a> | Restituisce un numero intero che rappresenta il numero della settimana in base allo standard ISO 8601                                                |
| <a href="#">year (page 1242)</a> | Restituisce un numero intero che rappresenta l'anno in cui l'espressione viene interpretata come data in base all'interpretazione numerica standard. |

### Esempio 1 - Settimane parziali

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per l'ultima settimana del 2020 e la prima settimana del 2021 che viene caricato in una tabella chiamata 'Transactions'.
- La variabile di sistema BrokenWeeks che è impostata su 1.
- Un'istruzione di caricamento precedente che contiene i seguenti elementi:
  - La funzione `weekyear()`, impostata come campo 'week\_year', che restituisce l'anno in cui sono avvenute le transazioni.
  - La funzione `week()`, impostata come campo 'week', che mostra il numero di settimana di ogni data di transazione.

#### Script di caricamento

```
SET BrokenWeeks=1;
```

```
Transactions:
```

```
    Load
    *,
    week(date) as week,
    weekyear(date) as week_year
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8176,12/28/2020,19.42
```

```
8177,12/29/2020,23.80
```

```
8178,12/30/2020,82.06
```

```
8179,12/31/2020,40.56
```

```
8180,01/01/2021,37.23
```

```
8181,01/02/2021,17.17
```

```
8182,01/03/2021,88.27
```

```
8183,01/04/2021,57.42
8184,01/05/2021,67.42
8185,01/06/2021,23.80
8186,01/07/2021,82.06
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- date
- week
- week\_year

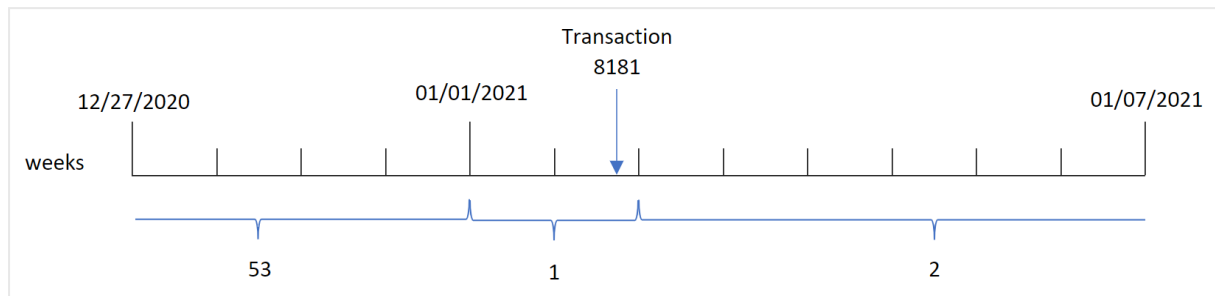
Tabella dei risultati

| id   | date       | settimana | settimana_anno |
|------|------------|-----------|----------------|
| 8176 | 28/12/2020 | 53        | 2020           |
| 8177 | 29/12/2020 | 53        | 2020           |
| 8178 | 30/12/2020 | 53        | 2020           |
| 8179 | 12/31/2020 | 53        | 2020           |
| 8180 | 01/01/2021 | 1         | 2021           |
| 8181 | 02/01/2021 | 1         | 2021           |
| 8182 | 03/01/2021 | 2         | 2021           |
| 8183 | 04/01/2021 | 2         | 2021           |
| 8184 | 05/01/2021 | 2         | 2021           |
| 8185 | 01/06/2021 | 2         | 2021           |
| 8186 | 01/07/2021 | 2         | 2021           |

Il campo 'week\_year' viene creato nell'istruzione di caricamento precedente mediante l'uso della funzione `weekyear()` e trasferendo il campo `date` come argomento della funzione.

La variabile di sistema `BrokenWeeks` è impostata su 1, il che significa che l'app utilizza settimane parziali. La settimana 1 inizia il 1° gennaio.

*Schema dell'intervallo della funzione weekyear() con l'uso di settimane parziali*



La transazione 8181 ha luogo il 2 gennaio, che fa parte della settimana 1. Pertanto, restituisce un valore di 2021 per il campo 'week\_year'.

### Esempio 2 - Settimane intere

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per l'ultima settimana del 2020 e la prima settimana del 2021 che viene caricato in una tabella chiamata 'transactions'.
- La variabile di sistema BrokenWeeks che è impostata su 0.
- Un'istruzione LOAD precedente che contiene i seguenti elementi:
  - La funzione weekyear(), impostata come campo 'week\_year', che restituisce l'anno in cui sono avvenute le transazioni.
  - La funzione week(), impostata come campo 'week', che mostra il numero di settimana di ogni data di transazione.

Tuttavia, in questo esempio, la politica aziendale prevede l'utilizzo di settimane intere.

#### Script di caricamento

```
SET BrokenWeeks=0;
```

```
Transactions:
```

```
    Load
    *,
    week(date) as week,
    weekyear(date) as week_year
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8176,12/28/2020,19.42
```

```
8177,12/29/2020,23.80
8178,12/30/2020,82.06
8179,12/31/2020,40.56
8180,01/01/2021,37.23
8181,01/02/2021,17.17
8182,01/03/2021,88.27
8183,01/04/2021,57.42
8184,01/05/2021,67.42
8185,01/06/2021,23.80
8186,01/07/2021,82.06
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- date
- week
- week\_year

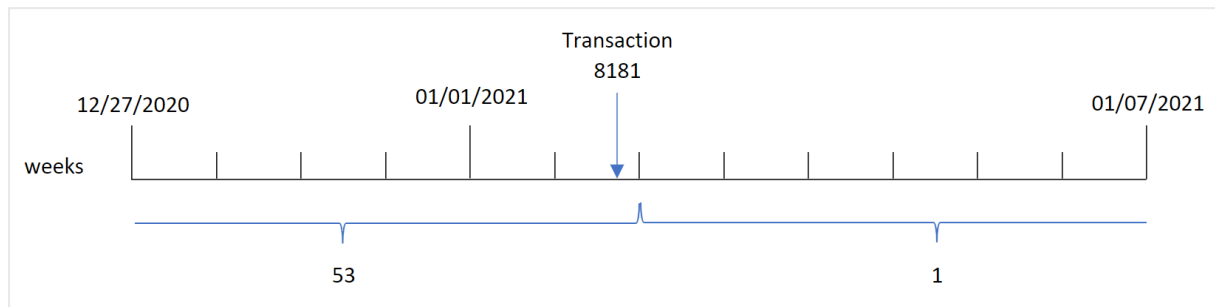
Tabella dei risultati

| id   | date       | settimana | settimana_anno |
|------|------------|-----------|----------------|
| 8176 | 28/12/2020 | 53        | 2020           |
| 8177 | 29/12/2020 | 53        | 2020           |
| 8178 | 30/12/2020 | 53        | 2020           |
| 8179 | 12/31/2020 | 53        | 2020           |
| 8180 | 01/01/2021 | 53        | 2020           |
| 8181 | 02/01/2021 | 53        | 2020           |
| 8182 | 03/01/2021 | 1         | 2021           |
| 8183 | 04/01/2021 | 1         | 2021           |
| 8184 | 05/01/2021 | 1         | 2021           |
| 8185 | 06/01/2021 | 1         | 2021           |
| 8186 | 07/01/2021 | 1         | 2021           |

La variabile di sistema `Brokenweeks` è impostata su 0, il che significa che l'applicazione utilizza settimane intere. Pertanto, non è necessario che la settimana 1 inizi il 1° gennaio.

La settimana 53 del 2020 continua fino alla fine del 2 gennaio 2021, mentre la settimana 1 del 2020 inizia domenica 3 gennaio 2021.

*Schema dell'intervallo della funzione `weekyear()` con l'uso di settimane intere*



La transazione 8181 ha luogo il 2 gennaio, che fa parte della settimana 1. Pertanto, restituisce un valore di 2021 per il campo 'week\_year'.

### Esempio 3 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, in questo esempio, il set di dati è invariato e viene caricato nell'applicazione. Il calcolo che restituisce il numero della settimana dell'anno in cui sono avvenute le transazioni viene creato come misura in un grafico dell'app.

#### Script di caricamento

```
SET BrokenWeeks=1;
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8176,12/28/2020,19.42
```

```
8177,12/29/2020,23.80
```

```
8178,12/30/2020,82.06
```

```
8179,12/31/2020,40.56
```

```
8180,01/01/2021,37.23
```

```
8181,01/02/2021,17.17
```

```
8182,01/03/2021,88.27
```

```
8183,01/04/2021,57.42
```

```
8184,01/05/2021,67.42
```

```
8185,01/06/2021,23.80
```

```
8186,01/07/2021,82.06
```

```
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- date

Per calcolare la settimana in cui avviene una transazione, creare la seguente misura:

- =week(date)

Per calcolare l'anno in cui avviene una transazione in base al numero della settimana, creare la seguente misura:

- =weekyear(date)

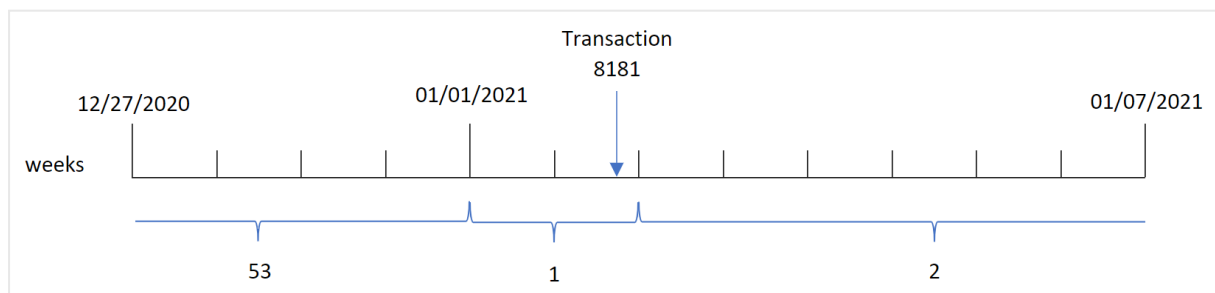
Tabella dei risultati

| id   | date       | settimana | settimana_anno |
|------|------------|-----------|----------------|
| 8176 | 28/12/2020 | 53        | 2020           |
| 8177 | 29/12/2020 | 53        | 2020           |
| 8178 | 30/12/2020 | 53        | 2020           |
| 8179 | 12/31/2020 | 53        | 2020           |
| 8180 | 01/01/2021 | 1         | 2021           |
| 8181 | 02/01/2021 | 1         | 2021           |
| 8182 | 03/01/2021 | 2         | 2021           |
| 8183 | 04/01/2021 | 2         | 2021           |
| 8184 | 05/01/2021 | 2         | 2021           |
| 8185 | 01/06/2021 | 2         | 2021           |
| 8186 | 01/07/2021 | 2         | 2021           |

Il campo 'week\_year' viene creato nell'istruzione di caricamento precedente mediante l'uso della funzione weekyear() e trasferendo il campo data come argomento della funzione.

La variabile di sistema Brokenweeks è impostata su 1, il che significa che l'applicazione utilizza settimane parziali. La settimana 1 inizia il 1° gennaio.

*Schema dell'intervallo della funzione weekyear() con l'uso di settimane parziali*



La transazione 8181 ha luogo il 2 gennaio, che fa parte della settimana 1. Pertanto, restituisce un valore di 2021 per il campo 'week\_year'.

### Esempio 4 - Scenario

#### Script di caricamento e risultati

##### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni per l'ultima settimana del 2020 e la prima settimana del 2021 che viene caricato in una tabella chiamata 'Transactions'.
- La variabile di sistema BrokenWeeks che è impostata su 0. Ciò significa che l'app utilizzerà settimane intere.
- La variabile di sistema ReferenceDay che è impostata su 2. Ciò significa che l'anno inizierà il 2 gennaio e conterrà un minimo di due giorni a gennaio.
- La variabile di sistema FirstWeekDay che è impostata su 1. Ciò significa che il primo giorno della settimana sarà il martedì.

La politica aziendale prevede l'utilizzo di settimane parziali. L'utente finale desidera un grafico che presenti le vendite totali per anno. L'app utilizza settimane intere, con la settimana 1 che contiene un minimo di due giorni a gennaio.

#### Script di caricamento

```
SET BrokenWeeks=0;  
SET ReferenceDay=2;  
SET FirstWeekDay=1;
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8176,12/28/2020,19.42
```

```
8177,12/29/2020,23.80
```

```
8178,12/30/2020,82.06
```

```
8179,12/31/2020,40.56
```

```
8180,01/01/2021,37.23
```

```
8181,01/02/2021,17.17
```

```
8182,01/03/2021,88.27
```

```
8183,01/04/2021,57.42
```

```
8184,01/05/2021,67.42
```

```
8185,01/06/2021,23.80
```

```
8186,01/07/2021,82.06
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella.

Per calcolare l'anno in cui avviene una transazione in base al numero della settimana, creare la seguente misura:

- `=weekyear(date)`

Per calcolare le vendite totali, creare la seguente misura:

- `sum(amount)`

Impostare la misura **Formattazione numero** su **Denaro**.

Tabella dei risultati

| <code>weekyear(data)</code> | <code>sum(amount)</code> |
|-----------------------------|--------------------------|
| 2020                        | 19.42                    |
| 2021                        | 373.37                   |

### year

Questa funzione restituisce un numero intero che rappresenta l'anno in cui **expression** viene interpretato come data in base all'interpretazione numerica standard.

#### Sintassi:

**year** (*expression*)

**Tipo di dati restituiti:** numero intero

La funzione `year()` è disponibile sia come funzione script che come grafico. La funzione restituisce l'anno per una data particolare. È comunemente utilizzata per creare un campo anno come dimensione in un Calendario principale.

### Casi di utilizzo

La funzione `year()` è utile quando si desidera confrontare le aggregazioni per anno. Ad esempio, la funzione potrebbe essere usata se si desidera visualizzare le vendite totali dei prodotti in base all'anno.

È possibile creare queste dimensioni anche nello script di caricamento utilizzando la funzione che consente di creare un campo in una tabella Calendario principale. In alternativa, può essere utilizzata direttamente in un grafico come dimensione calcolata.

Esempi di funzioni

| Esempio                           | Risultato                                  |
|-----------------------------------|--------------------------------------------|
| <code>year( '2012-10-12' )</code> | restituisce 2012                           |
| <code>year( '35648' )</code>      | restituisce 1997 poiché 35648 = 1997-08-06 |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - set di dati DateFormat (script)

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati delle date, caricato in una tabella denominata `Master_Calendar`.
- Viene utilizzata la variabile di sistema predefinita `DateFormat MM/GG/AAAA`.
- Un caricamento precedente, utilizzato per creare un campo aggiuntivo, denominato `year`, mediante la funzione `year()`.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Master_Calendar:
```

```
    Load
        date,
        year(date) as year
    ;
Load
date
Inline
[
date
12/28/2020
12/29/2020
12/30/2020
12/31/2020
```

```
01/01/2021
01/02/2021
01/03/2021
01/04/2021
01/05/2021
01/06/2021
01/07/2021
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- year

Tabella dei risultati

| date       | anno |
|------------|------|
| 12/28/2020 | 2020 |
| 12/29/2020 | 2020 |
| 12/30/2020 | 2020 |
| 12/31/2020 | 2020 |
| 01/01/2021 | 2021 |
| 01/02/2021 | 2021 |
| 01/03/2021 | 2021 |
| 01/04/2021 | 2021 |
| 01/05/2021 | 2021 |
| 01/06/2021 | 2021 |
| 01/07/2021 | 2021 |

### Esempio 2 - Date ANSI

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati delle date, caricato in una tabella denominata `Master_Calendar`.
- È utilizzata la variabile di sistema predefinita `DateFormat` (MM/GG/AAAA). Tuttavia, le date incluse nel set di dati sono nel formato data standard ANSI.

- Un caricamento precedente, utilizzato per creare un campo aggiuntivo, denominato `year`, mediante la funzione `year()`.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Master_Calendar:
```

```
    Load
```

```
        date,
```

```
        year(date) as year
```

```
    ;
```

```
Load
```

```
date
```

```
Inline
```

```
[
```

```
date
```

```
2020-12-28
```

```
2020-12-29
```

```
2020-12-30
```

```
2020-12-31
```

```
2021-01-01
```

```
2021-01-02
```

```
2021-01-03
```

```
2021-01-04
```

```
2021-01-05
```

```
2021-01-06
```

```
2021-01-07
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `date`
- `year`

Tabella dei risultati

| date       | anno |
|------------|------|
| 2020-12-28 | 2020 |
| 2020-12-29 | 2020 |
| 2020-12-30 | 2020 |
| 2020-12-31 | 2020 |
| 2021-01-01 | 2021 |
| 2021-01-02 | 2021 |
| 2021-01-03 | 2021 |

| date       | anno |
|------------|------|
| 2021-01-04 | 2021 |
| 2021-01-05 | 2021 |
| 2021-01-06 | 2021 |
| 2021-01-07 | 2021 |

### Esempio 3 - Date non formattate

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati delle date in formato numerico, caricato in una tabella denominata `Master_Calendar`.
- È utilizzata la variabile di sistema predefinita `DateFormat` (MM/GG/AAAA).
- Un caricamento precedente, utilizzato per creare un campo aggiuntivo, denominato `year`, mediante la funzione `year()`.

Viene caricata la data originale non formattata, denominata `unformatted_date`, e per maggiore chiarezza, viene utilizzato un ulteriore campo aggiuntivo, denominato `long_date`, per convertire la data numerica in un campo data formattato utilizzando la funzione `date()`.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Master_Calendar:
```

```
    Load
        unformatted_date,
        date(unformatted_date) as long_date,
        year(unformatted_date) as year
    ;
```

```
Load
unformatted_date
Inline
[
unformatted_date
44868
44898
44928
44958
44988
45018
45048
45078
```

```
45008  
45038  
45068  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- unformatted\_date
- long\_date
- year

Tabella dei risultati

| unformatted_date | long_date  | anno |
|------------------|------------|------|
| 44868            | 11/03/2022 | 2022 |
| 44898            | 12/03/2022 | 2022 |
| 44928            | 01/02/2023 | 2023 |
| 44958            | 02/01/2023 | 2023 |
| 44988            | 03/03/2023 | 2023 |
| 45008            | 03/23/2023 | 2023 |
| 45018            | 04/02/2023 | 2023 |
| 45038            | 04/22/2023 | 2023 |
| 45048            | 05/02/2023 | 2023 |
| 45068            | 05/22/2023 | 2023 |
| 45078            | 06/01/2023 | 2023 |

### Esempio 4 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor di caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

In questo esempio, un insieme di dati relativi agli ordini effettuati viene caricato in una tabella denominata Vendite. La tabella contiene tre campi:

- id
- sales\_date
- amount

Le garanzie sulle vendite dei prodotti durano due anni dalla data di vendita. L'attività consiste nel creare una misura in un grafico per determinare l'anno in cui scadrà ogni garanzia.

### Script di caricamento

```
sales:
Load
id,
sales_date,
amount
Inline
[
id,sales_date,amount
1,12/28/2020,231.24,
2,12/29/2020,567.28,
3,12/30/2020,364.28,
4,12/31/2020,575.76,
5,01/01/2021,638.68,
6,01/02/2021,785.38,
7,01/03/2021,967.46,
8,01/04/2021,287.67
9,01/05/2021,764.45,
10,01/06/2021,875.43,
11,01/07/2021,957.35
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: sales\_date.

Creare la seguente misura:

```
=year(sales_date+365*2)
```

Tabella dei risultati

| sales_date | =year(sales_date+365*2) |
|------------|-------------------------|
| 12/28/2020 | 2022                    |
| 12/29/2020 | 2022                    |
| 12/30/2020 | 2022                    |
| 12/31/2020 | 2022                    |
| 01/01/2021 | 2023                    |
| 01/02/2021 | 2023                    |
| 01/03/2021 | 2023                    |
| 01/04/2021 | 2023                    |

| sales_date | =year(sales_date+365*2) |
|------------|-------------------------|
| 01/05/2021 | 2023                    |
| 01/06/2021 | 2023                    |
| 01/07/2021 | 2023                    |

I risultati di questa misura sono riportati nella tabella precedente. Per aggiungere due anni a una data, moltiplicare 365 per 2 e aggiungere il risultato alla data di vendita. Pertanto, le vendite avvenute nel 2020 hanno come anno di scadenza il 2022.

## yearend

Questa funzione restituisce un valore corrispondente a un indicatore temporale recante l'ultimo millisecondo dell'ultimo giorno dell'anno contenente **date**. Il formato di output predefinito sarà il formato **DateFormat** impostato nello script.

### Sintassi:

```
YearEnd( date[, period_no[, first_month_of_year = 1]])
```

In altre parole, la funzione `yearend()` determina in quale anno cade la data. Quindi, restituisce data e ora, nel formato data, per l'ultimo millisecondo di quell'anno. Il primo mese dell'anno è, per impostazione predefinita, gennaio. Tuttavia, è possibile modificare il mese da impostare come primo utilizzando l'argomento `first_month_of_year` nella funzione `yearend()`.



*La funzione `yearend()` non considera la variabile di sistema `FirstMonthOfYear`. L'anno inizia il 1 gennaio a condizione che l'argomento `first_month_of_year` non venga utilizzato per modificarlo.*

Schema della funzione `yearend()`.



### Casi di utilizzo

La funzione `yearend()` viene utilizzata come parte di un'espressione quando si desidera che il calcolo utilizzi la frazione dell'anno non ancora trascorso. Ad esempio, se si vuole calcolare l'interesse totale non ancora maturato durante l'anno.

Tipo di dati restituiti: duale

#### Argomenti

| Argomento                  | Descrizione                                                                                                                                                                                                                 |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>date</b>                | La data o la data e ora da valutare.                                                                                                                                                                                        |
| <b>period_no</b>           | <b>period_no</b> è un numero intero, in cui il valore 0 indica l'anno che contiene <b>date</b> . I valori negativi di <b>period_no</b> indicano gli anni precedenti, mentre i valori positivi indicano gli anni successivi. |
| <b>first_month_of_year</b> | Se si intende utilizzare anni (fiscali) che non iniziano a gennaio, indicare un valore compreso tra 2 e 12 in <b>first_month_of_year</b> .                                                                                  |

È possibile utilizzare i seguenti valori per impostare il primo mese dell'anno nell'argomento `first_month_of_year`:

valori `first_month_of_year`

| Mese      | Valore |
|-----------|--------|
| Febbraio  | 2      |
| Marzo     | 3      |
| Aprile    | 4      |
| Maggio    | 5      |
| Giugno    | 6      |
| Luglio    | 7      |
| Agosto    | 8      |
| Settembre | 9      |
| Ottobre   | 10     |
| Novembre  | 11     |
| Dicembre  | 12     |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

Esempi di funzioni

| Esempio                                  | Risultato                        |
|------------------------------------------|----------------------------------|
| <code>yearend('10/19/2001')</code>       | Restituisce 12/31/2001 23:59:59. |
| <code>yearend('10/19/2001', -1)</code>   | Restituisce 12/31/2000 23:59:59. |
| <code>yearend('10/19/2001', 0, 4)</code> | Restituisce 03/31/2002 23:59:59. |

### Esempio 1 - Nessun argomento aggiuntivo

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni tra il 2020 e il 2022 viene caricato in una tabella denominata 'Transactions'.
- Il campo della data è stato fornito nel formato della variabile di sistema `DateFormat`, ossia (MM/DD/YYYY).
- Un'istruzione di caricamento precedente che contiene quanto segue:
  - `yearend()` funzione che è impostata come campo `year_end`.
  - `Timestamp()` funzione che è impostata come campo `year_end_timestamp`.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        yearend(date) as year_end,
        timestamp(yearend(date)) as year_end_timestamp
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,01/13/2020,37.23
```

```
8189,02/26/2020,17.17
8190,03/27/2020,88.27
8191,04/16/2020,57.42
8192,05/21/2020,53.80
8193,08/14/2020,82.06
8194,10/07/2020,40.39
8195,12/05/2020,87.21
8196,01/22/2021,95.93
8197,02/03/2021,45.89
8198,03/17/2021,36.23
8199,04/23/2021,25.66
8200,05/04/2021,82.77
8201,06/30/2021,69.98
8202,07/26/2021,76.11
8203,12/27/2021,25.12
8204,06/06/2022,46.23
8205,07/18/2022,84.21
8206,11/14/2022,96.24
8207,12/12/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- date
- year\_end
- year\_end\_timestamp

Tabella dei risultati

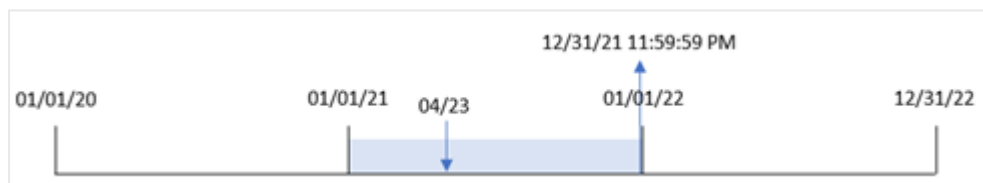
| id   | date       | year_end   | year_end_timestamp     |
|------|------------|------------|------------------------|
| 8188 | 01/13/2020 | 12/31/2020 | 12/31/2020 11:59:59 PM |
| 8189 | 02/26/2020 | 12/31/2020 | 12/31/2020 11:59:59 PM |
| 8190 | 03/27/2020 | 12/31/2020 | 12/31/2020 11:59:59 PM |
| 8191 | 04/16/2020 | 12/31/2020 | 12/31/2020 11:59:59 PM |
| 8192 | 05/21/2020 | 12/31/2020 | 12/31/2020 11:59:59 PM |
| 8193 | 08/14/2020 | 12/31/2020 | 12/31/2020 11:59:59 PM |
| 8194 | 10/07/2020 | 12/31/2020 | 12/31/2020 11:59:59 PM |
| 8195 | 12/05/2020 | 12/31/2020 | 12/31/2020 11:59:59 PM |
| 8196 | 01/22/2021 | 12/31/2021 | 12/31/2021 11:59:59 PM |
| 8197 | 02/03/2021 | 12/31/2021 | 12/31/2021 11:59:59 PM |
| 8198 | 03/17/2021 | 12/31/2021 | 12/31/2021 11:59:59 PM |

| id   | date       | year_end   | year_end_timestamp     |
|------|------------|------------|------------------------|
| 8199 | 04/23/2021 | 12/31/2021 | 12/31/2021 11:59:59 PM |
| 8200 | 05/04/2021 | 12/31/2021 | 12/31/2021 11:59:59 PM |
| 8201 | 06/30/2021 | 12/31/2021 | 12/31/2021 11:59:59 PM |
| 8202 | 07/26/2021 | 12/31/2021 | 12/31/2021 11:59:59 PM |
| 8203 | 12/27/2021 | 12/31/2021 | 12/31/2021 11:59:59 PM |
| 8204 | 06/06/2022 | 12/31/2022 | 12/31/2022 11:59:59 PM |
| 8205 | 07/18/2022 | 12/31/2022 | 12/31/2022 11:59:59 PM |
| 8206 | 11/14/2022 | 12/31/2022 | 12/31/2022 11:59:59 PM |
| 8207 | 12/12/2022 | 12/31/2022 | 12/31/2022 11:59:59 PM |

Il campo 'year\_end' viene creato nell'istruzione di caricamento precedente mediante l'uso della funzione `yearend()` e trasferendo il campo data come argomento della funzione.

La funzione `yearend()` inizialmente identifica in quale anno rientra il valore della data e restituisce data e ora per l'ultimo millisecondo di quell'anno.

*Diagramma della funzione `yearend()` con la transazione 8199 selezionata.*



La transazione 8199 è avvenuta il 23 aprile 2021. La funzione `yearend()` restituisce l'ultimo millisecondo di quell'anno, ovvero il 31 dicembre alle ore 11:59:59 PM.

### Esempio 2 - period\_no

Script di caricamento e risultati

#### Panoramica

Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, in questo esempio, l'attività consiste nel creare un campo, 'previous\_year\_end', che restituisce data e ora di fine dell'anno precedente all'anno in cui ha avuto luogo la transazione.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
  Load
```

```
*,
    yearend(date,-1) as previous_year_end,
    timestamp(yearend(date,-1)) as previous_year_end_timestamp
;

Load
*
Inline
[
id,date,amount
8188,01/13/2020,37.23
8189,02/26/2020,17.17
8190,03/27/2020,88.27
8191,04/16/2020,57.42
8192,05/21/2020,53.80
8193,08/14/2020,82.06
8194,10/07/2020,40.39
8195,12/05/2020,87.21
8196,01/22/2021,95.93
8197,02/03/2021,45.89
8198,03/17/2021,36.23
8199,04/23/2021,25.66
8200,05/04/2021,82.77
8201,06/30/2021,69.98
8202,07/26/2021,76.11
8203,12/27/2021,25.12
8204,06/06/2022,46.23
8205,07/18/2022,84.21
8206,11/14/2022,96.24
8207,12/12/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- date
- previous\_year\_end
- previous\_ year\_end\_timestamp

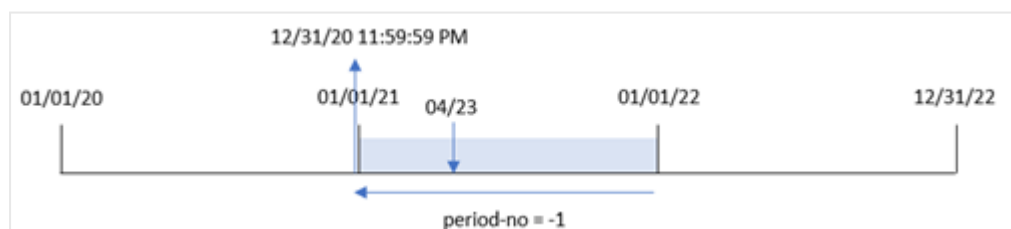
Tabella dei risultati

| id   | date       | previous_year_end | previous_year_end_timestamp |
|------|------------|-------------------|-----------------------------|
| 8188 | 01/13/2020 | 12/31/2019        | 12/31/2019 11:59:59 PM      |
| 8189 | 02/26/2020 | 12/31/2019        | 12/31/2019 11:59:59 PM      |
| 8190 | 03/27/2020 | 12/31/2019        | 12/31/2019 11:59:59 PM      |
| 8191 | 04/16/2020 | 12/31/2019        | 12/31/2019 11:59:59 PM      |
| 8192 | 05/21/2020 | 12/31/2019        | 12/31/2019 11:59:59 PM      |

| id   | date       | previous_year_end | previous_year_end_timestamp |
|------|------------|-------------------|-----------------------------|
| 8193 | 08/14/2020 | 12/31/2019        | 12/31/2019 11:59:59 PM      |
| 8194 | 10/07/2020 | 12/31/2019        | 12/31/2019 11:59:59 PM      |
| 8195 | 12/05/2020 | 12/31/2019        | 12/31/2019 11:59:59 PM      |
| 8196 | 01/22/2021 | 12/31/2020        | 12/31/2020 11:59:59 PM      |
| 8197 | 02/03/2021 | 12/31/2020        | 12/31/2020 11:59:59 PM      |
| 8198 | 03/17/2021 | 12/31/2020        | 12/31/2020 11:59:59 PM      |
| 8199 | 04/23/2021 | 12/31/2020        | 12/31/2020 11:59:59 PM      |
| 8200 | 05/04/2021 | 12/31/2020        | 12/31/2020 11:59:59 PM      |
| 8201 | 06/30/2021 | 12/31/2020        | 12/31/2020 11:59:59 PM      |
| 8202 | 07/26/2021 | 12/31/2020        | 12/31/2020 11:59:59 PM      |
| 8203 | 12/27/2021 | 12/31/2020        | 12/31/2020 11:59:59 PM      |
| 8204 | 06/06/2022 | 12/31/2021        | 12/31/2021 11:59:59 PM      |
| 8205 | 07/18/2022 | 12/31/2021        | 12/31/2021 11:59:59 PM      |
| 8206 | 11/14/2022 | 12/31/2021        | 12/31/2021 11:59:59 PM      |
| 8207 | 12/12/2022 | 12/31/2021        | 12/31/2021 11:59:59 PM      |

Poiché il valore `period_no` di `-1` è stato utilizzato come argomento `offset` nella funzione `yearend()`, la funzione per prima cosa identifica l'anno in cui avvengono le transazioni. Quindi, cerca l'anno precedente e identifica il primo millisecondo di quell'anno.

*Schema della funzione `yearend()` con un valore `period_no` di `-1`.*



La transazione 8199 avviene il 23 aprile 2021. La funzione `yearend()` restituisce l'ultimo millisecondo dell'anno precedente, ovvero il 31 dicembre 2020 alle ore 11:59:59 PM, per il campo `'previous_year_end'`.

### Esempio 3 - `first_month_of_year`

Script di caricamento e risultati

#### Panoramica

Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, in questo esempio, la politica aziendale prevede che l'anno inizi il 1 aprile.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*,
```

```
    yearend(date,0,4) as year_end,
```

```
    timestamp(yearend(date,0,4)) as year_end_timestamp
```

```
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,01/13/2020,37.23
```

```
8189,02/26/2020,17.17
```

```
8190,03/27/2020,88.27
```

```
8191,04/16/2020,57.42
```

```
8192,05/21/2020,53.80
```

```
8193,08/14/2020,82.06
```

```
8194,10/07/2020,40.39
```

```
8195,12/05/2020,87.21
```

```
8196,01/22/2021,95.93
```

```
8197,02/03/2021,45.89
```

```
8198,03/17/2021,36.23
```

```
8199,04/23/2021,25.66
```

```
8200,05/04/2021,82.77
```

```
8201,06/30/2021,69.98
```

```
8202,07/26/2021,76.11
```

```
8203,12/27/2021,25.12
```

```
8204,06/06/2022,46.23
```

```
8205,07/18/2022,84.21
```

```
8206,11/14/2022,96.24
```

```
8207,12/12/2022,67.67
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

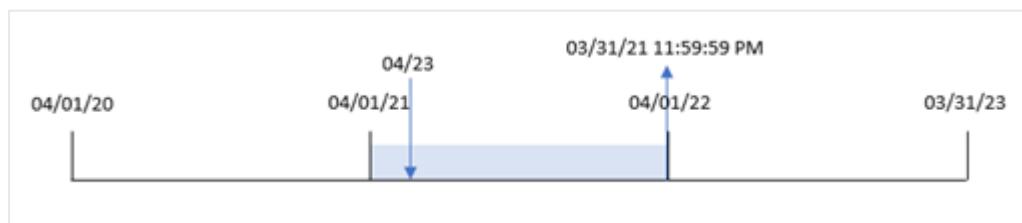
- id
- date
- year\_end
- year\_end\_timestamp

Tabella dei risultati

| id   | date       | year_end   | year_end_timestamp    |
|------|------------|------------|-----------------------|
| 8188 | 01/13/2020 | 03/31/2020 | 3/31/2020 11:59:59 PM |
| 8189 | 02/26/2020 | 03/31/2020 | 3/31/2020 11:59:59 PM |
| 8190 | 03/27/2020 | 03/31/2020 | 3/31/2020 11:59:59 PM |
| 8191 | 04/16/2020 | 03/31/2021 | 3/31/2021 11:59:59 PM |
| 8192 | 05/21/2020 | 03/31/2021 | 3/31/2021 11:59:59 PM |
| 8193 | 08/14/2020 | 03/31/2021 | 3/31/2021 11:59:59 PM |
| 8194 | 10/07/2020 | 03/31/2021 | 3/31/2021 11:59:59 PM |
| 8195 | 12/05/2020 | 03/31/2021 | 3/31/2021 11:59:59 PM |
| 8196 | 01/22/2021 | 03/31/2021 | 3/31/2021 11:59:59 PM |
| 8197 | 02/03/2021 | 03/31/2021 | 3/31/2021 11:59:59 PM |
| 8198 | 03/17/2021 | 03/31/2021 | 3/31/2021 11:59:59 PM |
| 8199 | 04/23/2021 | 03/31/2022 | 3/31/2022 11:59:59 PM |
| 8200 | 05/04/2021 | 03/31/2022 | 3/31/2022 11:59:59 PM |
| 8201 | 06/30/2021 | 03/31/2022 | 3/31/2022 11:59:59 PM |
| 8202 | 07/26/2021 | 03/31/2022 | 3/31/2022 11:59:59 PM |
| 8203 | 12/27/2021 | 03/31/2022 | 3/31/2022 11:59:59 PM |
| 8204 | 06/06/2022 | 03/31/2023 | 3/31/2023 11:59:59 PM |
| 8205 | 07/18/2022 | 03/31/2023 | 3/31/2023 11:59:59 PM |
| 8206 | 11/14/2022 | 03/31/2023 | 3/31/2023 11:59:59 PM |
| 8207 | 12/12/2022 | 03/31/2023 | 3/31/2023 11:59:59 PM |

Poiché l'argomento `first_month_of_year` di 4 viene utilizzato nella funzione `yearend()`, imposta il primo giorno dell'anno come il 1 aprile e l'ultimo giorno dell'anno come il 31 marzo.

Lo schema della funzione `yearend()` con aprile come primo mese dell'anno.



La transazione 8199 avviene il 23 aprile 2021. Poiché la funzione `yearend()` imposta l'inizio dell'anno sul 1 aprile, restituisce il 31 marzo 2022 come valore `'year_end'` per la transazione.

### Esempio 4 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, in questo esempio, il set di dati è invariato e viene caricato nell'applicazione. Il calcolo che restituisce data e ora della data di fine per l'anno in cui è avvenuta una transazione viene creato come misura in un oggetto grafico dell'applicazione.

#### Script di caricamento

```
Transactions:
Load
*
Inline
[
id,date,amount
8188,01/13/2020,37.23
8189,02/26/2020,17.17
8190,03/27/2020,88.27
8191,04/16/2020,57.42
8192,05/21/2020,53.80
8193,08/14/2020,82.06
8194,10/07/2020,40.39
8195,12/05/2020,87.21
8196,01/22/2021,95.93
8197,02/03/2021,45.89
8198,03/17/2021,36.23
8199,04/23/2021,25.66
8200,05/04/2021,82.77
8201,06/30/2021,69.98
8202,07/26/2021,76.11
8203,12/27/2021,25.12
8204,06/06/2022,46.23
8205,07/18/2022,84.21
8206,11/14/2022,96.24
8207,12/12/2022,67.67
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- date

Per calcolare in quale anno è avvenuta una transazione, creare le seguenti misure:

- =yearend(date)
- =timestamp(yearend(date))

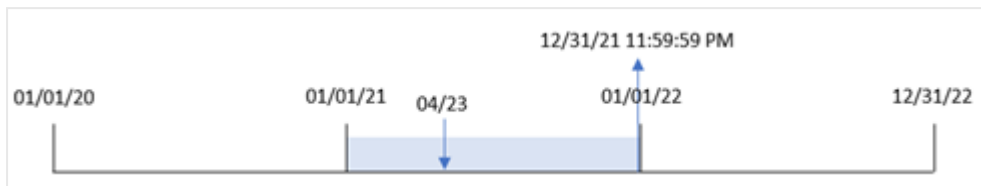
Tabella dei risultati

| id   | date       | =yearend(date) | =timestamp(yearend(date)) |
|------|------------|----------------|---------------------------|
| 8188 | 01/13/2020 | 12/31/2020     | 12/31/2020 11:59:59 PM    |
| 8189 | 02/26/2020 | 12/31/2020     | 12/31/2020 11:59:59 PM    |
| 8190 | 03/27/2020 | 12/31/2020     | 12/31/2020 11:59:59 PM    |
| 8191 | 04/16/2020 | 12/31/2020     | 12/31/2020 11:59:59 PM    |
| 8192 | 05/21/2020 | 12/31/2020     | 12/31/2020 11:59:59 PM    |
| 8193 | 08/14/2020 | 12/31/2020     | 12/31/2020 11:59:59 PM    |
| 8194 | 10/07/2020 | 12/31/2020     | 12/31/2020 11:59:59 PM    |
| 8195 | 12/05/2020 | 12/31/2020     | 12/31/2020 11:59:59 PM    |
| 8196 | 01/22/2021 | 12/31/2021     | 12/31/2021 11:59:59 PM    |
| 8197 | 02/03/2021 | 12/31/2021     | 12/31/2021 11:59:59 PM    |
| 8198 | 03/17/2021 | 12/31/2021     | 12/31/2021 11:59:59 PM    |
| 8199 | 04/23/2021 | 12/31/2021     | 12/31/2021 11:59:59 PM    |
| 8200 | 05/04/2021 | 12/31/2021     | 12/31/2021 11:59:59 PM    |
| 8201 | 06/30/2021 | 12/31/2021     | 12/31/2021 11:59:59 PM    |
| 8202 | 07/26/2021 | 12/31/2021     | 12/31/2021 11:59:59 PM    |
| 8203 | 12/27/2021 | 12/31/2021     | 12/31/2021 11:59:59 PM    |
| 8204 | 06/06/2022 | 12/31/2022     | 12/31/2022 11:59:59 PM    |
| 8205 | 07/18/2022 | 12/31/2022     | 12/31/2022 11:59:59 PM    |
| 8206 | 11/14/2022 | 12/31/2022     | 12/31/2022 11:59:59 PM    |
| 8207 | 12/12/2022 | 12/31/2022     | 12/31/2022 11:59:59 PM    |

La misura 'end\_of\_year' viene creata nell'oggetto grafico mediante l'utilizzo della funzione yearend() e trasferendo il campo data come argomento della funzione.

La funzione yearend() inizialmente identifica in quale anno rientra il valore della data, restituendo data e ora per l'ultimo millisecondo di quell'anno.

Schema della funzione `yearend()` che mostra la transazione 8199 avvenuta ad aprile.



La transazione 8199 avviene il 23 aprile 2021. La funzione `yearend()` restituisce l'ultimo millisecondo di quell'anno, ovvero il 31 dicembre alle ore 11:59:59 PM.

### Esempio 5 - Scenario

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati caricato in una tabella denominata 'Employee\_Expenses'. La tabella contiene i seguenti campi:
  - ID dipendenti
  - nome dipendenti
  - media delle richieste di rimborso spese giornaliero di ciascun dipendente

L'utente finale desidera un oggetto grafico che visualizzi, in base all'ID dipendente e al nome del dipendente, le richieste di rimborso spese stimate ancora da sostenere per il resto dell'anno. L'esercizio finanziario inizia a gennaio.

#### Script di caricamento

```
Employee_Expenses:
Load
*
Inline
[
employee_id,employee_name,avg_daily_claim
182,Mark, $15
183,Deryck, $12.5
184,Dexter, $12.5
185,Sydney,$27
186,Agatha,$18
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- employee\_id
- employee\_name

Per calcolare una proiezione delle richieste di rimborso spese, creare la seguente misura:

`=(yearend(today(1))-today(1))*avg_daily_claim`

Impostare la misura **Formattazione numero** su **Denaro**.

Tabella dei risultati

| employee_id | employee_name | =(yearend(today(1))-today(1))*avg_daily_claim |
|-------------|---------------|-----------------------------------------------|
| 182         | Contrassegno  | \$3240.00                                     |
| 183         | Deryck        | \$2700.00                                     |
| 184         | Dexter        | \$2700.00                                     |
| 185         | Sydney        | \$5832.00                                     |
| 186         | Agatha        | \$3888.00                                     |

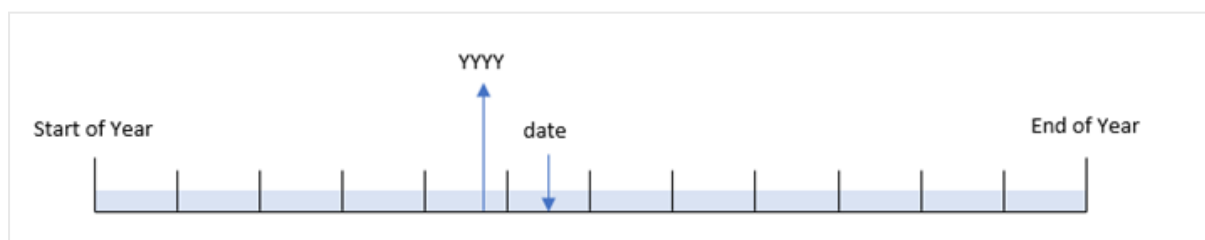
Utilizzando la data odierna come unico argomento, la funzione `yearend()` restituisce la data di fine dell'anno corrente. Quindi, sottraendo la data odierna dalla data di fine dell'anno, l'espressione restituisce il numero di giorni rimanenti per l'anno corrente.

Questo valore viene quindi moltiplicato per la media delle richieste di rimborso spese giornaliere di ciascun dipendente per calcolare il valore stimato delle richieste che ogni dipendente dovrebbe presentare durante il periodo rimanente dell'anno.

## yearname

Questa funzione restituisce un anno di quattro cifre come valore di visualizzazione con un valore numerico sottostante corrispondente a un indicatore temporale recante il primo millisecondo del primo giorno dell'anno contenente **date**.

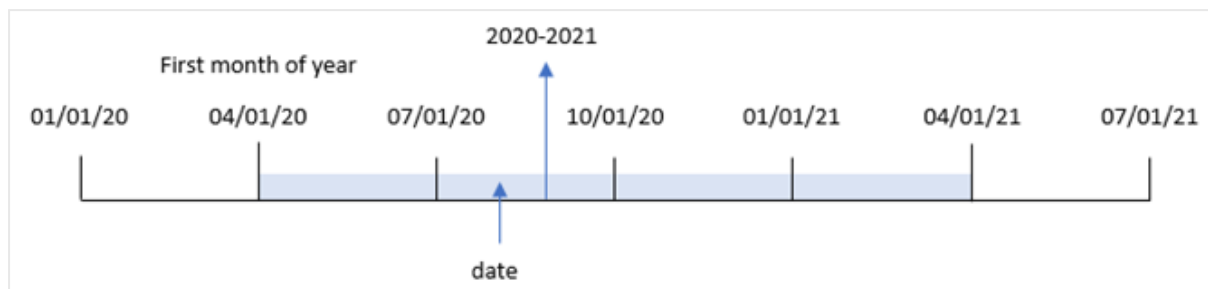
*Diagramma dell'intervallo di tempo della funzione `yearname()`.*



La funzione `yearname()` è diversa dalla funzione `year()` in quanto consente di compensare la data che si desidera valutare e di impostare il primo mese dell'anno.

Se il primo mese dell'anno non è gennaio, la funzione restituirà quattro cifre per i due anni nel periodo di dodici mesi che contengono la data. Ad esempio, se l'anno inizia ad aprile e la data da valutare è il giorno 30/06/2020, il risultato restituito è 2020-2021.

Lo schema della funzione `yearname()` con aprile impostato come primo mese dell'anno.



Sintassi:

```
YearName (date[, period_no[, first_month_of_year]] )
```

Tipo di dati restituiti: duale

| Argomento                  | Descrizione                                                                                                                                                                                                                 |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>date</b>                | La data o la data e ora da valutare.                                                                                                                                                                                        |
| <b>period_no</b>           | <b>period_no</b> è un numero intero, in cui il valore 0 indica l'anno che contiene <b>date</b> . I valori negativi di <b>period_no</b> indicano gli anni precedenti, mentre i valori positivi indicano gli anni successivi. |
| <b>first_month_of_year</b> | Se si intende utilizzare anni (fiscali) che non iniziano a gennaio, indicare un valore compreso tra 2 e 12 in <b>first_month_of_year</b> . Il valore visualizzato sarà quindi una stringa che mostrerà due anni.            |

È possibile utilizzare i seguenti valori per impostare il primo mese dell'anno nell'argomento `first_month_of_year`:

valori `first_month_of_year`

| Mese      | Valore |
|-----------|--------|
| Febbraio  | 2      |
| Marzo     | 3      |
| Aprile    | 4      |
| Maggio    | 5      |
| Giugno    | 6      |
| Luglio    | 7      |
| Agosto    | 8      |
| Settembre | 9      |
| Ottobre   | 10     |
| Novembre  | 11     |
| Dicembre  | 12     |

### Casi di utilizzo

La funzione `yearname()` è utile per confrontare le aggregazioni per anno. Ad esempio, può essere usata se si desidera visualizzare le vendite totali dei prodotti in base all'anno.

È possibile creare queste dimensioni nello script di caricamento utilizzando la funzione che consente di creare un campo in una tabella Calendario principale. Inoltre, possono essere create in un grafico come dimensioni calcolate.

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

Esempi di funzioni

| Esempio                                   | Risultato                |
|-------------------------------------------|--------------------------|
| <code>yearname('10/19/2001')</code>       | Restituisce '2001.'      |
| <code>yearname('10/19/2001', -1)</code>   | Restituisce '2000'.      |
| <code>yearname('10/19/2001', 0, 4)</code> | Restituisce '2001-2002'. |

Argomenti correlati

| Argomento                        | Descrizione                                                                                                                                                          |
|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">year (page 1242)</a> | Questa funzione restituisce un numero intero che rappresenta l'anno in cui l'espressione viene interpretata come data in base all'interpretazione numerica standard. |

### Esempio 1 - Nessun argomento aggiuntivo

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni tra il 2020 e il 2022 viene caricato in una tabella denominata 'Transactions'.
- La variabile di sistema DateFormat che è impostata su 'MM/DD/YYYY'.
- Un'istruzione di caricamento precedente che utilizza yearname() e ne imposta il valore come campo year\_name.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        yearname(date) as year_name
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'01/13/2020',37.23
```

```
8189,'02/26/2020',17.17
```

```
8190,'03/27/2020',88.27
```

```
8191,'04/16/2020',57.42
```

```
8192,'05/21/2020',53.80
```

```
8193,'08/14/2020',82.06
```

```
8194,'10/07/2020',40.39
```

```
8195,'12/05/2020',87.21
```

```
8196,'01/22/2021',95.93
```

```
8197,'02/03/2021',45.89
```

```
8198,'03/17/2021',36.23
```

```
8199,'04/23/2021',25.66
```

```
8200,'05/04/2021',82.77
```

```
8201,'06/30/2021',69.98
```

```
8202,'07/26/2021',76.11
```

```
8203,'12/27/2021',25.12
```

```
8204,'06/06/2022',46.23
```

```
8205,'07/18/2022',84.21
```

```
8206,'11/14/2022',96.24
```

```
8207,'12/12/2022',67.67
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- year\_name

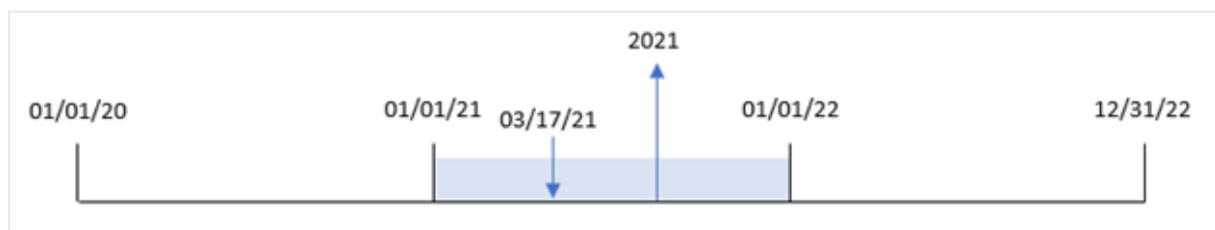
Tabella dei risultati

| date       | year_name |
|------------|-----------|
| 01/13/2020 | 2020      |
| 02/26/2020 | 2020      |
| 03/27/2020 | 2020      |
| 04/16/2020 | 2020      |
| 05/21/2020 | 2020      |
| 08/14/2020 | 2020      |
| 10/07/2020 | 2020      |
| 12/05/2020 | 2020      |
| 01/22/2021 | 2021      |
| 02/03/2021 | 2021      |
| 03/17/2021 | 2021      |
| 04/23/2021 | 2021      |
| 05/04/2021 | 2021      |
| 06/30/2021 | 2021      |
| 07/26/2021 | 2021      |
| 12/27/2021 | 2021      |
| 06/06/2022 | 2022      |
| 07/18/2022 | 2022      |
| 11/14/2022 | 2022      |
| 12/12/2022 | 2022      |

Il campo 'year\_name' viene creato nell'istruzione di caricamento precedente mediante l'uso della funzione `yearname()` e trasferendo il campo data come argomento della funzione.

La funzione `yearname()` identifica l'anno in cui rientra il valore della data e lo restituisce come valore anno a quattro cifre.

*Diagramma della funzione `yearname()` che mostra il 2021 come valore per l'anno.*



### Esempio 2 - period\_no

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni tra il 2020 e il 2022 caricato in una tabella denominata 'Transazioni'.
- La variabile di sistema DateFormat impostata su 'MM/DD/YYYY'.
- Un'istruzione di caricamento precedente che utilizza yearname() e ne imposta il valore come campo year\_name.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        yearname(date,-1) as prior_year_name
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'01/13/2020',37.23
```

```
8189,'02/26/2020',17.17
```

```
8190,'03/27/2020',88.27
```

```
8191,'04/16/2020',57.42
```

```
8192,'05/21/2020',53.80
```

```
8193,'08/14/2020',82.06
```

```
8194,'10/07/2020',40.39
```

```
8195,'12/05/2020',87.21
```

```
8196,'01/22/2021',95.93
```

```
8197,'02/03/2021',45.89
```

```
8198,'03/17/2021',36.23
```

```
8199,'04/23/2021',25.66
```

```
8200,'05/04/2021',82.77
```

```
8201,'06/30/2021',69.98
```

```
8202,'07/26/2021',76.11
```

```
8203,'12/27/2021',25.12
```

```
8204,'06/06/2022',46.23
```

```
8205,'07/18/2022',84.21
```

```
8206,'11/14/2022',96.24
```

```
8207,'12/12/2022',67.67
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

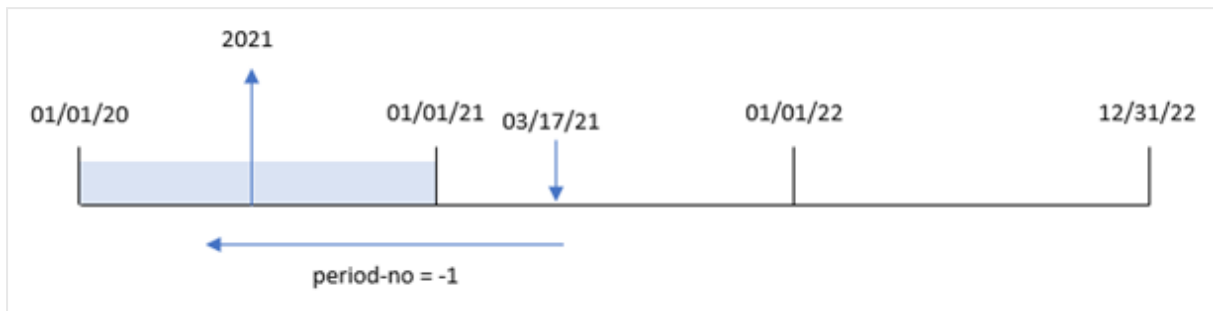
- date
- prior\_year\_name

Tabella dei risultati

| date       | prior_year_name |
|------------|-----------------|
| 01/13/2020 | 2019            |
| 02/26/2020 | 2019            |
| 03/27/2020 | 2019            |
| 04/16/2020 | 2019            |
| 05/21/2020 | 2019            |
| 08/14/2020 | 2019            |
| 10/07/2020 | 2019            |
| 12/05/2020 | 2019            |
| 01/22/2021 | 2020            |
| 02/03/2021 | 2020            |
| 03/17/2021 | 2020            |
| 04/23/2021 | 2020            |
| 05/04/2021 | 2020            |
| 06/30/2021 | 2020            |
| 07/26/2021 | 2020            |
| 12/27/2021 | 2020            |
| 06/06/2022 | 2021            |
| 07/18/2022 | 2021            |
| 11/14/2022 | 2021            |
| 12/12/2022 | 2021            |

Poiché il valore `period_no` di `-1` viene utilizzato come argomento `offset` nella funzione `yearname()`, la funzione per prima cosa identifica l'anno in cui avvengono le transazioni. La funzione quindi si sposta di un anno prima e restituisce l'anno risultante.

Schema della funzione `yearname()` con il valore `period_no` con un impostato su `-1`.



### Esempio 3 - first\_month\_of\_year

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Lo stesso set di dati dal primo esempio.
- La variabile di sistema `DateFormat` che è impostata su `'MM/DD/YYYY'`.
- Un'istruzione di caricamento precedente che utilizza `yearname()` e ne imposta il valore come campo `year_name`.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
*,
yearname(date,0,4) as year_name
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188, '01/13/2020', 37.23
```

```
8189, '02/26/2020', 17.17
```

```
8190, '03/27/2020', 88.27
```

```
8191, '04/16/2020', 57.42
```

```
8192, '05/21/2020', 53.80
```

```
8193, '08/14/2020', 82.06
```

```
8194, '10/07/2020', 40.39
```

```
8195, '12/05/2020', 87.21
```

```
8196, '01/22/2021', 95.93
```

```
8197, '02/03/2021', 45.89
```

```
8198, '03/17/2021', 36.23
```

```
8199, '04/23/2021', 25.66
8200, '05/04/2021', 82.77
8201, '06/30/2021', 69.98
8202, '07/26/2021', 76.11
8203, '12/27/2021', 25.12
8204, '06/06/2022', 46.23
8205, '07/18/2022', 84.21
8206, '11/14/2022', 96.24
8207, '12/12/2022', 67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- date
- year\_name

Tabella dei risultati

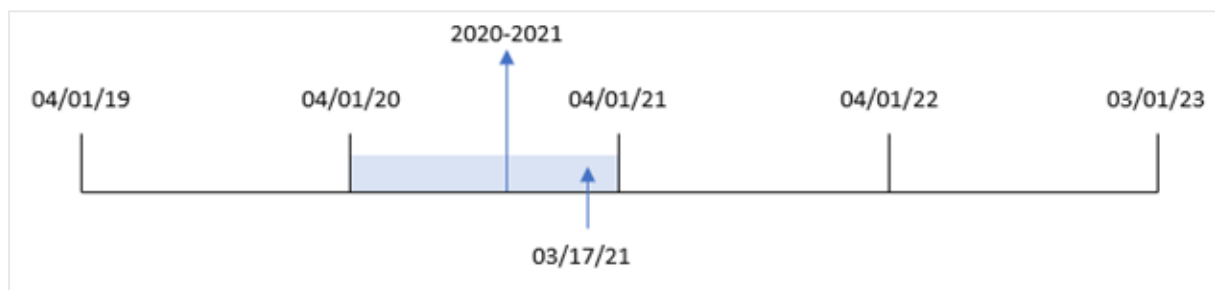
| date       | year_name |
|------------|-----------|
| 01/13/2020 | 2019-2020 |
| 02/26/2020 | 2019-2020 |
| 03/27/2020 | 2019-2020 |
| 04/16/2020 | 2020-2021 |
| 05/21/2020 | 2020-2021 |
| 08/14/2020 | 2020-2021 |
| 10/07/2020 | 2020-2021 |
| 12/05/2020 | 2020-2021 |
| 01/22/2021 | 2020-2021 |
| 02/03/2021 | 2020-2021 |
| 03/17/2021 | 2020-2021 |
| 04/23/2021 | 2021-2022 |
| 05/04/2021 | 2021-2022 |
| 06/30/2021 | 2021-2022 |
| 07/26/2021 | 2021-2022 |
| 12/27/2021 | 2021-2022 |
| 06/06/2022 | 2022-2023 |
| 07/18/2022 | 2022-2023 |

| date       | year_name |
|------------|-----------|
| 11/14/2022 | 2022-2023 |
| 12/12/2022 | 2022-2023 |

Poiché l'argomento `first_month_of_year` di 4 viene utilizzato nella funzione `yearname()`, l'inizio dell'anno va dal 1 gennaio al 1 aprile. Pertanto, ogni periodo di dodici mesi attraversa due anni civili e la funzione `yearname()` restituisce i due anni con formato a quattro cifre anni per le date valutate.

La transazione 8198 avviene il 17 marzo 2021. La funzione `yearname()` imposta l'inizio dell'anno il 1 aprile e la fine il 30 marzo. Pertanto, la transazione 8198 è avvenuta nel periodo dell'anno dal 1 aprile 2020 al 30 marzo 2021. Di conseguenza, la funzione `yearname()` restituisce il valore 2020-2021.

*Lo schema della funzione `yearname()` con marzo impostato come primo mese dell'anno.*



### Esempio 4 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Lo stesso set di dati dal primo esempio.
- La variabile di sistema `DateFormat` che è impostata su `'MM/DD/YYYY'`.

Tuttavia, il campo che restituisce l'anno in cui è avvenuta la transazione viene creato come misura in un oggetto grafico.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'01/13/2020',37.23
```

```
8189, '02/26/2020', 17.17
8190, '03/27/2020', 88.27
8191, '04/16/2020', 57.42
8192, '05/21/2020', 53.80
8193, '08/14/2020', 82.06
8194, '10/07/2020', 40.39
8195, '12/05/2020', 87.21
8196, '01/22/2021', 95.93
8197, '02/03/2021', 45.89
8198, '03/17/2021', 36.23
8199, '04/23/2021', 25.66
8200, '05/04/2021', 82.77
8201, '06/30/2021', 69.98
8202, '07/26/2021', 76.11
8203, '12/27/2021', 25.12
8204, '06/06/2022', 46.23
8205, '07/18/2022', 84.21
8206, '11/14/2022', 96.24
8207, '12/12/2022', 67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

date

Per calcolare il campo 'year\_name', creare questa misura:

=yearname(date)

Tabella dei risultati

| date       | =yearname(data) |
|------------|-----------------|
| 01/13/2020 | 2020            |
| 02/26/2020 | 2020            |
| 03/27/2020 | 2020            |
| 04/16/2020 | 2020            |
| 05/21/2020 | 2020            |
| 08/14/2020 | 2020            |
| 10/07/2020 | 2020            |
| 12/05/2020 | 2020            |
| 01/22/2021 | 2021            |
| 02/03/2021 | 2021            |
| 03/17/2021 | 2021            |
| 04/23/2021 | 2021            |

| date       | =yearname(data) |
|------------|-----------------|
| 05/04/2021 | 2021            |
| 06/30/2021 | 2021            |
| 07/26/2021 | 2021            |
| 12/27/2021 | 2021            |
| 06/06/2022 | 2022            |
| 07/18/2022 | 2022            |
| 11/14/2022 | 2022            |
| 12/12/2022 | 2022            |

La misura 'year\_name' viene creata nell'oggetto grafico utilizzando la funzione `yearname()` e trasferendo il campo data come argomento della funzione.

La funzione `yearname()` identifica l'anno in cui rientra il valore della data e lo restituisce come valore anno a quattro cifre.

*Diagramma della funzione `yearname()` con 2021 come valore per l'anno.*



### Esempio 5 - Scenario

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Lo stesso set di dati dal primo esempio.
- La variabile di sistema `DateFormat` che è impostata su `'MM/DD/YYYY'`.

L'utente finale desidera un grafico che presenta le vendite totali per trimestre per le transazioni. Utilizzare la funzione `yearname()` come dimensione calcolata per creare questo grafico quando la dimensione `yearname()` non è disponibile nel modello dati.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'01/13/2020',37.23
```

```
8189,'02/26/2020',17.17
```

```
8190,'03/27/2020',88.27
```

```
8191,'04/16/2020',57.42
```

```
8192,'05/21/2020',53.80
```

```
8193,'08/14/2020',82.06
```

```
8194,'10/07/2020',40.39
```

```
8195,'12/05/2020',87.21
```

```
8196,'01/22/2021',95.93
```

```
8197,'02/03/2021',45.89
```

```
8198,'03/17/2021',36.23
```

```
8199,'04/23/2021',25.66
```

```
8200,'05/04/2021',82.77
```

```
8201,'06/30/2021',69.98
```

```
8202,'07/26/2021',76.11
```

```
8203,'12/27/2021',25.12
```

```
8204,'06/06/2022',46.23
```

```
8205,'07/18/2022',84.21
```

```
8206,'11/14/2022',96.24
```

```
8207,'12/12/2022',67.67
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella.

Per confrontare le aggregazioni per anno, creare la seguente dimensione calcolata:

```
=yearname(date)
```

Creare questa misura:

```
=sum(amount)
```

Impostare la misura **Formattazione numero** su **Denaro**.

Tabella dei risultati

| yearname(date) | sum(amount) |
|----------------|-------------|
| 2020           | \$463.55    |
| 2021           | \$457.69    |
| 2022           | \$294.35    |

## yearstart

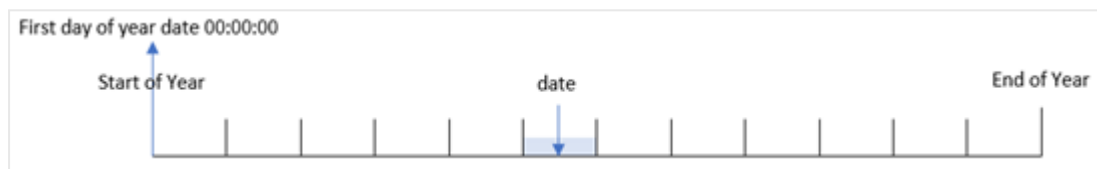
Questa funzione restituisce un indicatore temporale corrispondente all'inizio del primo giorno dell'anno contenente **date**. Il formato di output predefinito sarà il formato **DateFormat** impostato nello script.

### Sintassi:

**YearStart**(date[, period\_no[, first\_month\_of\_year]])

In altre parole, la funzione `yearstart()` determina in quale anno cade la data. Quindi, restituisce data e ora, nel formato data, per il primo millisecondo di quell'anno. Il primo mese dell'anno è, per impostazione predefinita, gennaio. Tuttavia, è possibile modificare il mese da impostare come primo utilizzando l'argomento `first_month_of_year` nella funzione `yearstart()`.

*Diagramma della funzione `yearstart()` che mostra l'intervallo di tempo che la funzione può coprire.*



### Casi di utilizzo

La funzione `yearstart()` viene utilizzata come parte di un'espressione quando si desidera che il calcolo utilizzi la frazione dell'anno trascorso fin'ora. Ad esempio, se si desidera calcolare l'interesse accumulato in un anno a oggi.

**Tipo di dati restituiti:** duale

#### Argomenti

| Argomento                  | Descrizione                                                                                                                                                                                                                 |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>date</b>                | La data o la data e ora da valutare.                                                                                                                                                                                        |
| <b>period_no</b>           | <b>period_no</b> è un numero intero, in cui il valore 0 indica l'anno che contiene <b>date</b> . I valori negativi di <b>period_no</b> indicano gli anni precedenti, mentre i valori positivi indicano gli anni successivi. |
| <b>first_month_of_year</b> | Se si intende utilizzare anni (fiscali) che non iniziano a gennaio, indicare un valore compreso tra 2 e 12 in <b>first_month_of_year</b> .                                                                                  |

I mesi seguenti possono essere utilizzati in `first_month_of_year` argument:

valori `first_month_of_year`

| Mese     | Valore |
|----------|--------|
| Febbraio | 2      |
| Marzo    | 3      |

| Mese      | Valore |
|-----------|--------|
| Aprile    | 4      |
| Maggio    | 5      |
| Giugno    | 6      |
| Luglio    | 7      |
| Agosto    | 8      |
| Settembre | 9      |
| Ottobre   | 10     |
| Novembre  | 11     |
| Dicembre  | 12     |

### Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione `SET DateFormat` nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema, a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

#### Esempi di funzioni

| Esempio                                    | Risultato                        |
|--------------------------------------------|----------------------------------|
| <code>yearstart('10/19/2001')</code>       | Restituisce 01/01/2001 00:00:00. |
| <code>yearstart('10/19/2001', -1)</code>   | Restituisce 01/01/2000 00:00:00. |
| <code>yearstart('10/19/2001', 0, 4)</code> | Restituisce 04/01/2001 00:00:00. |

### Esempio 1 - Esempio di base

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni tra il 2020 e il 2022 viene caricato in una tabella denominata 'Transactions'.
- Il campo della data è stato fornito nel formato della variabile di sistema DateFormat (MM/DD/YYYY).
- Un'istruzione di caricamento precedente che contiene quanto segue:
  - yearstart() funzione che è impostata come campo year\_start.
  - La funzione Timestamp() impostata come campo year\_start\_timestamp.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        yearstart(date) as year_start,
        timestamp(yearstart(date)) as year_start_timestamp
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,01/13/2020,37.23
```

```
8189,02/26/2020,17.17
```

```
8190,03/27/2020,88.27
```

```
8191,04/16/2020,57.42
```

```
8192,05/21/2020,53.80
```

```
8193,08/14/2020,82.06
```

```
8194,10/07/2020,40.39
```

```
8195,12/05/2020,87.21
```

```
8196,01/22/2021,95.93
```

```
8197,02/03/2021,45.89
```

```
8198,03/17/2021,36.23
```

```
8199,04/23/2021,25.66
```

```
8200,05/04/2021,82.77
```

```
8201,06/30/2021,69.98
```

```
8202,07/26/2021,76.11
```

```
8203,12/27/2021,25.12
```

```
8204,06/06/2022,46.23
```

```
8205,07/18/2022,84.21
```

```
8206,11/14/2022,96.24
```

```
8207,12/12/2022,67.67
```

```
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- date

- year\_start
- year\_start\_timestamp

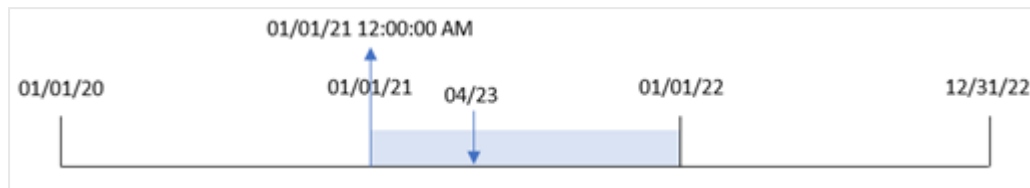
Tabella dei risultati

| id   | date       | year_start | year_start_timestamp  |
|------|------------|------------|-----------------------|
| 8188 | 01/13/2020 | 01/01/2020 | 1/1/2020 12:00:00 AM  |
| 8189 | 02/26/2020 | 01/01/2020 | 1/1/2020 12:00:00 AM  |
| 8190 | 03/27/2020 | 01/01/2020 | 1/1/2020 12:00:00 AM  |
| 8191 | 04/16/2020 | 01/01/2020 | 1/1/2020 12:00:00 AM  |
| 8192 | 05/21/2020 | 01/01/2020 | 1/1/2020 12:00:00 AM  |
| 8193 | 08/14/2020 | 01/01/2020 | 1/1/2020 12:00:00 AM  |
| 8194 | 10/07/2020 | 01/01/2020 | 1/1/2020 12:00:00 AM  |
| 8195 | 12/05/2020 | 01/01/2020 | 1/1/2020 12:00:00 AM  |
| 8196 | 01/22/2021 | 01/01/2021 | 11/1/2021 12:00:00 AM |
| 8197 | 02/03/2021 | 01/01/2021 | 11/1/2021 12:00:00 AM |
| 8198 | 03/17/2021 | 01/01/2021 | 11/1/2021 12:00:00 AM |
| 8199 | 04/23/2021 | 01/01/2021 | 11/1/2021 12:00:00 AM |
| 8200 | 05/04/2021 | 01/01/2021 | 11/1/2021 12:00:00 AM |
| 8201 | 06/30/2021 | 01/01/2021 | 11/1/2021 12:00:00 AM |
| 8202 | 07/26/2021 | 01/01/2021 | 11/1/2021 12:00:00 AM |
| 8203 | 12/27/2021 | 01/01/2021 | 11/1/2021 12:00:00 AM |
| 8204 | 06/06/2022 | 01/01/2022 | 1/1/2022 12:00:00 AM  |
| 8205 | 07/18/2022 | 01/01/2022 | 1/1/2022 12:00:00 AM  |
| 8206 | 11/14/2022 | 01/01/2022 | 1/1/2022 12:00:00 AM  |
| 8207 | 12/12/2022 | 01/01/2022 | 1/1/2022 12:00:00 AM  |

Il campo 'year\_start' viene creato nell'istruzione di caricamento precedente mediante l'uso della funzione `yearstart()` e trasferendo il campo data come argomento della funzione.

La funzione `yearstart()` inizialmente identifica in quale anno rientra il valore della data e restituisce data e ora per il primo millisecondo di quell'anno.

Schema della funzione `yearstart()` e della transazione 8199.



La transazione 8199 è avvenuta il 23 aprile 2021. La funzione `yearstart()` restituisce il primo millisecondo di quell'anno, ovvero il 1 gennaio alle ore 12:00:00 AM.

### Esempio 2 - `period_no`

Script di caricamento e risultati

#### Panoramica

Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, in questo esempio, l'attività consiste nel creare un campo, `'previous_year_start'`, che restituisce data e ora di inizio dell'anno precedente all'anno in cui ha avuto luogo la transazione.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

Transactions:

```
Load
    *,
    yearstart(date,-1) as previous_year_start,
    timestamp(yearstart(date,-1)) as previous_year_start_timestamp
;

Load
*
Inline
[
id,date,amount
8188,01/13/2020,37.23
8189,02/26/2020,17.17
8190,03/27/2020,88.27
8191,04/16/2020,57.42
8192,05/21/2020,53.80
8193,08/14/2020,82.06
8194,10/07/2020,40.39
8195,12/05/2020,87.21
8196,01/22/2021,95.93
8197,02/03/2021,45.89
8198,03/17/2021,36.23
8199,04/23/2021,25.66
8200,05/04/2021,82.77
8201,06/30/2021,69.98
8202,07/26/2021,76.11
8203,12/27/2021,25.12
```

```
8204,06/06/2022,46.23
8205,07/18/2022,84.21
8206,11/14/2022,96.24
8207,12/12/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

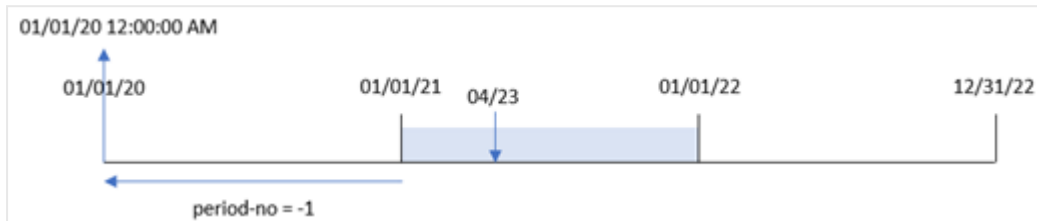
- id
- date
- previous\_year\_start
- previous\_year\_start\_timestamp

Tabella dei risultati

| id   | date       | previous_year_start | previous_year_start_timestamp |
|------|------------|---------------------|-------------------------------|
| 8188 | 01/13/2020 | 01/01/2019          | 1/1/2019 12:00:00 AM          |
| 8189 | 02/26/2020 | 01/01/2019          | 1/1/2019 12:00:00 AM          |
| 8190 | 03/27/2020 | 01/01/2019          | 1/1/2019 12:00:00 AM          |
| 8191 | 04/16/2020 | 01/01/2019          | 1/1/2019 12:00:00 AM          |
| 8192 | 05/21/2020 | 01/01/2019          | 1/1/2019 12:00:00 AM          |
| 8193 | 08/14/2020 | 01/01/2019          | 1/1/2019 12:00:00 AM          |
| 8194 | 10/07/2020 | 01/01/2019          | 1/1/2019 12:00:00 AM          |
| 8195 | 12/05/2020 | 01/01/2019          | 1/1/2019 12:00:00 AM          |
| 8196 | 01/22/2021 | 01/01/2020          | 1/1/2020 12:00:00 AM          |
| 8197 | 02/03/2021 | 01/01/2020          | 1/1/2020 12:00:00 AM          |
| 8198 | 03/17/2021 | 01/01/2020          | 1/1/2020 12:00:00 AM          |
| 8199 | 04/23/2021 | 01/01/2020          | 1/1/2020 12:00:00 AM          |
| 8200 | 05/04/2021 | 01/01/2020          | 1/1/2020 12:00:00 AM          |
| 8201 | 06/30/2021 | 01/01/2020          | 1/1/2020 12:00:00 AM          |
| 8202 | 07/26/2021 | 01/01/2020          | 1/1/2020 12:00:00 AM          |
| 8203 | 12/27/2021 | 01/01/2020          | 1/1/2020 12:00:00 AM          |
| 8204 | 06/06/2022 | 01/01/2021          | 11/1/2021 12:00:00 AM         |
| 8205 | 07/18/2022 | 01/01/2021          | 11/1/2021 12:00:00 AM         |
| 8206 | 11/14/2022 | 01/01/2021          | 11/1/2021 12:00:00 AM         |
| 8207 | 12/12/2022 | 01/01/2021          | 11/1/2021 12:00:00 AM         |

In questa istanza, poiché il valore `period_no` di -1 viene utilizzato come argomento offset nella funzione `yearstart()`, la funzione per prima cosa identifica l'anno in cui avvengono le transazioni. Quindi, cerca l'anno precedente e identifica l'ultimo millisecondo di quell'anno.

*Schema della funzione `yearstart()` con un valore `period_no` di -1.*



La transazione 8199 è avvenuta il 23 aprile 2021. La funzione `yearstart()` restituisce il primo millisecondo dell'anno precedente, ovvero il 1° gennaio 2020 alle ore 12:00:00 AM, per il campo `'previous_year_start'`.

### Esempio 3 - `first_month_of_year`

Script di caricamento e risultati

#### Panoramica

Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, in questo esempio, la politica aziendale prevede che l'anno inizi il 1 aprile.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        yearstart(date,0,4) as year_start,
        timestamp(yearstart(date,0,4)) as year_start_timestamp
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,01/13/2020,37.23
```

```
8189,02/26/2020,17.17
```

```
8190,03/27/2020,88.27
```

```
8191,04/16/2020,57.42
```

```
8192,05/21/2020,53.80
```

```
8193,08/14/2020,82.06
```

```
8194,10/07/2020,40.39
```

```
8195,12/05/2020,87.21
```

```
8196,01/22/2021,95.93
```

```
8197,02/03/2021,45.89
```

```
8198,03/17/2021,36.23
```

```
8199,04/23/2021,25.66
```

```
8200,05/04/2021,82.77
8201,06/30/2021,69.98
8202,07/26/2021,76.11
8203,12/27/2021,25.12
8204,06/06/2022,46.23
8205,07/18/2022,84.21
8206,11/14/2022,96.24
8207,12/12/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- date
- year\_start
- year\_start\_timestamp

Tabella dei risultati

| id   | date       | year_start | year_start_timestamp |
|------|------------|------------|----------------------|
| 8188 | 01/13/2020 | 04/01/2019 | 4/1/2019 12:00:00 AM |
| 8189 | 02/26/2020 | 04/01/2019 | 4/1/2019 12:00:00 AM |
| 8190 | 03/27/2020 | 04/01/2019 | 4/1/2019 12:00:00 AM |
| 8191 | 04/16/2020 | 04/01/2020 | 4/1/2020 12:00:00 AM |
| 8192 | 05/21/2020 | 04/01/2020 | 4/1/2020 12:00:00 AM |
| 8193 | 08/14/2020 | 04/01/2020 | 4/1/2020 12:00:00 AM |
| 8194 | 10/07/2020 | 04/01/2020 | 4/1/2020 12:00:00 AM |
| 8195 | 12/05/2020 | 04/01/2020 | 4/1/2020 12:00:00 AM |
| 8196 | 01/22/2021 | 04/01/2020 | 4/1/2020 12:00:00 AM |
| 8197 | 02/03/2021 | 04/01/2020 | 4/1/2020 12:00:00 AM |
| 8198 | 03/17/2021 | 04/01/2020 | 4/1/2020 12:00:00 AM |
| 8199 | 04/23/2021 | 04/01/2021 | 01/04/2021 00:00:00  |
| 8200 | 05/04/2021 | 04/01/2021 | 01/04/2021 00:00:00  |
| 8201 | 06/30/2021 | 04/01/2021 | 01/04/2021 00:00:00  |
| 8202 | 07/26/2021 | 04/01/2021 | 01/04/2021 00:00:00  |
| 8203 | 12/27/2021 | 04/01/2021 | 01/04/2021 00:00:00  |
| 8204 | 06/06/2022 | 04/01/2022 | 4/1/2022 12:00:00 AM |

| id   | date       | year_start | year_start_timestamp |
|------|------------|------------|----------------------|
| 8205 | 07/18/2022 | 04/01/2022 | 4/1/2022 12:00:00 AM |
| 8206 | 11/14/2022 | 04/01/2022 | 4/1/2022 12:00:00 AM |
| 8207 | 12/12/2022 | 04/01/2022 | 4/1/2022 12:00:00 AM |

In questa istanza, poiché l'argomento `first_month_of_year` di 4 viene utilizzato nella funzione `yearstart()`, imposta il primo giorno dell'anno come il 1 aprile e l'ultimo giorno dell'anno come il 31 marzo.

Lo schema della funzione `yearstart()` con il primo mese impostato su aprile.



La transazione 8199 è avvenuta il 23 aprile 2021. Poiché la funzione `yearstart()` imposta l'inizio dell'anno sul 1 aprile, lo restituisce come valore 'year\_start' per la transazione.

### Esempio 4 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, in questo esempio, il set di dati è invariato e viene caricato nell'applicazione. Il calcolo che restituisce data e ora della data di inizio per l'anno in cui è avvenuta una transazione viene creato come misura in un oggetto grafico dell'applicazione.

#### Script di caricamento

```

Transactions:
Load
*
Inline
[
id,date,amount
8188,01/13/2020,37.23
8189,02/26/2020,17.17
8190,03/27/2020,88.27
8191,04/16/2020,57.42
8192,05/21/2020,53.80
8193,08/14/2020,82.06
8194,10/07/2020,40.39
8195,12/05/2020,87.21
8196,01/22/2021,95.93

```

```
8197,02/03/2021,45.89
8198,03/17/2021,36.23
8199,04/23/2021,25.66
8200,05/04/2021,82.77
8201,06/30/2021,69.98
8202,07/26/2021,76.11
8203,12/27/2021,25.12
8204,06/06/2022,46.23
8205,07/18/2022,84.21
8206,11/14/2022,96.24
8207,12/12/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- id
- date

Per calcolare in quale anno è avvenuta una transazione, creare le seguenti misure:

- =yearstart(date)
- =timestamp(yearstart(date))

Tabella dei risultati

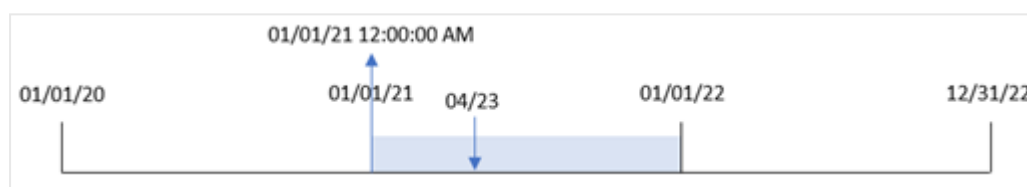
| id   | date       | =yearstart(date) | =timestamp(yearstart(date)) |
|------|------------|------------------|-----------------------------|
| 8188 | 06/06/2022 | 01/01/2022       | 1/1/2022 12:00:00 AM        |
| 8189 | 07/18/2022 | 01/01/2022       | 1/1/2022 12:00:00 AM        |
| 8190 | 11/14/2022 | 01/01/2022       | 1/1/2022 12:00:00 AM        |
| 8191 | 12/12/2022 | 01/01/2022       | 1/1/2022 12:00:00 AM        |
| 8192 | 01/22/2021 | 01/01/2021       | 11/1/2021 12:00:00 AM       |
| 8193 | 02/03/2021 | 01/01/2021       | 11/1/2021 12:00:00 AM       |
| 8194 | 03/17/2021 | 01/01/2021       | 11/1/2021 12:00:00 AM       |
| 8195 | 04/23/2021 | 01/01/2021       | 11/1/2021 12:00:00 AM       |
| 8196 | 05/04/2021 | 01/01/2021       | 11/1/2021 12:00:00 AM       |
| 8197 | 06/30/2021 | 01/01/2021       | 11/1/2021 12:00:00 AM       |
| 8198 | 07/26/2021 | 01/01/2021       | 11/1/2021 12:00:00 AM       |
| 8199 | 12/27/2021 | 01/01/2021       | 11/1/2021 12:00:00 AM       |
| 8200 | 01/13/2020 | 01/01/2020       | 1/1/2020 12:00:00 AM        |
| 8201 | 02/26/2020 | 01/01/2020       | 1/1/2020 12:00:00 AM        |

| id   | date       | =yearstart(date) | =timestamp(yearstart(date)) |
|------|------------|------------------|-----------------------------|
| 8202 | 03/27/2020 | 01/01/2020       | 1/1/2020 12:00:00 AM        |
| 8203 | 04/16/2020 | 01/01/2020       | 1/1/2020 12:00:00 AM        |
| 8204 | 05/21/2020 | 01/01/2020       | 1/1/2020 12:00:00 AM        |
| 8205 | 08/14/2020 | 01/01/2020       | 1/1/2020 12:00:00 AM        |
| 8206 | 10/07/2020 | 01/01/2020       | 1/1/2020 12:00:00 AM        |
| 8207 | 12/05/2020 | 01/01/2020       | 1/1/2020 12:00:00 AM        |

La misura 'start\_of\_year' viene creata nell'oggetto grafico mediante l'utilizzo della funzione `yearstart()` e trasferendo il campo `date` come argomento della funzione.

La funzione `yearstart()` inizialmente identifica in quale anno rientra il valore della `date` e restituisce `date` e ora per il primo millisecondo di quell'anno.

*Schema della funzione `yearstart()` e della transazione 8199.*



La transazione 8199 è avvenuta il 23 aprile 2021. La funzione `yearstart()` restituisce il primo millisecondo di quell'anno, ovvero il 1 gennaio alle ore 12:00:00 AM.

### Esempio 5 - Scenario

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati caricato in una tabella denominata 'Loans'. La tabella contiene i seguenti campi:
  - ID del prestito.
  - Il saldo all'inizio dell'anno.
  - Il tasso di interesse semplice applicato su ciascun prestito ogni anno.

L'utente finale desidera un oggetto grafico che mostri, in base all'ID del prestito, gli interessi correnti che sono stati maturato per ciascun prestito nell'anno in corso.

### Script di caricamento

```
Loans:
Load
*
Inline
[
loan_id,start_balance,rate
8188,$10000.00,0.024
8189,$15000.00,0.057
8190,$17500.00,0.024
8191,$21000.00,0.034
8192,$90000.00,0.084
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- loan\_id
- start\_balance

Per calcolare gli interessi maturati, creare la seguente misura:

`=start_balance*(rate*(today(1)-yearstart(today(1)))/365)`

Impostare la misura **Formattazione numero** su **Denaro**.

Tabella dei risultati

| loan_id | start_balance | =start_balance*(rate*(today(1)-yearstart(today(1)))/365) |
|---------|---------------|----------------------------------------------------------|
| 8188    | \$10000.00    | \$39.73                                                  |
| 8189    | \$15000.00    | \$339.66                                                 |
| 8190    | \$17500.00    | \$166.85                                                 |
| 8191    | \$21000.00    | \$283.64                                                 |
| 8192    | \$90000.00    | \$3003.29                                                |

Utilizzando la data odierna come unico argomento, la funzione `yearstart()` restituisce la data di inizio dell'anno corrente. Sottraendo tale risultato dalla data corrente, l'espressione restituisce il numero di giorni trascorsi fino ad ora quest'anno.

Questo valore viene quindi moltiplicato per il tasso di interesse e diviso per 365, per restituire il tasso di interesse effettivo per questo periodo. Il tasso di interesse effettivo per il periodo viene quindi moltiplicato per il saldo iniziale del prestito, per restituire gli interessi maturati finora nel corso di quest'anno.

## yeartodate

Questa funzione stabilisce se l'indicatore temporale di input ricade all'interno dell'anno in cui lo script è stato caricato per l'ultima volta e restituisce True in caso affermativo e False in caso negativo.

### Sintassi:

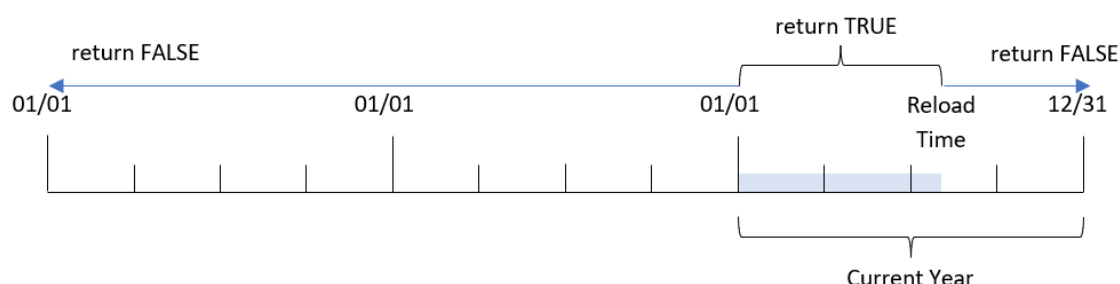
```
YearToDate(timestamp[ , yearoffset [ , firstmonth [ , todaydate] ] ])
```

Tipo di dati restituiti: Boolean



*In Qlik Sense, il valore booleano vero è rappresentato da -1 e il valore falso è rappresentato da 0.*

Schema esemplificativo della funzione yeartodate()



Se non si utilizza alcun parametro opzionale, l'anno rispetto alla data attuale sarà qualsiasi data che rientra in un anno di calendario a partire dal 1° gennaio fino all'ultima data di esecuzione dello script inclusa.

In altre parole, la funzione yeartodate(), quando viene attivata senza parametri aggiuntivi, viene utilizzata per valutare un timestamp e restituire un risultato booleano in base al fatto che la data si sia verificata nell'anno solare fino alla data di ricaricamento inclusa.

Tuttavia, è anche possibile sostituire la data di inizio dell'anno utilizzando l'argomento firstmonth, nonché fare confronti con gli anni precedenti o successivi utilizzando l'argomento yearoffset.

Infine, nei casi di insiemi di dati cronologici, la funzione yeartodate() fornisce un parametro per impostare todaydate, che confronterà invece il timestamp con l'anno solare fino alla data fornita nell'argomento todaydate.

### Argomenti

| Argomento | Descrizione                                        |
|-----------|----------------------------------------------------|
| timestamp | Il timestamp da valutare, ad esempio '12/10/2012'. |

| Argomento  | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| yearoffset | Specificando un <b>yearoffset</b> , <b>yeartodate</b> restituisce True per lo stesso periodo di un altro anno. Un <b>yearoffset</b> negativo indica un anno precedente, mentre un differimento positivo indica un anno successivo. È possibile ottenere l'anno alla data odierna più recente specificando yearoffset = -1. Se omissso, viene utilizzato 0.                                                                |
| firstmonth | Se si specifica un valore <b>firstmonth</b> compreso tra 1 e 12 (1 se omissso), l'inizio dell'anno potrà essere spostato in avanti al primo giorno di qualsiasi mese. Ad esempio, se si vuole lavorare con un anno fiscale che inizia il 1° maggio, specificare <b>firstmonth</b> = 5. Il valore 1 indica un anno fiscale che inizia il 1° gennaio, mentre il valore 12 indica un anno fiscale che inizia il 1° dicembre. |
| todaydate  | Se si specifica un valore <b>todaydate</b> (timestamp dell'ultima esecuzione dello script, se omissso), è possibile spostare il giorno utilizzato come limite superiore del periodo.                                                                                                                                                                                                                                      |

## Casi di utilizzo

La funzione yeartodate() restituisce un risultato booleano. In genere, questo tipo di funzione viene utilizzato come condizione in un'espressione if. Questo restituisce un'aggregazione o un calcolo che dipende dal fatto che la data valutata si sia verificata nell'anno fino all'ultima data di ricaricamento dell'applicazione.

Ad esempio, la funzione YearToDate() può essere utilizzata per identificare tutte le apparecchiature prodotte fino a quel momento nell'anno corrente.

Gli esempi seguenti presuppongono che la data dell'ultimo ricaricamento sia il 18/11/2011.

### Esempi di funzioni

| Esempio                                       | Risultato         |
|-----------------------------------------------|-------------------|
| yeartodate( '11/18/2010')                     | restituisce False |
| yeartodate( '02/01/2011')                     | restituisce True  |
| yeartodate( '11/18/2011')                     | restituisce True  |
| yeartodate( '11/19/2011')                     | restituisce False |
| yeartodate( '11/19/2011', 0, 1, '12/31/2011') | restituisce True  |
| yeartodate( '11/18/2010', -1)                 | restituisce True  |
| yeartodate( '11/18/2011', -1)                 | restituisce False |
| yeartodate( '04/30/2011', 0, 5)               | restituisce False |
| yeartodate( '05/01/2011', 0, 5)               | restituisce True  |

## Impostazioni locali

Se non diversamente specificato, gli esempi di questo argomento utilizzano il seguente formato di data: MM/GG/AAAA. Il formato della data viene specificato nell'istruzione SET DateFormat nello script di caricamento dei dati. La formattazione predefinita della data potrebbe essere diversa nel proprio sistema,

a causa delle impostazioni regionali e di altri fattori. È possibile modificare i formati degli esempi seguenti in base alle proprie esigenze. In alternativa, è possibile modificare i formati nel proprio script di caricamento per adattarli a questi esempi.

Le impostazioni regionali predefinite delle app si basano sulle impostazioni regionali del sistema del computer o del server in cui risulta installato Qlik Sense. Se il server Qlik Sense a cui si accede è impostato in Svezia, l'editor caricamento dati utilizzerà le impostazioni regionali svedesi per date, ora e valuta. Queste impostazioni di formato regionale non riguardano la lingua visualizzata nell'interfaccia utente Qlik Sense. Qlik Sense verrà visualizzato nella stessa lingua del browser utilizzato.

### Esempio 1 - Esempio di base

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni tra il 2020 e il 2022, caricato in una tabella denominata `Transactions`.
- Il campo della data fornito nel formato della variabile di sistema `DateFormat` (MM/GG/AAAA).
- La creazione di un campo, `year_to_date`, che determina quali transazioni sono state effettuate nell'anno solare fino alla data dell'ultimo ricaricamento.

Al momento in cui scriviamo, la data è il 26 aprile 2022.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
  Load
    *,
    yeartodate(date) as year_to_date
  ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
id,date,amount
8188,01/10/2020,37.23
8189,02/28/2020,17.17
8190,04/09/2020,88.27
8191,04/16/2020,57.42
8192,05/21/2020,53.80
8193,08/14/2020,82.06
8194,10/07/2020,40.39
8195,12/05/2020,87.21
```

```
8196,01/22/2021,95.93
8197,02/03/2021,45.89
8198,03/17/2021,36.23
8199,04/23/2021,25.66
8200,05/04/2021,82.77
8201,06/30/2021,69.98
8202,07/26/2021,76.11
8203,12/27/2021,25.12
8204,02/02/2022,46.23
8205,02/26/2022,84.21
8206,03/07/2022,96.24
8207,03/11/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

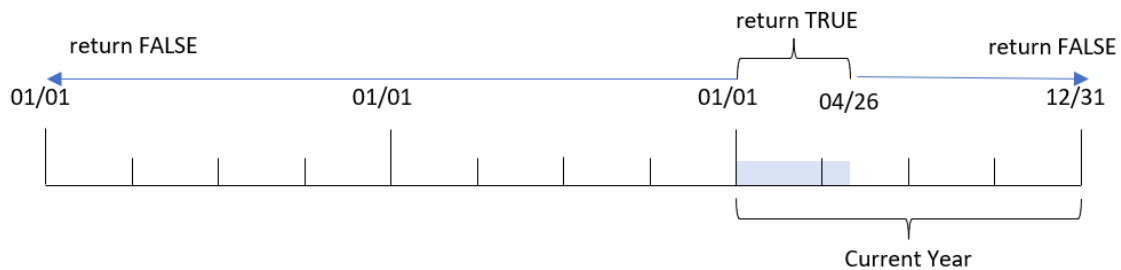
- date
- year\_to\_date

Tabella dei risultati

| date       | anno_alla_data_odierna |
|------------|------------------------|
| 01/10/2020 | 0                      |
| 02/28/2020 | 0                      |
| 04/09/2020 | 0                      |
| 04/16/2020 | 0                      |
| 05/21/2020 | 0                      |
| 08/14/2020 | 0                      |
| 10/07/2020 | 0                      |
| 12/05/2020 | 0                      |
| 01/22/2021 | 0                      |
| 02/03/2021 | 0                      |
| 03/17/2021 | 0                      |
| 04/23/2021 | 0                      |
| 05/04/2021 | 0                      |
| 06/30/2021 | 0                      |
| 07/26/2021 | 0                      |
| 12/27/2021 | 0                      |
| 02/02/2022 | -1                     |

| date       | anno_alla_data_odierna |
|------------|------------------------|
| 02/26/2022 | -1                     |
| 03/07/2022 | -1                     |
| 03/11/2022 | -1                     |

Schema della funzione `yeartodate()`, esempio di base



Il campo `year_to_date` viene creato nell'istruzione di caricamento precedente mediante l'uso della funzione `yeartodate()` e trasferendo il campo `date` come argomento della funzione.

Poiché non vengono passati altri parametri alla funzione, la funzione `yeartodate()` identifica inizialmente la data di ricaricamento e quindi i confini dell'anno solare in corso (a partire dal 1° gennaio) che restituirà un risultato booleano di `TRUE`.

Pertanto, qualsiasi transazione effettuata tra il 1° gennaio e il 26 aprile, data di ricaricamento, restituirà un risultato booleano di `TRUE`. Qualsiasi transazione che si verifichi prima dell'inizio del 2022 restituirà un risultato booleano di `FALSE`.

### Esempio 2 - `yearoffset`

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.
- La creazione di un campo, `two_years_prior`, che determina quali transazioni hanno avuto luogo due anni prima dell'anno solare in corso.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
  Load
```

```
*,
    yeartodate(date,-2) as two_years_prior
;

Load
*
Inline
[
id,date,amount
8188,01/10/2020,37.23
8189,02/28/2020,17.17
8190,04/09/2020,88.27
8191,04/16/2020,57.42
8192,05/21/2020,53.80
8193,08/14/2020,82.06
8194,10/07/2020,40.39
8195,12/05/2020,87.21
8196,01/22/2021,95.93
8197,02/03/2021,45.89
8198,03/17/2021,36.23
8199,04/23/2021,25.66
8200,05/04/2021,82.77
8201,06/30/2021,69.98
8202,07/26/2021,76.11
8203,12/27/2021,25.12
8204,02/02/2022,46.23
8205,02/26/2022,84.21
8206,03/07/2022,96.24
8207,03/11/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- two\_years\_prior

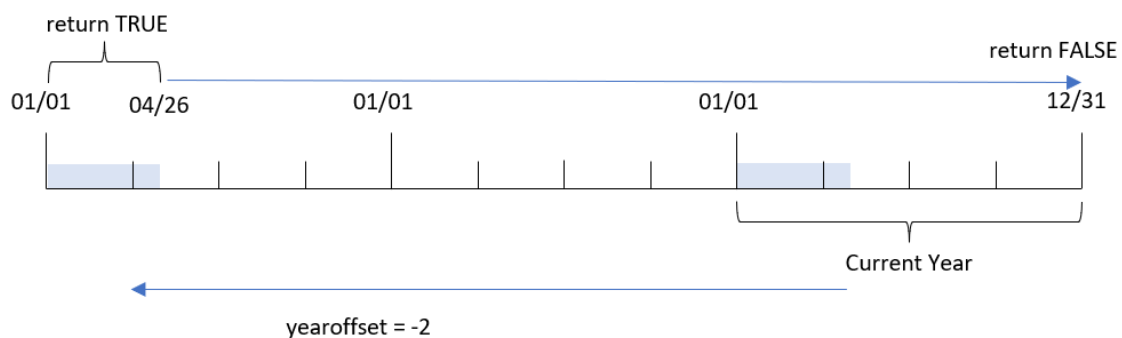
Tabella dei risultati

| date       | due anni prima |
|------------|----------------|
| 01/10/2020 | -1             |
| 02/28/2020 | -1             |
| 04/09/2020 | -1             |
| 04/16/2020 | -1             |
| 05/21/2020 | 0              |
| 08/14/2020 | 0              |
| 10/07/2020 | 0              |

| date       | due anni prima |
|------------|----------------|
| 12/05/2020 | 0              |
| 01/22/2021 | 0              |
| 02/03/2021 | 0              |
| 03/17/2021 | 0              |
| 04/23/2021 | 0              |
| 05/04/2021 | 0              |
| 06/30/2021 | 0              |
| 07/26/2021 | 0              |
| 12/27/2021 | 0              |
| 02/02/2022 | 0              |
| 02/26/2022 | 0              |
| 03/07/2022 | 0              |
| 03/11/2022 | 0              |

Utilizzando -2 come argomento `yearoffset` della funzione `yeartodate()`, questa sposta i confini del segmento dell'anno solare di confronto di ben due anni. Inizialmente, il segmento di anno corrisponde a un periodo compreso tra il 1° gennaio e il 26 aprile 2022. L'argomentazione `yearoffset` compensa poi questo segmento con quello di due anni prima. La data limite sarà quindi compresa tra il 1° gennaio e il 26 aprile 2020.

*Schema della funzione `yeartodate()`, esempio di `yearoffset`*



Pertanto, qualsiasi transazione effettuata tra il 1° gennaio e il 26 aprile 2020 restituirà un risultato booleano di `TRUE`. Tutte le transazioni che appaiono prima o dopo questo segmento restituiranno `FALSE`.

### Esempio 3 - firstmonth

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.
- La creazione di un campo, `year_to_date`, che determina quali transazioni sono state effettuate nell'anno solare fino alla data dell'ultimo ricaricamento.

In questo esempio, abbiamo impostato l'inizio dell'anno fiscale al 1° luglio.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        yeartodate(date,0,7) as year_to_date
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
id,date,amount
8188,01/10/2020,37.23
8189,02/28/2020,17.17
8190,04/09/2020,88.27
8191,04/16/2020,57.42
8192,05/21/2020,53.80
8193,08/14/2020,82.06
8194,10/07/2020,40.39
8195,12/05/2020,87.21
8196,01/22/2021,95.93
8197,02/03/2021,45.89
8198,03/17/2021,36.23
8199,04/23/2021,25.66
8200,05/04/2021,82.77
8201,06/30/2021,69.98
8202,07/26/2021,76.11
8203,12/27/2021,25.12
8204,02/02/2022,46.23
8205,02/26/2022,84.21
8206,03/07/2022,96.24
8207,03/11/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

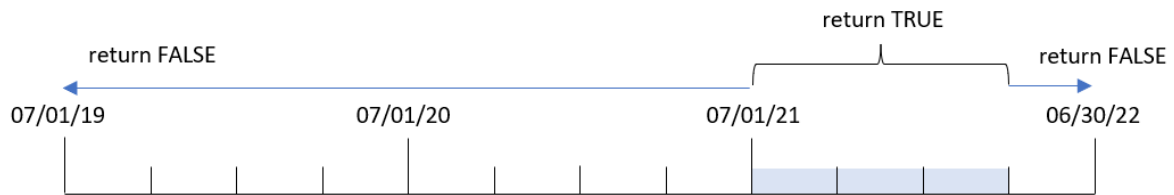
- date
- year\_to\_date

Tabella dei risultati

| date       | anno_alla_data_odierna |
|------------|------------------------|
| 01/10/2020 | 0                      |
| 02/28/2020 | 0                      |
| 04/09/2020 | 0                      |
| 04/16/2020 | 0                      |
| 05/21/2020 | 0                      |
| 08/14/2020 | 0                      |
| 10/07/2020 | 0                      |
| 12/05/2020 | 0                      |
| 01/22/2021 | 0                      |
| 02/03/2021 | 0                      |
| 03/17/2021 | 0                      |
| 04/23/2021 | 0                      |
| 05/04/2021 | 0                      |
| 06/30/2021 | 0                      |
| 07/26/2021 | -1                     |
| 12/27/2021 | -1                     |
| 02/02/2022 | -1                     |
| 02/26/2022 | -1                     |
| 03/07/2022 | -1                     |
| 03/11/2022 | -1                     |

In questa istanza, poiché l'argomento `firstmonth` di 7 viene utilizzato nella funzione `yeartodate()`, imposta il primo giorno dell'anno al 1° luglio e l'ultimo giorno dell'anno al 30 giugno.

Schema della funzione `yeartodate()`, esempio `firstmonth`



Pertanto, qualsiasi transazione effettuata tra il 1° luglio 2021 e il 26 aprile 2022, data di ricaricamento, restituirà un risultato booleano di `TRUE`. Qualsiasi transazione effettuata prima del 1° luglio 2021 restituirà un risultato booleano di `FALSE`.

### Esempio 4 - `todaydate`

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Vengono utilizzati lo stesso set di dati e lo stesso scenario del primo esempio.
- La creazione di un campo, `year_to_date`, che determina quali transazioni sono state effettuate nell'anno solare fino alla data dell'ultimo ricaricamento.

Tuttavia, in questo esempio, dobbiamo identificare tutte le transazioni avvenute nell'anno solare fino al 1° marzo 2022 incluso.

#### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

Transactions:

```
Load
    *,
    yeartodate(date, 0, 1, '03/01/2022') as year_to_date
;
Load
*
Inline
[
id,date,amount
8188,01/10/2020,37.23
8189,02/28/2020,17.17
8190,04/09/2020,88.27
8191,04/16/2020,57.42
8192,05/21/2020,53.80
8193,08/14/2020,82.06
```

```
8194,10/07/2020,40.39
8195,12/05/2020,87.21
8196,01/22/2021,95.93
8197,02/03/2021,45.89
8198,03/17/2021,36.23
8199,04/23/2021,25.66
8200,05/04/2021,82.77
8201,06/30/2021,69.98
8202,07/26/2021,76.11
8203,12/27/2021,25.12
8204,02/02/2022,46.23
8205,02/26/2022,84.21
8206,03/07/2022,96.24
8207,03/11/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- date
- year\_to\_date

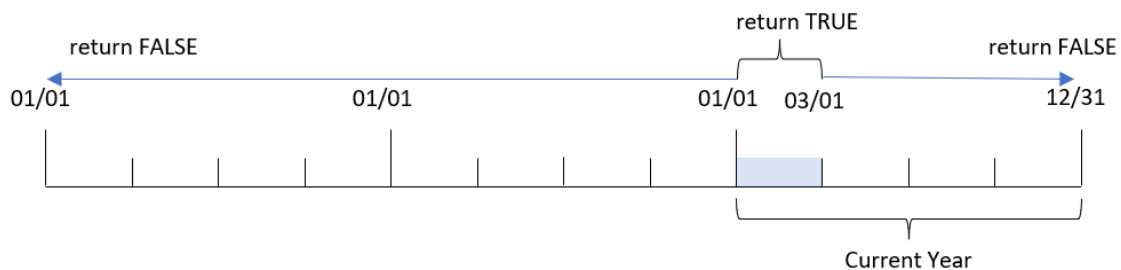
Tabella dei risultati

| date       | anno_alla_data_odierna |
|------------|------------------------|
| 01/10/2020 | 0                      |
| 02/28/2020 | 0                      |
| 04/09/2020 | 0                      |
| 04/16/2020 | 0                      |
| 05/21/2020 | 0                      |
| 08/14/2020 | 0                      |
| 10/07/2020 | 0                      |
| 12/05/2020 | 0                      |
| 01/22/2021 | 0                      |
| 02/03/2021 | 0                      |
| 03/17/2021 | 0                      |
| 04/23/2021 | 0                      |
| 05/04/2021 | 0                      |
| 06/30/2021 | 0                      |
| 07/26/2021 | 0                      |
| 12/27/2021 | 0                      |

| date       | anno_alla_data_odierna |
|------------|------------------------|
| 02/02/2022 | -1                     |
| 02/26/2022 | -1                     |
| 03/07/2022 | 0                      |
| 03/11/2022 | 0                      |

In questo caso, poiché la funzione `yeartodate()` utilizza l'argomento `todaydate` di 01/03/2022, imposta il limite finale del segmento dell'anno solare di confronto al 1° marzo 2022. È fondamentale fornire il parametro `firstmonth` (compreso tra 1 e 12), altrimenti la funzione restituirà risultati null.

*Schema della funzione `yeartodate()`, esempio di utilizzo dell'argomento `todaydate`*



Pertanto, qualsiasi transazione che si verifichi tra il 1° gennaio 2022 e il 1° marzo 2022, con il parametro `todaydate`, restituirà un risultato booleano di `TRUE`. Qualsiasi transazione avvenuta prima del 1° gennaio 2022 o dopo il 1° marzo 2022 restituirà un risultato booleano di `FALSE`.

### Esempio 5 - Esempio di oggetto grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor di caricamento dati e aggiungere lo script di caricamento sotto in una nuova scheda.

Lo script di caricamento contiene lo stesso set di dati e lo stesso scenario del primo esempio.

Tuttavia, in questo esempio, il set di dati invariato viene caricato nell'applicazione. Il calcolo che determina quali transazioni hanno avuto luogo nell'anno solare fino alla data dell'ultimo ricaricamento viene creato come misura in un oggetto grafico dell'applicazione.

#### Script di caricamento

Transactions:

Load

\*

Inline

[

```
id,date,amount
8188,01/10/2020,37.23
8189,02/28/2020,17.17
8190,04/09/2020,88.27
8191,04/16/2020,57.42
8192,05/21/2020,53.80
8193,08/14/2020,82.06
8194,10/07/2020,40.39
8195,12/05/2020,87.21
8196,01/22/2021,95.93
8197,02/03/2021,45.89
8198,03/17/2021,36.23
8199,04/23/2021,25.66
8200,05/04/2021,82.77
8201,06/30/2021,69.98
8202,07/26/2021,76.11
8203,12/27/2021,25.12
8204,02/02/2022,46.23
8205,02/26/2022,84.21
8206,03/07/2022,96.24
8207,03/11/2022,67.67
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione: date.

Aggiungere la misura seguente:

=yeartodate(date)

Tabella dei risultati

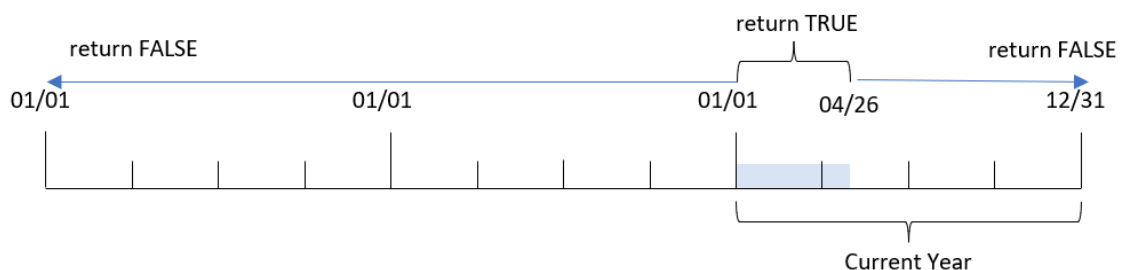
| date       | =annoalladataodierna(date) |
|------------|----------------------------|
| 01/10/2020 | 0                          |
| 02/28/2020 | 0                          |
| 04/09/2020 | 0                          |
| 04/16/2020 | 0                          |
| 05/21/2020 | 0                          |
| 08/14/2020 | 0                          |
| 10/07/2020 | 0                          |
| 12/05/2020 | 0                          |
| 01/22/2021 | 0                          |
| 02/03/2021 | 0                          |

| date       | =annoalladataodierna(date) |
|------------|----------------------------|
| 03/17/2021 | 0                          |
| 04/23/2021 | 0                          |
| 05/04/2021 | 0                          |
| 06/30/2021 | 0                          |
| 07/26/2021 | 0                          |
| 12/27/2021 | 0                          |
| 02/02/2022 | -1                         |
| 02/26/2022 | -1                         |
| 03/07/2022 | -1                         |
| 03/11/2022 | -1                         |

La misura `year_to_date` viene creata nell'oggetto grafico mediante l'utilizzo della funzione `yeartodate()` e trasferendo il campo `date` come argomento della funzione.

Poiché non vengono passati altri parametri alla funzione, la funzione `yeartodate()` identifica inizialmente la data di ricaricamento e quindi i confini dell'anno solare in corso (a partire dal 1° gennaio) che restituirà un risultato booleano di `TRUE`.

*Schema della funzione `yeartodate()`, esempio di utilizzo dell'oggetto grafico*



Qualsiasi transazione effettuata tra il 1° gennaio e il 26 aprile, data di ricaricamento, restituirà un risultato booleano di `TRUE`. Qualsiasi transazione che si verifichi prima dell'inizio del 2022 restituirà un risultato booleano di `FALSE`.

### Esempio 6 - Scenario

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati contenente un insieme di transazioni tra il 2020 e il 2022, caricato in una tabella denominata `Transactions`.
- Il campo della data fornito nel formato della variabile di sistema `DateFormat` (MM/GG/AAAA).

L'utente finale desidera un oggetto KPI che presenti le vendite totali per il periodo equivalente nel 2021 come l'anno corrente fino ad oggi, all'ultimo orario di ricaricamento.

Al momento in cui scriviamo, la data è il 16 giugno 2022.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,01/10/2020,37.23
```

```
8189,02/28/2020,17.17
```

```
8190,04/09/2020,88.27
```

```
8191,04/16/2020,57.42
```

```
8192,05/21/2020,53.80
```

```
8193,08/14/2020,82.06
```

```
8194,10/07/2020,40.39
```

```
8195,12/05/2020,87.21
```

```
8196,01/22/2021,95.93
```

```
8197,02/03/2021,45.89
```

```
8198,03/17/2021,36.23
```

```
8199,04/23/2021,25.66
```

```
8200,05/04/2021,82.77
```

```
8201,06/30/2021,69.98
```

```
8202,07/26/2021,76.11
```

```
8203,12/27/2021,25.12
```

```
8204,02/02/2022,46.23
```

```
8205,02/26/2022,84.21
```

```
8206,03/07/2022,96.24
```

```
8207,03/11/2022,67.67
```

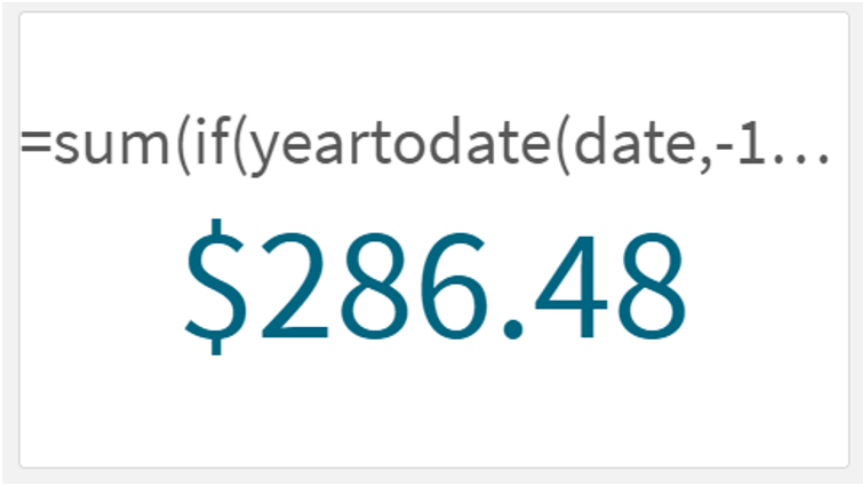
```
];
```

### Risultati

**Procedere come indicato di seguito:**

1. Creare un oggetto KPI.
2. Creare la seguente misura di aggregazione per calcolare le vendite totali:  
`=sum(if(yeartodate(date,-1),amount,0))`
3. Impostare la misura **Formattazione numero** su **Denaro**.

Grafico `yeartodate()` KPI per il 2021



=sum(if(yeartodate(date,-1...  
**\$286.48**

La funzione `yeartodate()` restituisce un valore booleano quando si valutano le date di ciascun ID transazione. Poiché il ricaricamento è avvenuto il 16 giugno 2022, la funzione `yeartodate` segmenta il periodo dell'anno tra il 01/01/2022 e il 16/06/2022. Tuttavia, poiché nella funzione `period_no` è stato utilizzato il valore -1, questi confini vengono spostati all'anno precedente. Pertanto, per qualsiasi transazione che si verifichi tra il 01/01/2021 e il 06/16/2021, la funzione `yeartodate()` restituisce un valore booleano di `TRUE` e somma l'importo.

### 8.8 Funzioni esponenziali e logaritmiche

In questa sezione vengono descritte le funzioni correlate ai calcoli esponenziali e logaritmici. Tutte le funzioni possono essere utilizzate sia nello script di caricamento dei dati che nelle espressioni del grafico.

Nelle seguenti funzioni i parametri sono espressioni dove **x** e **y** devono essere interpretati come numeri reali valutati.

#### **exp**

La funzione esponenziale naturale,  $e^x$ , che utilizza l'algoritmo naturale **e** come base. Il risultato è un numero positivo.

**exp** ( *x* )

#### **Esempi e risultati:**

`exp(3)` restituisce 20,085.

#### **log**

Il logaritmo naturale di **x**. La funzione viene definita solo se  $x > 0$ . Il risultato è un numero.

**log** ( *x* )

### Esempi e risultati:

`log(3)` restituisce 1,0986

### **log10**

Il logaritmo comune (base 10) di **x**. La funzione viene definita solo se **x** > 0. Il risultato è un numero.

```
log10 ( x )
```

### Esempi e risultati:

`log10(3)` restituisce 0,4771

### **pow**

Restituisce **x** alla potenza di **y**. Il risultato è un numero.

```
pow ( x, y )
```

### Esempi e risultati:

`pow(3, 3)` restituisce 27

### **sqr**

**x** quadrato (**x** alla potenza di 2). Il risultato è un numero.

```
sqr ( x )
```

### Esempi e risultati:

`sqr(3)` restituisce 9

### **sqrt**

Radice quadrata di **x**. La funzione viene definita solo se **x** >= 0. Il risultato è un numero positivo.

```
sqrt ( x )
```

### Esempi e risultati:

`sqrt(3)` restituisce 1,732

## 8.9 Funzioni di campo

Queste funzioni possono essere utilizzate solo nelle espressioni grafiche.

Le funzioni di campo restituiscono numeri interi o stringhe che identificano aspetti differenti delle selezioni dei campi.

## Funzioni di conteggio

### GetAlternativeCount

**GetAlternativeCount()** viene utilizzato per trovare il numero di valori alternativi (grigio chiaro) nel campo identificato.

**GetAlternativeCount - funzione per grafici** (field\_name)

### GetExcludedCount

**GetExcludedCount()** trova il numero di valori esclusi nel campo identificato. Sono conteggiati solo i campi esclusi (di colore grigio scuro). I valori alternativi (di colore grigio chiaro) e quelli esclusi selezionati (in grigio scuro con un segno di spunta) non vengono conteggiati.

**GetExcludedCount - funzione per grafici** (field\_name)

### GetPossibleCount

**GetPossibleCount()** viene utilizzato per trovare il numero di valori possibili nel campo identificato. Se il campo identificato include selezioni, i campi selezionati (verdi) vengono conteggiati. In caso contrario, vengono conteggiati i valori associati (bianco).

**GetPossibleCount - funzione per grafici** (field\_name)

### GetSelectedCount

**GetSelectedCount()** trova il numero di valori selezionati (verdi) in un campo.

**GetSelectedCount - funzione per grafici** (field\_name [, include\_excluded])

### GetStateCounts

La funzione grafico **GetStateCounts()** viene utilizzata per calcolare il numero totale di valori unici corrispondenti agli stati di selezione specificati.

**GetStateCounts - funzione per grafici** (field\_name, state\_name [, state\_type1,...state\_typeN])

## Funzioni di campo e di selezione

### GetCurrentSelections

**GetCurrentSelections()** restituisce un elenco delle selezioni attuali all'interno dell'app. Se invece le selezioni vengono effettuate usando una stringa di ricerca all'interno di una casella di ricerca, **GetCurrentSelections()** restituisce la stringa di ricerca.

**GetCurrentSelections - funzione per grafici** ([record\_sep [, tag\_sep [, value\_sep [, max\_values]]]])

### GetFieldSelections

**GetFieldSelections()** restituisce una **stringa** con le selezioni attuali in un campo.

**GetFieldSelections - funzione per grafici** ( field\_name [, value\_sep [, max\_values]])

### GetObjectDimension

**GetObjectDimension()** restituisce il nome della dimensione. **Index** è un numero intero facoltativo che indica la dimensione da restituire.

[GetObjectDimension - funzione per grafici](#) ([index])

### GetObjectField

**GetObjectField()** restituisce l'espressione di campo della dimensione. **Index** è un numero intero opzionale che indica la dimensione che deve essere restituita.

[GetObjectField - funzione per grafici](#) ([index])

### GetObjectMeasure

**GetObjectMeasure()** restituisce il nome della misura. **Index** è un numero intero opzionale che indica la misura da restituire.

[GetObjectMeasure - funzione per grafici](#) ([index])

## GetAlternativeCount - funzione per grafici

**GetAlternativeCount()** viene utilizzato per trovare il numero di valori alternativi (grigio chiaro) nel campo identificato.

### Sintassi:

**GetAlternativeCount** (field\_name)

**Tipo di dati restituiti:** numero intero



*I colori utilizzati nella barra di selezione e per ogni stato di selezione possono essere modificati tramite un tema personalizzato. Se si utilizza un'app che utilizza un tema personalizzato, si potrebbe notare che le selezioni non vengono visualizzate con gli stessi colori descritti in questo argomento della guida.*

### Argomenti

| Argomento  | Descrizione                                       |
|------------|---------------------------------------------------|
| field_name | Il campo contenente la scala di dati da misurare. |

La tabella seguente elenca altre funzioni correlate a questa funzione.

## Funzioni correlate

| Funzione                                                                      | Interazione                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#"><u>GetStateCounts - funzione per grafici (page 1343)</u></a>      | Utilizzando <b>GetStateCounts()</b> , è possibile combinare il calcolo dei seguenti conteggi con un'unica chiamata di funzione: <ul style="list-style-type: none"> <li>• Conteggio dei valori selezionati inclusi.</li> <li>• Conteggio dei valori possibili.</li> <li>• Conteggio dei valori alternativi.</li> <li>• Conteggio dei valori esclusi, senza includere i valori esclusi alternativi e selezionati.</li> <li>• Conteggio dei valori selezionati esclusi.</li> </ul> |
| <a href="#"><u>GetSelectedCount - funzione per grafici (page 1338)</u></a>    | Restituisce il conteggio dei valori selezionati inclusi.                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <a href="#"><u>GetPossibleCount - funzione per grafici (page 1334)</u></a>    | Restituisce il conteggio dei valori possibili.                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <a href="#"><u>GetAlternativeCount - funzione per grafici (page 1304)</u></a> | Restituisce il conteggio dei valori esclusi, senza includere i valori esclusi alternativi e selezionati.                                                                                                                                                                                                                                                                                                                                                                        |

## Esempio: espressioni del grafico

| Esempio                            | Risultato                                                                                                                                                                                                                                |
|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GetAlternativeCount ([First name]) | Restituisce 0 se non è stata effettuata alcuna selezione per il campo First Name. Se sono state effettuate delle selezioni, il valore restituito è il numero totale di valori distinti disponibili meno il numero di valori selezionati. |

## Esempio: principi fondamentali della funzione GetAlternativeCount

Espressione del grafico

## Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `FirstName`
  - `LastName`
  - `Initials`
  - `HasCellphone`

### Script di caricamento

Example:

```
LOAD * inline [
FirstName|LastName|Initials|HasCellphone
John|Anderson|JA|Yes
Sue|Brown|SB|Yes
Mark|Carr|MC|No
Peter|Devonshire|PD|No
Jane|Elliot|JE|Yes
Peter|Franc|PF|Yes ] (delimiter is '|');
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova casella di filtro e aggiungere questo campo come dimensione:

- `FirstName`

Aggiungere un KPI, quindi creare la seguente misura:

- `=GetAlternativeCount (FirstName)`, per trovare il numero di valori alternativi nel campo `FirstName`.

Il KPI con la funzione `GetAlternativeCount` appare con il valore 0. Se si seleziona `Jane` dalla casella di filtro `FirstName`, il KPI restituisce il valore 4 perché vi sono quattro valori univoci ed esclusi (grigi). Ora selezionare `Mark`, in modo da selezionare sia `Jane` che `Mark`. Il valore KPI per la misura `GetAlternativeCount` restituisce 3 perché vi sono 3 valori univoci ed esclusi, `John`, `Peter` e `Sue`.

*I risultati KPI mostrano 3 valori univoci ed esclusi dopo aver selezionato Jane e Mark dalla casella di filtro FirstName.*

🔍 **FirstName**

|       |   |
|-------|---|
| Jane  | ✓ |
| Mark  | ✓ |
| John  |   |
| Peter |   |
| Sue   |   |

`=GetAlternativeCount (FirstName)`

3

È possibile vedere come i valori restituiti dalla funzione `GetAlternativeCount` cambiano man mano che vengono effettuate altre selezioni.

### Esempio: scenario per l'applicazione della funzione GetAlternativeCount

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata CustomerTypeData.
- I seguenti campi nella tabella dati:
  - CustomerID
  - CustomerType

#### Script di caricamento

```
CustomerTypeData:
LOAD * INLINE [
CustomerID, CustomerType
C001, Retail
C002, Wholesale
C003, Distributor
C004, Retail
C005, Wholesale
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova casella di filtro e aggiungere questo campo come dimensione:

- CustomerType

Aggiungere un KPI, quindi creare la seguente misura:

- =GetAlternativeCount (CustomerType), per trovare il numero di valori alternativi nel campo CustomerType.  
Il KPI con la funzione GetAlternativeCount appare con il valore 0 perché non è stato selezionato alcun valore nel campo CustomerType.

Selezionare Retail dalla casella di filtro CustomerType. GetAlternativeCount restituisce il valore 2 perché nel campo CustomerType vi sono due valori univoci ed esclusi (grigio chiaro).

I risultati del KPI che mostrano 2 valori univoci ed esclusi dopo aver selezionato Retail dalla casella di filtro CustomerType.



I risultati mostrano che, sebbene vi siano più valori, la funzione `GetAlternativeCount` visualizza solo il conteggio univoco degli elementi che non sono selezionati.

### Esempio: scenario per l'applicazione della funzione `GetAlternativeCount` con i valori disponibili

Espressione del grafico

#### Panoramica

Questo esempio si basa sullo scenario precedente. Il set di dati include una seconda tabella di dati per dimostrare come le selezioni in ogni campo cambino i risultati restituiti dalla funzione `GetAlternativeCount`.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato nelle 2 tabelle dati inline seguenti e nei campi corrispondenti:
  - CustomerTypeData: include i campi CustomerID e CustomerType.
  - CustomerSegment: include i campi CustomerType e Segment.

#### Script di caricamento

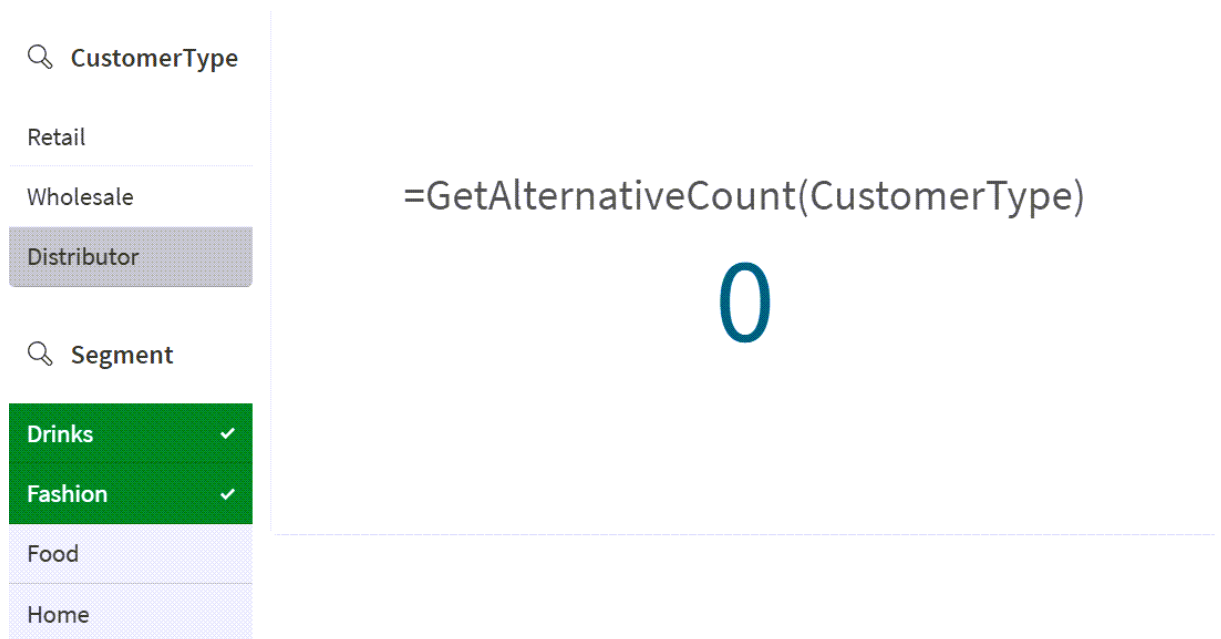
```
CustomerTypeData:
LOAD * INLINE [
CustomerID, CustomerType
C001, Retail
C002, Wholesale
C003, Distributor
C004, Retail
C005, Wholesale
];
```

```
CustomerSegment:
Load * Inline [
CustomerType, Segment
Retail, Fashion
Retail, Home
Wholesale, Fashion
Wholesale, Drinks
Distributor, Food
];
```

### Risultati

1. Caricare i dati e aprire un foglio. Creare una nuova casella di filtro e aggiungere questi campi come dimensioni:
  - CustomerType
  - Segment
2. Aggiungere un KPI, quindi creare la seguente misura:
  - =GetAlternativeCount (CustomerType), per trovare il numero di valori alternativi nel campo CustomerType.  
Il KPI con la funzione GetAlternativeCount appare con il valore 0 perché non è stato selezionato alcun valore per i campi CustomerType o Segment.
3. Selezionare Drinks e Fashion dalla casella di filtro segment. Il KPI continua a restituire 0 perché non è stato selezionato alcun valore nel campo CustomerType.

*I risultati del KPI mostrano zero valori univoci ed esclusi dalle selezioni nella casella di filtro Segment.*



4. Selezionare Retail nel campo CustomerType e mantenere le selezioni nel filtro segment. La funzione GetAlternativeCount ora restituisce il valore 1 perché nel campo CustomerType c'è solo un valore univoco ed escluso (grigio chiaro), wholesale.

I risultati del KPI mostrano 1 valore univoco ed escluso dalle selezioni nelle caselle di filtro *CustomerType* e *Segment*.

=GetAlternativeCount(CustomerType)

1

Sebbene due valori, *wholesale* e *distributor*, non siano selezionati nel campo *CustomerType* (questi appaiono in grigio chiaro e grigio scuro), solo un valore, *wholesale*, è *disponibile* per la selezione in *CustomerType*, a causa delle selezioni nel filtro *Segment*.

### GetCurrentSelections - funzione per grafici

**GetCurrentSelections()** restituisce un elenco delle selezioni attuali all'interno dell'app. Se invece le selezioni vengono effettuate usando una stringa di ricerca all'interno di una casella di ricerca, **GetCurrentSelections()** restituisce la stringa di ricerca.

Se vengono utilizzate opzioni, è necessario specificare `record_sep`. Per specificare una nuova riga, impostare `record_sep` su `chr(13)&chr(10)`.

Se vengono selezionati tutti i valori meno due o tutti i valori meno uno, verrà utilizzato rispettivamente il formato "NOT x,y" o "NOT y". Se si selezionano tutti i valori e il conteggio dei valori è superiore a `max_values`, verrà restituito il testo ALL.

#### Sintassi:

```
GetCurrentSelections ([record_sep [, tag_sep [, value_sep [, max_values [, state_name]]]])
```

**Tipo di dati restituiti:** stringa

### Argomenti

| Argomenti         | Descrizione                                                                                                                                                                                                                          |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>record_sep</b> | Il separatore da inserire tra i record del campo. Il valore predefinito è <CR><LF> che significa una nuova linea.                                                                                                                    |
| <b>tag_sep</b>    | Il separatore da inserire tra il tag del nome di campo e i valori del campo. Il valore predefinito è ": ".                                                                                                                           |
| <b>value_sep</b>  | Il separatore da inserire tra i valori di campo. Il valore predefinito è ', '.                                                                                                                                                       |
| <b>max_values</b> | Il numero massimo di valori di campo da elencare singolarmente. Se si seleziona un numero maggiore di valori, verrà utilizzato il formato 'x di y valori'. Il valore predefinito è 6.                                                |
| <b>state_name</b> | Il nome di uno stato alternato che è stato scelto per la specifica visualizzazione. Se viene utilizzato l'argomento <b>state_name</b> , saranno prese in considerazione solo le selezioni associate al nome dello stato specificato. |

### Esempio: espressioni dei grafico

| Esempio                     | Risultato                                                                                                                                                                                                                       |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GetCurrentSelections<br>( ) | Restituisce (-) se non è stata effettuata alcuna selezione per nessuno dei campi. Se sono state effettuate delle selezioni, i valori restituiti sono il nome del campo e il valore, ad esempio <code>First Name : John</code> . |

## Esempio: principi fondamentali della funzione GetCurrentSelections

Espressione del grafico

### Panoramica

Questo esempio mostra come i valori restituiti dalla funzione `GetCurrentSelections` cambiano in base alle selezioni effettuate.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `FirstName`
  - `LastName`
  - `Initials`
  - `HasCellphone`

### Script di caricamento

Example:

```
LOAD * inline [
FirstName|LastName|Initials|HasCellphone
John|Anderson|JA|Yes
Sue|Brown|SB|Yes
Mark|Carr|MC|No
Peter|Devonshire|PD|No
Jane|Elliot|JE|Yes
Peter|Franc|PF|Yes ] (delimiter is '|');
```

### Risultati

1. Caricare i dati e aprire un foglio. Creare una nuova casella di filtro e aggiungere questi campi come dimensioni:
  - `FirstName`
  - `Initials`
2. Aggiungere un oggetto KPI, quindi creare la seguente misura:
  - `=GetCurrentSelections ()`, per visualizzare le selezioni correnti.Il KPI restituisce (-) perché non sono stati selezionati valori.



*In questo esempio, la funzionalità di applicazione di stili ai KPI è stata regolata in modo da utilizzare un comportamento di presentazione reattivo, per semplificare la visualizzazione dell'output. .*

3. Fare clic su **Modifica foglio** per passare alla modalità di analisi quindi, nel campo `FirstName`, selezionare John. Il KPI restituisce il seguente valore: `FirstName: John`.
  - Ora selezionare Peter, in modo da selezionare sia John che Peter. Il KPI restituisce il seguente valore: `FirstName: John, Peter`.
  - Aggiungere un valore alla selezione selezionando JA dal campo `Initials`. Il KPI restituisce il seguente valore: `First name: John, Peter Initials: JA`.

| Q FirstName | Q Initials |
|-------------|------------|
| John ✓      | JA ✓       |
| Peter ✓     | PD         |
| Jane        | PF         |
| Mark        | JE         |
| Sue         | MC         |
|             | SB         |

`=GetCurrentSelections()`

**FirstName: John, Peter Initials: JA**

4. Annullare la selezione ed eseguire una delle seguenti operazioni:
  - Selezionare John dal campo `FirstName` e JA dal campo `Initials`.
  - Modificare l'espressione KPI come segue: `=GetCurrentSelections ( chr(13)&chr(10) , ' = ' )` per cambiare il separatore di tag nel segno uguale (=).



*Il KPI non supporta le interruzioni di riga e aggiunge uno spazio come separatore dei record.*

5. Annullare la selezione ed eseguire una delle seguenti operazioni:

- Selezionare tutti i nomi tranne sue nel campo `FirstName`.
- Modificare l'espressione KPI come segue: `=GetCurrentSelections (chr(13)&chr(10), '=', ' ', 3)` per aggiungere un separatore di tag (=), un separatore di valori (,) e il numero massimo di valori da elencare (3).

| Q FirstName | Q Initials |
|-------------|------------|
| Jane ✓      | JA         |
| John ✓      | JE         |
| Mark ✓      | MC         |
| Peter ✓     | PD         |
| Sue         | PF         |
|             | SB         |

`=GetCurrentSelections(chr(13)&chr(10), '=', ' ', 3)`  
**FirstName=NOT Sue**

Il KPI restituisce il valore `FirstName=NOT sue`, perché sono stati selezionati quattro nomi, ma l'argomento `max_values` ne può elencare solo tre. Se si cambia l'argomento `max_values` in 4, tutti e quattro i valori saranno elencati separatamente con la stessa selezione.

| Q FirstName | Q Initials |
|-------------|------------|
| Jane ✓      | JA         |
| John ✓      | JE         |
| Mark ✓      | MC         |
| Peter ✓     | PD         |
| Sue         | PF         |
|             | SB         |

`=GetCurrentSelections(chr(13)&chr(10), '=', ' ', 4)`  
**FirstName=Jane,John,Mark,Peter**

### Esempio: personalizzare il titolo di un grafico per mostrare le selezioni

Espressione del grafico

#### Panoramica

Questo esempio mostra come utilizzare la funzione `GetCurrentSelections` per creare titoli di grafici dinamici in base alla selezione dell'utente.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:

- SalesID
- Product
- Year
- Region
- SalesAmount

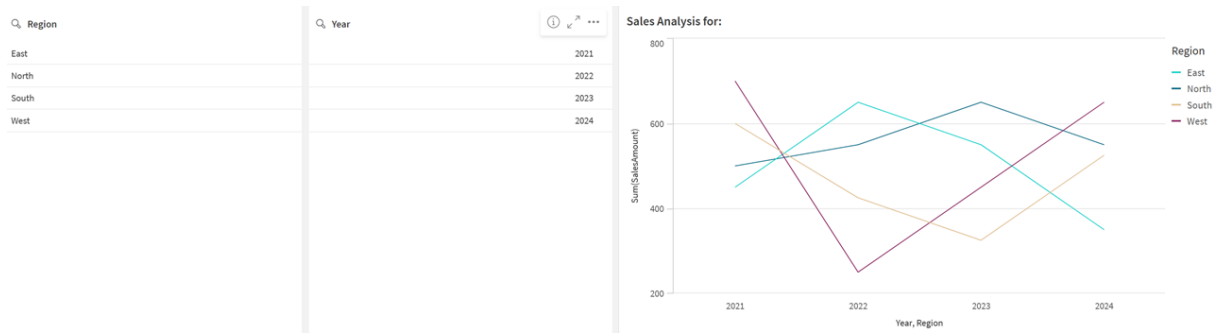
### Script di caricamento

Example:

```
LOAD * INLINE [  
SalesID, Product, Year, Region, SalesAmount  
1, widgetA, 2021, North, 500  
2, widgetB, 2021, South, 600  
3, widgetC, 2021, East, 450  
4, widgetA, 2021, West, 700  
5, widgetB, 2022, North, 550  
6, widgetC, 2022, South, 425  
7, widgetB, 2022, East, 650  
8, widgetC, 2022, West, 250  
9, widgetB, 2023, North, 650  
10, widgetC, 2023, South, 325  
11, widgetB, 2023, East, 550  
12, widgetC, 2023, West, 450  
13, widgetB, 2024, North, 550  
14, widgetC, 2024, South, 525  
15, widgetB, 2024, East, 350  
16, widgetC, 2024, West, 650  
];
```

### Risultati

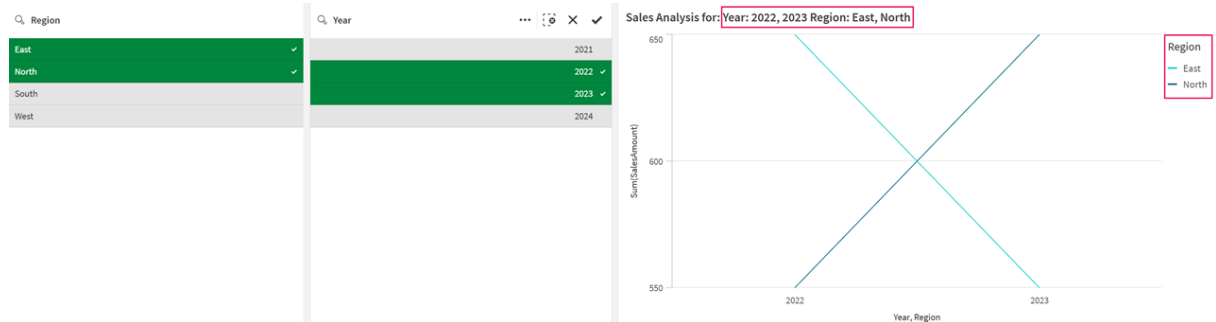
1. Caricare i dati e aprire un foglio. Creare una nuova casella di filtro e aggiungere questi campi come dimensioni:
  - Region
  - Year
2. Creare un grafico lineare e aggiungere le seguenti dimensioni:
  - Year
  - Region
3. Creare la seguente misura:
  - =Sum(SalesAmount), per visualizzare il totale delle vendite.
4. Nel pannello delle proprietà della misura, in **Aspetto > Generale > Titolo**, inserire la seguente espressione: ='Sales Analysis for: ' & GetCurrentSelections().



I risultati mostrano la somma delle vendite per tutti gli anni e le regioni. Il titolo del grafico appare come Sales Analysis for:.

5. Fare clic su **Modifica foglio** per passare alla modalità di analisi, quindi effettuare le seguenti selezioni:

- Nel campo **Regione** selezionare East e North.
- Nel campo **Anno** selezionare 2022 e 2023.



Il titolo del grafico cambia per mostrare le selezioni del filtro per Region e Year.

### GetExcludedCount - funzione per grafici

**GetExcludedCount()** trova il numero di valori esclusi nel campo identificato. Sono conteggiati solo i campi esclusi (di colore grigio scuro). I valori alternativi (di colore grigio chiaro) e quelli esclusi selezionati (in grigio scuro con un segno di spunta) non vengono conteggiati.

**Sintassi:**

```
GetExcludedCount (field_name)
```

**Tipo di dati restituiti:** numero intero



*I colori utilizzati nella barra di selezione e per ogni stato di selezione possono essere modificati tramite un tema personalizzato. Se si utilizza un'app che utilizza un tema personalizzato, si potrebbe notare che le selezioni non vengono visualizzate con gli stessi colori descritti in questo argomento della guida.*

## Argomenti

| Argomenti               | Descrizione                                       |
|-------------------------|---------------------------------------------------|
| <code>field_name</code> | Il campo contenente la scala di dati da misurare. |

La tabella seguente elenca altre funzioni correlate a questa funzione.

## Funzioni correlate

| Funzione                                                                             | Interazione                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|--------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#"><u><i>GetStateCounts - funzione per grafici (page 1343)</i></u></a>      | Utilizzando <b>GetStateCounts()</b> , è possibile combinare il calcolo dei seguenti conteggi con un'unica chiamata di funzione: <ul style="list-style-type: none"> <li>• Conteggio dei valori selezionati inclusi.</li> <li>• Conteggio dei valori possibili.</li> <li>• Conteggio dei valori alternativi.</li> <li>• Conteggio dei valori esclusi, senza includere i valori esclusi alternativi e selezionati.</li> <li>• Conteggio dei valori selezionati esclusi.</li> </ul> |
| <a href="#"><u><i>GetSelectedCount - funzione per grafici (page 1338)</i></u></a>    | Restituisce il conteggio dei valori selezionati inclusi.                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <a href="#"><u><i>GetPossibleCount - funzione per grafici (page 1334)</i></u></a>    | Restituisce il conteggio dei valori possibili.                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <a href="#"><u><i>GetAlternativeCount - funzione per grafici (page 1304)</i></u></a> | Restituisce il conteggio dei valori alternativi.                                                                                                                                                                                                                                                                                                                                                                                                                                |

## Esempio: espressioni dei grafico

| Esempio                                 | Risultato                                                                                                                                                                                                          |
|-----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>GetExcludedCount(Initials)</code> | Restituisce 0 se non è stata effettuata alcuna selezione. Se è stata effettuata una selezione, il valore restituito è il numero di valori esclusi in <code>Initials</code> , visualizzati con colore grigio scuro. |

## Esempio: principi fondamentali della funzione GetExcludedCount

Espressione del grafico

## Panoramica

Questo esempio mostra come i valori che vengono restituiti dalla funzione `GetExcludedCount` cambiano man mano che vengono effettuate altre selezioni.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `FirstName`
  - `LastName`
  - `Initials`
  - `HasCellphone`

### Script di caricamento

Example:

```
LOAD * inline [
FirstName|LastName|Initials|HasCellphone
John|Anderson|JA|Yes
Sue|Brown|SB|Yes
Mark|Carr|MC|No
Peter|Devonshire|PD|No
Jane|Elliot|JE|Yes
Peter|Franc|PF|Yes ] (delimiter is '|');
```

### Risultati

1. Caricare i dati e aprire un foglio. Creare una nuova casella di filtro e aggiungere questi campi come dimensioni:
  - `FirstName`
  - `LastName`
  - `Initials`
2. Creare tre oggetti KPI e aggiungere le seguenti espressioni come misure, una per ogni oggetto KPI:
  - a. `=GetExcludedCount(FirstName)`, per contare il numero di valori esclusi nel campo `FirstName`.
  - b. `=GetExcludedCount(Initials)`, per contare il numero di valori esclusi nel campo `Initials`.
  - c. `=GetExcludedCount(LastName)`, per contare il numero di valori esclusi nel campo `LastName`.

Il foglio apparirà come l'immagine seguente, perché non è stato selezionato alcun valore. La funzione `GetExcludedCount` restituisce 0 per ogni KPI.

| Q1: FirstName | Q1: Initials | Q1: LastName | =GetExcludedCount (FirstN... | =GetExcludedCount (Initials) | =GetExcludedCount (LastN... |
|---------------|--------------|--------------|------------------------------|------------------------------|-----------------------------|
| Jane          | JA           | Anderson     | 0                            | 0                            | 0                           |
| John          | JE           | Brown        |                              |                              |                             |
| Mark          | MC           | Carr         |                              |                              |                             |
| Peter         | PD           | Devonshire   |                              |                              |                             |
| Sue           | PF           | Elliot       |                              |                              |                             |
|               | SB           | Franc        |                              |                              |                             |

3. Fare clic su **Modifica foglio** per passare alla modalità di analisi, quindi effettuare le seguenti selezioni in questo ordine:

Filtrare le selezioni e i risultati in modalità di analisi

### Selezione della casella di filtro

### Risultati KPI

Selezionare John nel filtro  
FirstName.

- `GetExcludedCount (Initials)` restituisce il valore 5 perché sono presenti 5 valori esclusi in `Initials` con colore grigio scuro. Il valore JA rimane in bianco, in quanto è associato alla selezione John in `FirstName`.
- `GetExcludedCount (LastName)` restituisce il valore 5 perché sono presenti 5 valori esclusi in `LastName` con colore grigio scuro. Il valore Anderson rimane in bianco, in quanto è associato alla selezione John in `FirstName`.

Selezionare John e Peter nel filtro  
FirstName.

- `GetExcludedCount (Initials)` restituisce il valore 3 perché sono presenti 3 valori esclusi in `Initials` con colore grigio scuro.
- `GetExcludedCount (LastName)` restituisce il valore 3 perché sono presenti 3 valori esclusi in `LastName` con colore grigio scuro.

Dopo aver selezionato John e Peter, selezionare Franc nel filtro  
LastName.

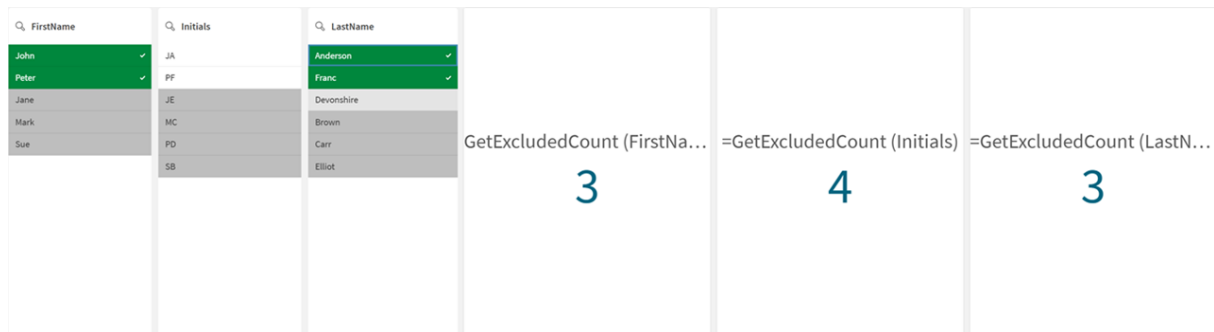
- `GetExcludedCount (FirstName)` restituisce il valore 3 perché vi sono 3 valori esclusi in `FirstName` con colore grigio scuro e nessun segno di selezione. La funzione `The GetExcludedCount` conta solo i valori esclusi. I valori alternativi e quelli esclusi selezionati non sono inclusi nel conteggio. John è un valore escluso selezionato.
- `GetExcludedCount (Initials)` restituisce il valore 5 perché sono presenti 5 valori esclusi in `Initials` con colore grigio scuro.
- `GetExcludedCount (LastName)` restituisce il valore 3 perché sono presenti 3 valori esclusi in `LastName` con colore grigio scuro. I 2 valori di colore grigio chiaro, Anderson e Devonshire, sono valori alternativi.

Mantenere le selezioni precedenti e poi, nel filtro  
LastName,  
selezionare  
Anderson.

- `GetExcludedCount (FirstName)` restituisce il valore 3 perché sono presenti 3 valori esclusi in `FirstName` con colore grigio scuro.
- `GetExcludedCount (Initials)` restituisce il valore 4 perché sono presenti 4 valori esclusi in `Initials` con colore grigio scuro.
- `GetExcludedCount (LastName)` restituisce il valore 3 perché sono presenti 3 valori esclusi in `LastName` con colore grigio scuro.

È possibile vedere i risultati finali delle selezioni della casella di filtro nell'immagine seguente.

Risultati del KPI per la selezione dei nomi, John e Peter, e dei cognomi, Franc e Anderson.



### Esempio: scenario per l'applicazione della funzione GetExcludedCount

Espressione del grafico

#### Panoramica

Un direttore vendite utilizza la funzione `GetExcludedCount` per creare un grafico KPI per analizzare i prodotti venduti da ogni addetto alle vendite.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Salesperson`
  - `Product`
  - `SalesAmount`

#### Script di caricamento

Example:

```
LOAD * INLINE [  
Salesperson, Product, SalesAmount  
John, Laptops, 1000  
Alice, Smartphones, 800  
Bob, Laptops, 1200  
Carol, Tablets, 700  
Dave, Tablets, 900  
];
```

#### Risultati

1. Caricare i dati e aprire un foglio. Creare una nuova casella di filtro e aggiungere questi campi come dimensioni:
  - `Salesperson`

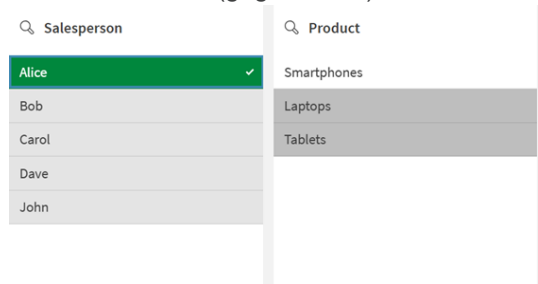
- Product

2. Creare un nuovo oggetto KPI, quindi aggiungere la seguente misura:

- `=GetExcludedCount(Product)`, per visualizzare il numero di valori esclusi nel campo Product.

3. Fare clic su **Modifica foglio** per passare alla modalità di analisi, quindi selezionare Alice dal filtro Salesperson.

La funzione `GetExcludedCount` restituisce il valore 2 perché nel campo Product vi sono 2 valori univoci ed esclusi (grigio chiaro).



`=GetExcludedCount (Product)`

2

I risultati mostrano come è possibile utilizzare la funzione `GetExcludedCount` per analizzare i dati. L'addetta alle vendite Alice non ha venduto alcun articolo in 2 dei 3 gruppi di prodotti, i prodotti Laptop e Tablet.

### GetFieldSelections - funzione per grafici

`GetFieldSelections()` restituisce una **stringa** con le selezioni attuali in un campo.

Se si selezionano tutti i valori meno due o tutti i valori meno uno, verrà utilizzato rispettivamente il formato 'NOT x,y' o il formato 'NOT y'. Se si selezionano tutti i valori e il conteggio dei valori è superiore a `max_values`, verrà restituito il testo ALL.

**Sintassi:**

```
GetFieldSelections ( field_name [, value_sep [, max_values [, state_name]]])
```

**Tipo di dati restituiti:** stringa

Restituisci formati stringa

| Formato       | Descrizione                                                                                                                                                                                                                               |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 'a, b, c'     | Se il numero dei valori selezionati è <code>max_values</code> o meno, la stringa restituita è un elenco dei valori selezionati.<br><br>I valori sono separati con <code>value_sep</code> come delimitatore.                               |
| 'NOT a, b, c' | Se il numero dei valori non selezionati è <code>max_values</code> o meno, la stringa restituita è un elenco dei valori non selezionati con NOT come prefisso.<br><br>I valori sono separati con <code>value_sep</code> come delimitatore. |

| Formato         | Descrizione                                                                                                                                                           |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 'x of y'        | x = il numero di valori selezionati<br><br>y = il numero totale di valori<br><br>Questo viene restituito quando $\text{max\_values} < x < (y - \text{max\_values})$ . |
| 'ALL'           | Restituito se tutti i valori risultano selezionati.                                                                                                                   |
| '.'             | Restituito se nessun valore risulta selezionato.                                                                                                                      |
| <search string> | Se si è effettuato una selezione usando la ricerca, la stringa di ricerca viene restituita.                                                                           |

### Argomenti

| Argomenti         | Descrizione                                                                                                                                                                                                                          |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>field_name</b> | Il campo contenente la scala di dati da misurare.                                                                                                                                                                                    |
| <b>value_sep</b>  | Il separatore da inserire tra i valori di campo. Il valore predefinito è ', '.                                                                                                                                                       |
| <b>max_values</b> | Il numero massimo di valori di campo da elencare singolarmente. Se si seleziona un numero maggiore di valori, verrà utilizzato il formato 'x di y valori'. Il valore predefinito è 6.                                                |
| <b>state_name</b> | Il nome di uno stato alternato che è stato scelto per la specifica visualizzazione. Se viene utilizzato l'argomento <b>state_name</b> , saranno prese in considerazione solo le selezioni associate al nome dello stato specificato. |

### Esempio: espressioni del grafico

| Esempio                                     | Risultato                                                                                                                                                                   |
|---------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>GetFieldSelections(FirstName )</code> | Restituisce "-" (NULL) se non è stata effettuata alcuna selezione per il campo <code>FirstName</code> . Altrimenti, restituisce il valore delle selezioni, ad esempio John. |

## Esempio: principi fondamentali della funzione GetFieldSelections

Espressione del grafico

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:

- FirstName
- LastName
- Initials
- HasCellphone

### Script di caricamento

Example:

```
LOAD * inline [
FirstName|LastName|Initials|HasCellphone
John|Anderson|JA|Yes
Sue|Brown|SB|Yes
Mark|Carr|MC|No
Peter|Devonshire|PD|No
Jane|Elliot|JE|Yes
Peter|Franc|PF|Yes ] (delimiter is '|');
```

### Risultati

1. Caricare i dati e aprire un foglio. Creare una nuova casella di filtro e aggiungere questo campo come dimensione:
  - FirstName
2. Creare un oggetto KPI, quindi aggiungere la seguente misura:
  - =GetFieldSelections(FirstName), per restituire i valori della selezione effettuata nel campo FirstName.



*In questo esempio, la funzionalità di applicazione di stili ai KPI è stata regolata in modo da utilizzare un comportamento di presentazione reattivo, per semplificare la visualizzazione dell'output.*

3. Fare clic su **Modifica foglio** per passare alla modalità di analisi, quindi effettuare le seguenti operazioni nell'ordine in cui appaiono nella tabella:

Azioni e risultati in modalità di analisi

| Azione                                                                                                                                                                        | Risultati KPI                                                           |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------|
| Selezionare John nel filtro FirstName.                                                                                                                                        | La misura restituisce il valore John.                                   |
| Selezionare John e Peter nel filtro FirstName.                                                                                                                                | La misura restituisce il valore John, Peter.                            |
| Fare clic su <b>Modifica foglio</b> per passare alla modalità di modifica, quindi modificare l'espressione della misura KPI come segue: =GetFieldSelections(FirstName, ';' ') | La misura restituisce i valori il nuovo formato value_sep: John; Peter. |
| Modificare l'espressione di misura del KPI                                                                                                                                    | La misura restituisce i valori il nuovo formato                         |

### Azione

come segue: `=GetFieldSelections (FirstName,',' , 2)`

Fare clic su **Modifica foglio** per tornare alla modalità di analisi. Annullare le selezioni dal filtro `FirstName`, quindi selezionare John, Sue e Mark.

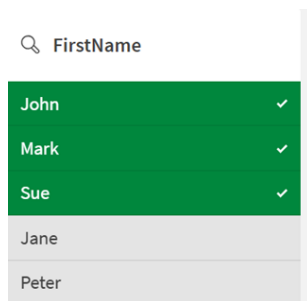
### Risultati KPI

`value_sep` per la spaziatura: John;Peter.

La misura restituisce `NOT Jane;Peter`, perché il numero di selezioni supera l'argomento `max_value` di 2. Se non è impostato `max_value`, verranno visualizzati tutti i nomi selezionati: John; Sue; Mark.

L'immagine che segue mostra i risultati dell'ultima azione nella tabella in cui il numero di selezioni supera l'argomento `max_value`.

*Risultati del KPI per la selezione dei nomi John, Sue, e Mark*



`=GetFieldSelections (FirstName,',' , 2)`

NOT Jane;Peter

In questo esempio è possibile vedere come i valori restituiti dalla funzione `GetFieldSelections` cambiano man mano che vengono effettuate le selezioni.

## Esempio: scenario per l'applicazione della funzione `GetFieldSelections`

Espressione del grafico

### Panoramica

La funzione `GetFieldSelections` viene utilizzata per evidenziare i nomi dei clienti quando vengono selezionati.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `OrderID`
  - `CustomerName`
  - `Product`
  - `Quantity`

### Script di caricamento

Example:

```
LOAD * INLINE [  
OrderID, CustomerName, Product, Quantity  
1, John Doe, Laptop, 2  
2, Jane Smith, Smartphone, 1  
3, Ahmed Khan, Desk, 3  
4, Maria Garcia, Chair, 1  
5, Li Wei, Blender, 2  
];
```

### Risultati

1. Caricare i dati e aprire un foglio. Creare una nuova casella di filtro e aggiungere questi campi come dimensioni:
  - CustomerName
  - Product
2. Creare un oggetto KPI, quindi aggiungere la seguente misura:
  - =GetFieldSelections(CustomerName).



*In questo esempio, la funzionalità di applicazione di stili ai KPI è stata regolata in modo da utilizzare un comportamento di presentazione reattivo, per semplificare la visualizzazione dell'output. .*

3. Fare clic su **Modifica foglio** per passare alla modalità di analisi, quindi selezionare Jane Smith e John Doe dal filtro CustomerName.

La funzione GetFieldSelections restituisce i seguenti valori: Jane Smith, John Doe.

| CustomerName | Product     |
|--------------|-------------|
| Jane Smith ✓ | Laptop      |
| John Doe ✓   | Smartphone  |
| Ahmed Khan   | Blender     |
| Li Wei       | Chair       |
| Maria Garcia | Desk        |
|              | Laptops     |
|              | Smartphones |
|              | Tablets     |

=GetFieldSelections (CustomerName)  
**Jane Smith, John Doe**

È possibile visualizzare i clienti selezionati nella casella di filtro del KPI.

### GetObjectDimension - funzione per grafici

**GetObjectDimension()** restituisce il nome della dimensione. **Index** è un numero intero facoltativo che indica la dimensione da restituire.



Non è possibile utilizzare questa funzione in un grafico nelle posizioni seguenti: titolo, sottotitolo, piè di pagina, espressione linea di riferimento ed espressioni min/max.



Non è possibile fare riferimento al nome di una dimensione o misura in un altro oggetto usando l'Object ID.

### Sintassi:

```
GetObjectDimension ([index])
```

Tipo di dati restituiti: Stringa

Esempio: espressioni del grafico

| Esempio                                                                   | Risultato                                                 |
|---------------------------------------------------------------------------|-----------------------------------------------------------|
| <code>GetObjectDimension ()</code><br><code>GetObjectDimension (0)</code> | Restituisce il nome della prima dimensione del grafico.   |
| <code>GetObjectDimension (1)</code>                                       | Restituisce il nome della seconda dimensione del grafico. |

### Esempio: principi fondamentali della funzione GetObjectDimension

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `TransactionDate`
  - `CustomerID`
  - `TransactionQuantity`

#### Script di caricamento

Example:

```
LOAD * INLINE [  
TransactionDate, CustomerID, TransactionQuantity  
2018/08/30, 049681, 13  
2018/08/30, 203521, 6  
2018/08/30, 203521, 21  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- TransactionDate
- CustomerID
- TransactionQuantity

Creare le seguenti misure:

- =GetObjectDimension (), per visualizzare la prima dimensione nella tabella.
- =GetObjectDimension (0), per visualizzare la prima dimensione nella tabella.
- =GetObjectDimension (1), per visualizzare la seconda dimensione nella tabella.

| Tabella dei risultati |            |                     |                       |                        |                        |
|-----------------------|------------|---------------------|-----------------------|------------------------|------------------------|
| Transaction Date      | CustomerID | TransactionQuantity | GetObjectDimension () | GetObjectDimension (0) | GetObjectDimension (1) |
| 2018/08/30            | 049681     | 13                  | TransactionDate       | TransactionDate        | CustomerID             |
| 2018/08/30            | 203521     | 6                   | TransactionDate       | TransactionDate        | CustomerID             |
| 2018/08/30            | 203521     | 21                  | TransactionDate       | TransactionDate        | CustomerID             |

Osservando i risultati, si può vedere come la funzione `GetObjectDimension` restituisca il nome della dimensione indicata nei parametri della funzione.

### Esempio: scenario avanzato per l'applicazione della funzione `GetObjectDimension`

Espressione del grafico

#### Panoramica

Questo esempio utilizza la funzione `GetObjectDimension` in combinazione con le selezioni dei filtri. Il calcolo delle dimensioni e delle misure che appare nell'oggetto grafico cambia in base al filtro selezionato.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in due tabelle dati inline: `sales` e `filter`. La prima tabella, `sales`, include i dati relativi alle vendite. La seconda tabella, `filter`, contiene un elenco di nomi di campi che verranno utilizzati come interruttore per la dimensione. L'interruttore permette di passare da una dimensione all'altra in un grafico e di personalizzare il calcolo della misura che appare in base

alla dimensione selezionata.

- La `sales` tabella contiene i seguenti campi:
  - `Country`
  - `Salesperson`
  - `SalesValue`
- La tabella `Filter` contiene il seguente campo: `FilterField`.

### Script di caricamento

```
Sales:
LOAD * INLINE [
Country, SalesPerson, SalesValue
USA, John, 500
USA, Alice, 700
Canada, Bob, 300
Canada, Carol, 400
Mexico, Dave, 200
];
```

```
Filter:
LOAD * INLINE [
FilterField
Country
SalesPerson
];
```

### Risultati

1. Caricare i dati e aprire un foglio. Creare una nuova casella di filtro e aggiungere questo campo come dimensione:
  - `FilterField`
2. Nel pannello delle risorse, in **Campi**, fare clic con il pulsante destro del mouse su **Campo filtro** e selezionare **Impostazioni campo**. Selezionare la casella di controllo **Sempre un valore selezionato**, quindi fare clic su **Salva**.
3. Creare una tabella, quindi aggiungere una dimensione inserendo la seguente espressione:
  - `=$(= FilterField)`
  - In **Etichetta**, inserire la seguente espressione per definire il nome della colonna per la dimensione: `='$(= FilterField)'`
4. Creare la seguente misura:
  - `=If(GetObjectDimension() = 'Country', Sum(SalesValue), Avg(SalesValue))`, per sommare i valori di vendita se la dimensione selezionata è `Country`, oppure per restituire la media del valore di vendita quando la dimensione selezionata è `SalesPerson`.
  - In **Label**, inserire la seguente espressione per definire il nome della colonna per la misura in base alla selezione del filtro: `=If(GetObjectDimension() = 'Country', 'Sum(SalesValue)', 'Avg(SalesValue)')`

In modalità di analisi, quando si passa da `SalesPerson` a `Country` nel filtro `FieldFilter`, la tabella cambia per includere la dimensione selezionata e il calcolo della misura corrispondente per quella dimensione. Ad esempio, se si seleziona `Country` nel filtro, la tabella mostra `Country` come dimensione nella prima colonna. La misura utilizza quindi la funzione `GetObjectDimension`, la equipara a `Country` e restituisce `Sum(SalesValue)`.

Casella di filtro e risultati della tabella per `Country`

|               |  |
|---------------|--|
| 🔍 FilterField |  |
| Country ✓     |  |
| SalesPerson   |  |

| Country | Sum(SalesValue) |
|---------|-----------------|
| Totals  | 2100            |
| Canada  | 700             |
| Mexico  | 200             |
| USA     | 1200            |

La tabella seguente mostra i risultati quando si seleziona `Country` nel filtro `FilterField`.

Tabella dei risultati per `Country`

| Country | Sum(SalesValue) |
|---------|-----------------|
| Totals  | 2100            |
| Canada  | 700             |
| Mexico  | 200             |
| USA     | 1200            |

La tabella seguente mostra i risultati quando si seleziona `SalesPerson` nel filtro `FilterField`.

Tabella dei risultati per `SalesPerson`

| SalesPerson | Avg(SalesValue) |
|-------------|-----------------|
| Totals      | 420             |
| Alice       | 700             |
| Bob         | 300             |
| Carol       | 400             |
| Dave        | 200             |
| John        | 500             |

## GetObjectField - funzione per grafici

`GetObjectField()` restituisce l'espressione di campo della dimensione. **Index** è un numero intero opzionale che indica la dimensione che deve essere restituita.



Non è possibile utilizzare questa funzione in un grafico nelle posizioni seguenti: titolo, sottotitolo, piè di pagina, espressione linea di riferimento ed espressioni min/max.



Non è possibile fare riferimento al nome di una dimensione o misura in un altro oggetto usando l'Object ID.

### Sintassi:

```
GetObjectField ([index])
```

Tipo di dati restituiti: stringa

Esempio: espressioni del grafico

| Esempio                                 | Risultato                                                                 |
|-----------------------------------------|---------------------------------------------------------------------------|
| GetObjectField ()<br>GetObjectField (0) | Restituisce l'espressione di campo per la prima dimensione del grafico.   |
| GetObjectField (1)                      | Restituisce l'espressione di campo per la seconda dimensione del grafico. |

Se si desidera restituire il nome di una misura, usare invece la funzione **GetObjectMeasure**.

### Esempio: principi fondamentali della funzione GetObjectField

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `TransactionDate`
  - `CustomerID`
  - `TransactionQuantity`

#### Script di caricamento

Example:

```
LOAD * INLINE [  
TransactionDate, CustomerID, TransactionQuantity  
2018/08/30, 049681, 13  
2018/08/30, 203521, 6  
2018/08/30, 203521, 21  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `TransactionDate`
- `CustomerID`
- `TransactionQuantity`

Creare le seguenti misure:

- `=GetObjectField()`, per visualizzare la prima dimensione nella tabella.
- `=GetObjectField(0)`, per visualizzare la prima dimensione nella tabella.
- `=GetObjectField(1)`, per visualizzare la seconda dimensione nella tabella.

| Tabella dei risultati |            |                     |                   |                    |                    |  |
|-----------------------|------------|---------------------|-------------------|--------------------|--------------------|--|
| TransactionDate       | CustomerID | TransactionQuantity | GetObjectField () | GetObjectField (0) | GetObjectField (1) |  |
| 2018/08/30            | 049681     | 13                  | TransactionDate   | TransactionDate    | CustomerID         |  |
| 2018/08/30            | 203521     | 6                   | TransactionDate   | TransactionDate    | CustomerID         |  |
| 2018/08/30            | 203521     | 21                  | TransactionDate   | TransactionDate    | CustomerID         |  |

Osservando i risultati, si può vedere come la funzione `GetObjectField` restituisca il nome del campo indicato nei parametri della funzione.

### Esempio: scenario avanzato per l'applicazione della funzione `GetObjectField`

Espressione del grafico

#### Panoramica

Questo esempio utilizza la funzione `GetObjectField` in combinazione con le selezioni dei filtri. Il calcolo delle dimensioni e delle misure che appare nell'oggetto grafico cambia in base al filtro selezionato.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in due tabelle dati inline: `sa1es` e `Fi1ter`. La prima tabella, `sa1es`, include i dati relativi alle vendite. La seconda tabella, `Fi1ter`, contiene un elenco di nomi di campi che verranno utilizzati come interruttore per la dimensione. L'interruttore permette di passare da una dimensione all'altra in un grafico e di personalizzare il calcolo della misura che appare in base alla dimensione selezionata.
- La `sa1es` tabella contiene i seguenti campi:

- Country
  - Salesperson
  - SalesValue
- La tabella `Filter` contiene il seguente campo: `FilterField`.

### Script di caricamento

```
Sales:
LOAD * INLINE [
Country, SalesPerson, SalesValue
USA, John, 500
USA, Alice, 700
Canada, Bob, 300
Canada, Carol, 400
Mexico, Dave, 200
];
```

```
Filter:
LOAD * INLINE [
FilterField
Country
SalesPerson
];
```

### Risultati

1. Caricare i dati e aprire un foglio. Creare una nuova casella di filtro e aggiungere questo campo come dimensione:
  - `FilterField`
2. Nel pannello delle risorse, in **Campi**, fare clic con il pulsante destro del mouse su **Campo filtro** e selezionare **Impostazioni campo**. Selezionare la casella di controllo **Sempre un valore selezionato**, quindi fare clic su **Salva**.
3. Creare una tabella, quindi aggiungere una dimensione inserendo la seguente espressione:
  - `=$(= FilterField)`
  - In **Etichetta**, inserire la seguente espressione per definire il nome della colonna per la dimensione: `'$(= FilterField)'`
4. Creare la seguente misura:
  - `=If(GetObjectField() = 'Country', Sum(SalesValue), Avg(SalesValue))`, per sommare i valori di vendita se la dimensione selezionata è `Country`, oppure per restituire la media del valore di vendita quando la dimensione selezionata è `SalesPerson`.
  - In **Label**, inserire la seguente espressione per definire il nome della colonna che visualizzerà la misura in base alla selezione del filtro: `=If(GetObjectField() = 'Country', 'Sum (SalesValue)', 'Avg(SalesValue)')` Ad esempio, se il valore `Country` è selezionato nel filtro, il nome della misura restituirà il valore `sum(SalesValue)`.

In modalità di analisi, quando si passa da `SalesPerson` a `Country` nel filtro `FieldFilter`, la tabella cambia per includere la dimensione selezionata e il calcolo della misura corrispondente per quella dimensione. Ad esempio, se si seleziona `Country` nel filtro, la tabella mostra `Country` come dimensione nella prima colonna. La misura utilizza quindi la funzione `GetObjectField`, la equipara a `Country` e restituisce `sum(SalesValue)`.

Casella di filtro e risultati della tabella per `Country`

|             |  |
|-------------|--|
| FilterField |  |
| Country ✓   |  |
| SalesPerson |  |

| Country | Sum(SalesValue) |
|---------|-----------------|
| Totals  | 2100            |
| Canada  | 700             |
| Mexico  | 200             |
| USA     | 1200            |

La tabella seguente mostra i risultati quando si seleziona `Country` nel filtro `FieldFilter`.

Tabella dei risultati per `Country`

| Country | Sum(SalesValue) |
|---------|-----------------|
| Canada  | 700             |
| Mexico  | 200             |
| USA     | 1200            |

La tabella seguente mostra i risultati quando si seleziona `SalesPerson` nel filtro `FieldFilter`.

Tabella dei risultati per `SalesPerson`

| SalesPerson | Avg(SalesValue) |
|-------------|-----------------|
| Alice       | 700             |
| Bob         | 300             |
| Carol       | 400             |
| Dave        | 200             |
| John        | 500             |

### GetObjectMeasure - funzione per grafici

`GetObjectMeasure()` restituisce il nome della misura. `Index` è un numero intero opzionale che indica la misura da restituire.



Non è possibile utilizzare questa funzione in un grafico nelle posizioni seguenti: titolo, sottotitolo, piè di pagina, espressione linea di riferimento ed espressioni min/max.



Non è possibile fare riferimento al nome di una dimensione o misura in un altro oggetto usando l'Object ID.

### Sintassi:

```
GetObjectMeasure ([index])
```

Tipo di dati restituiti: Stringa

Esempio: espressioni del grafico

| Esempio                                                               | Risultato                                             |
|-----------------------------------------------------------------------|-------------------------------------------------------|
| <code>GetObjectMeasure ()</code><br><code>GetObjectMeasure (0)</code> | Restituisce il nome della prima misura del grafico.   |
| <code>GetObjectMeasure (1)</code>                                     | Restituisce il nome della seconda misura del grafico. |

### Esempio: principi fondamentali della funzione GetObjectMeasure

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `CustomerID`
  - `TransactionQuantity`

#### Script di caricamento

Example:

```
LOAD * INLINE [  
CustomerID, TransactionQuantity  
049681, 13  
203521, 6  
203521, 21  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- CustomerID

Creare le seguenti misure:

- Sum(TransactionQuantity): valore della prima misura.
- Avg(TransactionQuantity): valore della seconda misura.
- =GetObjectMeasure(), per visualizzare la prima misura nella tabella.
- =GetObjectMeasure(0), per visualizzare la prima misura nella tabella.
- =GetObjectMeasure(1), per visualizzare la seconda misura nella tabella.

Tabella dei risultati

| Custom<br>erID | Sum<br>(TransactionQ<br>uantity) | Avg<br>(TransactionQ<br>uantity) | GetObjectMea<br>sure ()           | GetObjectMea<br>sure (0)          | GetObjectMea<br>sure (1)          |
|----------------|----------------------------------|----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|
| Totals         | 40                               | 13.333333                        | Sum<br>(TransactionQ<br>uantity)  | Sum<br>(TransactionQ<br>uantity)  | Avg<br>(TransactionQ<br>uantity)  |
| 049681         | 13                               | 13                               | =Sum<br>(TransactionQu<br>antity) | =Sum<br>(TransactionQu<br>antity) | =Avg<br>(TransactionQu<br>antity) |
| 203521         | 27                               | 13.5                             | =Sum<br>(TransactionQu<br>antity) | =Sum<br>(TransactionQu<br>antity) | =Avg<br>(TransactionQu<br>antity) |

Osservando i risultati, si può vedere come la funzione GetObjectMeasure restituisca il nome della misura indicata nei parametri della funzione.

### GetPossibleCount - funzione per grafici

**GetPossibleCount()** viene utilizzato per trovare il numero di valori possibili nel campo identificato. Se il campo identificato include selezioni, i campi selezionati (verdi) vengono conteggiati. In caso contrario, vengono conteggiati i valori associati (bianco).

Per i campi con selezioni, **GetPossibleCount()** restituisce il numero di campi selezionati (verdi).

#### Sintassi:

```
GetPossibleCount (field_name)
```

**Tipo di dati restituiti:** numero intero



*I colori utilizzati nella barra di selezione e per ogni stato di selezione possono essere modificati tramite un tema personalizzato. Se si utilizza un'app che utilizza un tema personalizzato, si potrebbe notare che le selezioni non vengono visualizzate con gli stessi colori descritti in questo argomento della guida.*

#### Argomenti

| Argomenti         | Descrizione                                       |
|-------------------|---------------------------------------------------|
| <b>field_name</b> | Il campo contenente la scala di dati da misurare. |

La tabella seguente elenca altre funzioni correlate a questa funzione.

#### Funzioni correlate

| Funzione                                                                      | Interazione                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#"><u>GetStateCounts - funzione per grafici (page 1343)</u></a>      | Utilizzando <b>GetStateCounts()</b> , è possibile combinare il calcolo dei seguenti conteggi con un'unica chiamata di funzione: <ul style="list-style-type: none"> <li>• Conteggio dei valori selezionati inclusi.</li> <li>• Conteggio dei valori possibili.</li> <li>• Conteggio dei valori alternativi.</li> <li>• Conteggio dei valori esclusi, senza includere i valori esclusi alternativi e selezionati.</li> <li>• Conteggio dei valori selezionati esclusi.</li> </ul> |
| <a href="#"><u>GetSelectedCount - funzione per grafici (page 1338)</u></a>    | Restituisce il conteggio dei valori selezionati inclusi.                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <a href="#"><u>GetAlternativeCount - funzione per grafici (page 1304)</u></a> | Restituisce il conteggio dei valori alternativi.                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <a href="#"><u>GetPossibleCount - funzione per grafici (page 1334)</u></a>    | Restituisce il conteggio dei valori esclusi, senza includere i valori esclusi alternativi e selezionati.                                                                                                                                                                                                                                                                                                                                                                        |

#### Esempio: espressioni del grafico

| Esempio                                 | Risultato                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>GetPossibleCount(Initials)</code> | Se non vengono effettuate selezioni, restituisce il numero totale di valori univoci nel campo <code>Initials</code> . Se viene selezionato un valore in questo campo, il valore restituito è 1. Se viene effettuata una selezione in un altro campo, il valore restituito è il numero di valori univoci nel campo <code>Initials</code> che si riferiscono ai valori selezionati nel secondo campo. |

### Esempio: principi fondamentali della funzione GetPossibleCount

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `FirstName`
  - `LastName`
  - `Initials`
  - `HasCellphone`

#### Script di caricamento

Example:

```
LOAD * inline [
FirstName|LastName|Initials|HasCellphone
John|Anderson|JA|Yes
Sue|Brown|SB|Yes
Mark|Carr|MC|No
Peter|Devonshire|PD|No
Jane|Elliot|JE|Yes
Peter|Franc|PF|Yes ] (delimiter is '|');
```

#### Risultati

1. Caricare i dati e aprire un foglio. Creare una nuova casella di filtro e aggiungere questi campi come dimensioni:
  - `FirstName`
  - `Initials`
2. Creare due oggetti KPI, uno per ciascuna delle seguenti espressioni di misura:
  - KPI 1: `=GetPossibleCount(FirstName)` per restituire il numero di valori univoci possibili nel campo `FirstName`.
  - KPI 2: `=GetPossibleCount(Initials)` per restituire il numero di valori univoci possibili nel campo `Initials`.

Ogni KPI restituisce il numero di valori univoci possibili. Se non si effettua alcuna selezione nel filtro, il KPI 1 restituisce il valore 5 e il KPI 2 restituisce il valore 6.



*In questo esempio, la funzionalità di applicazione di stili ai KPI è stata regolata in modo da utilizzare un comportamento di presentazione reattivo, per semplificare la visualizzazione dell'output.*

3. Fare clic su **Modifica foglio** per passare alla modalità di analisi, quindi selezionare John nel filtro `FirstName`.  
Entrambi i KPI restituiscono il valore 1 perché c'è un valore selezionato nel campo `FirstName`, mentre `JA` è l'unico valore associato nel campo `Initials`.
4. Annullare la selezione del filtro `FirstName`, quindi selezionare Peter.  
Il KPI 1 restituisce il valore 1 perché c'è one unico valore possibile disponibile nel campo `FirstName`.  
Il KPI 2 restituisce 2 perché ci sono two valori possibili disponibili nel campo `Initials`.



### Esempio: scenario per l'applicazione della funzione `GetPossibleCount`

Espressione del grafico

#### Panoramica

Un manager vuole filtrare i dati dei dipendenti per determinare quali dipendenti hanno esperienza in diverse aree di competenza.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Project`
  - `Employee`
  - `Skill`

#### Script di caricamento

Example:

```
LOAD * INLINE [  
Project, Employee, Skill  
Project X, John, Java  
Project X, Mary, SQL  
Project Y, James, Python  
Project Y, Linda, Java  
Project Z, Robert, SQL  
];
```

### Risultati

1. Caricare i dati e aprire un foglio. Creare una nuova casella di filtro e aggiungere questo campo come dimensione:

- skill

2. Creare un oggetto KPI, quindi aggiungere la seguente misura:

- `=GetPossibleCount(Employee)` per determinare il numero di dipendenti.

Se non si effettua alcuna selezione nella casella di filtro, il KPI restituisce il valore 5 perché ci sono five dipendenti.



*In questo esempio, la funzionalità di applicazione di stili ai KPI è stata regolata in modo da utilizzare un comportamento di presentazione reattivo, per semplificare la visualizzazione dell'output.*

3. Fare clic su **Modifica foglio** per passare alla modalità di analisi, quindi selezionare Java nel filtro skill.

Il KPI restituisce il valore 2 perché ci sono two dipendenti che hanno la competenza Java.

 Skill

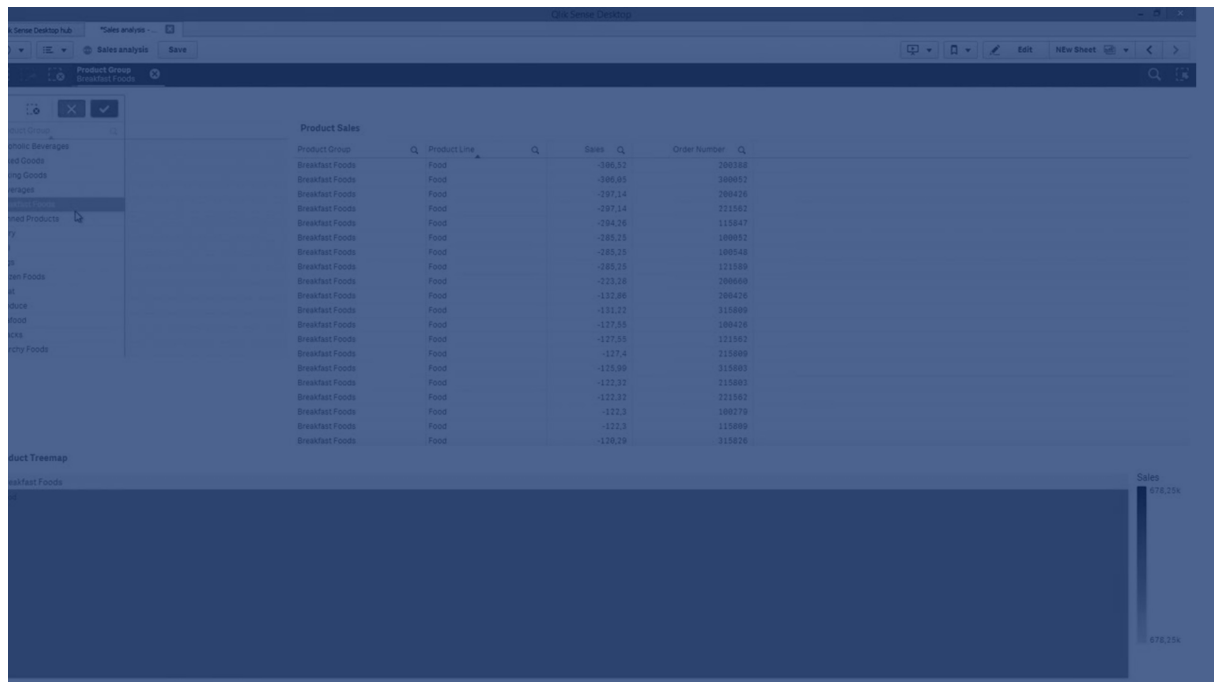
|        |   |
|--------|---|
| Java   | ✓ |
| Python |   |
| SQL    |   |

`=GetPossibleCount(Employee)`

2

### GetSelectedCount - funzione per grafici

`GetSelectedCount()` trova il numero di valori selezionati (verdi) in un campo.

**Sintassi:**

```
GetSelectedCount (field_name [, include_excluded [, state_name]])
```

**Tipo di dati restituiti:** numero intero



*I colori utilizzati nella barra di selezione e per ogni stato di selezione possono essere modificati tramite un tema personalizzato. Se si utilizza un'app che utilizza un tema personalizzato, si potrebbe notare che le selezioni non vengono visualizzate con gli stessi colori descritti in questo argomento della guida.*

**Argomenti**

| Argomenti               | Descrizione                                                                                                                                                                                                                          |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>field_name</b>       | Il campo contenente la scala di dati da misurare.                                                                                                                                                                                    |
| <b>include_excluded</b> | Se impostato su <b>True()</b> , il conteggio includerà i valori selezionati, i quali sono attualmente esclusi dalle selezioni in altri campi. Se invece è <b>False</b> o omesso, questi valori non verranno inclusi.                 |
| <b>state_name</b>       | Il nome di uno stato alternato che è stato scelto per la specifica visualizzazione. Se viene utilizzato l'argomento <b>state_name</b> , saranno prese in considerazione solo le selezioni associate al nome dello stato specificato. |

La tabella seguente elenca altre funzioni correlate a questa funzione.

## Funzioni correlate

| Funzione                                                                      | Interazione                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#"><u>GetStateCounts - funzione per grafici (page 1343)</u></a>      | Utilizzando <b>GetStateCounts()</b> , è possibile combinare il calcolo dei seguenti conteggi con un'unica chiamata di funzione: <ul style="list-style-type: none"> <li>• Conteggio dei valori selezionati inclusi.</li> <li>• Conteggio dei valori possibili.</li> <li>• Conteggio dei valori alternativi.</li> <li>• Conteggio dei valori esclusi, senza includere i valori esclusi alternativi e selezionati.</li> <li>• Conteggio dei valori selezionati esclusi.</li> </ul> |
| <a href="#"><u>GetPossibleCount - funzione per grafici (page 1334)</u></a>    | Restituisce il conteggio dei valori possibili.                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <a href="#"><u>GetAlternativeCount - funzione per grafici (page 1304)</u></a> | Restituisce il conteggio dei valori alternativi.                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <a href="#"><u>GetSelectedCount - funzione per grafici (page 1338)</u></a>    | Restituisce il conteggio dei valori esclusi, senza includere i valori esclusi alternativi e selezionati.                                                                                                                                                                                                                                                                                                                                                                        |

## Esempio: espressioni del grafico

| Esempio                               | Risultato                                                                                                                                                                                                                                                                                                                                 |
|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GetSelectedCount (FirstName)          | Se non vengono effettuate selezioni, viene restituito 0. Se sono state effettuate delle selezioni nel campo <code>FirstName</code> , viene restituito il numero di articoli selezionati.                                                                                                                                                  |
| GetSelectedCount (FirstName, True ()) | Se non vengono effettuate selezioni, viene restituito 0. Se sono state effettuate delle selezioni nel campo <code>FirstName</code> , il numero di articoli selezionati viene restituito indipendentemente da eventuali selezioni successive in altri campi che potrebbero ridurre i valori disponibili nel campo <code>FirstName</code> . |

## Esempio: principi fondamentali della funzione GetSelectedCount

Espressione del grafico

## Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:

- `FirstName`
- `LastName`
- `Initials`
- `HasCellphone`

### Script di caricamento

Example:

```
LOAD * inline [
FirstName|LastName|Initials|HasCellphone
John|Anderson|JA|Yes
Sue|Brown|SB|Yes
Mark|Carr|MC|No
Peter|Devonshire|PD|No
Jane|Elliot|JE|Yes
Peter|Franc|PF|Yes ] (delimiter is '|');
```

### Risultati

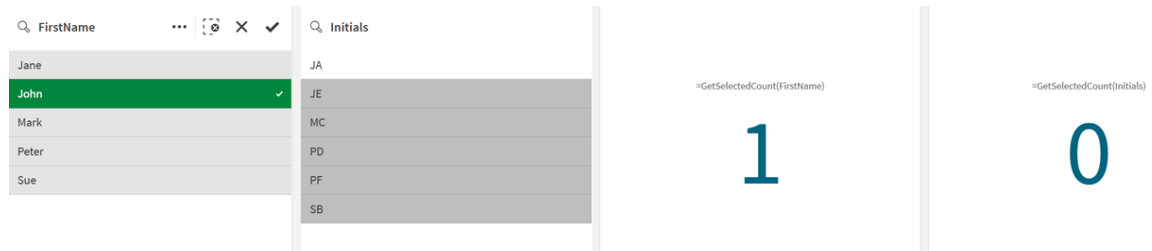
1. Caricare i dati e aprire un foglio. Creare una nuova casella di filtro e aggiungere questi campi come dimensioni:
  - `FirstName`
  - `Initials`
2. Creare due oggetti KPI, uno per ciascuna delle seguenti espressioni di misura:
  - KPI 1: `=getSelectedCount(FirstName)` per restituire il numero di elementi selezionati nel campo `FirstName`.
  - KPI 2: `=getSelectedCount(Initials)` per restituire il numero di elementi selezionati nel campo `Initials`.

Ogni KPI restituisce il numero di elementi selezionati in ogni campo. Se non si seleziona alcun filtro, entrambi i KPI restituiscono il valore 0.



*In questo esempio, la funzionalità di applicazione di stili ai KPI è stata regolata in modo da utilizzare un comportamento di presentazione reattivo, per semplificare la visualizzazione dell'output.*

3. Fare clic su **Modifica foglio** per passare alla modalità di analisi, quindi selezionare `John` nel filtro `FirstName`.  
Il KPI 1 restituisce il valore 1 perché c'è un elemento selezionato nel filtro `FirstName`. Il KPI 2 restituisce ancora 0 perché il filtro `Initials` non ha elementi selezionati.



### Esempio: utilizzo della funzione GetSelectedCount con il parametro include\_excluded

Espressione del grafico

#### Panoramica

Questo esempio utilizza lo stesso set di dati dell'esempio precedente, ma dimostra la differenza di risultati quando il parametro `include_excluded` è impostato su `True()`.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `FirstName`
  - `LastName`
  - `Initials`
  - `HasCellphone`

#### Script di caricamento

Example:

```
LOAD * inline [
FirstName|LastName|Initials|HasCellphone
John|Anderson|JA|Yes
Sue|Brown|SB|Yes
Mark|Carr|MC|No
Peter|Devonshire|PD|No
Jane|Elliot|JE|Yes
Peter|Franc|PF|Yes ] (delimiter is '|');
```

#### Risultati

1. Caricare i dati e aprire un foglio. Creare una nuova casella di filtro e aggiungere questi campi come dimensioni:
  - `Initials`
  - `HasCellPhone`

2. Creare due oggetti KPI, uno per ciascuna delle seguenti espressioni di misura:

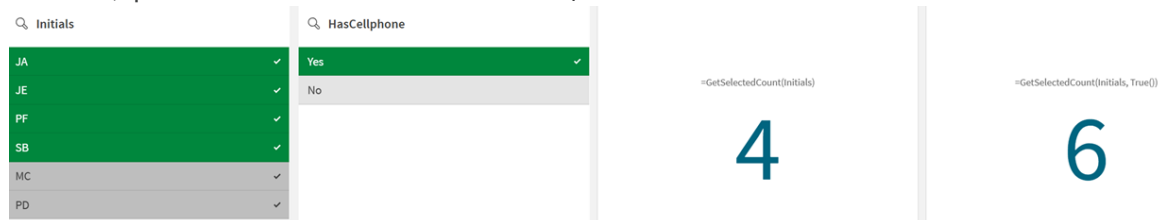
- KPI 1: `=GetSelectedCount(Initials)` per determinare il numero di valori che sono stati selezionati nel campo `Initials`.
- KPI 2: `=GetSelectedCount([Initials], True())` per includere i valori selezionati, che sono attualmente esclusi dalle selezioni in altri campi.

Se non si effettua alcuna selezione nella casella di filtro, entrambi i KPI restituiscono il valore 0.



*In questo esempio, la funzionalità di applicazione di stili ai KPI è stata regolata in modo da utilizzare un comportamento di presentazione reattivo, per semplificare la visualizzazione dell'output.*

3. Fare clic su **Modifica foglio** per passare alla modalità di analisi. Selezionare tutti i valori nel filtro `Initials`, quindi selezionare `Yes` nel filtro `HasCellphone`.



Quando si confrontano i risultati di ogni KPI, si può vedere come ognuno restituisca un valore diverso a seconda che il parametro `include_excluded` sia impostato su `True()` o meno. Il KPI 2 restituisce il valore 6 perché include tutti i valori selezionati nel filtro `Initials`, anche se la selezione nel filtro `HasCellphone` ha ridotto gli elementi disponibili a 4. Il KPI 1 restituisce 4 perché non conta i valori esclusi selezionati in `Initials`.

### GetStateCounts - funzione per grafici

La funzione grafico **GetStateCounts()** viene utilizzata per calcolare il numero totale di valori unici corrispondenti agli stati di selezione specificati.

Con **GetStateCounts()**, è possibile combinare i calcoli delle seguenti Funzioni in un'unica chiamata di funzione: **GetSelectedCount()**, **GetNotSelectedCount()**, **GetAlternativeCount()**, **GetPossibleCount()** e **GetExcludedCount()**. È disponibile anche il conteggio dei valori esclusi selezionati da aggiungere al calcolo. È possibile specificare se ogni calcolo della funzione aggiunge oppure sottrae dal totale restituito.

**Sintassi:**

```
GetStateCounts (field_name, state_name [, state_type1,...state_typeN])
```

Tipo di dati restituiti: numero intero

#### Argomenti

| Argomento         | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>field_name</b> | Il campo per il quale si sta calcolando lo stato di selezione. Un nome di campo inesistente produce un risultato nullo.                                                                                                                                                                                                                                                                                                                                                                         |
| <b>state_name</b> | Il nome dello Stato alternativo. Se l'argomento è vuoto ( ' ' ) oppure nullo, viene utilizzato lo stato alternativo ereditato. Utilizzare \$ per impiegare esplicitamente lo stato predefinito. Un nome di stato (non vuoto) che non corrisponde a uno stato esistente dà come risultato NULL.                                                                                                                                                                                                  |
| <b>state_type</b> | <p>Un elenco di uno o più tipi di campo per il valore del campo. Questi tipi di stato saranno aggregati in un conteggio. Un tipo di stato viene specificato tramite una chiave. Inserire ogni chiave con virgolette singole.</p> <p>Se questo argomento viene omissso, la funzione restituisce una stringa con tutti i conteggi di stato disponibili per il campo, nello stesso ordine dell'enumerazione.</p> <p>Per un elenco degli stati utilizzabili, consultare la tabella sottostante.</p> |

I tipi di stato sono referenziati mediante chiavi specifiche. È possibile utilizzare la versione numerica oppure testuale del tasto. Combinare più di un tasto nella stessa espressione per personalizzare ulteriormente il risultato. È possibile sottrarre il conteggio dello stato dal totale, anziché aggiungerlo. A tale scopo, utilizzare la chiave testuale e anteporre al tipo di stato il simbolo meno (-).

Il tipo di stato ed eventualmente il simbolo meno devono essere racchiusi in una serie di virgolette singole.

#### Chiavi per ogni tipo di campo stato

| Tipo di stato campo | Descrizione                                                                                                                                                                         | Chiave numerica | Chiave testuale |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|-----------------|
| Selezionati         | Include i valori selezionati nel calcolo. Per la funzione equivalente, vedere <a href="#">GetSelectedCount - funzione per grafici (page 1338)</a> .                                 | 1               | S               |
| Opzionale           | Include nel calcolo valori opzionali (non selezionati, ma selezionabili). Per la funzione equivalente, vedere <a href="#">GetPossibleCount - funzione per grafici (page 1334)</a> . | 2               | O               |
| Deselezionato       | Include nel calcolo i valori non selezionati. Questo tipo di stato è disponibile solo quando il campo è in modalità AND.                                                            | 3               | D               |
| Alternativo         | Include valori alternativi nel calcolo. Per la funzione equivalente, vedere <a href="#">GetAlternativeCount - funzione per grafici (page 1304)</a> .                                | 4               | Una             |

| Tipo di stato campo | Descrizione                                                                                                                                                       | Chiave numerica | Chiave testuale |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|-----------------|
| Escluso             | Include nel calcolo i valori esclusi (non selezionati). Per la funzione equivalente, vedere <a href="#">GetExcludedCount - funzione per grafici (page 1315)</a> . | 5               | X               |
| Selezionato escluso | Include nel calcolo i valori esclusi selezionati.                                                                                                                 | 6               | XS              |

## Casi di utilizzo

Con **GetStateCounts()**, è possibile calcolare uno stato di selezione personalizzato. La funzione consente di consolidare più chiamate di funzione in un'unica chiamata di funzione, semplificando il processo di scrittura dell'espressione.

Ad esempio, potrebbe essere necessario calcolare il numero totale di valori esclusi, alternativi e selezionati per un campo. Per calcolare questo totale è possibile utilizzare **GetStateCounts()**.

### Esempi: espressioni del grafico

| Esempi                                                                                                     | Risultati                                                                                                                                                                |
|------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>=GetStateCounts(ProductName, Null(), 'S')</code>                                                     | Restituisce il conteggio selezionato per <code>ProductName</code> , nello stato alternato ereditato.                                                                     |
| <code>=GetStateCounts(ProductName, '', 'X', 'A', 'XS')</code>                                              | Restituisce il conteggio totale dei valori esclusi, esclusi selezionati e alternativi per <code>ProductName</code> . Viene utilizzato lo stato alternato ereditato.      |
| <code>=GetStateCounts(ProductName, '', 'S', 'XS')</code>                                                   | Restituisce il conteggio totale delle selezioni dell'utente per <code>ProductName</code> , nello stato ereditato.                                                        |
| Il campo <i>ProductName</i> è in modalità and.<br><code>=GetStateCounts(ProductName, '', 'D', '-O')</code> | Restituisce il numero di valori non selezionati, sottratti dal numero di valori possibili, per <code>ProductName</code> . Viene utilizzato lo stato alternato ereditato. |
| <code>=GetStateCounts(ProductName, '', 'X', 'A', 'XS')</code>                                              | Restituisce il conteggio totale dei valori esclusi, esclusi selezionati e alternativi per <code>ProductName</code> . Viene utilizzato lo stato alternato ereditato.      |
| <code>=GetStateCounts(ProductName, '\$', 'O')</code>                                                       | Restituisce il conteggio possibile per <code>ProductName</code> , nello stato alternato predefinito.                                                                     |
| <code>=GetStateCounts(ProductName, 'StateA', 'S')</code>                                                   | Restituisce il conteggio selezionato di <code>ProductName</code> , nello stato alternato denominato <code>StateA</code> .                                                |

### Esempio 1 - Conteggio del numero totale di selezioni dell'utente (compresi i valori selezionati esclusi)

Espressione del grafico

#### Panoramica

La funzione **GetSelectedCount()** restituisce il numero totale di valori selezionati, senza includere il numero totale di valori esclusi. Con **GetExcludedCount()**, è possibile includere nel calcolo del grafico il conteggio dei valori esclusi selezionati.

Aprire Editor caricamento dati, quindi aggiungere il seguente script di caricamento in una nuova sezione.

#### Script di caricamento

```
Names:
LOAD * inline [
First name|Last name|Initials|Has cellphone
John|Anderson|JA|Yes
Sue|Brown|SB|Yes
Mark|Carr|MC|No
Peter|Devonshire|PD|No
Jane|Elliot|JE|Yes
Peter|Franc|PF|Yes ] (delimiter is '|');
```

#### Risultati

Procedere come indicato di seguito:

1. Caricare i dati e aprire un foglio.
2. Creare due caselle di filtro.
3. Aggiungere **First name** come dimensione nella prima casella di filtro. Aggiungere **Last name** come dimensione nella seconda casella di filtro.
4. Quindi, creare un KPI con questa misura:  
`=GetStateCounts([First Name], '', 'S', 'XS')`
5. Nella casella di filtro **First name**, selezionare i valori **Mark** e **Peter**. Nella casella di filtro **Last name**, selezionare i valori **Carr**.
6. Osservare che il KPI mostra un valore di 2. Questo riflette il fatto che tutti i valori selezionati, incluso il valore escluso **Peter**, sono stati conteggiati.

### Esempio 2 - Combinazione di conteggi esclusi, esclusi selezionati e alternativi

Espressioni del grafico

#### Panoramica

La funzione **GetExcludedCount()** restituisce il numero totale di valori unici non selezionati ed esclusi. Se si desidera includere nella visualizzazione i conteggi selezionati esclusi e alternativi, è possibile utilizzare la funzione **GetStateCounts()** come segue.

Aprire Editor caricamento dati, quindi aggiungere il seguente script di caricamento in una nuova sezione.

#### Script di caricamento

```
Names:
LOAD * inline [
First name|Last name|Initials|Has cellphone
John|Anderson|JA|Yes
Sue|Brown|SB|Yes
Mark|Carr|MC|No
Peter|Devonshire|PD|No
Jane|Elliot|JE|Yes
Peter|Franc|PF|Yes ] (delimiter is '|');
```

#### Risultati

Procedere come indicato di seguito:

1. Caricare i dati e aprire un foglio.
2. Creare due caselle di filtro.
3. Aggiungere **First name** come dimensione nella prima casella di filtro. Aggiungere **Last name** come dimensione nella seconda casella di filtro.
4. Quindi, creare un KPI con questa misura:  
`=GetStateCounts([First name], '', 'X', 'A', 'XS')`
5. Nella casella di filtro **First name**, selezionare i valori **John** e **Peter**. Nella casella di filtro **Last name**, selezionare i valori **Franc**.
6. Osservare che il KPI mostra un valore di 4. Ciò riflette il fatto che il valore escluso selezionato **John** è incluso nel conteggio, oltre al conteggio dei valori esclusi.
7. Modificare il KPI e sostituire la misura esistente con la seguente:  
`=GetStateCounts([Last name], '', 'X', 'A', 'XS')`
8. Mantenendo le selezioni esistenti, selezionare il valore **Anderson** nella casella di filtro **Last name**.
9. Osservare che il KPI mostra un valore di 4. Ciò riflette il fatto che il valore alternativo **Devonshire** è incluso nel conteggio, oltre al conteggio dei valori esclusi.

## 8.10 Funzioni di file

Le funzioni di file (disponibili solo nelle espressioni di script) restituiscono informazioni sul file tabellare in corso di lettura. Queste funzioni restituiranno un valore NULL per tutte le sorgenti dati tranne che per i file tabella (eccezione: **ConnectionString()**).

### Prospetto delle funzioni di file

Ciascuna funzione viene descritta ulteriormente dopo la panoramica. È inoltre possibile fare clic sul nome della funzione nella sintassi per accedere immediatamente ai dettagli per tale funzione specifica.

#### Attribute

Questa funzione di script restituisce il valore dei metatag di diversi file multimediali come testo. Sono supportati i formati di file seguenti: MP3, WMA, WMV, PNG e JPG. Se il file **filename** non esiste, il formato di file non è supportato oppure non è presente il metatag **attributename**, verrà restituito NULL.

```
Attribute (filename, attributename)
```

#### ConnectionString

La funzione **ConnectionString()** restituisce il nome della connessione dati attiva per le connessioni ODBC o OLE DB. La funzione restituisce una stringa vuota se non è stata eseguita alcuna istruzione **connect** o dopo un'istruzione **disconnect**.

```
ConnectionString ()
```

#### FileName

La funzione **FileName** restituisce una stringa contenente il nome del file tabella in corso di lettura, senza percorso o estensione.

```
FileName ()
```

#### FileDir

La funzione **FileDir** restituisce una stringa contenente il percorso della directory del file tabella in corso di lettura.

```
FileDir ()
```

#### FileExtension

La funzione **FileExtension** restituisce una stringa contenente l'estensione del file tabella in corso di lettura.

```
FileExtension ()
```

#### FullName

La funzione **FullName** restituisce una stringa contenente il nome del file tabella in corso di lettura, senza percorso ma includendo l'estensione.

```
FullName ()
```

### FilePath

La funzione **FilePath** restituisce una stringa contenente il percorso completo del file tabella in corso di lettura.

```
FilePath ()
```

### FileSize

La funzione **FileSize** restituisce un valore intero contenente le dimensioni in byte del file filename oppure, se non viene specificato alcun filename, del file tabella in corso di lettura.

```
FileSize ()
```

### FileTime

La funzione **FileTime** restituisce data e ora in formato UTC dell'ultima modifica apportata al file specificato. Se non viene specificato nessun file, la funzione restituisce la data e l'ora in formato UTC dell'ultima modifica apportata al file tabella attualmente letto.

```
FileTime ([ filename ])
```

### GetFolderPath

La funzione **GetFolderPath** restituisce il valore della funzione Microsoft Windows *SHGetFolderPath*. Questa funzione utilizza come input il nome di una cartella Microsoft Windows e restituisce il percorso completo della cartella.

```
GetFolderPath ()
```

### QvdCreateTime

Questa funzione di script restituisce l'intestazione XML relativa alla data e ora da un file QVD, se disponibile, altrimenti restituisce NULL. Nella data e ora, l'ora è fornita in UTC.

```
QvdCreateTime (filename)
```

### QvdFieldName

La funzione script restituisce il nome del numero campo **fieldno** in un file QVD. Se il campo non esiste, viene restituito NULL.

```
QvdFieldName (filename , fieldno)
```

### QvdNoOfFields

Questa funzione dello script restituisce il numero di campi all'interno di file QVD.

```
QvdNoOfFields (filename)
```

### QvdNoOfRecords

Questa funzione dello script restituisce il numero di record attualmente presente in un file QVD.

```
QvdNoOfRecords (filename)
```

### QvdTableName

Questa funzione di script restituisce il nome della tabella memorizzata in un file QVD.

```
QvdTableName (filename)
```

## Attribute

Questa funzione di script restituisce il valore dei metatag di diversi file multimediali come testo. Sono supportati i formati di file seguenti: MP3, WMA, WMV, PNG e JPG. Se il file **filename** non esiste, il formato di file non è supportato oppure non è presente il metatag **attributename**, verrà restituito NULL.

### Sintassi:

```
Attribute(filename, attributename)
```

È possibile leggere un elevato numero di metatag. Negli esempi di questo argomento viene mostrato quali tag è possibile leggere per i relativi tipi di file supportati.



*È possibile leggere solo i metatag salvati nel file in base alla specifica pertinente, ad esempio ID2v3 per file MP3 o EXIF per file JPG, non metainformazioni salvate in **Esplora file Windows**.*

### Argomenti:

#### Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| filename      | <p>Il nome di un file multimediale comprensivo del percorso, se necessario, come connessione dati di una cartella.</p> <p><b>Esempio: 'lib://Table Files'</b></p> <p>Nella modalità di creazione degli script legacy sono supportati anche i seguenti formati di percorso:</p> <ul style="list-style-type: none"> <li>• assoluto</li> </ul> <p><b>Esempio: c:\data\</b></p> <ul style="list-style-type: none"> <li>• relativo alla directory di lavoro dell'app Qlik Sense.</li> </ul> <p><b>Esempio: data\</b></p> |
| attributename | Il nome di un metatag.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

Negli esempi viene utilizzata la funzione **GetFolderPath** per trovare i percorsi dei file multimediali. Poiché la funzione **GetFolderPath** è supportata solo nella modalità legacy, è necessario sostituire i riferimenti a **GetFolderPath** con un percorso di connessione dei dati lib:// quando si utilizza questa funzione in modalità standard o in Qlik Sense SaaS.

[Restrizione dell'accesso al file system \(page 1919\)](#)

**Example 1: File MP3**

Lo script legge tutti i metatag MP3 possibili nella cartella *MyMusic*.

```
// Script to read MP3 meta tags
for each vExt in 'mp3'
for each vFoundFile in filelist( GetFolderPath('MyMusic') & '\*.' & vExt )
FileList:
LOAD FileLongName,
    subfield(FileLongName,'\',-1) as FileShortName,
    num(FileSize(FileLongName),'# ### ### ##',',',' ') as FileSize,
    FileTime(FileLongName) as FileTime,
    // ID3v1.0 and ID3v1.1 tags
    Attribute(FileLongName, 'Title') as Title,
    Attribute(FileLongName, 'Artist') as Artist,
    Attribute(FileLongName, 'Album') as Album,
    Attribute(FileLongName, 'Year') as Year,
    Attribute(FileLongName, 'Comment') as Comment,
    Attribute(FileLongName, 'Track') as Track,
    Attribute(FileLongName, 'Genre') as Genre,

    // ID3v2.3 tags
    Attribute(FileLongName, 'AENC') as AENC, // Audio encryption
    Attribute(FileLongName, 'APIC') as APIC, // Attached picture
    Attribute(FileLongName, 'COMM') as COMM, // Comments
    Attribute(FileLongName, 'COMR') as COMR, // Commercial frame
    Attribute(FileLongName, 'ENCR') as ENCR, // Encryption method registration
    Attribute(FileLongName, 'EQUA') as EQUA, // Equalization
    Attribute(FileLongName, 'ETCO') as ETCO, // Event timing codes
    Attribute(FileLongName, 'GEOB') as GEOB, // General encapsulated object
    Attribute(FileLongName, 'GRID') as GRID, // Group identification registration
    Attribute(FileLongName, 'IPLS') as IPLS, // Involved people list
    Attribute(FileLongName, 'LINK') as LINK, // Linked information
    Attribute(FileLongName, 'MCDI') as MCDI, // Music CD identifier
    Attribute(FileLongName, 'MLLT') as MLLT, // MPEG location lookup table
    Attribute(FileLongName, 'OWNE') as OWNE, // Ownership frame
    Attribute(FileLongName, 'PRIV') as PRIV, // Private frame
    Attribute(FileLongName, 'PCNT') as PCNT, // Play counter
    Attribute(FileLongName, 'POPM') as POPM, // Popularimeter

    Attribute(FileLongName, 'POSS') as POSS, // Position synchronisation frame
    Attribute(FileLongName, 'RBUF') as RBUF, // Recommended buffer size
    Attribute(FileLongName, 'RVAD') as RVAD, // Relative volume adjustment
    Attribute(FileLongName, 'RVRB') as RVRB, // Reverb
    Attribute(FileLongName, 'SYLT') as SYLT, // Synchronized lyric/text
    Attribute(FileLongName, 'SYTC') as SYTC, // Synchronized tempo codes
    Attribute(FileLongName, 'TALB') as TALB, // Album/Movie/Show title
    Attribute(FileLongName, 'TBPM') as TBPM, // BPM (beats per minute)
    Attribute(FileLongName, 'TCOM') as TCOM, // Composer
    Attribute(FileLongName, 'TCO') as TCO, // Content type
    Attribute(FileLongName, 'TCOP') as TCOP, // Copyright message
    Attribute(FileLongName, 'TDAT') as TDAT, // Date
    Attribute(FileLongName, 'TDLY') as TDLY, // Playlist delay
```

```
Attribute(FileLongName, 'TENC') as TENC, // Encoded by
Attribute(FileLongName, 'TEXT') as TEXT, // Lyricist/Text writer
Attribute(FileLongName, 'TFLT') as TFLT, // File type
Attribute(FileLongName, 'TIME') as TIME, // Time
Attribute(FileLongName, 'TIT1') as TIT1, // Content group description
Attribute(FileLongName, 'TIT2') as TIT2, // Title/songname/content description
Attribute(FileLongName, 'TIT3') as TIT3, // Subtitle/Description refinement
Attribute(FileLongName, 'TKEY') as TKEY, // Initial key
Attribute(FileLongName, 'TLAN') as TLAN, // Language(s)
Attribute(FileLongName, 'TLEN') as TLEN, // Length
Attribute(FileLongName, 'TMED') as TMED, // Media type

Attribute(FileLongName, 'TOAL') as TOAL, // Original album/movie/show title
Attribute(FileLongName, 'TOFN') as TOFN, // Original filename
Attribute(FileLongName, 'TOLY') as TOLY, // Original lyricist(s)/text writer(s)
Attribute(FileLongName, 'TOPE') as TOPE, // Original artist(s)/performer(s)
Attribute(FileLongName, 'TORY') as TORY, // Original release year
Attribute(FileLongName, 'TOWN') as TOWN, // File owner/licensee
Attribute(FileLongName, 'TPE1') as TPE1, // Lead performer(s)/Soloist(s)
Attribute(FileLongName, 'TPE2') as TPE2, // Band/orchestra/accompaniment

Attribute(FileLongName, 'TPE3') as TPE3, // Conductor/performer refinement
Attribute(FileLongName, 'TPE4') as TPE4, // Interpreted, remixed, or otherwise modified by
Attribute(FileLongName, 'TPOS') as TPOS, // Part of a set
Attribute(FileLongName, 'TPUB') as TPUB, // Publisher
Attribute(FileLongName, 'TRCK') as TRCK, // Track number/Position in set
Attribute(FileLongName, 'TRDA') as TRDA, // Recording dates
Attribute(FileLongName, 'TRSN') as TRSN, // Internet radio station name
Attribute(FileLongName, 'TRSO') as TRSO, // Internet radio station owner

Attribute(FileLongName, 'TSIZ') as TSIZ, // Size
Attribute(FileLongName, 'TSRC') as TSRC, // ISRC (international standard recording code)
Attribute(FileLongName, 'TSSE') as TSSE, // Software/Hardware and settings used for
encoding
Attribute(FileLongName, 'TYER') as TYER, // Year
Attribute(FileLongName, 'TXXX') as TXXX, // User defined text information frame
Attribute(FileLongName, 'UFID') as UFID, // Unique file identifier
Attribute(FileLongName, 'USER') as USER, // Terms of use
Attribute(FileLongName, 'USLT') as USLT, // Unsynchronized lyric/text transcription
Attribute(FileLongName, 'WCOM') as WCOM, // Commercial information
Attribute(FileLongName, 'WCOP') as WCOP, // Copyright/Legal information

Attribute(FileLongName, 'WOAF') as WOAF, // Official audio file webpage
Attribute(FileLongName, 'WOAR') as WOAR, // Official artist/performer webpage
Attribute(FileLongName, 'WOAS') as WOAS, // Official audio source webpage
Attribute(FileLongName, 'WORS') as WORS, // Official internet radio station homepage
Attribute(FileLongName, 'WPAY') as WPAY, // Payment
Attribute(FileLongName, 'WPUB') as WPUB, // Publishers official webpage
Attribute(FileLongName, 'WXXX') as WXXX; // User defined URL link frame
LOAD @1:n as FileLongName Inline "$ (vFoundFile)" (fix, no labels);
Next vFoundFile
Next vExt
```

### Example 2: JPEG

Lo script legge tutti i metatag EXIF possibili dai file JPG nella cartella *MyPictures*.

```
// Script to read Jpeg Exif meta tags
for each vExt in 'jpg', 'jpeg', 'jpe', 'jfif', 'jif', 'jfi'
for each vFoundFile in filelist( GetFolderPath('MyPictures') & '\*.' & vExt )

FileList:
LOAD FileLongName,
    subfield(FileLongName,'\\',-1) as FileShortName,
    num(FileSize(FileLongName),'# ### ##',',',' ') as FileSize,
    FileTime(FileLongName) as FileTime,
    // ***** Exif Main (IFD0) Attributes *****
    Attribute(FileLongName, 'ImageWidth') as ImageWidth,
    Attribute(FileLongName, 'ImageLength') as ImageLength,
    Attribute(FileLongName, 'BitsPerSample') as BitsPerSample,
    Attribute(FileLongName, 'Compression') as Compression,

    // examples: 1=uncompressed, 2=CCITT, 3=CCITT 3, 4=CCITT 4,

    // 5=LZW, 6=JPEG (old style), 7=JPEG, 8=Deflate, 32773=PackBits RLE,
    Attribute(FileLongName, 'PhotometricInterpretation') as PhotometricInterpretation,

    // examples: 0=whiteIsZero, 1=BlackIsZero, 2=RGB, 3=Palette, 5=CMYK, 6=YCbCr,
    Attribute(FileLongName, 'ImageDescription') as ImageDescription,
    Attribute(FileLongName, 'Make') as Make,
    Attribute(FileLongName, 'Model') as Model,
    Attribute(FileLongName, 'StripOffsets') as StripOffsets,
    Attribute(FileLongName, 'Orientation') as Orientation,

    // examples: 1=TopLeft, 2=TopRight, 3=BottomRight, 4=BottomLeft,

    // 5=LeftTop, 6=RightTop, 7=RightBottom, 8=LeftBottom,
    Attribute(FileLongName, 'SamplesPerPixel') as SamplesPerPixel,
    Attribute(FileLongName, 'RowsPerStrip') as RowsPerStrip,
    Attribute(FileLongName, 'StripByteCounts') as StripByteCounts,
    Attribute(FileLongName, 'XResolution') as XResolution,
    Attribute(FileLongName, 'YResolution') as YResolution,
    Attribute(FileLongName, 'PlanarConfiguration') as PlanarConfiguration,

    // examples: 1=chunky format, 2=planar format,
    Attribute(FileLongName, 'ResolutionUnit') as ResolutionUnit,

    // examples: 1=none, 2=inches, 3=centimeters,
    Attribute(FileLongName, 'TransferFunction') as TransferFunction,
    Attribute(FileLongName, 'Software') as Software,
    Attribute(FileLongName, 'DateTime') as DateTime,
    Attribute(FileLongName, 'Artist') as Artist,
    Attribute(FileLongName, 'HostComputer') as HostComputer,
    Attribute(FileLongName, 'WhitePoint') as WhitePoint,
    Attribute(FileLongName, 'PrimaryChromaticities') as PrimaryChromaticities,
    Attribute(FileLongName, 'YCbCrCoefficients') as YCbCrCoefficients,
```

```
Attribute(FileLongName, 'YCbCrSubSampling') as YCbCrSubSampling,
Attribute(FileLongName, 'YCbCrPositioning') as YCbCrPositioning,

// examples: 1=centered, 2=co-sited,
Attribute(FileLongName, 'ReferenceBlackWhite') as ReferenceBlackWhite,
Attribute(FileLongName, 'Rating') as Rating,
Attribute(FileLongName, 'RatingPercent') as RatingPercent,
Attribute(FileLongName, 'ThumbnailFormat') as ThumbnailFormat,

// examples: 0=Raw Rgb, 1=Jpeg,
Attribute(FileLongName, 'Copyright') as Copyright,
Attribute(FileLongName, 'ExposureTime') as ExposureTime,
Attribute(FileLongName, 'FNumber') as FNumber,
Attribute(FileLongName, 'ExposureProgram') as ExposureProgram,

// examples: 0=Not defined, 1=Manual, 2=Normal program, 3=Aperture priority, 4=Shutter
priority,

// 5=Creative program, 6=Action program, 7=Portrait mode, 8=Landscape mode, 9=Bulb,
Attribute(FileLongName, 'ISOSpeedRatings') as ISOSpeedRatings,
Attribute(FileLongName, 'TimeZoneOffset') as TimeZoneOffset,
Attribute(FileLongName, 'SensitivityType') as SensitivityType,

// examples: 0=Unknown, 1=Standard output sensitivity (SOS), 2=Recommended exposure index
(REI),

// 3=ISO speed, 4=Standard output sensitivity (SOS) and Recommended exposure index (REI),

//5=Standard output sensitivity (SOS) and ISO Speed, 6=Recommended exposure index (REI)
and ISO Speed,

// 7=Standard output sensitivity (SOS) and Recommended exposure index (REI) and ISO speed,
Attribute(FileLongName, 'ExifVersion') as ExifVersion,
Attribute(FileLongName, 'DateTimeOriginal') as DateTimeOriginal,
Attribute(FileLongName, 'DateTimeDigitized') as DateTimeDigitized,
Attribute(FileLongName, 'ComponentsConfiguration') as ComponentsConfiguration,

// examples: 1=Y, 2=Cb, 3=Cr, 4=R, 5=G, 6=B,
Attribute(FileLongName, 'CompressedBitsPerPixel') as CompressedBitsPerPixel,
Attribute(FileLongName, 'ShutterSpeedValue') as ShutterSpeedValue,
Attribute(FileLongName, 'ApertureValue') as ApertureValue,
Attribute(FileLongName, 'BrightnessValue') as BrightnessValue, // examples: -1=Unknown,
Attribute(FileLongName, 'ExposureBiasValue') as ExposureBiasValue,
Attribute(FileLongName, 'MaxApertureValue') as MaxApertureValue,
Attribute(FileLongName, 'SubjectDistance') as SubjectDistance,

// examples: 0=Unknown, -1=Infinity,
Attribute(FileLongName, 'MeteringMode') as MeteringMode,

// examples: 0=Unknown, 1=Average, 2=CenterWeightedAverage, 3=Spot,

// 4=MultiSpot, 5=Pattern, 6=Partial, 255=Other,
Attribute(FileLongName, 'LightSource') as LightSource,

// examples: 0=Unknown, 1=Daylight, 2=Fluorescent, 3=Tungsten, 4=Flash, 9=Fine weather,
```

```
// 10=Cloudy weather, 11=Shade, 12=Daylight fluorescent,

// 13=Day white fluorescent, 14=Cool white fluorescent,

// 15=white fluorescent, 17=Standard light A, 18=Standard light B, 19=Standard light C,

// 20=D55, 21=D65, 22=D75, 23=D50, 24=ISO studio tungsten, 255=other light source,
Attribute(FileLongName, 'Flash') as Flash,
Attribute(FileLongName, 'FocalLength') as FocalLength,
Attribute(FileLongName, 'SubjectArea') as SubjectArea,
Attribute(FileLongName, 'MakerNote') as MakerNote,
Attribute(FileLongName, 'UserComment') as UserComment,
Attribute(FileLongName, 'SubSecTime') as SubSecTime,

Attribute(FileLongName, 'SubsecTimeOriginal') as SubsecTimeOriginal,
Attribute(FileLongName, 'SubsecTimeDigitized') as SubsecTimeDigitized,
Attribute(FileLongName, 'XPTitle') as XPTitle,
Attribute(FileLongName, 'XPComment') as XPComment,

Attribute(FileLongName, 'XPAuthor') as XPAuthor,
Attribute(FileLongName, 'XPKeywords') as XPKeywords,
Attribute(FileLongName, 'XPSubject') as XPSubject,
Attribute(FileLongName, 'FlashpixVersion') as FlashpixVersion,
Attribute(FileLongName, 'ColorSpace') as ColorSpace, // examples: 1=sRGB,
65535=Uncalibrated,
Attribute(FileLongName, 'PixelXDimension') as PixelXDimension,
Attribute(FileLongName, 'PixelYDimension') as PixelYDimension,
Attribute(FileLongName, 'RelatedSoundFile') as RelatedSoundFile,

Attribute(FileLongName, 'FocalPlaneXResolution') as FocalPlaneXResolution,
Attribute(FileLongName, 'FocalPlaneYResolution') as FocalPlaneYResolution,
Attribute(FileLongName, 'FocalPlaneResolutionUnit') as FocalPlaneResolutionUnit,

// examples: 1=None, 2=Inch, 3=Centimeter,
Attribute(FileLongName, 'ExposureIndex') as ExposureIndex,
Attribute(FileLongName, 'SensingMethod') as SensingMethod,

// examples: 1=Not defined, 2=One-chip color area sensor, 3=Two-chip color area sensor,

// 4=Three-chip color area sensor, 5=Color sequential area sensor,

// 7=Trilinear sensor, 8=Color sequential linear sensor,
Attribute(FileLongName, 'FileSource') as FileSource,

// examples: 0=Other, 1=Scanner of transparent type,

// 2=Scanner of reflex type, 3=Digital still camera,
Attribute(FileLongName, 'SceneType') as SceneType,

// examples: 1=A directly photographed image,
Attribute(FileLongName, 'CFAPattern') as CFAPattern,
Attribute(FileLongName, 'CustomRendered') as CustomRendered,

// examples: 0=Normal process, 1=Custom process,
Attribute(FileLongName, 'ExposureMode') as ExposureMode,
```

```
// examples: 0=Auto exposure, 1=Manual exposure, 2=Auto bracket,
Attribute(FileLongName, 'WhiteBalance') as WhiteBalance,

// examples: 0=Auto white balance, 1=Manual white balance,
Attribute(FileLongName, 'DigitalZoomRatio') as DigitalZoomRatio,
Attribute(FileLongName, 'FocalLengthIn35mmFilm') as FocalLengthIn35mmFilm,
Attribute(FileLongName, 'SceneCaptureType') as SceneCaptureType,

// examples: 0=Standard, 1=Landscape, 2=Portrait, 3=Night scene,
Attribute(FileLongName, 'GainControl') as GainControl,

// examples: 0=None, 1=Low gain up, 2=High gain up, 3=Low gain down, 4=High gain down,
Attribute(FileLongName, 'Contrast') as Contrast,

// examples: 0=Normal, 1=Soft, 2=Hard,
Attribute(FileLongName, 'Saturation') as Saturation,

// examples: 0=Normal, 1=Low saturation, 2=High saturation,
Attribute(FileLongName, 'Sharpness') as Sharpness,

// examples: 0=Normal, 1=Soft, 2=Hard,
Attribute(FileLongName, 'SubjectDistanceRange') as SubjectDistanceRange,

// examples: 0=Unknown, 1=Macro, 2=Close view, 3=Distant view,
Attribute(FileLongName, 'ImageUniqueID') as ImageUniqueID,
Attribute(FileLongName, 'BodySerialNumber') as BodySerialNumber,
Attribute(FileLongName, 'CMNT_GAMMA') as CMNT_GAMMA,
Attribute(FileLongName, 'PrintImageMatching') as PrintImageMatching,
Attribute(FileLongName, 'OffsetSchema') as OffsetSchema,

// ***** Interoperability Attributes *****
Attribute(FileLongName, 'InteroperabilityIndex') as InteroperabilityIndex,
Attribute(FileLongName, 'InteroperabilityVersion') as InteroperabilityVersion,
Attribute(FileLongName, 'InteroperabilityRelatedImageFileFormat') as
InteroperabilityRelatedImageFileFormat,
Attribute(FileLongName, 'InteroperabilityRelatedImageWidth') as
InteroperabilityRelatedImageWidth,
Attribute(FileLongName, 'InteroperabilityRelatedImageLength') as
InteroperabilityRelatedImageLength,
Attribute(FileLongName, 'InteroperabilityColorSpace') as InteroperabilityColorSpace,

// examples: 1=sRGB, 65535=Uncalibrated,
Attribute(FileLongName, 'InteroperabilityPrintImageMatching') as
InteroperabilityPrintImageMatching,
// ***** GPS Attributes *****
Attribute(FileLongName, 'GPSVersionID') as GPSVersionID,
Attribute(FileLongName, 'GPSLatitudeRef') as GPSLatitudeRef,
Attribute(FileLongName, 'GPSLatitude') as GPSLatitude,
Attribute(FileLongName, 'GPSLongitudeRef') as GPSLongitudeRef,
Attribute(FileLongName, 'GPSLongitude') as GPSLongitude,
Attribute(FileLongName, 'GPSAltitudeRef') as GPSAltitudeRef,

// examples: 0=Above sea level, 1=Below sea level,
Attribute(FileLongName, 'GPSAltitude') as GPSAltitude,
Attribute(FileLongName, 'GPSTimeStamp') as GPSTimeStamp,
```

```
Attribute(FileLongName, 'GPSSatellites') as GPSSatellites,
Attribute(FileLongName, 'GPSStatus') as GPSStatus,
Attribute(FileLongName, 'GPSMeasureMode') as GPSMeasureMode,
Attribute(FileLongName, 'GPSDOP') as GPSDOP,
Attribute(FileLongName, 'GPSSpeedRef') as GPSSpeedRef,

Attribute(FileLongName, 'GPSSpeed') as GPSSpeed,
Attribute(FileLongName, 'GPSTrackRef') as GPSTrackRef,
Attribute(FileLongName, 'GPSTrack') as GPSTrack,
Attribute(FileLongName, 'GPSImgDirectionRef') as GPSImgDirectionRef,
Attribute(FileLongName, 'GPSImgDirection') as GPSImgDirection,
Attribute(FileLongName, 'GPSMapDatum') as GPSMapDatum,
Attribute(FileLongName, 'GPSDestLatitudeRef') as GPSDestLatitudeRef,

Attribute(FileLongName, 'GPSDestLatitude') as GPSDestLatitude,
Attribute(FileLongName, 'GPSDestLongitudeRef') as GPSDestLongitudeRef,
Attribute(FileLongName, 'GPSDestLongitude') as GPSDestLongitude,
Attribute(FileLongName, 'GPSDestBearingRef') as GPSDestBearingRef,
Attribute(FileLongName, 'GPSDestBearing') as GPSDestBearing,
Attribute(FileLongName, 'GPSDestDistanceRef') as GPSDestDistanceRef,

Attribute(FileLongName, 'GPSDestDistance') as GPSDestDistance,
Attribute(FileLongName, 'GPSProcessingMethod') as GPSProcessingMethod,
Attribute(FileLongName, 'GPSAreaInformation') as GPSAreaInformation,
Attribute(FileLongName, 'GPSDateStamp') as GPSDateStamp,
Attribute(FileLongName, 'GPSDifferential') as GPSDifferential;

// examples: 0=No correction, 1=Differential correction,
LOAD @1:n as FileLongName Inline "$(vFoundFile)" (fix, no labels);
Next vFoundFile
Next vExt
```

### Example 3: File multimediali di Windows

Lo script legge tutti i metatag WMA/WMV ASF possibili nella cartella *MyMusic*.

```
/ Script to read WMA/WMV ASF meta tags
for each vExt in 'asf', 'wma', 'wmv'
for each vFoundFile in filelist( GetFolderPath('MyMusic') & '\*.' & vExt )

FileList:
LOAD FileLongName,
    subfield(FileLongName,'\ ', -1) as FileShortName,
    num(FileSize(FileLongName), '# ### ##') as FileSize,
    FileTime(FileLongName) as FileTime,
    Attribute(FileLongName, 'Title') as Title,
    Attribute(FileLongName, 'Author') as Author,
    Attribute(FileLongName, 'Copyright') as Copyright,
    Attribute(FileLongName, 'Description') as Description,

    Attribute(FileLongName, 'Rating') as Rating,
    Attribute(FileLongName, 'PlayDuration') as PlayDuration,
    Attribute(FileLongName, 'MaximumBitrate') as MaximumBitrate,
    Attribute(FileLongName, 'WMFSDKVersion') as WMFSDKVersion,
    Attribute(FileLongName, 'WMFSDKNeeded') as WMFSDKNeeded,
```

```
Attribute(FileLongName, 'IsVBR') as IsVBR,
Attribute(FileLongName, 'ASFLeakyBucketPairs') as ASFLeakyBucketPairs,

Attribute(FileLongName, 'PeakValue') as PeakValue,
Attribute(FileLongName, 'AverageLevel') as AverageLevel;
LOAD @1:n as FileLongName Inline "$(vFoundFile)" (fix, no labels);
Next vFoundFile
Next vExt
```

### Example 4: PNG

Lo script legge tutti i metatag PNG possibili nella cartella *MyPictures*.

```
// Script to read PNG meta tags
for each vExt in 'png'
for each vFoundFile in filelist( GetFolderPath('MyPictures') & '\*.' & vExt )

FileList:
LOAD FileLongName,
    subfield(FileLongName, '\', -1) as FileShortName,
    num(FileSize(FileLongName), '# ### ### ##', ',', ' ') as FileSize,
    FileTime(FileLongName) as FileTime,
    Attribute(FileLongName, 'Comment') as Comment,

    Attribute(FileLongName, 'Creation Time') as Creation_Time,
    Attribute(FileLongName, 'Source') as Source,
    Attribute(FileLongName, 'Title') as Title,
    Attribute(FileLongName, 'Software') as Software,
    Attribute(FileLongName, 'Author') as Author,
    Attribute(FileLongName, 'Description') as Description,

    Attribute(FileLongName, 'Copyright') as Copyright;
LOAD @1:n as FileLongName Inline "$(vFoundFile)" (fix, no labels);
Next vFoundFile
Next vExt
```

## ConnectString

La funzione **ConnectString()** restituisce il nome della connessione dati attiva per le connessioni ODBC o OLE DB. La funzione restituisce una stringa vuota se non è stata eseguita alcuna istruzione **connect** o dopo un'istruzione **disconnect**.

**Sintassi:**

```
ConnectString()
```

Esempi e risultati:

Esempi di script

| Esempio                                                                                                              | Risultato                                                                                                                                                                |
|----------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>LIB CONNECT TO 'Tutorial ODBC';  ConnectionString:  Load ConnectString() as ConnectString AutoGenerate 1;</pre> | <p>Restituisce 'Tutorial ODBC' nel campo ConnectString.</p> <p>Nel seguente esempio si presuppone che sia disponibile una connessione dati denominata Tutorial ODBC.</p> |

## FileName

La funzione **FileName** restituisce una stringa contenente il nome del file tabella in corso di lettura, senza percorso o estensione.

Sintassi:

```
FileName()
```

Esempi e risultati:

Esempi di script

| Esempio                                                       | Risultato                                                    |
|---------------------------------------------------------------|--------------------------------------------------------------|
| <pre>LOAD *, filename( ) as X from C:\UserFiles\abc.txt</pre> | Restituisce 'abc' nel campo X per la lettura di ogni record. |

## FileDir

La funzione **FileDir** restituisce una stringa contenente il percorso della directory del file tabella in corso di lettura.

Sintassi:

```
FileDir()
```



*Questa funzione supporta solo le connessioni dati della cartella in modalità standard.*

Esempi e risultati:

Esempi di script

| Esempio                                                      | Risultato                                                    |
|--------------------------------------------------------------|--------------------------------------------------------------|
| <pre>Load *, filedir( ) as X from C:\UserFiles\abc.txt</pre> | Restituisce 'C:\UserFiles' nel campo X in ogni record letto. |

## FileExtension

La funzione **FileExtension** restituisce una stringa contenente l'estensione del file tabella in corso di lettura.

Sintassi:

```
FileExtension()
```

Esempi e risultati:

Esempi di script

| Esempio                                                        | Risultato                                                    |
|----------------------------------------------------------------|--------------------------------------------------------------|
| LOAD *, FileExtension( ) as X from<br><br>C:\UserFiles\abc.txt | Restituisce 'txt' nel campo X per la lettura di ogni record. |

## FileName

La funzione **FileName** restituisce una stringa contenente il nome del file tabella in corso di lettura, senza percorso ma includendo l'estensione.

Sintassi:

```
FileName()
```

Esempi e risultati:

Esempi di script

| Esempio                                                   | Risultato                                               |
|-----------------------------------------------------------|---------------------------------------------------------|
| LOAD *, FileName( ) as X from<br><br>C:\UserFiles\abc.txt | Restituisce 'abc.txt' nel campo X in ogni record letto. |

## FilePath

La funzione **FilePath** restituisce una stringa contenente il percorso completo del file tabella in corso di lettura.

Sintassi:

```
FilePath()
```



*Questa funzione supporta solo le connessioni dati della cartella in modalità standard.*

Esempi e risultati:

Esempi di script

| Esempio                                                   | Risultato                                                            |
|-----------------------------------------------------------|----------------------------------------------------------------------|
| Load *, FilePath( ) as X from<br><br>C:\UserFiles\abc.txt | Restituisce 'C:\UserFiles\abc.txt' nel campo X in ogni record letto. |

### FileSize

La funzione **FileSize** restituisce un valore intero contenente le dimensioni in byte del file filename oppure, se non viene specificato alcun filename, del file tabella in corso di lettura.

**Sintassi:**

**FileSize**( [filename] )

**Argomenti:**

Argomenti

| Argomento | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| filename  | <p>Il nome di un file, se necessario incluso il percorso, come la connessione dati di una cartella o di un file Web. Se non si specifica un nome file, viene utilizzato il file tabella attualmente letto.</p> <p><b>Esempio: 'lib://Table Files/'</b></p> <p>Nella modalità di creazione degli script legacy sono supportati anche i seguenti formati di percorso:</p> <ul style="list-style-type: none"> <li>assoluto</li> </ul> <p><b>Esempio: c:\data\</b></p> <ul style="list-style-type: none"> <li>relativo alla directory di lavoro dell'app Qlik Sense.</li> </ul> <p><b>Esempio: data\</b></p> <ul style="list-style-type: none"> <li>indirizzo dell'URL (HTTP o FTP), che punta a una posizione in Internet o su una Intranet</li> </ul> <p><b>Esempio: http://www.qlik.com</b></p> |

Esempi e risultati:

Esempi di script

| Esempio                                   | Risultato                                                                                                                 |
|-------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| LOAD *, FileSize( ) as X<br>from abc.txt; | Restituisce la dimensione del file specificato (abc.txt) come un numero intero nel campo X per la lettura di ogni record. |
| FileSize(<br>'lib://DataFiles/xyz.xls' )  | Restituisce la dimensione del file xyz.xls.                                                                               |

## FileTime

La funzione **FileTime** restituisce data e ora in formato UTC dell'ultima modifica apportata al file specificato. Se non viene specificato nessun file, la funzione restituisce la data e l'ora in formato UTC dell'ultima modifica apportata al file tabella attualmente letto.

Sintassi:

```
FileTime( [ filename ] )
```

Argomenti:

Argomenti

| Argomento | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| filename  | <p>Il nome di un file, con il relativo percorso se necessario, come connessione dati di una cartella o di un file Web.</p> <p><b>Esempio: 'lib://Table Files/'</b></p> <p>Nella modalità di creazione degli script legacy sono supportati anche i seguenti formati di percorso:</p> <ul style="list-style-type: none"> <li>• assoluto</li> </ul> <p><b>Esempio: c:\data\</b></p> <ul style="list-style-type: none"> <li>• relativo alla directory di lavoro dell'app Qlik Sense.</li> </ul> <p><b>Esempio: data\</b></p> <ul style="list-style-type: none"> <li>• indirizzo dell'URL (HTTP o FTP), che punta a una posizione in Internet o su una Intranet</li> </ul> <p><b>Esempio: http://www.qlik.com</b></p> |

## Esempi e risultati:

## Esempi di script

| Esempio                                                 | Risultato                                                                                                   |
|---------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| <code>LOAD *, FileTime( ) as X<br/>from abc.txt;</code> | Restituisce il timestamp dell'ultima modifica del file (abc.txt) nel campo X per la lettura di ogni record. |
| <code>FileTime( 'xyz.xls' )</code>                      | Restituisce l'indicazione di data e ora dell'ultima modifica del file xyz.xls.                              |

## GetFolderPath

La funzione **GetFolderPath** restituisce il valore della funzione Microsoft Windows *SHGetFolderPath*. Questa funzione utilizza come input il nome di una cartella Microsoft Windows e restituisce il percorso completo della cartella.



*Questa funzione non è supportata in modalità standard. .*

## Sintassi:

```
GetFolderPath(foldername)
```

## Argomenti:

## Argomenti

| Argomento         | Descrizione                                                                                                                                                                                                                                                   |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>foldername</b> | <p>Nome della cartella Microsoft Windows.</p> <p>Il nome della cartella non deve contenere spazi. Occorre rimuovere qualsiasi spazio presente nel nome della cartella in Windows Explorer.</p> <p>Esempi:</p> <p><i>MyMusic</i></p> <p><i>MyDocuments</i></p> |

## Esempi e risultati:

Lo scopo di questo esempio è ottenere i percorsi delle seguenti cartelle Microsoft Windows: *MyMusic*, *MyPictures* e *Windows*. Aggiungere lo script di esempio all'app e ricaricarla.

```
LOAD
  GetFolderPath('MyMusic') as MyMusic,
  GetFolderPath('MyPictures') as MyPictures,
  GetFolderPath('windows') as windows
AutoGenerate 1;
```

Una volta ricaricata l'app, i campi *MyMusic*, *MyPictures* e *Windows* vengono aggiunti al modello dati. Ogni campo contiene il percorso della cartella definita nell'input. Ad esempio:

- *C:\Users\smul\Music* for the folder *MyMusic*
- *C:\Users\smul\Pictures* for the folder *MyPictures*
- *C:\Windows* for the folder *Windows*

### QvdCreateTime

Questa funzione di script restituisce l'intestazione XML relativa alla data e ora da un file QVD, se disponibile, altrimenti restituisce NULL. Nella data e ora, l'ora è fornita in UTC.

Sintassi:

```
QvdCreateTime(filename)
```

Argomenti:

Argomenti

| Argomento | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| filename  | <p>Il nome di un file QVD, includendo il percorso, se necessario come connessione dati della cartella o Web.</p> <p><b>Esempio: 'lib://Table Files/'</b></p> <p>Nella modalità di creazione degli script legacy sono supportati anche i seguenti formati di percorso:</p> <ul style="list-style-type: none"><li>• assoluto</li></ul> <p><b>Esempio: c:\data\</b></p> <ul style="list-style-type: none"><li>• relativo alla directory di lavoro dell'app Qlik Sense.</li></ul> <p><b>Esempio: data\</b></p> <ul style="list-style-type: none"><li>• indirizzo dell'URL (HTTP o FTP), che punta a una posizione in Internet o su una Intranet</li></ul> <p><b>Esempio: http://www.qlik.com</b></p> |

Esempio:

```
QvdCreateTime('MyFile.qvd')
```

```
QvdCreateTime('C:\MyDir\MyFile.qvd')
```

```
QvdCreateTime('lib://DataFiles/MyFile.qvd')
```

## QvdFieldName

La funzione script restituisce il nome del numero campo **fieldno** in un file QVD. Se il campo non esiste, viene restituito NULL.

Sintassi:

```
QvdFieldName(filename , fieldno)
```

Argomenti:

| Argomenti |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Argomento | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| filename  | <p>Il nome di un file QVD, includendo il percorso, se necessario come connessione dati della cartella o Web.</p> <p><b>Esempio: 'lib://Table Files/'</b></p> <p>Nella modalità di creazione degli script legacy sono supportati anche i seguenti formati di percorso:</p> <ul style="list-style-type: none"><li>• assoluto</li></ul> <p><b>Esempio: c:\data\</b></p> <ul style="list-style-type: none"><li>• relativo alla directory di lavoro dell'app Qlik Sense.</li></ul> <p><b>Esempio: data\</b></p> <ul style="list-style-type: none"><li>• indirizzo dell'URL (HTTP o FTP), che punta a una posizione in Internet o su una Intranet</li></ul> <p><b>Esempio: http://www.qlik.com</b></p> |
| fieldno   | Il numero del campo all'interno della tabella contenuta nel file QVD.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

Esempi:

```
QvdFieldName ('MyFile.qvd', 5)
```

```
QvdFieldName ('C:\MyDir\MyFile.qvd', 5)
```

```
QvdFieldName ('lib://DataFiles/MyFile.qvd', 5)
```

Tutti e tre gli esempi restituiscono il nome del quinto campo della tabella contenuta nel file QVD.

## QvdNoOfFields

Questa funzione dello script restituisce il numero di campi all'interno di file QVD.

### Sintassi:

```
QvdNoOfFields (filename)
```

### Argomenti:

| Argomenti |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Argomento | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| filename  | <p>Il nome di un file QVD, includendo il percorso, se necessario come connessione dati della cartella o Web.</p> <p><b>Esempio: 'lib://Table Files/'</b></p> <p>Nella modalità di creazione degli script legacy sono supportati anche i seguenti formati di percorso:</p> <ul style="list-style-type: none"><li>• assoluto</li></ul> <p><b>Esempio: c:\data\</b></p> <ul style="list-style-type: none"><li>• relativo alla directory di lavoro dell'app Qlik Sense.</li></ul> <p><b>Esempio: data\</b></p> <ul style="list-style-type: none"><li>• indirizzo dell'URL (HTTP o FTP), che punta a una posizione in Internet o su una Intranet</li></ul> <p><b>Esempio: http://www.qlik.com</b></p> |

### Esempi:

```
QvdNoOfFields ('MyFile.qvd')
```

```
QvdNoOfFields ('C:\MyDir\MyFile.qvd')
```

```
QvdNoOfFields ('lib://DataFiles/MyFile.qvd')
```

## QvdNoOfRecords

**Esempio:** Questa funzione dello script restituisce il numero di record attualmente presente in un file QVD.

### Sintassi:

```
QvdNoOfRecords (filename)
```

**Argomenti:**

| Argomenti |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Argomento | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| filename  | <p>Il nome di un file QVD, includendo il percorso, se necessario come connessione dati della cartella o Web.</p> <p><b>Esempio: <i>'lib://Table Files'</i></b></p> <p>Nella modalità di creazione degli script legacy sono supportati anche i seguenti formati di percorso:</p> <ul style="list-style-type: none"> <li>• assoluto</li> </ul> <p><b>Esempio: <i>c:\data\</i></b></p> <ul style="list-style-type: none"> <li>• relativo alla directory di lavoro dell'app Qlik Sense.</li> </ul> <p><b>Esempio: <i>data\</i></b></p> <ul style="list-style-type: none"> <li>• indirizzo dell'URL (HTTP o FTP), che punta a una posizione in Internet o su una Intranet</li> </ul> <p><b>Esempio: <i>http://www.qlik.com</i></b></p> |

**Esempi:**

```
QvdNoOfRecords ('MyFile.qvd')
```

```
QvdNoOfRecords ('C:\MyDir\MyFile.qvd')
```

```
QvdNoOfRecords ('lib://DataFiles/MyFile.qvd')
```

**QvdTableName**

Questa funzione di script restituisce il nome della tabella memorizzata in un file QVD.

**Sintassi:**

```
QvdTableName (filename)
```

## Argomenti:

| Argomenti |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Argomento | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| filename  | <p>Il nome di un file QVD, includendo il percorso, se necessario come connessione dati della cartella o Web.</p> <p><b>Esempio: 'lib://Table Files/'</b></p> <p>Nella modalità di creazione degli script legacy sono supportati anche i seguenti formati di percorso:</p> <ul style="list-style-type: none"> <li>• assoluto</li> </ul> <p><b>Esempio: c:\data\</b></p> <ul style="list-style-type: none"> <li>• relativo alla directory di lavoro dell'app Qlik Sense.</li> </ul> <p><b>Esempio: data\</b></p> <ul style="list-style-type: none"> <li>• indirizzo dell'URL (HTTP o FTP), che punta a una posizione in Internet o su una Intranet</li> </ul> <p><b>Esempio: http://www.qlik.com</b></p> |

## Esempi:

```
QvdTableName ('MyFile.qvd')
```

```
QvdTableName ('C:\MyDir\MyFile.qvd')
```

```
QvdTableName ('lib://data\MyFile.qvd')
```

## 8.11 Funzioni finanziarie

Le funzioni finanziarie possono essere utilizzate nello script di caricamento dei dati e nelle espressioni grafiche per calcolare i pagamenti e i tassi di interesse.

Per tutti gli argomenti, le uscite sono rappresentate da numeri negativi. Il denaro in entrata viene rappresentato da numeri positivi.

Di seguito è riportato un elenco degli argomenti utilizzati nelle funzioni finanziarie (tranne quelli che iniziano con **range-**).



*Per tutte le funzioni finanziarie è importante essere coerenti quando si specificano le unità per **rate** e **nper**. Se si eseguono pagamenti mensili su un prestito quinquennale al tasso di interesse annuale del 6%, utilizzare 0,005 (6%/12) per **rate** e 60 (5\*12) per **nper**. Se si effettuano pagamenti annuali sullo stesso prestito, utilizzare il 6% per **rate** e il 5 per **nper**.*

### Panoramica sulle funzioni finanziarie

Ciascuna funzione viene descritta ulteriormente dopo la panoramica. È inoltre possibile fare clic sul nome della funzione nella sintassi per accedere immediatamente ai dettagli per tale funzione specifica.

#### FV

Questa funzione restituisce il valore futuro di un investimento basato su pagamenti periodici e costanti e un tasso di interesse annuale semplice.

```
FV (rate, nper, pmt [ ,pv [ , type ] ])
```

#### nPer

Questa funzione restituisce il numero dei periodi per un investimento basato su pagamenti periodici e costanti con un tasso di interesse costante.

```
nPer (rate, pmt, pv [ ,fv [ , type ] ])
```

#### Pmt

Questa funzione restituisce il pagamento di un prestito basato su versamenti periodici e costanti e un tasso di interesse costante. Non può essere modificato per tutta la durata dell'annualità. Un pagamento viene indicato con un numero negativo, ad esempio, -20.

```
Pmt (rate, nper, pv [ ,fv [ , type ] ] )
```

#### PV

Questa funzione restituisce il valore corrente di un investimento.

```
PV (rate, nper, pmt [ ,fv [ , type ] ])
```

#### Rate

Questa funzione restituisce il tasso di interesse per periodo di un anno. Il risultato presenta un formato numerico predefinito **Fix** con due decimali e %.

```
Rate (nper, pmt , pv [ ,fv [ , type ] ])
```

### BlackAndSchole

Il modello Black and Scholes è un modello matematico per gli strumenti derivati del mercato finanziario. La formula consente di calcolare il valore teorico di una stock option. In Qlik Sense la funzione **BlackAndSchole** restituisce il valore in base alla formula Black and Scholes non modificata (opzioni in stile europeo).

```
BlackAndSchole (strike , time_left , underlying_price , vol , risk_free_rate , type)
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

### Argomenti

| Argomento        | Descrizione                                                                                                                                                        |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| strike           | Il prezzo futuro di acquisto dell'azione.                                                                                                                          |
| time_left        | Il numero di intervalli di tempo rimanenti.                                                                                                                        |
| underlying_price | Il valore attuale dell'azione.                                                                                                                                     |
| vol              | La volatilità (del prezzo dell'azione) espressa come percentuale in forma decimale, per periodo di tempo.                                                          |
| risk_free_rate   | Il tasso senza rischi espresso come percentuale in forma decimale, per periodo di tempo.                                                                           |
| call_or_put      | Il tipo di opzione:<br><br>'c', 'call' o qualsiasi valore numerico diverso da zero per le opzioni di chiamata<br><br>'p', 'put' o 0 per le opzioni di inserimento. |

**Limiti:**

Il valore di strike, time\_left e underlying\_price deve essere >0.

Il valore di vol e risk\_free\_rate deve essere: <0 o >0.

**Esempi e risultati:**

### Esempi di script

| Esempio                                                                                                                                                                                                                                                                                                                                | Risultato             |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
| <code>BlackAndSchole(130, 4, 68.5, 0.4, 0.04, 'call')</code><br><br>Consente di calcolare il prezzo teorico di un'opzione per acquistare un'azione che ha attualmente un valore pari a 68,5, a un valore di 130 in 4 anni. La formula utilizza una volatilità di 0,4 (40%) all'anno e un tasso di interesse senza rischi di 0,04 (4%). | Restituisce<br>11,245 |

## FV

Questa funzione restituisce il valore futuro di un investimento basato su pagamenti periodici e costanti e un tasso di interesse annuale semplice.

**Sintassi:**

```
FV(rate, nper, pmt [ ,pv [ , type ] ])
```

**Tipo di dati restituiti:** numerico. Per impostazione predefinita, il risultato sarà formattato come valuta..

**Argomenti:**

Argomenti

| Argomento | Descrizione                                                                                                                                                                  |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| rate      | Il tasso di interesse per periodo.                                                                                                                                           |
| nper      | Il numero totale dei periodi di pagamento in un'annualità.                                                                                                                   |
| pmt       | Il pagamento effettuato per ogni periodo. Non può essere modificato per tutta la durata dell'annualità. Un pagamento viene indicato con un numero negativo, ad esempio, -20. |
| pv        | Il valore attuale, o l'ammontare della somma forfettaria, che verrà saldato da una serie di pagamenti futuri. Se <b>p</b> <b>v</b> viene omissso, viene utilizzato 0 (zero). |
| type      | Deve essere 0 se i pagamenti sono in scadenza alla fine del periodo oppure 1 se sono in scadenza all'inizio del periodo. Se <b>type</b> viene omissso, viene utilizzato 0.   |

**Esempi e risultati:**

Esempio di script

| Esempio                                                                                                                                                                                                                                                                    | Risultato            |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|
| Un nuovo elettrodomestico viene pagato con 36 rate mensili di \$20. Il tasso di interesse annuo è del 6%. La fattura deve essere pagata alla fine di ogni mese. Qual è il valore totale investito, al momento del pagamento dell'ultima fattura?<br><br>FV(0.005, 36, -20) | Restituisce \$786.72 |

### nPer

Questa funzione restituisce il numero dei periodi per un investimento basato su pagamenti periodici e costanti con un tasso di interesse costante.

**Sintassi:**

```
nPer (rate, pmt, pv [ ,fv [ , type ] ])
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

Argomenti

| Argomento | Descrizione                                                |
|-----------|------------------------------------------------------------|
| rate      | Il tasso di interesse per periodo.                         |
| nper      | Il numero totale dei periodi di pagamento in un'annualità. |

| Argomento | Descrizione                                                                                                                                                                                        |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| pmt       | Il pagamento effettuato per ogni periodo. Non può essere modificato per tutta la durata dell'annualità. Un pagamento viene indicato con un numero negativo, ad esempio, -20.                       |
| pv        | Il valore attuale, o l'ammontare della somma forfettaria, che verrà saldato da una serie di pagamenti futuri. Se <b>p</b> <b>v</b> viene omissso, viene utilizzato 0 (zero).                       |
| fv        | Il valore futuro, o un saldo di cassa, che si desidera raggiungere dopo avere effettuato l'ultimo pagamento. Se <b>f</b> <b>v</b> viene omissso, viene utilizzato 0.                               |
| type      | Deve essere 0 se i pagamenti sono in scadenza alla fine del periodo oppure 1 se sono in scadenza all'inizio del periodo. Se <b>t</b> <b>y</b> <b>p</b> <b>e</b> viene omissso, viene utilizzato 0. |

## Esempi e risultati:

## Esempio di script

| Esempio                                                                                                                                                                                                                                                                                                                      | Risultato            |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|
| Si desidera vendere un elettrodomestico a rate mensili di \$20. Il tasso di interesse annuo è del 6%. La fattura deve essere pagata alla fine di ogni mese. Quanti periodi sono richiesti se il valore del denaro che si è ricevuto dopo l'ultima fattura pagata deve corrispondere a \$800?<br><br>nPer(0.005, -20, 0, 800) | Restituisce<br>36,56 |

## Pmt

Questa funzione restituisce il pagamento di un prestito basato su versamenti periodici e costanti e un tasso di interesse costante. Non può essere modificato per tutta la durata dell'annualità. Un pagamento viene indicato con un numero negativo, ad esempio, -20.

```
Pmt(rate, nper, pv [ ,fv [ , type ] ] )
```

**Tipo di dati restituiti:** numerico. Per impostazione predefinita, il risultato sarà formattato come valuta..

Per trovare la quantità totale pagata per la durata del prestito, moltiplicare il valore di **pmt** restituito da **nper**.

## Argomenti:

## Argomenti

| Argomento  | Descrizione                                                                                                                                                                  |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| rate       | Il tasso di interesse per periodo.                                                                                                                                           |
| nper       | Il numero totale dei periodi di pagamento in un'annualità.                                                                                                                   |
| p <b>v</b> | Il valore attuale, o l'ammontare della somma forfettaria, che verrà saldato da una serie di pagamenti futuri. Se <b>p</b> <b>v</b> viene omissso, viene utilizzato 0 (zero). |

| Argomento | Descrizione                                                                                                                                                                |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| fv        | Il valore futuro, o un saldo di cassa, che si desidera raggiungere dopo avere effettuato l'ultimo pagamento. Se <b>fv</b> viene omissso, viene utilizzato 0.               |
| type      | Deve essere 0 se i pagamenti sono in scadenza alla fine del periodo oppure 1 se sono in scadenza all'inizio del periodo. Se <b>type</b> viene omissso, viene utilizzato 0. |

Esempi e risultati:

#### Esempi di script

| Esempio                                                                                                                                                                          | Risultato                |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| La seguente formula restituisce i pagamenti mensili su un prestito di \$20.000 con tasso di interesse al 10%, da saldare in 8 mensilità:<br><br><code>Pmt(0.1/12,8,20000)</code> | Restituisce - \$2,594.66 |
| Per lo stesso prestito, se i pagamenti avvengono a inizio mensilità, il totale sarà:<br><br><code>Pmt(0.1/12,8,20000,0,1)</code>                                                 | Restituisce - \$2,573.21 |

## PV

Questa funzione restituisce il valore corrente di un investimento.

```
PV(rate, nper, pmt [ ,fv [ , type ] ])
```

**Tipo di dati restituiti:** numerico. Per impostazione predefinita, il risultato sarà formattato come valuta..

Il presente valore corrisponde all'importo totale attuale di una serie di pagamenti futuri. Ad esempio, quando si richiede un prestito, l'entità del prestito è il valore attuale di chi concede il prestito.

**Argomenti:**

#### Argomenti

| Argomento | Descrizione                                                                                                                                                                  |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| rate      | Il tasso di interesse per periodo.                                                                                                                                           |
| nper      | Il numero totale dei periodi di pagamento in un'annualità.                                                                                                                   |
| pmt       | Il pagamento effettuato per ogni periodo. Non può essere modificato per tutta la durata dell'annualità. Un pagamento viene indicato con un numero negativo, ad esempio, -20. |
| fv        | Il valore futuro, o un saldo di cassa, che si desidera raggiungere dopo avere effettuato l'ultimo pagamento. Se <b>fv</b> viene omissso, viene utilizzato 0.                 |
| type      | Deve essere 0 se i pagamenti sono in scadenza alla fine del periodo oppure 1 se sono in scadenza all'inizio del periodo. Se <b>type</b> viene omissso, viene utilizzato 0.   |

Esempi e risultati:

Esempio di script

| Esempio                                                                                                                                                                                               | Risultato                 |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|
| Qual è il valore attuale di un debito quando occorre pagare \$100 alla fine di ogni mese in un periodo di 5 anni, con un tasso di interesse del 7%?<br><br><code>PV(0.07/12, 12*5, -100, 0, 0)</code> | Restituisce<br>\$5,050.20 |

## Rate

Questa funzione restituisce il tasso di interesse per periodo di un anno. Il risultato presenta un formato numerico predefinito **Fix** con due decimali e %.

Sintassi:

**Rate**(*nper*, *pmt*, *pv* [, *fv* [, *type* ] ])

Tipo di dati restituiti: numerico.

Il valore di **rate** viene calcolato mediante ripetizione e può avere zero o più soluzioni. Se i risultati di **rate** successivi non convergono, verrà restituito un valore NULL.

Argomenti:

Argomenti

| Argomento   | Descrizione                                                                                                                                                                  |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>nper</i> | Il numero totale dei periodi di pagamento in un'annualità.                                                                                                                   |
| <i>pmt</i>  | Il pagamento effettuato per ogni periodo. Non può essere modificato per tutta la durata dell'annualità. Un pagamento viene indicato con un numero negativo, ad esempio, -20. |
| <i>pv</i>   | Il valore attuale, o l'ammontare della somma forfettaria, che verrà saldato da una serie di pagamenti futuri. Se <b>pv</b> viene omissso, viene utilizzato 0 (zero).         |
| <i>fv</i>   | Il valore futuro, o un saldo di cassa, che si desidera raggiungere dopo avere effettuato l'ultimo pagamento. Se <b>fv</b> viene omissso, viene utilizzato 0.                 |
| <i>type</i> | Deve essere 0 se i pagamenti sono in scadenza alla fine del periodo oppure 1 se sono in scadenza all'inizio del periodo. Se <b>type</b> viene omissso, viene utilizzato 0.   |

Esempi e risultati:

Esempio di script

| Esempio                                                                                                                                                  | Risultato            |
|----------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|
| Qual è il tasso di interesse di un prestito in cinque anni di \$10.000 annuali con pagamenti mensili di \$300?<br><br><code>Rate(60, -300, 10000)</code> | Restituisce<br>2.00% |

## 8.12 Funzioni di formattazione

Le funzioni di formattazione impongono il formato di visualizzazione ai campi numerici o alle espressioni di input. A seconda del tipo di dati, è possibile specificare i caratteri per il separatore decimale, il separatore delle migliaia e così via.

Tutte le funzioni restituiscono un valore duale che riporta sia la stringa che il valore numerico, ma possono essere interpretate come una conversione da numero a stringa. **Dual()** è un caso speciale, tuttavia le altre funzioni di formattazione generano una stringa che rappresenta il numero a partire dal valore numerico dell'espressione di input.

Le funzioni di interpretazioni invece si comportano nel modo opposto: le espressioni delle stringhe vengono valutate come numeri specificando il formato del numero risultante.

Le funzioni possono essere utilizzate sia negli script di caricamento dei dati che nelle espressioni grafiche.



*Tutte le rappresentazioni numeriche vengono fornite con un punto decimale come separatore decimale.*

### Panoramica sulle funzioni di formattazione

Ciascuna funzione viene descritta ulteriormente dopo la panoramica. È inoltre possibile fare clic sul nome della funzione nella sintassi per accedere immediatamente ai dettagli per tale funzione specifica.

#### ApplyCodepage

**ApplyCodepage()** applica il set di caratteri di una pagina codici differente al campo o al testo dichiarato nell'espressione. L'argomento **codepage** deve essere in formato numerico.

**ApplyCodepage** (text, codepage)

#### Date

**Date()** consente di formattare un'espressione come una data utilizzando il formato impostato nelle variabili di sistema nello script di caricamento dei dati o nel sistema operativo oppure una stringa di formattazione, se disponibile.

**Date** (number[, format])

#### Dual

**Dual()** combina un numero e una stringa in un unico record in modo che la rappresentazione numerica del record possa essere utilizzata per l'ordinamento e il calcolo, mentre il valore della stringa possa essere utilizzato per la visualizzazione.

**Dual** (text, number)

### Interval

**Interval()** consente di formattare un numero come un intervallo di tempo utilizzando il formato impostato nelle variabili di sistema nello script di caricamento dei dati o nel sistema operativo oppure una stringa di formattazione, se disponibile.

```
Interval (number[, format])
```

### Money

**Money()** consente di formattare un'espressione numericamente come valore di valuta nel formato delle variabili di sistema impostato nello script di caricamento dei dati o nel sistema operativo, a meno che non vengano forniti una stringa di formattazione e separatori decimali e delle migliaia opzionali.

```
Money (number[, format[, dec_sep [, thou_sep]])
```

### Num

**Num()** formatta un numero, ovvero converte il valore numerico dell'input per visualizzare il testo usando il formato specificato nel secondo parametro. Se il secondo parametro viene omissso, utilizza i separatori decimali e delle migliaia impostati nello script di caricamento dei dati. I simboli dei separatori decimali e delle migliaia personalizzati sono parametri opzionali.

```
Num (number[, format[, dec_sep [, thou_sep]])
```

### Time

**Time()** consente di formattare un'espressione come valore ora nel formato dell'ora impostato nelle variabili di sistema nello script di caricamento dei dati o nel sistema operativo, a meno che non venga fornita una stringa di formattazione.

```
Time (number[, format])
```

### Timestamp

**TimeStamp()** consente di formattare un'espressione come valore data e ora nel formato dell'indicatore temporale impostato nelle variabili di sistema nello script di caricamento dei dati o nel sistema operativo, a meno che non venga fornita una stringa di formattazione.

```
Timestamp (number[, format])
```

---

### Vedere anche:

p [Funzioni di interpretazione \(page 1433\)](#)

## ApplyCodepage

**ApplyCodepage()** applica il set di caratteri di una pagina codici differente al campo o al testo dichiarato nell'espressione. L'argomento **codepage** deve essere in formato numerico.



*Sebbene sia possibile utilizzare `ApplyCodepage` nelle espressioni grafiche, viene utilizzato più di frequente come funzione di script nell'Editor caricamento dati. Ad esempio, dal momento che si caricano file che potrebbero essere stati salvati con set di caratteri differenti non controllati dall'utente, è possibile applicare la pagina codici che rappresenta il set di caratteri richiesto.*

#### Sintassi:

**`ApplyCodepage`** (*text*, *codepage*)

Tipo di dati restituiti: stringa

#### Argomenti

| Argomento       | Descrizione                                                                                                   |
|-----------------|---------------------------------------------------------------------------------------------------------------|
| <b>text</b>     | Campo o testo a cui si desidera applicare una pagina codici differente, data dall'argomento <b>codepage</b> . |
| <b>codepage</b> | Numero che rappresenta la pagina dei codici da applicare al campo o all'espressione data da <b>text</b> .     |

#### Esempio: script di caricamento

##### Esempio

```
LOAD
ApplyCodepage(ROWX,1253) as GreekProduct,
ApplyCodepage (ROWY, 1255) as HebrewProduct,
ApplyCodepage (ROWZ, 65001) as EnglishProduct;
SQL SELECT ROWX, ROWY, ROWZ From Products;
```

##### Risultato

Durante il caricamento da SQL la sorgente potrebbe presentare una combinazione di set di caratteri differenti: cirillico, ebraico e così via, del formato UTF-8. Questi devono essere caricati riga per riga, applicano una pagina codici differente per ciascuna riga. Il valore **codepage** 1253 rappresenta il set di caratteri

**Esempio****Risultato**

greco Windows,  
il valore 1255  
rappresenta  
l'ebraico e il  
valore 65001  
rappresenta i  
caratteri UTF-8  
latini standard.

Vedere anche: [Set di caratteri \(page 175\)](#)

**Date**

**Date()** consente di formattare un'espressione come una data utilizzando il formato impostato nelle variabili di sistema nello script di caricamento dei dati o nel sistema operativo oppure una stringa di formattazione, se disponibile.

**Sintassi:**

```
Date (number[, format])
```

**Tipo di dati restituiti:** duale

**Argomenti**

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>number</b> | Il numero da formattare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>format</b> | <p>Stringa che descrive il formato della stringa risultante. Se non viene fornita la stringa di formattazione, verrà utilizzato il formato della data impostato nelle variabili di sistema nello script di caricamento dei dati o nel sistema operativo.</p> <p>La funzione <code>date</code> supporta molte opzioni di formato per diverse strutture della data, ad esempio:</p> <p>DD/MM/YYYY: Giorno/Mese/Anno (ad esempio, 15/02/2022)</p> <p>MM-DD-YYYY: Mese-Giorno-Anno separati da trattini (ad esempio, 02-15-2022).</p> <p>YYYY-MM-DD: Anno-Mese-Giorno in formato ISO (per esempio, 2022-02-15)</p> <p>D-MMM-YYYY: Giorno, nome breve del Mese e Anno separati da trattini (ad esempio, 15-Feb-2022)</p> <p>MMMM YYYY: nome completo del Mese e Anno (ad esempio, February 2022)</p> <p>MM/YYYY: Mese e Anno (ad esempio, 02/2022)</p> <p>DD MMMM YYYY: Giorno, nome completo del Mese e Anno (ad esempio, 15 February 2022)</p> |

Esempio: espressioni dei grafici

| Esempio                                   | Risultato                                                                                             |
|-------------------------------------------|-------------------------------------------------------------------------------------------------------|
| <code>Date( 35648 )</code>                | Restituisce 97-08-06 quando l'impostazione <code>DateFormat</code> è <code>YY-MM-DD</code> .          |
| <code>Date( 35648, 'YY.MM.DD' )</code>    | Restituisce 97.08.06 quando l'impostazione <code>DateFormat</code> è <code>YY-MM-DD</code> .          |
| <code>Date( '8/6/97', 'YY.MM.DD' )</code> | Restituisce <code>NULL</code> quando l'impostazione <code>DateFormat</code> è <code>YY-MM-DD</code> . |

### Esempio: principi fondamentali per la data

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `RowNo`
  - `InputDate`
- La variabile `DateFormat`, che utilizza il formato `YY-MM-DD`.

#### Script di caricamento

```
SET DateFormat='YY-MM-DD';
Example:
Load * inline [
RowNo, InputDate
1, 35648
2, 35648.375
3, 8/6/97
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `RowNo`
- `InputDate`

Creare le seguenti misure:

- `=Date(InputDate)`, per visualizzare il valore come una data formattata utilizzando le impostazioni della variabile `DateFormat`.
- `=Date(InputDate, 'DD.MM.YYYY')`, per visualizzare il valore come una data formattata secondo il formato fornito nella funzione.

- `=Date(InputDate, 'YY.MM.DD')`, per visualizzare il valore come una data formattata secondo il formato fornito nella funzione.
- `=Num(InputDate)`, per restituire il valore numerico della prima data.

Results table

| RowNo | InputDate | Date (InputDate) | Date(InputDate, 'DD.MM.YYYY') | Date(InputDate, 'YY.MM.DD') | Num (InputDate) |
|-------|-----------|------------------|-------------------------------|-----------------------------|-----------------|
| 1     | 35648     | 97-08-06         | 06.08.1997                    | 97.08.06                    | 35648           |
| 2     | 35648.375 | 97-08-06         | 06.08.1997                    | 97.08.06                    | 35648.375       |
| 3     | 8/6/97    | -                | -                             | -                           | -               |

I risultati mostrano come la funzione `Date` possa essere utilizzata per visualizzare e formattare le date. In questo esempio, `DateFormat` è impostato come `YY-MM-DD` nello script di caricamento. Il sistema prevede che l'espressione per la data utilizzi questo formato. Si può vedere che il valore per `InputDate` nella riga 3 (8/6/97) non segue questo formato, quindi il sistema non lo riconosce come data e restituisce `NULL`.

Modificare l'impostazione `DateFormat` nello script di caricamento per `SET DateFormat='M/D/YY'`; e ricaricare i dati. Il valore nella riga 3 ora corrisponde al formato previsto. Il sistema la riconoscerà come una data valida e la formatterà di conseguenza. La tabella che segue mostra i risultati con la nuova impostazione per `DateFormat`.

Results table with variable statement `set DateFormat='M/D/YY'`

| RowNo | InputDate | Date (InputDate) | Date(InputDate, 'DD.MM.YYYY') | Date(InputDate, 'YY.MM.DD') | Num (InputDate) |
|-------|-----------|------------------|-------------------------------|-----------------------------|-----------------|
| 1     | 35648     | 97-08-06         | 06.08.1997                    | 97.08.06                    | 35648           |
| 2     | 35648.375 | 97-08-06         | 06.08.1997                    | 97.08.06                    | 35648.375       |
| 3     | 35648     | 97-08-06         | 06.08.1997                    | 97.08.06                    | 35648.375       |

### Esempio: scenario per la data

Espressione del grafico

#### Panoramica

Un set di dati contiene dati sulle vendite che utilizzano il formato della data `MM-DD-YYYY`. Si desidera utilizzare il formato della data `DD/MM/YYYY`.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:

- Customer
  - SalesDate
  - Amount
- La variabile DateFormat, che utilizza il formato MM/DD/YYYY.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';  
Example:  
Load * inline [  
Customer, SalesDate, Amount  
Customer A, 2/15/2024, 100008  
Customer B, 9/30/2024, 12500  
Customer C, 12/28/2024, 15000  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- Customer

Creare le seguenti misure:

- =Date(SalesDate , 'DD/MM/YYYY') per visualizzare il valore come una data formattata secondo il formato fornito nella funzione.
- =Sum(Amount) per calcolare il valore totale del costo delle vendite.

Results table

| Customer   | Date(SalesDate , 'DD/MM/YYYY') | Sum(Amount) |
|------------|--------------------------------|-------------|
| Customer A | 15/02/2024                     | 100008      |
| Customer B | 30/09/2024                     | 12500       |
| Customer C | 28/12/2024                     | 15000       |

Osservando i risultati, si può notare che sebbene le date nel set di dati utilizzino un formato, la funzione date consente di presentare queste date in un formato diverso.

## Dual

**Dual()** combina un numero e una stringa in un unico record in modo che la rappresentazione numerica del record possa essere utilizzata per l'ordinamento e il calcolo, mentre il valore della stringa possa essere utilizzato per la visualizzazione.

### Sintassi:

```
Dual (text, number)
```

Tipo di dati restituiti: duale



*Tutti i valori duali restituiti sono allineati a destra.*

#### Argomenti

| Argomento     | Descrizione                                                              |
|---------------|--------------------------------------------------------------------------|
| <b>text</b>   | Il valore della stringa da usare unitamente all'argomento del numero.    |
| <b>number</b> | Il numero da usare unitamente alla stringa nell'argomento della stringa. |

In Qlik Sense, tutti i valori di campo sono potenzialmente valori duali. Questo significa che i valori del campo possono avere sia un valore numerico che un valore testuale. Un esempio è una data che potrebbe avere un valore numerico di 40908 e la rappresentazione testuale '2011-12-31'.



*Quando diversi dati letti da un campo presentano rappresentazioni di stringhe differenti per la stessa rappresentazione numerica valida, tutti condividono la prima rappresentazione a stringa rilevata.*



*La funzione **dual** viene in genere utilizzata all'inizio dello script, prima che venga eseguita la lettura di altri dati nel campo interessato, affinché sia possibile creare la prima rappresentazione di stringa, che verrà poi visualizzata nelle caselle di filtro.*

#### Esempio: espressioni dei grafici

| Esempio                                                                                                                                                                                                                                         | Descrizione                                                                                                                                                                                                             |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>Dual(IF(DayOfWeek = 0, 'Monday', IF(DayOfWeek = 1, 'Tuesday', If(DayOfWeek = 2, 'Wednesday', If(DayOfWeek = 3, 'Thursday', If(DayOfWeek = 4, 'Friday', If(DayOfWeek = 5, 'Saturday', If(DayOfWeek = 6, 'Sunday'))))))), DayOfWeek)</code> | Il campo DayOfWeek può essere utilizzato come dimensione in una visualizzazione. In una tabella, i giorni della settimana sono ordinati automaticamente nella sequenza numerica corretta, anziché in ordine alfabetico. |
| <code>Dual(Pick( Priority, 'Low', 'Medium', 'High'), Priority)</code>                                                                                                                                                                           | Il campo Priority può essere utilizzato come dimensione in una visualizzazione. In una tabella, il campo Priority è ordinato automaticamente nella sequenza numerica corretta, anziché in ordine alfabetico.            |

| Esempio                                                                        | Descrizione                                                                                                                                                                                                                                                                                                          |
|--------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>Dual('Q' &amp; Ceil(Month(Date)/3), Ceil(Month(Date)/3))</code>          | Il campo <code>Date</code> può essere utilizzato come dimensione in una visualizzazione. I valori di testo da Q1 a Q4 vengono creati e vengono assegnati loro i valori numerici da 1 a 4.                                                                                                                            |
| <code>Dual(WeekYear(Date) &amp; '-w' &amp; Week(Date), WeekStart(Date))</code> | Il campo <code>Date</code> può essere utilizzato come dimensione in una visualizzazione. In questo esempio vengono creati valori testuali per il modulo '2012-w22' e, allo stesso tempo, viene assegnato un valore numerico corrispondente al numero della data del primo giorno della settimana, ad esempio: 41057. |

### Esempio: script di caricamento

| Esempio                                                                                                                                                                                                                        | Descrizione                                                                                                                                                                                                                                                                                                                                 |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Aggiungere gli esempi seguenti allo script ed eseguirlo.</p> <pre>Load dual ( NameDay,NumDay ) as DayOfWeek inline  [ NameDay,NumDay  Monday,0  Tuesday,1  Wednesday,2  Thursday,3  Friday,4  Saturday,5  Sunday,6 ];</pre> | Il campo <code>DayOfWeek</code> può essere utilizzato in una visualizzazione, ad esempio come dimensione. In una tabella, i giorni della settimana sono ordinati automaticamente nella sequenza numerica corretta, anziché in ordine alfabetico.                                                                                            |
| <pre>Load Dual('Q' &amp; Ceil( Month(Now())/3), Ceil(Month(Now ())/3)) as Quarter AutoGenerate 1;</pre>                                                                                                                        | Questo esempio individua il trimestre attuale. Viene visualizzato come Q1 quando la funzione <b>Now</b> viene eseguita nei primi tre mesi dell'anno, come Q2 per i secondi tre mesi e così via. Tuttavia, quando viene utilizzato nell'ordinamento, il campo <code>quarter</code> si comporterà come il relativo valore numerico: da 1 a 4. |
| <pre>Dual('Q' &amp; Ceil( Month(Date)/3), Ceil(Month(Date)/3)) as Quarter</pre>                                                                                                                                                | Come nell'esempio precedente, il campo <code>quarter</code> viene creato con i valori di testo da Q1 a Q4 a cui vengono assegnati i valori numerici da 1 a 4. Per poter utilizzarli nello script, i valori di <code>Date</code> devono venire caricati.                                                                                     |

| Esempio                                                                                    | Descrizione                                                                                                                                                                                                                                                                                                                     |
|--------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>Dual(WeekYear(Date) &amp; '-w' &amp; Week(Date), WeekStart(Date)) as Yearweek</code> | Questo esempio crea un campo <code>yearweek</code> con i valori testuali in forma di 2012-w22 e allo stesso tempo assegna un valore numerico corrispondente al numero della data del primo giorno della settimana, ad esempio: 41057. Per poter utilizzarli nello script, i valori di <code>Date</code> devono venire caricati. |

### Esempio: principi fondamentali della funzione Dual

Espressione del grafico

#### Panoramica

Un set di dati contiene i dati del servizio clienti. Si desidera esaminare il numero di chiamate in base al giorno della settimana.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `DayOfWeek`
  - `NoOfCalls`

#### Script di caricamento

Example:

```
Load * inline [
DayOfWeek, NoOfCalls
0, 25000
1, 30000
2, 27500
3, 32000
4, 23250
5, 28950
6, 32350
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare un nuovo grafico a barre, quindi aggiungere una dimensione utilizzando uno dei seguenti metodi:

- Nell'espressione della dimensione, aggiungere quanto segue:  
`=Dual(IF(DayOfWeek = 0, 'Monday',  
IF(DayOfWeek = 1, 'Tuesday',  
If(DayOfWeek = 2, 'Wednesday',  
If(DayOfWeek = 3, 'Thursday',`

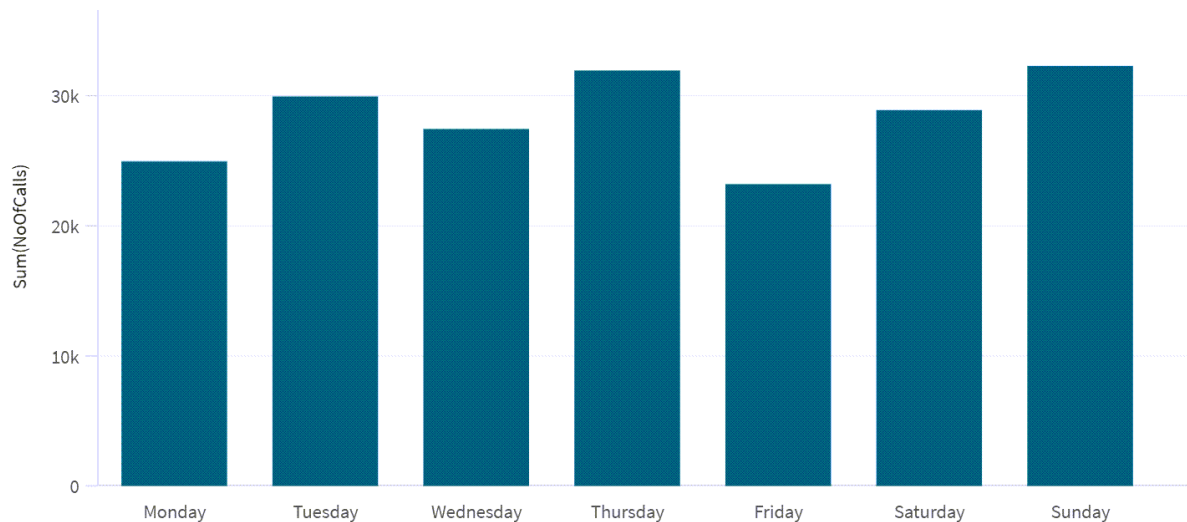
```
If(DayOfWeek = 4 , 'Friday',  
If(DayOfWeek = 5 , 'Saturday',  
If(DayOfWeek = 6 , 'Sunday'  
))))), DayOfWeek)
```

- b. Questo metodo è più efficiente. Nell'espressione della dimensione, aggiungere quanto segue:
- ```
=Dual(pick( DayOfWeek +1 , 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday',  
'Saturday', 'Sunday' ), DayOfWeek)
```

Creare la seguente misura:

- `=Sum(NoOfCalls)`, per calcolare il numero di chiamate.

Risultati: grafico a barre che mostra il numero di chiamate per giorno della settimana.



Il grafico a barre mostra il numero di chiamate in base al giorno della settimana. Ordina automaticamente i giorni della settimana nell'ordine corretto.

### Esempio: scenario per l'applicazione della funzione Dual

Espressione del grafico

#### Panoramica

Un set di dati contiene i feedback sul servizio clienti. Si desidera ordinare e visualizzare i feedback mostrando prima le risposte negative e poi quelle positive.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `ResponseType`
  - `NoOfResponses`

### Script di caricamento

Example:

```
Load * inline [  
  ResponseType, NoOfResponses  
  1, 350  
  2, 560  
  3, 4350  
  4, 2350  
  5, 5350  
];
```

### Risultati

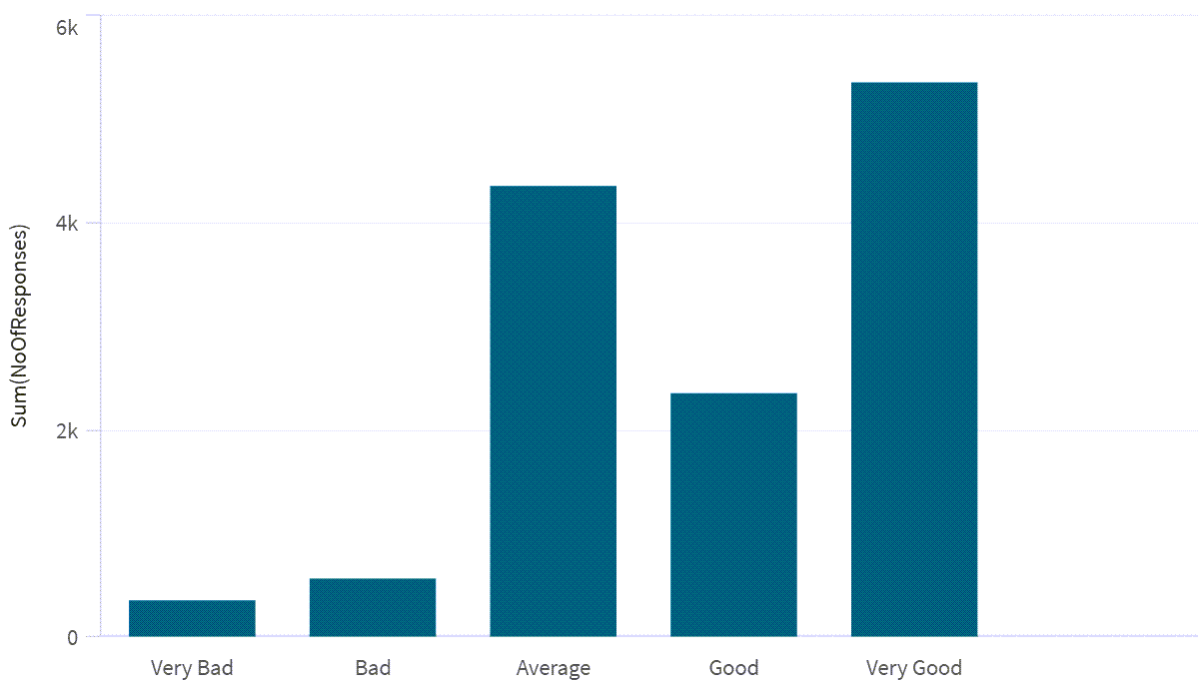
Caricare i dati e aprire un foglio. Creare un nuovo grafico a barre e aggiungere la seguente espressione a una dimensione.

- `=Dual(pick( ResponseType , 'Very Bad','Bad','Average', 'Good','Very Good'), ResponseType)`

Creare la seguente misura:

- `=Sum(NoOfResponses)`, per calcolare il numero di risposte.

*Risultati: grafico a barre che mostra il numero di risposte in base al tipo di risposta.*



Il grafico a barre mostra il numero di risposte in base al tipo di risposta. Notare che le barre vengono ordinate automaticamente da un valore negativo a uno positivo. Al tipo di risposta, che è un valore numerico, è stata assegnata una descrizione testuale.

## Interval

**Interval()** consente di formattare un numero come un intervallo di tempo utilizzando il formato impostato nelle variabili di sistema nello script di caricamento dei dati o nel sistema operativo oppure una stringa di formattazione, se disponibile.

Gli intervalli possono essere formattati come un'ora, come giorni oppure come combinazione di giorni, ore, minuti, secondi e frazioni di secondo.

### Sintassi:

```
Interval (number[, format])
```

**Tipo di dati restituiti:** duale

| Argomenti     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>number</b> | Il numero da formattare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>format</b> | Stringa che descrive come formattare la stringa dell'intervallo risultante. Se viene omessa, si utilizzerà il formato dell'ora e il separatore decimale impostati nel sistema operativo. Se viene omessa, si utilizzerà il formato dell'ora e il separatore decimale impostati nel sistema operativo.<br>La funzione <code>Interval</code> supporta molte opzioni di formato per diverse strutture per l'ora, ad esempio:<br><br>hh:mm: solo ore e minuti<br><br>hh:mm:ss: ore, minuti e secondi<br><br>d hh:mm: giorni, ore e minuti |

Esempio: espressioni dei grafici

### Esempio

### Risultato

```
Interval( 0.375 )
```

Restituisce 9:00:00 quando il valore `TimeFormat` è `h:mm:ss`.

```
Interval(.375, 'hh:mm' )
```

Restituisce 9:00 quando il formato fornito nella funzione è `hh.mm`.

```
Interval( 0.375, 'd hh' )
```

Restituisce 0 09 quando il formato fornito nella funzione è `d hh`.

## Esempio: principi fondamentali della funzione Interval

Espressione del grafico

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Il seguente campo in una tabella dati:
  - `IntervalValue`
- La variabile `TimeFormat`, che utilizza il formato `h:mm:ss`.

### Script di caricamento

```
SET TimeFormat='h:mm:ss';  
Example:  
Load * inline [  
IntervalValue  
0.375  
1.375  
35.648  
36.875  
368.75  
468.95  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `IntervalValue`

Creare le seguenti misure:

- `=Interval(IntervalValue)`, per visualizzare il valore come un formato intervallo che utilizza l'impostazione di sistema.
- `=Interval(IntervalValue, 'd hh')`, per visualizzare il valore come un formato intervallo che utilizza il formato fornito nella funzione.
- `=Interval(IntervalValue, 'hh:mm')`, per visualizzare il valore come un formato intervallo che utilizza il formato fornito nella funzione.
- `=Interval(IntervalValue, 'd hh:mm')`, per visualizzare il valore come un formato intervallo che utilizza il formato fornito nella funzione.

Tabella dei risultati

| IntervalValue | Interval<br>(IntervalValue) | Interval<br>(IntervalValue, 'd<br>hh') | Interval<br>(IntervalValue,<br>'hh:mm') | Interval<br>(IntervalValue, 'd<br>hh:mm') |
|---------------|-----------------------------|----------------------------------------|-----------------------------------------|-------------------------------------------|
| 0.375         | 9:00:00                     | 0 09                                   | 09:00                                   | 0 09:00                                   |
| 1.375         | 33:00:00                    | 1 09                                   | 33:00                                   | 1 09:00                                   |
| 35.648        | 855:33:07                   | 35 15                                  | 855:33                                  | 35 15:33                                  |
| 36.875        | 885:00:00                   | 36 21                                  | 885:00                                  | 36 21:00                                  |
| 368.75        | 8850:00:00                  | 368 18                                 | 8850:00                                 | 368 18:00                                 |
| 468.95        | 11254:48:00                 | 468 22                                 | 11254:48                                | 468 22:48                                 |

I risultati mostrano come la funzione `Interval` visualizza i valori in base al parametro di formato della funzione. Per esempio, nell'ultima riga, il valore 468.95, se formattato utilizzando la funzione `Interval` e il parametro di formato, `d hh:mm`, restituisce il valore 468 giorni, 22 ore e 48 minuti.

### Esempio: scenario per l'applicazione della funzione `Interval`

Espressione del grafico

#### Panoramica

I dati dei ticket di assistenza calcolano il tempo di risoluzione per ticket in secondi. Si desidera convertire questo valore in un altro formato.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `TicketID`
  - `ResolutionTime`

#### Script di caricamento

Example:

```
Load * inline [
TicketID, ResolutionTime
Ticket A, 4350
Ticket B, 560
Ticket C, 86400
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `TicketID`

Creare la seguente misura:

- `=Interval(ResolutionTime / 86400, 'hh:mm:ss')`, per convertire i valori in giorni, dividendo `ResolutionTime` per 86,400, e poi formattarli utilizzando il formato fornito dalla funzione.

Tabella dei risultati

| TicketID | Interval(ResolutionTime / 86400, 'hh:mm:ss') |
|----------|----------------------------------------------|
| Ticket A | 01:12:30                                     |
| Ticket B | 00:09:20                                     |
| Ticket C | 24:00:00                                     |

I valori temporali di risoluzione, originariamente forniti in secondi, sono ora convertiti in una frazione di giorno e quindi formattati in ore, minuti e secondi. Ad esempio, la tabella mostra che `Ticket B` ha impiegato 9 ore e 20 minuti per essere risolto.

## Money

**Money()** consente di formattare un'espressione numericamente come valore di valuta nel formato delle variabili di sistema impostato nello script di caricamento dei dati o nel sistema operativo, a meno che non vengano forniti una stringa di formattazione e separatori decimali e delle migliaia opzionali.

### Sintassi:

```
Money (number[, format[, dec_sep[, thou_sep]])
```

Tipo di dati restituiti: duale

| Argomenti     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>number</b> | Il numero da formattare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>format</b> | <p>Stringa che descrive come formattare la stringa della valuta risultante. Se viene omessa, verrà utilizzato il valore impostato nello script di caricamento per <code>MoneyFormat</code>. La funzione <code>Money</code> supporta molte opzioni di formato per diverse strutture della data, ad esempio:</p> <p><code>\$#,##0.00</code>: aggiunge il simbolo del dollaro e due cifre decimali.</p> <p><code>€ #,##0</code>: aggiunge il simbolo dell'euro e nessun decimale, con uno spazio dopo il simbolo dell'euro.</p> <p><code>¥#,##0</code>: aggiunge il simbolo dello yen giapponese senza cifre decimali.</p> <p><code>\$#,##0;(\$#,##0)</code>: formatta i valori negativi tra parentesi.</p> <p><code>₣ #.##0,00</code>: aggiunge il simbolo del franco svizzero con punti come separatori delle migliaia e una virgola per i decimali (ad esempio, <code>₣ 1.000,00</code>).</p> <p><code>R #,##0.00</code>: aggiunge il simbolo del rand sudafricano con due cifre decimali e uno spazio dopo il simbolo del rand.</p> <p><code>\$ #,##0.000</code>: aggiunge tre cifre decimali al formato del dollaro.</p> |

| Argomento       | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>dec_sep</b>  | <p>Stringa che specifica il separatore decimale dei numeri. Se viene omessa, verrà utilizzato il valore <code>MoneyDecimalSep</code> impostato nello script di caricamento dei dati. La funzione <code>Money</code> supporta molti formati per il separatore decimale, ad esempio:</p> <p>Virgola ', ': ad esempio, \$1,000,00 per mille dollari.</p> <p>Punto '. ': ad esempio, € 1,000.00.</p> <p>Trattino '-': se utilizzato al posto del punto o della virgola, utilizzati di norma per il separatore decimale, mostra valori come ¥1,000-00.</p>                                                                                                                                                                                                                                                                                       |
| <b>thou_sep</b> | <p>Stringa che specifica il separatore delle migliaia dei numeri. Se viene omessa, verrà utilizzato il valore <code>MoneyThousandSep</code> impostato nello script di caricamento dei dati. La funzione <code>Money</code> supporta molte opzioni per il separatore delle migliaia:</p> <p>Punto ' . ': utilizza un punto per il separatore di migliaia (comune in alcuni formati europei), visualizzandolo come \$1.000,00 se combinato con una virgola come separatore decimale.</p> <p>Spazio ' ': utilizza uno spazio per il separatore delle migliaia, che è comune in alcuni Paesi europei. Viene visualizzato come € 1 000.00.</p> <p>Trattino basso '_': utilizza un trattino basso per il separatore delle migliaia, che a volte viene utilizzato in formati tecnici o regionali specifici. Visualizza valori come \$1_000.00.</p> |

Se gli argomenti **format**, **dec\_sep** e **thou\_sep** vengono omessi, il formato della valuta impostato nel sistema operativo viene utilizzato per la formattazione di ogni parte specifica dell'output. Le impostazioni del sistema operativo vengono automaticamente incluse come variabili di sistema al momento della creazione dello script, assicurando così che siano coerenti in tutta l'applicazione.

Queste variabili di sistema sono **MoneyFormat**, **MoneyDecimalSep** e **MoneyThousandSep** e sono mappate rispettivamente agli argomenti delle funzioni **format**, **dec\_sep** e **thou\_sep**.

Esempio: espressioni dei grafici

| Esempio                                               | Risultati                                                                                                                                                                                                                                               |
|-------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>Money( 35648 )</code>                           | Restituisce kr 35 648,00 quando l'impostazione <code>MoneyFormat</code> è kr ##0,00 e il valore per <code>MoneyThousandSep</code> è un punto ' ' .                                                                                                      |
| <code>Money( 35648, '#,##0 ¥', ' . ' , ' , ' )</code> | Restituisce 3,564,800 ¥ ignorando l'impostazione <code>MoneyFormat</code> e l'impostazione <code>MoneyThousandSep</code> , poiché i parametri opzionali <code>Format</code> , <code>dec_sep</code> e <code>thou_sep</code> sono forniti nella funzione. |
| <code>Money(35648)</code>                             | Restituisce \$ 35,648.00 quando l'impostazione <code>MoneyFormat</code> è \$ #,##0.00 e il valore per <code>MoneyThousandSep</code> è una virgola ', ' .                                                                                                |

### Esempio: principi fondamentali della funzione Money

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Un campo della tabella dati denominata `InputValue`.
- I seguenti formati per le variabili:
  - La variabile `MoneyFormat`, che utilizza il formato `kr ##0,00`
  - La variabile `MoneyThousandSep` impostata per utilizzare uno spazio ' '

#### Script di caricamento

```
SET MoneyFormat='kr ##0,00';
SET MoneyThousandSep=' ';
Example:
Load * inline [
InputValue
35648
3564800
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `InputValue`

Creare le seguenti misure:

- `=Money(InputValue)`, per visualizzare il valore come valuta formattata utilizzando l'impostazione di sistema.
- `=Money(InputValue, '#,##0 ¥', '.', ',')`, per visualizzare il valore come valuta formattata secondo il formato fornito nella funzione.
- `=Num(InputValue)`, per restituire il valore numerico.

| Results table           |                                |                                                     |                              |
|-------------------------|--------------------------------|-----------------------------------------------------|------------------------------|
| <code>InputValue</code> | <code>Money(InputValue)</code> | <code>Money(InputValue, '#,##0 ¥', '.', ',')</code> | <code>Num(InputValue)</code> |
| 35648                   | kr35 648,00                    | 35,648 ¥                                            | 35648                        |
| 3564800                 | kr3 564 800,00                 | 3,564,800 ¥                                         | 3564800                      |

In questo esempio, è possibile vedere come la funzione `Money` visualizza i valori e come i valori del parametro di formato possono modificare l'output.

### Esempio: principi fondamentali della funzione Money che utilizza un'impostazione di formato differente

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Un campo della tabella dati denominata `InputValue`.
- I seguenti formati per le variabili:
  - La variabile `MoneyFormat`, che utilizza il formato `$ #,##0.00`
  - La variabile `MoneyThousandSep` impostata per utilizzare una virgola `,`

#### Script di caricamento

```
SET MoneyFormat='$ #,##0.00';
SET MoneyThousandSep=',';
Example:
Load * inline [
InputValue
35648
3564800
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `InputValue`

Creare le seguenti misure:

- `=Money(InputValue)`, per visualizzare il valore come valuta formattata utilizzando l'impostazione di sistema.
- `=Money(InputValue, '#,##0 ¥', '.', ',')`, per visualizzare il valore come valuta formattata secondo il formato fornito nella funzione.
- `=Num(InputValue)`, per restituire il valore numerico.

| Results table |                   |                                        |                 |  |
|---------------|-------------------|----------------------------------------|-----------------|--|
| InputValue    | Money(InputValue) | Money(InputValue, '#,##0 ¥', '.', ',') | Num(InputValue) |  |
| 35648         | \$ 35,648.00      | 35,648 ¥                               | 35648           |  |
| 3564800       | \$ 3,564,800.00   | 3,564,800 ¥                            | 3564800         |  |

In questo esempio, è possibile vedere come la funzione `Money` visualizza i valori e come i valori del parametro di formato possono modificare l'output.

### Esempio: scenario per la funzione Money

Espressione del grafico

#### Panoramica

I dati di vendita dei clienti vengono ricevuti in diverse valute. Si desidera visualizzare i dati in una tabella che utilizza il formato di valuta pertinente.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Customer`
  - `CurrencyCode`
  - `Sales`
- I seguenti formati per le variabili:
  - La variabile `MoneyFormat`, che utilizza il formato `$ ###0.00;-$ ###0.00`
  - La variabile `MoneyThousandSep` impostata per utilizzare una virgola `,`
  - La variabile `MoneyDecimalSep` impostata in modo da utilizzare un punto `.`

#### Script di caricamento

```
SET MoneyThousandSep=',';
SET MoneyDecimalSep='.';
SET MoneyFormat='$ ###0.00;-$ ###0.00';
Example:
Load * inline [
Customer, CurrencyCode, Sales
Customer A, USD, 100008
Customer B, EUR, 12500
Customer C, YEN, 15000
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `Customer`

Creare la seguente misura:

- `=Money(Sales, If(CurrencyCode = 'USD', '$#,##0.00', If(CurrencyCode = 'EUR', '€#,##0.00', '¥#,##0')))`, per visualizzare la valuta nel formato di valuta corretto.

Results table

| Customer   | Money(Sales, If(CurrencyCode = 'USD', '\$#,##0.00', If(CurrencyCode = 'EUR', '€#,##0.00', '¥#,##0'))) |
|------------|-------------------------------------------------------------------------------------------------------|
| Customer A | \$100,008.00                                                                                          |
| Customer B | € 12,500.00                                                                                           |
| Customer C | ¥15,000                                                                                               |

I dati di vendita utilizzano ora il formato di valuta definito nell'espressione di misura. I parametri per i separatori decimali e delle migliaia utilizzano le impostazioni dello script di caricamento.

## Num

**Num()** formatta un numero, ovvero converte il valore numerico dell'input per visualizzare il testo usando il formato specificato nel secondo parametro. Se il secondo parametro viene omissso, utilizza i separatori decimali e delle migliaia impostati nello script di caricamento dei dati. I simboli dei separatori decimali e delle migliaia personalizzati sono parametri opzionali.

### Sintassi:

```
Num (number[, format[, dec_sep [, thou_sep]])
```

### Tipo di dati restituiti: duale

La funzione Num restituisce un valore duale contenente sia la stringa che il valore numerico. La funzione prende il valore numerico dell'espressione di input e genera una stringa che rappresenta il numero.

Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>number</b> | La stringa di testo da valutare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>format</b> | <p>Stringa che specifica come formattare la stringa risultante. Se omissi, verranno utilizzati i separatori di decimali e migliaia impostati nello script di caricamento dei dati. La funzione Num supporta molte opzioni di formato per diverse strutture numeriche, ad esempio:</p> <p>#,##0: visualizza un numero senza cifre decimali.</p> <p>#,##0.00: visualizza un numero con due cifre decimali.</p> <p>##0.00%: visualizza un numero come una percentuale con due cifre decimali.</p> <p>#,##0;(#,##0): formatta i numeri negativi in modo da essere visualizzati tra parentesi.</p> <p>\$#,##0.00: formatta i numeri come valuta con il simbolo \$.</p> <p>0.00E+0: visualizza i numeri in una notazione scientifica.</p> |

| Argomento       | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>dec_sep</b>  | <p>Stringa che specifica il separatore decimale dei numeri. Se omissso, viene utilizzato il valore della variabile <code>DecimalSep</code> impostato nello script di caricamento dei dati.</p> <p>La funzione <code>Num</code> supporta molti formati per il separatore decimale, ad esempio:</p> <p>Virgola ', ': ad esempio, \$1,000,00 per mille dollari.</p> <p>Punto '. ': ad esempio, € 1,000.00.</p> <p>Trattino '-': se utilizzato al posto del punto o della virgola, utilizzati di norma per il separatore decimale, mostra valori come ¥1,000-00.</p> <p>Per ulteriori informazioni, vedere <a href="#">DecimalSep (page 239)</a>.</p>                                                                                                                                                                                                                                                                                          |
| <b>thou_sep</b> | <p>Stringa che specifica il separatore delle migliaia dei numeri. Se omissso, viene utilizzato il valore della variabile <code>ThousandSep</code> impostato nello script di caricamento dei dati.</p> <p>La funzione <code>Num</code> supporta molte opzioni per il separatore delle migliaia:</p> <p>Punto ' . ': utilizza un punto per il separatore di migliaia (comune in alcuni formati europei), visualizzandolo come \$1.000,00 se combinato con una virgola come separatore decimale.</p> <p>Spazio ' ': utilizza uno spazio per il separatore delle migliaia, che è comune in alcuni Paesi europei. Viene visualizzato come € 1 000.00.</p> <p>Trattino basso ' _ ': utilizza un trattino basso per il separatore delle migliaia, che a volte viene utilizzato in formati tecnici o regionali specifici. Visualizza valori come \$1_000.00.</p> <p>Per ulteriori informazioni, vedere <a href="#">ThousandSep (page 277)</a>.</p> |

Esempio: espressioni dei grafici

### Esempio

### Risultati (valore stringa restituito)

|                                                      |                                                                       |
|------------------------------------------------------|-----------------------------------------------------------------------|
| <code>Num(35648.312)</code>                          | Restituisce 35648.312 (dipende dalle variabili ambiente nello script) |
| <code>Num(35648.312, '0.0', ' . ')</code>            | Restituisce 35648.3                                                   |
| <code>Num(35648.312, '0,00', ' , ')</code>           | Restituisce 3,56,48                                                   |
| <code>Num(35648.312, '#,##0.0', ' . ', ' , ')</code> | Restituisce 35,648.3                                                  |
| <code>Num(35648.312, '# ##0', ' , ', ' _ ')</code>   | Restituisce 35 648                                                    |

### Esempio: principi fondamentali della funzione Num

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Il seguente campo in una tabella dati:
  - `InputValue`

#### Script di caricamento

Example:

```
Load * inline [  
InputValue  
35648  
3564800  
356.48  
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `InputValue`

Creare le seguenti misure:

- `=Num(InputDate)`, per visualizzare il valore come numero formattato utilizzando le impostazioni di sistema.
- `=Num(InputValue, '0.0', '.', '')`, per visualizzare il valore come un numero formattato secondo il formato fornito nella funzione.
- `=Num(InputValue, '0,00', ',', '')`, per visualizzare il valore come un numero formattato secondo il formato fornito nella funzione.
- `=Num(InputValue, '#,##0.0', ' ', ' ', '')`, per visualizzare il valore come un numero formattato secondo il formato fornito nella funzione.
- `=Num(InputValue, '# ##0', ' ', ' ', ' ')`, per visualizzare il valore come un numero formattato secondo il formato fornito nella funzione.

Results table

| InputValue | Num<br>(InputDate) | Num<br>(InputValue,<br>'0.0', '.') | Num<br>(InputValue,<br>'0,00', ',') | Num<br>(InputValue,<br>'#,##0.0', '!', ',') | Num<br>(InputValue, '#<br>##0', '!', ',') |
|------------|--------------------|------------------------------------|-------------------------------------|---------------------------------------------|-------------------------------------------|
| 356.48     | 356.48             | 356.5                              | 3,56                                | 356.5                                       | 356                                       |
| 35648      | 35648              | 35648.0                            | 3,56,48                             | 35,648.0                                    | 35 648                                    |
| 3564800    | 3564800            | 3564800.0                          | 3,56,48,00                          | 3,564,800.0                                 | 3 564 800                                 |

I risultati mostrano come la funzione `Num` può visualizzare i valori per il campo `InputValue` in base al formato fornito nell'espressione della funzione.

### Script di caricamento

La funzione `Num` può essere utilizzata in uno script di caricamento per formattare un numero, anche se i separatori delle migliaia e dei decimali sono già impostati nello script. Lo script di caricamento mostrato di seguito include specifici separatori delle migliaia e dei decimali, ma utilizza `Num` per formattare i dati in modi diversi.

### Script di caricamento

In Editor caricamento dati, creare una nuova sezione, quindi aggiungere lo script di esempio ed eseguirlo. Aggiungere quindi almeno i campi elencati nella colonna dei risultati a un foglio nell'app per visualizzare il risultato.

```
SET ThousandSep=',';
SET DecimalSep='.';
Transactions:
Load
*,
Num(transaction_amount) as [No formatting],
Num(transaction_amount,'0') as [0],
Num(transaction_amount,'#,#0') as [#,#0],
Num(transaction_amount,'# ###,00') as [# ###,00],
Num(transaction_amount,'# ###,00',' ',' ') as [# ###,00 , ' ' , ' '],
Num(transaction_amount,'# ,###.00','.',',') as [# ,###.00 , '.' , ','],
Num(transaction_amount,'$# ,###.00') as [$# ,###.00],
;
Load * Inline [
transaction_id, transaction_date, transaction_amount, transaction_quantity, discount,
customer_id, size, color_code
3750, 20180830, 12423.56, 23, 0,2038593, L, Red
3751, 20180907, 5356.31, 6, 0.1, 203521, m, orange
3752, 20180916, 15.75, 1, 0.22, 5646471, s, blue
3753, 20180922, 1251, 7, 0, 3036491, l, black
3754, 20180922, 21484.21, 1356, 75, 049681, xs, Red
3756, 20180922, -59.18, 2, 0.3333333333333333, 2038593, M, Blue
3757, 20180923, 3177.4, 21, .14, 203521, xL, black
];
```

### Risultati

Tabella Qlik Sense che mostra i risultati di diversi utilizzi della funzione *Num* nello script di caricamento. La quarta colonna della tabella mostra una formattazione incorretta, a scopo esemplificativo.

| No<br>formatting | 0     | #,##0  | # ###,00    | # ###,00 , ','<br>, '' | #,###.00 , ','<br>, '' | \$#,###.00  |
|------------------|-------|--------|-------------|------------------------|------------------------|-------------|
| -59.18           | -59   | -59    | -59###,00   | -59,18                 | -59.18                 | \$-59,18    |
| 15.75            | 16    | 16     | 16###,00    | 15,75                  | 15.75                  | \$15,75     |
| 1251             | 1251  | 1,251  | 1251###,00  | 1 251,00               | 1,251.00               | \$1,251.00  |
| 3177.4           | 3177  | 3,177  | 3177###,00  | 3 177,40               | 3,177.40               | \$3,177.40  |
| 5356.31          | 5356  | 5,356  | 5356###,00  | 5 356,31               | 5,356.31               | \$5,356.31  |
| 12423.56         | 12424 | 12,424 | 12424###,00 | 12 423,56              | 12,423.56              | \$12,423.56 |
| 21484.21         | 21484 | 21,484 | 21484###,00 | 21 484,21              | 21,484.21              | \$21,484.21 |

### Esempio: scenario per la funzione Num

Espressione del grafico

#### Panoramica

Un set di dati contiene dati sulle vendite che si desidera presentare con un formato numerico più semplice.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Customer`
  - `Sales`

#### Script di caricamento

Example:

```
Load * inline [
Customer, Sales
Customer A, 2550000
Customer B, 12500
Customer C, 15000
Customer D, 250
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- Customer

Creare la seguente misura:

- `=If(Sales > 1000000, Num(Sales/1000000, '#,##0.00M'), If(Sales > 1000, Num(Sales/1000, '#,##0.00K'), Num(Sales, '#,##0.00')))`, per impostare la rappresentazione numerica dei valori.

Results table

| Customer   | If(Sales > 1000000, Num(Sales/1000000, '#,##0.00M'), If(Sales > 1000, Num(Sales/1000, '#,##0.00K'), Num(Sales, '#,##0.00')) |
|------------|-----------------------------------------------------------------------------------------------------------------------------|
| Customer A | 2.55M                                                                                                                       |
| Customer B | 12.50K                                                                                                                      |
| Customer C | 15.00K                                                                                                                      |
| Customer D | 250.00                                                                                                                      |

L'espressione di misura nella colonna 2 utilizza la funzione `Num` per presentare i dati in un formato numerico semplificato.

### Script di caricamento

La funzione `Num` può essere utilizzata in uno script di caricamento per formattare un numero come percentuale.

### Script di caricamento

Nell'editor caricamento dati creare una nuova sezione, aggiungere lo script di esempio ed eseguirlo. Per visualizzare il risultato, aggiungere quindi i campi elencati nella colonna dei risultati a un foglio nell'app.

```
SET ThousandSep='';
SET DecimalSep='.';
Transactions:
Load
*,
Num(discount,'#,#0%') as [Discount #,#0%]
;
Load * Inline [
transaction_id, transaction_date, transaction_amount, transaction_quantity, discount,
customer_id, size, color_code
3750, 20180830, 12423.56, 23, 0,2038593, L, Red
3751, 20180907, 5356.31, 6, 0.1, 203521, m, orange
3752, 20180916, 15.75, 1, 0.22, 5646471, s, blue
3753, 20180922, 1251, 7, 0, 3036491, l, black
3754, 20180922, 21484.21, 1356, 75, 049681, xs, Red
3756, 20180922, -59.18, 2, 0.3333333333333333, 2038593, M, Blue
3757, 20180923, 3177.4, 21, .14, 203521, XL, Black
];
```

Tabella Qlik Sense che mostra i risultati della funzione `Num` utilizzata nello script di caricamento per formattare le percentuali.

| Discount           | Discount #,##0% |
|--------------------|-----------------|
| 0                  | 0%              |
| 0.1                | 10%             |
| 0.22               | 22%             |
| 0                  | 0%              |
| 75                 | 7,500%          |
| 0.3333333333333333 | 33%             |
| .14                | 14%             |

## Time

**Time()** consente di formattare un'espressione come valore ora nel formato dell'ora impostato nelle variabili di sistema nello script di caricamento dei dati o nel sistema operativo, a meno che non venga fornita una stringa di formattazione.

**Sintassi:**

**Time** (number[, format])

**Tipo di dati restituiti:** duale

### Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>number</b> | Il numero da formattare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>format</b> | <p>Stringa che descrive come formattare la stringa dell'ora risultante. Se viene omessa, si utilizza il formato dell'ora e il separatore decimale impostati nel sistema operativo.</p> <p>La funzione <code>Time</code> supporta molte opzioni di formato per diverse strutture per l'ora, ad esempio:</p> <p><code>hh:mm:ss</code>: formatta un valore orario in un formato orario standard e visualizza ore, minuti e secondi con zeri iniziali.</p> <p><code>hh:mm TT</code>: utilizza un formato di 12 ore con il suffisso AM/PM.</p> <p><code>hh:mm</code>: mostra solo le ore e i minuti, omettendo i secondi.</p> <p><code>hh.mm.ss</code>: formato orario con separatori diversi, come ad esempio un punto (.) anziché i due punti.</p> <p><code>hh TT</code>: include solo l'ora e AM/PM.</p> <p><code>hh "Hrs" mm "Mins"</code>: mostra l'ora con un testo personalizzato, ad esempio Hrs.</p> |

### Esempio: espressioni dei grafici

| Esempio                               | Risultato                                                                                                    |
|---------------------------------------|--------------------------------------------------------------------------------------------------------------|
| <code>Time( 0.375 )</code>            | Restituisce 09:00:00 quando l'impostazione <code>TimeFormat</code> è 'hh:mm:ss'.                             |
| <code>Time( 35648.375 )</code>        | Restituisce 09:00:00 quando il valore <code>Timeformat</code> fornito è 'hh:mm:ss'.                          |
| <code>Time( 0.99999, 'hh-mm' )</code> | Restituisce 23-59 ignorando l'impostazione <code>TimeFormat</code> poiché il formato è fornito nella misura. |

### Esempio: principi fondamentali della funzione Time

#### Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Il seguente campo in una tabella dati:
  - `TimeValue`
- La variabile `TimeFormat`, che utilizza il formato `hh:mm:ss`.

#### Script di caricamento

```
SET TimeFormat='hh:mm:ss';
Example:
Load * inline [
TimeValue
0.375
35648.375
0.99999
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `TimeValue`

Creare le seguenti misure:

- `=Time(TimeValue )`, per visualizzare il valore come ora formattata utilizzando l'impostazione di sistema.
- `=Time(TimeValue, 'hh-mm' )`, per visualizzare il valore come ora nel formato fornito nella funzione.
- `=Time(TimeValue, 'hh:mm:ss' )`, per visualizzare il valore come ora nel formato fornito nella funzione.

- `=Time(TimeValue, 'hh:mm TT' )`, per visualizzare il valore come ora nel formato fornito nella funzione.
- `=Time(TimeValue, 'hh:mm' )`, per visualizzare il valore come ora nel formato fornito nella funzione.
- `=Time(TimeValue, 'hh.mm.ss' )`, per visualizzare il valore come ora nel formato fornito nella funzione.
- `=Time(TimeValue, 'hh TT' )`, per visualizzare il valore come ora nel formato fornito nella funzione.
- `=Time(TimeValue, 'hh "Hrs" mm "Mins"' )`, per visualizzare il valore come ora nel formato fornito nella funzione.

Tabella dei risultati

| TimeV<br>alue | Time<br>(TimeVa<br>lue) | Time<br>(TimeVa<br>lue, 'hh-<br>mm') | Time<br>(TimeVa<br>lue,<br>'hh:mm:<br>ss') | Time<br>(TimeVa<br>lue,<br>'hh:mm<br>TT') | Time<br>(TimeVa<br>lue,<br>'hh:mm') | Time<br>(TimeVa<br>lue,<br>'hh.mm.<br>ss') | Time<br>(TimeVa<br>lue, 'hh<br>TT') | Time<br>(TimeVa<br>lue, 'hh<br>"Hrs"<br>mm<br>"Mins"') |
|---------------|-------------------------|--------------------------------------|--------------------------------------------|-------------------------------------------|-------------------------------------|--------------------------------------------|-------------------------------------|--------------------------------------------------------|
| 0.375         | 09:00:00                | 09-00                                | 09:00:00                                   | 09:00<br>AM                               | 09:00                               | 09.00.00                                   | 09 AM                               | 09 Ore<br>00 Min                                       |
| 0.99999       | 23:59:59                | 23-59                                | 23:59:59                                   | 11:59<br>PM                               | 23:59                               | 23.59.59                                   | 11 PM                               | 23 Hrs<br>59 Mins                                      |
| 35648.<br>375 | 09:00:00                | 09-00                                | 09:00:00                                   | 09:00<br>AM                               | 09:00                               | 09.00.00                                   | 09 AM                               | 09 Hrs<br>00 Mins                                      |

I risultati mostrano come la funzione `Time` può visualizzare i valori in base al formato fornito nell'espressione della funzione.

### Esempio: principi fondamentali della funzione `Time` che utilizza un'impostazione `TimeFormat` differente

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Il seguente campo in una tabella dati:
  - `TimeValue`
- La variabile `TimeFormat`, che utilizza il formato `hh.mm.ss`.

### Script di caricamento

```
SET TimeFormat='hh.mm.ss';
Example:
Load * inline [
TimeValue
0.375
35648.375
0.99999
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- TimeValue

Creare le seguenti misure:

- =Time(TimeValue ), per visualizzare il valore come ora formattata utilizzando l'impostazione di sistema.
- =Time(TimeValue, 'hh-mm' ), per visualizzare il valore come ora nel formato fornito nella funzione.
- =Time(TimeValue, 'hh:mm:ss' ), per visualizzare il valore come ora nel formato fornito nella funzione.
- =Time(TimeValue, 'hh:mm TT' ), per visualizzare il valore come ora nel formato fornito nella funzione.
- =Time(TimeValue, 'hh:mm' ), per visualizzare il valore come ora nel formato fornito nella funzione.
- =Time(TimeValue, 'hh.mm.ss' ), per visualizzare il valore come ora nel formato fornito nella funzione.
- =Time(TimeValue, 'hh TT' ), per visualizzare il valore come ora nel formato fornito nella funzione.
- =Time(TimeValue, 'hh "Hrs" mm "Mins"' ), per visualizzare il valore come ora nel formato fornito nella funzione.

Tabella dei risultati

| TimeV<br>alue | Time<br>(TimeVa<br>lue) | Time<br>(TimeVa<br>lue, 'hh-<br>mm') | Time<br>(TimeVa<br>lue,<br>'hh:mm:<br>ss') | Time<br>(TimeVa<br>lue,<br>'hh:mm<br>TT') | Time<br>(TimeVa<br>lue,<br>'hh:mm') | Time<br>(TimeVa<br>lue,<br>'hh.mm.<br>ss') | Time<br>(TimeVa<br>lue, 'hh<br>TT') | Time<br>(TimeVa<br>lue, 'hh<br>"Hrs"<br>mm<br>"Mins") |
|---------------|-------------------------|--------------------------------------|--------------------------------------------|-------------------------------------------|-------------------------------------|--------------------------------------------|-------------------------------------|-------------------------------------------------------|
| 0.375         | 09.00.00                | 09-00                                | 09:00:00                                   | 09:00<br>AM                               | 09:00                               | 09.00.00                                   | 09 AM                               | 09 Hrs<br>00 Mins                                     |
| 0.99999       | 23.59.59                | 23-59                                | 23:59:59                                   | 11:59<br>PM                               | 23:59                               | 23.59.59                                   | 11 PM                               | 23 Hrs<br>59 Mins                                     |
| 35648.<br>375 | 09.00.00                | 09-00                                | 09:00:00                                   | 09:00<br>AM                               | 09:00                               | 09.00.00                                   | 09 AM                               | 09 Hrs<br>00 Mins                                     |

I risultati mostrano come la funzione `Time` può visualizzare i valori in base al formato fornito nell'espressione della funzione.

Timestamp

`TimeStamp()` consente di formattare un'espressione come valore data e ora nel formato dell'indicatore temporale impostato nelle variabili di sistema nello script di caricamento dei dati o nel sistema operativo, a meno che non venga fornita una stringa di formattazione.

Sintassi:

```
TimeStamp(number[, format])
```

Tipo di dati restituiti: duale

| Argomenti |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Argomento | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| number    | Il numero da formattare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| format    | <p>Stringa che descrive come formattare la stringa dell'indicatore temporale risultante. Se viene omessa, si utilizzerà il formato della data breve, il formato dell'ora e il separatore decimale impostati nel sistema operativo.</p> <p>La funzione <code>TimeStamp</code> supporta molte opzioni di formato per diverse strutture per l'ora, ad esempio:</p> <p><code>DD/MM/YYYY</code>: si concentra solo sulla data (ad esempio: 15/11/2024).</p> <p><code>YYYY-MM-DD hh:mm:ss</code>: visualizza il timestamp completo (ad esempio: 2024-11-15 13:30:45).</p> <p><code>MMM DD, YYYY</code>: visualizza una versione breve del mese e l'anno completo (ad esempio: Nov 15, 2024).</p> <p><code>hh:mm:ss TT</code>: mostra solo l'ora nel formato di 12 ore (per esempio: 01:30:45 PM)</p> <p><code>HH:mm:ss</code>: visualizza l'ora nel formato a 24 ore (ad esempio: 13:30:45).</p> <p><code>www, MMM DD YYYY</code>: mostra il giorno completo della settimana e la data (ad esempio: Friday, Nov 15 2024).</p> <p><code>YYYY-MM-DD hh:mm:ss.fff</code>: include i millisecondi (ad esempio: 2024-11-15 13:30:45.123).</p> |

Esempio: espressioni dei grafici

| Esempio                                   | Risultato                                                                                                      |
|-------------------------------------------|----------------------------------------------------------------------------------------------------------------|
| <code>TimeStamp( 35648 )</code>           | Restituisce 97-08-06 00:00:00 quando il valore <code>TimeStampFormat</code> è <code>YY-MM-DD hh:mm:ss</code> . |
| <code>TimeStamp(35648, 'YYYY-MM-DD</code> | Restituisce 1997-08-06 00.00 quando il formato fornito è <code>YYYY-MM-</code>                                 |

| Esempio                                                     | Risultato                                                                                    |
|-------------------------------------------------------------|----------------------------------------------------------------------------------------------|
| <code>hh.mm')</code>                                        | <code>DD hh.mm.</code>                                                                       |
| <code>Timestamp( 35648, 'M/D/YYYY h:mm:ss[.fff] TT')</code> | Restituisce 8/6/1997 12:00:00 AM quando il formato fornito è M/D/YYYY h:mm:ss[.fff] TT.      |
| <code>Timestamp( 35648, 'www, DD MMM YYYY hh:mm TT')</code> | Restituisce wed, 06 Aug 1997 12:00 AM quando il formato fornito è www, DD MMM YYYY hh:mm TT. |

### Esempio: principi fondamentali della funzione Timestamp

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Il seguente campo in una tabella dati:
  - `TimestampValue`
- La variabile `TimestampFormat`, che utilizza il formato `YY-MM-DD hh:mm:ss`.

#### Script di caricamento

```
SET TimestampFormat='YY-MM-DD hh:mm:ss';
Example:
Load * inline [
TimestampValue
35648.375
35648
36648.875
36648.8756789
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `TimestampValue`

Creare le seguenti misure:

- `=Timestamp(TimestampValue)` per visualizzare il valore come formato timestamp utilizzando l'impostazione di sistema.
- `=Timestamp(TimestampValue, 'YYYY-MM-DD hh.mm')` per visualizzare il valore come formato timestamp che utilizza il formato fornito nella funzione.

- `=Timestamp(TimestampValue, 'M/D/YYYY h:mm:ss[.fff] TT')` per visualizzare il valore come formato timestamp che utilizza il formato fornito nella funzione.
- `=Timestamp(TimestampValue, 'YYYY-MM-DD hh:mm:ss.fff')` per visualizzare il valore come formato timestamp che utilizza il formato fornito nella funzione.
- `=Timestamp(TimestampValue, 'www, DD MMM YYYY hh:mm TT')` per visualizzare il valore come formato timestamp che utilizza il formato fornito nella funzione.
- `=Timestamp(TimestampValue, 'www, MMM DD YYYY')` per visualizzare il valore come formato timestamp che utilizza il formato fornito nella funzione.

Tabella dei risultati

| TimeSt<br>ampVal<br>ue | TimeSta<br>mp<br>(TimeSta<br>mpValu<br>e) | Timestamp<br>(Timestamp<br>Value, 'YYY<br>Y-MM-DD<br>hh.mm') | Timestamp<br>(TimestampV<br>alue, 'M/D/YYYY<br>h:mm:ss[.fff]<br>TT') | Timestamp<br>(Timestamp<br>Value, 'YYY<br>Y-MM-DD<br>hh:mm:ss.fff<br>'f') | Timestamp<br>(TimeStam<br>pValue, 'ww<br>w, DD MMM<br>YYYY<br>hh:mm TT') | Timestamp<br>(TimeStam<br>pValue, 'www<br>w, MMM DD<br>YYYY') |
|------------------------|-------------------------------------------|--------------------------------------------------------------|----------------------------------------------------------------------|---------------------------------------------------------------------------|--------------------------------------------------------------------------|---------------------------------------------------------------|
| 35648                  | 97-08-06<br>00:00:00                      | 1997-08-06<br>00.00                                          | 8/6/1997<br>12:00:00 AM                                              | 1997-08-06<br>00:00:00.000                                                | Mer, 06 ago<br>1997 12:00<br>AM                                          | Mercoledì, 06<br>agosto 1997                                  |
| 35648.3<br>75          | 97-08-06<br>09:00:00                      | 1997-08-06<br>09.00                                          | 8/6/1997<br>9:00:00 AM                                               | 1997-08-06<br>09:00:00.000                                                | Mer, 06 ago<br>1997 09:00<br>AM                                          | Mercoledì, 06<br>agosto 1997                                  |
| 36648.8<br>75          | 00-05-02<br>21:00:00                      | 2000-05-02<br>21.00                                          | 5/2/2000<br>9:00:00 PM                                               | 2000-05-02<br>21:00:00.000                                                | Mar, 02<br>maggio<br>2000 09:00<br>PM                                    | Martedì, 02<br>maggio 2000                                    |
| 36648.8<br>756789      | 00-05-02<br>21:00:58                      | 2000-05-02<br>21.00                                          | 5/2/2000<br>9:00:58 PM                                               | 2000-05-02<br>21:00:58.657                                                | Mar, 02<br>maggio<br>2000 09:00<br>PM                                    | Martedì, 02<br>maggio 2000                                    |

I risultati mostrano come la funzione `TimestampFormat` può visualizzare i valori utilizzando diversi parametri di formato.

### Esempio: principi fondamentali della funzione `Timestamp` che utilizza un'impostazione `TimestampFormat` differente

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Il seguente campo in una tabella dati:
  - `TimestampValue`
- La variabile `TimestampFormat`, che utilizza il formato `YY-MM-DD hh:mm:ss`.

### Script di caricamento

```
SET TimestampFormat='M/D/YY hh:mm:ss';
Example:
Load * inline [
TimestampValue
35648.375
35648
36648.875
36648.8756789
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `TimestampValue`

Creare le seguenti misure:

- `=Timestamp(TimestampValue)` per visualizzare il valore come formato timestamp utilizzando l'impostazione di sistema.
- `=Timestamp(TimestampValue, 'YYYY-MM-DD hh.mm')` per visualizzare il valore come formato timestamp che utilizza il formato fornito nella funzione.
- `=Timestamp(TimestampValue, 'M/D/YYYY h:mm:ss[.fff] TT')` per visualizzare il valore come formato timestamp che utilizza il formato fornito nella funzione.
- `=Timestamp(TimestampValue, 'YYYY-MM-DD hh:mm:ss.fff')` per visualizzare il valore come formato timestamp che utilizza il formato fornito nella funzione.
- `=Timestamp(TimestampValue, 'www, DD MMM YYYY hh:mm TT')` per visualizzare il valore come formato timestamp che utilizza il formato fornito nella funzione.
- `=Timestamp(TimestampValue, 'www, MMM DD YYYY')` per visualizzare il valore come formato timestamp che utilizza il formato fornito nella funzione.

Tabella dei risultati

| TimeSt<br>ampVal<br>ue | Timesta<br>mp<br>(TimeSta<br>mpValu<br>e) | Timestamp<br>(TimeStamp<br>Value,'YYY<br>Y-MM-DD<br>hh.mm') | Timestamp<br>(TimeStampV<br>alue,'M/D/YYYY<br>Y h:mm:ss[.fff]<br>TT') | Timestamp<br>(TimeStamp<br>Value,'YYY<br>Y-MM-DD<br>hh:mm:ss.ff<br>f') | Timestamp<br>(TimeStam<br>pValue,'ww<br>w, DD MMM<br>YYYY<br>hh:mm TT') | Timestamp<br>(TimeStamp<br>Value,'www<br>w, MMM DD<br>YYYY') |
|------------------------|-------------------------------------------|-------------------------------------------------------------|-----------------------------------------------------------------------|------------------------------------------------------------------------|-------------------------------------------------------------------------|--------------------------------------------------------------|
| 35648                  | 8/6/97<br>00:00:00                        | 1997-08-06<br>00.00                                         | 8/6/1997<br>12:00:00 AM                                               | 1997-08-06<br>00:00:00.000                                             | Mer, 06 ago<br>1997 12:00<br>AM                                         | Mercoledì, 06<br>agosto 1997                                 |
| 35648.3<br>75          | 8/6/97<br>09:00:00                        | 1997-08-06<br>09.00                                         | 8/6/1997<br>9:00:00 AM                                                | 1997-08-06<br>09:00:00.000                                             | Mer, 06 ago<br>1997 09:00<br>AM                                         | Mercoledì, 06<br>agosto 1997                                 |
| 36648.8<br>75          | 5/2/00<br>21:00:00                        | 2000-05-02<br>21.00                                         | 5/2/2000<br>9:00:00 PM                                                | 2000-05-02<br>21:00:00.000                                             | Mar, 02<br>maggio<br>2000 09:00<br>PM                                   | Martedì, 02<br>maggio 2000                                   |
| 36648.8<br>756789      | 5/2/00<br>21:00:58                        | 2000-05-02<br>21.00                                         | 5/2/2000<br>9:00:58 PM                                                | 2000-05-02<br>21:00:58.657                                             | Mar, 02<br>maggio<br>2000 09:00<br>PM                                   | Martedì, 02<br>maggio 2000                                   |

I risultati mostrano come la funzione `TimestampFormat` può visualizzare i valori utilizzando diversi parametri di formato.

## Esempio: modifica del formato del timestamp per gli ordini dei clienti

Espressione del grafico

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Customer`
  - `CustomerName`
  - `OrderTimestamp`

### Script di caricamento

Example:

```
Load * inline [
```

```
OrderID, CustomerName, OrderTimestamp
1, Alice, 2024-02-20 11:00:00
2, Bob, 2024-03-01 15:30:00
3, Charlie, 2024-04-12 09:45:00
4, David, 2024-05-21 21:10:00
5, Eve, 2024-06-30 14:25:00
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- OrderID
- CustomerName

Creare la seguente misura:

- =Timestamp(OrderTimestamp, 'www, DD MMM YYYY hh:mm TT'), per visualizzare il timestamp dell'ordine in un formato che mostra il giorno della settimana, la data e l'ora.

Tabella dei risultati

| OrderID | CustomerName | Timestamp(OrderTimestamp, 'WWW, DD MMM YYYY hh:mm TT') |
|---------|--------------|--------------------------------------------------------|
| 1       | Alice        | Martedì, 20 feb 2024 11:00 AM                          |
| 2       | Bob          | Venerdì, 01 Mar 2024 03:30 PM                          |
| 3       | Charlie      | Venerdì 12 apr 2024 09:45 AM                           |
| 4       | David        | Martedì, 21 maggio 2024 09:10 PM                       |
| 5       | Eve          | Domenica, 30 giu 2024 02:25 PM                         |

Utilizzando la funzione `Timestamp`, i dati del timestamp vengono ora formattati per includere il giorno della settimana e il nuovo formato della data e dell'ora.

## 8.13 Funzioni numeriche generiche

Nelle seguenti funzioni numeriche generiche gli argomenti sono espressioni, dove **x** può essere interpretato come un numero reale valutato. Tutte le funzioni possono essere utilizzate sia negli script di caricamento dei dati che nelle espressioni grafiche.

### Panoramica delle funzioni numeriche generiche

Ciascuna funzione viene descritta ulteriormente dopo la panoramica. È inoltre possibile fare clic sul nome della funzione nella sintassi per accedere immediatamente ai dettagli per tale funzione specifica.

### bitcount

**BitCount()** restituisce quanti bit nell'equivalente binario di un numero decimale sono impostati su 1, ossia la funzione restituisce il numero di bit impostati in **integer\_number**, dove **integer\_number** viene interpretato come numero intero a 32 bit con segno.

```
BitCount(integer_number)
```

### div

**Div()** restituisce la parte intera della divisione aritmetica del primo argomento per il secondo argomento. Entrambi i parametri vengono interpretati come numeri reali, ossia, non devono essere numeri interi.

```
Div(integer_number1, integer_number2)
```

### fabs

**Fabs()** restituisce il valore assoluto di **x**. Il risultato è un numero positivo.

```
Fabs(x)
```

### fact

**Fact()** restituisce il fattoriale di un numero intero positivo **x**.

```
Fact(x)
```

### frac

**Frac()** restituisce la parte frazionaria di **x**.

```
Frac(x)
```

### sign

**Sign()** restituisce 1, 0 o -1 a seconda che **x** sia un numero positivo, 0 o un numero negativo.

```
Sign(x)
```

## Funzioni di combinazione e permutazione

### combin

**Combin()** restituisce il numero di combinazioni di elementi **q** che può essere scelto da un gruppo di voci **p**. È rappresentata dalla formula:  $\text{combin}(p,q) = p! / q!(p-q)!$  L'ordine con cui vengono selezionate le voci non è significativo.

```
Combin(p, q)
```

### permut

**Permut()** restituisce il numero di permutazioni di elementi **q** che può essere selezionato da una serie di voci **p**. È rappresentata dalla formula:  $\text{Permut}(p,q) = (p)! / (p - q)!$  L'ordine con cui vengono selezionate le voci è significativo.

```
Permut(p, q)
```

## Funzioni modulo

### fmod

**fmod()** è una funzione di modulo generalizzato che restituisce la parte rimanente della divisione di un numero intero del primo argomento (il dividendo) per il secondo argomento (il divisore). Il risultato è un numero reale. Entrambi gli argomenti vengono interpretati come numeri reali, ossia, non devono essere numeri interi.

```
Fmod (a, b)
```

### mod

**Mod()** è una funzione di modulo matematico che restituisce la parte restante non negativa di una divisione di numero intero. Il primo argomento è il dividendo il secondo argomento è il divisore ed entrambi gli argomenti devono essere valori interi.

```
Mod (integer_number1, integer_number2)
```

## Funzioni di parità

### even

**Even()** restituisce True (-1), se **integer\_number** è un numero intero pari o zero. Restituisce False (0), se **integer\_number** è un numero intero dispari e NULL se **integer\_number** non è un numero intero.

```
Even (integer_number)
```

### odd

**Odd()** restituisce True (-1), se **integer\_number** è un numero intero dispari o zero. Restituisce False (0), se **integer\_number** è un numero intero pari e NULL se **integer\_number** non è un numero intero.

```
Odd (integer_number)
```

## Funzioni di arrotondamento

### ceil

**Ceil()** arrotonda per eccesso un numero al multiplo più vicino di **step** modificato in base al numero di **offset** .

```
Ceil (x[, step[, offset]])
```

### floor

**Floor()** arrotonda per eccesso un numero al multiplo più vicino di **step** modificato in base al numero di **offset** .

```
Floor (x[, step[, offset]])
```

### round

**Round()** restituisce il risultato dell'arrotondamento di un numero per eccesso o per difetto al multiplo più vicino di **step** modificato in base al numero di **offset** .

```
Round (( x [ , step [ , offset ] ] )
```

## BitCount

**BitCount()** restituisce quanti bit nell'equivalente binario di un numero decimale sono impostati su 1, ossia la funzione restituisce il numero di bit impostati in **integer\_number**, dove **integer\_number** viene interpretato come numero intero a 32 bit con segno.

Sintassi:

```
BitCount(integer_number)
```

Tipo di dati restituiti: numero intero

Esempi e risultati:

Esempi e risultati

| Esempi          | Risultati                                     |
|-----------------|-----------------------------------------------|
| BitCount ( 3 )  | 3 è 11 binario, pertanto restituisce 2        |
| BitCount ( -1 ) | -1 è 64 uno in binario, quindi restituisce 64 |

## Ceil

**Ceil()** arrotonda per eccesso un numero al multiplo più vicino di **step** modificato in base al numero di **offset**.

Confrontare con la funzione **floor**, la quale consente di arrotondare i numeri di input per difetto.

Sintassi:

```
Ceil(x[, step[, offset]])
```

Tipo di dati restituiti: numerico

Argomenti:

Argomenti

| Argomento     | Descrizione                                                              |
|---------------|--------------------------------------------------------------------------|
| <b>x</b>      | Numero di input.                                                         |
| <b>step</b>   | Incremento intervallo. Il valore predefinito è 1.                        |
| <b>offset</b> | Definisce la base dell'intervallo della fase. Il valore predefinito è 0. |

## Esempi e risultati:

| Esempi e risultati              |                                                                                                                                                                                                                                                                                                    |
|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Esempi                          | Risultati                                                                                                                                                                                                                                                                                          |
| <code>ceil(2.4 )</code>         | Restituisce 3<br><br>In questo esempio, la dimensione della fase è uno e la base dell'intervallo della fase è 0.<br><br>Gli intervalli sono ... $0 < x \leq 1$ , $1 < x \leq 2$ , <b><math>2 &lt; x \leq 3</math></b> , $3 < x \leq 4$ ...                                                         |
| <code>ceil(4.2 )</code>         | Restituisce 5                                                                                                                                                                                                                                                                                      |
| <code>ceil(3.88 ,0.1)</code>    | Restituisce 3,9<br><br>In questo esempio, la dimensione dell'intervallo è 0.1 e la base dell'intervallo è 0.<br><br>Gli intervalli sono ... $3.7 < x \leq 3.8$ , <b><math>3.8 &lt; x \leq 3.9</math></b> , $3.9 < x \leq 4.0$ ...                                                                  |
| <code>ceil(3.88 ,5)</code>      | Restituisce 5                                                                                                                                                                                                                                                                                      |
| <code>ceil(1.1 ,1)</code>       | Restituisce 2                                                                                                                                                                                                                                                                                      |
| <code>ceil(1.1 ,1,0.5)</code>   | Restituisce 1,5<br><br>In questo esempio, la dimensione della fase è 1 e l'offset è 0,5. Significa che la base dell'intervallo della fase è 0,5 e non 0.<br><br>Gli intervalli sono ... <b><math>0.5 &lt; x \leq 1.5</math></b> , $1.5 < x \leq 2.5$ , $2.5 < x \leq 3.5$ , $3.5 < x \leq 4.5$ ... |
| <code>ceil(1.1 ,1,-0.01)</code> | Restituisce 1,99<br><br>Gli intervalli sono ... $-0.01 < x \leq 0.99$ , <b><math>0.99 &lt; x \leq 1.99</math></b> , $1.99 < x \leq 2.99$ ...                                                                                                                                                       |

## Combin

**Combin()** restituisce il numero di combinazioni di elementi **q** che può essere scelto da un gruppo di voci **p**. È rappresentata dalla formula:  $\text{combin}(p,q) = p! / q!(p-q)!$  L'ordine con cui vengono selezionate le voci non è significativo.

## Sintassi:

```
Combin (p, q)
```

**Tipo di dati restituiti:** numero intero

## Limiti:

Le voci che non sono numeri interi vengono troncate.

**Esempi e risultati:**

Esempi e risultati

| Esempi                                                                                                                             | Risultati                |
|------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| Quante combinazioni di 7 numeri possono essere scelte da un totale di 35 numeri a disposizione?<br><br><code>Combin( 35,7 )</code> | Restituisce<br>6.724.520 |

## Div

**Div()** restituisce la parte intera della divisione aritmetica del primo argomento per il secondo argomento. Entrambi i parametri vengono interpretati come numeri reali, ossia, non devono essere numeri interi.

**Sintassi:**

```
Div(integer_number1, integer_number2)
```

**Tipo di dati restituiti:** numero intero

**Esempi e risultati:**

Esempi e risultati

| Esempi                      | Risultati      |
|-----------------------------|----------------|
| <code>Div( 7,2 )</code>     | Restituisce 3  |
| <code>Div( 7.1,2.3 )</code> | Restituisce 3  |
| <code>Div( 9,3 )</code>     | Restituisce 3  |
| <code>Div( -4,3 )</code>    | Restituisce -1 |
| <code>Div( 4,-3 )</code>    | Restituisce -1 |
| <code>Div( -4,-3 )</code>   | Restituisce 1  |

## Even

**Even()** restituisce True (-1), se **integer\_number** è un numero intero pari o zero. Restituisce False (0), se **integer\_number** è un numero intero dispari e NULL se **integer\_number** non è un numero intero.

**Sintassi:**

```
Even(integer_number)
```

**Tipo di dati restituiti:** Booleano

**Esempi e risultati:**

Esempi e risultati

| Esempi         | Risultati            |
|----------------|----------------------|
| Even( 3 )      | Restituisce 0, False |
| Even( 2 * 10 ) | Restituisce -1, True |
| Even( 3.14 )   | Restituisce NULL     |

## Fabs

**Fabs()** restituisce il valore assoluto di **x**. Il risultato è un numero positivo.

**Sintassi:**

**fabs (x)**

**Tipo di dati restituiti:** numerico

**Esempi e risultati:**

Esempi e risultati

| Esempi       | Risultati       |
|--------------|-----------------|
| fabs( 2.4 )  | Restituisce 2,4 |
| fabs( -3.8 ) | Restituisce 3,8 |

## Fact

**Fact()** restituisce il fattoriale di un numero intero positivo **x**.

**Sintassi:**

**Fact (x)**

**Tipo di dati restituiti:** numero intero

**Limiti:**

Se il numero **x** non è un numero intero, verrà troncato. I numeri non positivi restituiranno NULL.

## Esempi e risultati:

Esempi e risultati

| Esempi     | Risultati                                     |
|------------|-----------------------------------------------|
| Fact( 1 )  | Restituisce 1                                 |
| Fact( 5 )  | Restituisce 120 ( $1 * 2 * 3 * 4 * 5 = 120$ ) |
| Fact( -5 ) | Restituisce NULL                              |

## Floor

**Floor()** arrotonda per eccesso un numero al multiplo più vicino di **step** modificato in base al numero di **offset**.

Confrontarla con la funzione **ceil**, che arrotonda per eccesso i numeri in input.

## Sintassi:

```
Floor(x[, step[, offset]])
```

Tipo di dati restituiti: numerico

## Argomenti:

Argomenti

| Argomento     | Descrizione                                                              |
|---------------|--------------------------------------------------------------------------|
| <b>x</b>      | Numero di input.                                                         |
| <b>step</b>   | Incremento intervallo. Il valore predefinito è 1.                        |
| <b>offset</b> | Definisce la base dell'intervallo della fase. Il valore predefinito è 0. |

## Esempi e risultati:

Esempi e risultati

| Esempi     | Risultati                                                                                                                                                                                                                        |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Floor(2.4) | Restituisce 2<br><br>In this example, the size of the step is 1 and the base of the step interval is 0.<br><br>The intervals are ... $0 \leq x < 1$ , $1 \leq x < 2$ , <b><math>2 \leq x &lt; 3</math></b> , $3 \leq x < 4$ .... |
| Floor(4.2) | Restituisce 4                                                                                                                                                                                                                    |

| Esempi                         | Risultati                                                                                                                                                                                                                                                                      |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>Floor(3.88 ,0.1)</code>  | Restituisce 3,8<br><br>In questo esempio la dimensione dell'intervallo è 0,1 e la base dell'intervallo è 0.<br><br>Gli intervalli sono ... $3.7 \leq x < 3.8$ , <b><math>3.8 \leq x &lt; 3.9</math></b> , $3.9 \leq x < 4.0$ ...                                               |
| <code>Floor(3.88 ,5)</code>    | Restituisce 0                                                                                                                                                                                                                                                                  |
| <code>Floor(1.1 ,1)</code>     | Restituisce 1                                                                                                                                                                                                                                                                  |
| <code>Floor(1.1 ,1,0.5)</code> | Restituisce 0,5<br><br>In questo esempio, la dimensione della fase è 1 e l'offset è 0,5. Significa che la base dell'intervallo della fase è 0,5 e non 0.<br><br>Gli intervalli sono ... <b><math>0.5 \leq x &lt; 1.5</math></b> , $1.5 \leq x < 2.5$ , $2.5 \leq x < 3.5$ ,... |

## Fmod

**fmod()** è una funzione di modulo generalizzato che restituisce la parte rimanente della divisione di un numero intero del primo argomento (il dividendo) per il secondo argomento (il divisore). Il risultato è un numero reale. Entrambi gli argomenti vengono interpretati come numeri reali, ossia, non devono essere numeri interi.

### Sintassi:

```
fmod (a, b)
```

**Tipo di dati restituiti:** numerico

### Argomenti:

| Argomenti |             |
|-----------|-------------|
| Argomento | Descrizione |
| <b>a</b>  | Dividendo   |
| <b>b</b>  | Divisore    |

### Esempi e risultati:

| Esempi e risultati         |                 |
|----------------------------|-----------------|
| Esempi                     | Risultati       |
| <code>fmod( 7,2 )</code>   | Restituisce 1   |
| <code>fmod( 7.5,2 )</code> | Restituisce 1,5 |
| <code>fmod( 9,3 )</code>   | Restituisce 0   |

| Esempi                      | Risultati      |
|-----------------------------|----------------|
| <code>fmod( -4, 3 )</code>  | Restituisce -1 |
| <code>fmod( 4, -3 )</code>  | Restituisce 1  |
| <code>fmod( -4, -3 )</code> | Restituisce -1 |

## Frac

**Frac()** restituisce la parte frazionaria di **x**.

La frazione viene definita in modo tale che  $\text{Frac}(x) + \text{Floor}(x) = x$ . In termini semplici questo significa che la parte frazionaria di un numero positivo è la differenza tra il numero (**x**) e il numero intero che precede la parte frazionaria.

Ad esempio: la parte frazionaria di 11,43 = 11,43 - 11 = 0,43

Per un numero negativo, ad esempio -1,4,  $\text{Floor}(-1.4) = -2$ , il quale produce il seguente risultato:

La parte frazionaria di -1,4 = 1,4 - (-2) = -1,4 + 2 = 0,6

### Sintassi:

```
Frac (x)
```

**Tipo di dati restituiti:** numerico

### Argomenti:

#### Argomenti

| Argomento | Descrizione                            |
|-----------|----------------------------------------|
| <b>x</b>  | Numero per cui restituire la frazione. |

### Esempi e risultati:

#### Esempi e risultati

| Esempi                                                                                                                                                    | Risultati              |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| <code>Frac( 11.43 )</code>                                                                                                                                | Restituisce 0,43       |
| <code>Frac( -1.4 )</code>                                                                                                                                 | Restituisce 0,6        |
| Estrarre il componente temporale dalla rappresentazione numerica di un timestamp, omettendo così la data.<br><br><code>Time(Frac(44518.663888889))</code> | Restituisce 3:56:00 PM |

## Mod

**Mod()** è una funzione di modulo matematico che restituisce la parte restante non negativa di una divisione di numero intero. Il primo argomento è il dividendo il secondo argomento è il divisore ed entrambi gli argomenti devono essere valori interi.

### Sintassi:

```
Mod(integer_number1, integer_number2)
```

**Tipo di dati restituiti:** numero intero

### Limiti:

**integer\_number2** deve essere maggiore di 0.

### Esempi e risultati:

Esempi e risultati

| Esempi       | Risultati        |
|--------------|------------------|
| Mod( 7,2 )   | Restituisce 1    |
| Mod( 7.5,2 ) | Restituisce NULL |
| Mod( 9,3 )   | Restituisce 0    |
| Mod( -4,3 )  | Restituisce 2    |
| Mod( 4,-3 )  | Restituisce NULL |
| Mod( -4,-3 ) | Restituisce NULL |

## Odd

**Odd()** restituisce True (-1), se **integer\_number** è un numero intero dispari o zero. Restituisce False (0), se **integer\_number** è un numero intero pari e NULL se **integer\_number** non è un numero intero.

### Sintassi:

```
Odd(integer_number)
```

**Tipo di dati restituiti:** Booleano

### Esempi e risultati:

Esempi e risultati

| Esempi        | Risultati            |
|---------------|----------------------|
| odd( 3 )      | Restituisce -1, True |
| odd( 2 * 10 ) | Restituisce 0, False |
| odd( 3.14 )   | Restituisce NULL     |

## Permut

**Permut()** restituisce il numero di permutazioni di elementi **q** che può essere selezionato da una serie di voci **p**. È rappresentata dalla formula:  $\text{Permut}(p, q) = (p)! / (p - q)!$  L'ordine con cui vengono selezionate le voci è significativo.

### Sintassi:

```
Permut (p, q)
```

**Tipo di dati restituiti:** numero intero

### Limiti:

Gli argomenti che non sono numeri interi vengono troncati.

### Esempi e risultati:

#### Esempi e risultati

| Esempi                                                                                                                                                                  | Risultati          |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|
| In quanti modi possono essere distribuite le medaglie d'oro, di argento e di bronzo dopo una finale dei 100 metri con 8 partecipanti?<br><br><code>Permut( 8,3 )</code> | Restituisce<br>336 |

## Round

**Round()** restituisce il risultato dell'arrotondamento di un numero per eccesso o per difetto al multiplo più vicino di **step** modificato in base al numero di **offset**.

Se il numero da arrotondare è esattamente alla metà di un intervallo, viene arrotondato per eccesso.

### Sintassi:

```
Round (x[, step[, offset]])
```

**Tipo di dati restituiti:** numerico



*Se si sta eseguendo l'arrotondamento di un numero a virgola mobile, si potrebbero ottenere risultati non corretti. Questi errori di arrotondamento sono dovuti al fatto che i numeri a virgola mobile vengono rappresentati da un numero finito di cifre binarie. Pertanto, i risultati vengono calcolati utilizzando un numero già arrotondato. Se questi errori di arrotondamento influiscono sul proprio lavoro, moltiplicare i numeri per convertirli in numeri interi prima dell'arrotondamento.*

## Argomenti:

## Argomenti

| Argomento     | Descrizione                                                              |
|---------------|--------------------------------------------------------------------------|
| <b>x</b>      | Numero di input.                                                         |
| <b>step</b>   | Incremento intervallo. Il valore predefinito è 1.                        |
| <b>offset</b> | Definisce la base dell'intervallo della fase. Il valore predefinito è 0. |

## Esempi e risultati:

## Esempi e risultati

| Esempi        | Risultati                                                                                                                                                                                                                                                                                                              |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Round(3.8 )   | Restituisce 4<br><br>In questo esempio, la dimensione della fase è uno e la base dell'intervallo della fase è 0.<br><br>Gli intervalli sono ... $0 \leq x < 1$ , $1 \leq x < 2$ , $2 \leq x < 3$ , <b><math>3 \leq x &lt; 4</math></b> ...                                                                             |
| Round(3.8,4 ) | Restituisce 4                                                                                                                                                                                                                                                                                                          |
| Round(2.5 )   | Restituisce 3.<br><br>In questo esempio, la dimensione della fase è uno e la base dell'intervallo della fase è 0.<br><br>Gli intervalli sono ... $0 \leq x < 1$ , $1 \leq x < 2$ , <b><math>2 \leq x &lt; 3</math></b> ...                                                                                             |
| Round(2,4 )   | Restituisce 4. Viene arrotondato in eccesso perché 2 è esattamente la metà dell'intervallo della fase di 4.<br><br>In questo esempio, la dimensione della fase è 4 e la base dell'intervallo della fase è 0.<br><br>Gli intervalli sono ... <b><math>0 \leq x &lt; 4</math></b> , $4 \leq x < 8$ , $8 \leq x < 12$ ... |
| Round(2,6 )   | Restituisce 0. Viene arrotondato per difetto perché 2 è meno della metà dell'intervallo della fase di 6.<br><br>In questo esempio, la dimensione della fase è 6 e la base dell'intervallo della fase è 0.<br><br>Gli intervalli sono ... <b><math>0 \leq x &lt; 6</math></b> , $6 \leq x < 12$ , $12 \leq x < 18$ ...  |

| Esempi                | Risultati                                                                                                                                                                                                       |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Round(3.88 ,0.1)      | Restituisce 3.9<br><br>In questo esempio, la dimensione della fase è 0.1 e la base dell'intervallo della fase è 0.<br><br>Gli intervalli sono ... 3.7 <= x <3.8, <b>3.8 &lt;= x &lt;3.9</b> , 3.9 <= x < 4.0... |
| Round(3.88875,1/1000) | Restituisce 3.889<br><br>In questo esempio, la dimensione del passo è 0,001, che arrotonda il numero per eccesso e lo limita a tre cifre decimali.                                                              |
| Round(3.88 ,5)        | Restituisce 5                                                                                                                                                                                                   |
| Round(1.1 ,1,0.5)     | Restituisce 1,5<br><br>In questo esempio, la dimensione della fase è 1 e la base dell'intervallo della fase è 0,5.<br><br>Gli intervalli sono ... <b>0.5 &lt;= x &lt;1.5</b> , 1.5 <= x <2.5, 2.5<= x <3.5...   |

## Sign

**Sign()** restituisce 1, 0 o -1 a seconda che **x** sia un numero positivo, 0 o un numero negativo.

**Sintassi:**

**Sign(x)**

**Tipo di dati restituiti:** numerico

**Limiti:**

Se non viene trovato nessun valore numerico, viene restituito NULL.

**Esempi e risultati:**

Esempi e risultati

| Esempi        | Risultati      |
|---------------|----------------|
| sign( 66 )    | Restituisce 1  |
| sign( 0 )     | Restituisce 0  |
| sign( - 234 ) | Restituisce -1 |

## 8.14 Funzioni geospaziali

Queste funzioni vengono utilizzate per gestire i dati geospaziali nelle visualizzazioni delle mappe. Qlik Sense segue le specifiche GeoJSON per i dati geospaziali e supporta i tipi di geometria seguenti:

- Point
- Linestring
- Polygon
- Multipolygon

Per ulteriori informazioni sulle specifiche GeoJSON, vedere:

≤ [GeoJSON.org](https://geojson.org)

### Panoramica delle funzioni geospaziali

Ciascuna funzione viene descritta ulteriormente dopo la panoramica. È inoltre possibile fare clic sul nome della funzione nella sintassi per accedere immediatamente ai dettagli per tale funzione specifica.

Sono disponibili due categorie di funzioni geospaziali: funzioni di aggregazione e funzioni di non aggregazione.

Le funzioni di aggregazione prendono un set geometrico (punti o aree) come input e restituiscono una singola geometria. Ad esempio, più aree possono essere unite e sulla mappa può essere disegnato un singolo limite per l'aggregazione.

Una funzione di non aggregazione prende una singola geometria e restituisce una geometria. Ad esempio, per la funzione `GeoGetPolygonCenter()`, se la geometria del limite di un'area viene impostata come input, per il centro di tale area viene restituita la geometria del punto (longitudine e latitudine).

Le funzioni di aggregazione sono le seguenti:

#### GeoAggrGeometry

La funzione **GeoAggrGeometry()** viene utilizzata per aggregare un numero di aree in un'area più estesa, aggregando ad esempio un numero di sottoregioni in una singola regione.

```
GeoAggrGeometry (field_name)
```

#### GeoBoundingBox

La funzione **GeoBoundingBox()** viene utilizzata per aggregare una geometria in un'area e calcolare la casella di delimitazione più piccola contenente tutte le coordinate.

```
GeoBoundingBox (field_name)
```

#### GeoCountVertex

La funzione **GeoCountVertex()** viene utilizzata per trovare il numero di vertici contenuti nella geometria di un poligono.

```
GeoCountVertex (field_name)
```

### GeoInvProjectGeometry

La funzione **GeoInvProjectGeometry()** viene utilizzata per aggregare una geometria in un'area e applicare l'inverso di una proiezione.

```
GeoInvProjectGeometry(type, field_name)
```

### GeoProjectGeometry

La funzione **GeoProjectGeometry()** viene utilizzata per aggregare una geometria in un'area e applicare una proiezione.

```
GeoProjectGeometry(type, field_name)
```

### GeoReduceGeometry

La funzione **GeoReduceGeometry()** viene utilizzata per ridurre il numero di vertici di una geometria e per aggregare un certo numero di aree in una singola area, continuando a visualizzare le linee di confine delle singole aree.

```
GeoReduceGeometry(geometry)
```

Le funzioni di non aggregazione sono le seguenti:

### GeoGetBoundingBox

La funzione **GeoGetBoundingBox()** viene utilizzata negli script e nelle espressioni grafiche per calcolare la casella di delimitazione geospaziale più piccola contenente tutte le coordinate di una geometria.

```
GeoGetBoundingBox(geometry)
```

### GeoGetPolygonCenter

La funzione **GeoGetPolygonCenter()** viene utilizzata negli script e nelle espressioni grafiche per calcolare e restituire il punto centrale di una geometria.

```
GeoGetPolygonCenter(geometry)
```

### GeoMakePoint

La funzione **GeoMakePoint()** viene utilizzata negli script e nelle espressioni grafiche per creare e assegnare un tag a un punto con latitudine e longitudine.

```
GeoMakePoint(lat_field_name, lon_field_name)
```

### GeoProject

La funzione **GeoProject()** viene utilizzata negli script e nelle espressioni grafiche per applicare una proiezione a una geometria.

```
GeoProject(type, field_name)
```

### GeoAggrGeometry

La funzione **GeoAggrGeometry()** viene utilizzata per aggregare un numero di aree in un'area più estesa, aggregando ad esempio un numero di sottoregioni in una singola regione.

### Sintassi:

```
GeoAggrGeometry(field_name)
```

Tipo di dati restituiti: stringa

### Argomenti:

#### Argomenti

| Argomento  | Descrizione                                                                                                                                                                                                       |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| field_name | Un campo o un'espressione che fa riferimento a un campo contenente la geometria da rappresentare. Potrebbe trattarsi di un punto (o di una serie di punti) che fornisce la longitudine e la latitudine o un'area. |

In genere, la funzione **GeoAggrGeometry()** può essere utilizzata per combinare i dati relativi ai limiti geospaziali. Ad esempio, si potrebbe disporre di aree di codice di avviamento postale per le parti periferiche di una città e dei ricavi delle vendite per ciascuna area. Se l'area del venditore copre più aree di codice di avviamento postale, potrebbe risultare utile presentare le vendite totali in base all'area di vendita, anziché singole aree, e visualizzare i risultati su una mappa codificata con i colori.

**GeoAggrGeometry()** può calcolare l'aggregazione delle singole geometrie della periferia e generare la geometria dei territori uniti in un modello dati. Quindi, se i limiti dell'area di vendita vengono modificati, quando i dati vengono ricaricati i nuovi limiti uniti e i ricavi verranno riprodotti nella mappa.

Poiché **GeoAggrGeometry()** è una funzione di aggregazione, se viene utilizzata nello script è necessario utilizzare un'istruzione **LOAD** con una clausola **Group by**.



*Le linee dei limiti delle mappe create mediante **GeoAggrGeometry()** sono quelle delle aree unite. Se si desidera visualizzare le linee dei singoli limiti delle aree preaggregate, utilizzare **GeoReduceGeometry()**.*

### Esempi:

In questo esempio viene caricato un file KML con dati di area e quindi una tabella con i dati di area aggregati.

```
[MapSource]: LOAD [world.Name], [world.Point], [world.Area] FROM [lib://Downloads/world.kml]
(kml, Table is [world.shp/Features]); Map: LOAD world.Name, GeoAggrGeometry(world.Area) as
[AggrArea] resident MapSource Group By world.Name;
```

```
Drop Table MapSource;
```

## GeoBoundingBox

La funzione **GeoBoundingBox()** viene utilizzata per aggregare una geometria in un'area e calcolare la casella di delimitazione più piccola contenente tutte le coordinate.

Una funzione **GeoBoundingBox** è rappresentata come elenco di quattro valori: sinistro, destro, superiore, inferiore.

### Sintassi:

```
GeoBoundingBox(field_name)
```

Tipo di dati restituiti: stringa

### Argomenti:

#### Argomenti

| Argomento  | Descrizione                                                                                                                                                                                                       |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| field_name | Un campo o un'espressione che fa riferimento a un campo contenente la geometria da rappresentare. Potrebbe trattarsi di un punto (o di una serie di punti) che fornisce la longitudine e la latitudine o un'area. |

GeoBoundingBox() aggrega una serie di geometrie e restituisce quattro coordinate per il rettangolo più piccolo che contiene tutte le coordinate della geometria aggregata.

Per visualizzare il risultato su una mappa, è necessario trasferire la stringa risultante di quattro coordinate in un formato di poligono, contrassegnare il campo trasferito con un formato di geopoligono e trascinare tale campo nell'oggetto mappa. Le caselle rettangolari verranno quindi visualizzate nella visualizzazione della mappa.

## GeoCountVertex

La funzione **GeoCountVertex()** viene utilizzata per trovare il numero di vertici contenuti nella geometria di un poligono.

### Sintassi:

```
GeoCountVertex(field_name)
```

Tipo di dati restituiti: numero intero

### Argomenti:

#### Argomenti

| Argomento  | Descrizione                                                                                                                                                                                                       |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| field_name | Un campo o un'espressione che fa riferimento a un campo contenente la geometria da rappresentare. Potrebbe trattarsi di un punto (o di una serie di punti) che fornisce la longitudine e la latitudine o un'area. |

## GeoGetBoundingBox

La funzione **GeoGetBoundingBox()** viene utilizzata negli script e nelle espressioni grafiche per calcolare la casella di delimitazione geospaziale più piccola contenente tutte le coordinate di una geometria.

Una casella di delimitazione geospaziale, creata dalla funzione `GeoBoundingBox()` è rappresentata come elenco di quattro valori: sinistro, destro, superiore, inferiore.

### Sintassi:

```
GeoBoundingBox (field_name)
```

Tipo di dati restituiti: stringa

### Argomenti:

#### Argomenti

| Argomento  | Descrizione                                                                                                                                                                                                       |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| field_name | Un campo o un'espressione che fa riferimento a un campo contenente la geometria da rappresentare. Potrebbe trattarsi di un punto (o di una serie di punti) che fornisce la longitudine e la latitudine o un'area. |



Non utilizzare la clausola **Group by** nell'editor caricamento dati con questa funzione e altre funzioni geospaziali non aggreganti, in quanto verrebbe generato un errore di caricamento.

## GeoGetPolygonCenter

La funzione `GeoGetPolygonCenter()` viene utilizzata negli script e nelle espressioni grafiche per calcolare e restituire il punto centrale di una geometria.

In alcuni casi il requisito è disegnare un punto anziché un riempimento di colore su una mappa. Se il dato geospaziale esistente è unicamente disponibile sotto forma di geometria dell'area (ad esempio un limite), utilizzare `GeoGetPolygonCenter()` per recuperare una coppia di longitudine e latitudine per il centro dell'area.

### Sintassi:

```
GeoGetPolygonCenter (field_name)
```

Tipo di dati restituiti: stringa

### Argomenti:

#### Argomenti

| Argomento  | Descrizione                                                                                                                                                                                                       |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| field_name | Un campo o un'espressione che fa riferimento a un campo contenente la geometria da rappresentare. Potrebbe trattarsi di un punto (o di una serie di punti) che fornisce la longitudine e la latitudine o un'area. |



Non utilizzare la clausola **Group by** nell'editor caricamento dati con questa funzione e altre funzioni geospaziali non aggreganti, in quanto verrebbe generato un errore di caricamento.

## GeoInvProjectGeometry

La funzione **GeoInvProjectGeometry()** viene utilizzata per aggregare una geometria in un'area e applicare l'inverso di una proiezione.

Sintassi:

```
GeoInvProjectGeometry(type, field_name)
```

Tipo di dati restituiti: stringa

Argomenti:

### Argomenti

| Argomento  | Descrizione                                                                                                                                                                                                                                                   |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| type       | Il tipo di proiezione utilizzato nella trasformazione della geometria della mappa. Questo può assumere uno di due valori: "unit", (valore predefinito), che dà come risultato una proiezione 1:1, o "mercator", che utilizza la proiezione standard Mercator. |
| field_name | Un campo o un'espressione che fa riferimento a un campo contenente la geometria da rappresentare. Potrebbe trattarsi di un punto (o di una serie di punti) che fornisce la longitudine e la latitudine o un'area.                                             |

Esempio:

### Esempio di script

| Esempio                                                                                                | Risultato                                                                                                                                                                                                                         |
|--------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| In un'istruzione Load:<br>GeoInvProjectGeometry<br>( 'mercator', AreaPolygon)<br>as InvProjectGeometry | La geometria caricata come <b>AreaPolygon</b> viene utilizzata mediante la trasformazione inversa della proiezione Mercator, quindi viene memorizzata come <b>InvProjectGeometry</b> per essere utilizzata nelle visualizzazioni. |

## GeoMakePoint

La funzione **GeoMakePoint()** viene utilizzata negli script e nelle espressioni grafiche per creare e assegnare un tag a un punto con latitudine e longitudine. GeoMakePoint restituisce punti nell'ordine longitudine e latitudine.

Sintassi:

```
GeoMakePoint(lat_field_name, lon_field_name)
```

**Tipo di dati restituiti:** stringa, formattata [longitudine, latitudine]

**Argomenti:**

Argomenti

| Argomento      | Descrizione                                                                                       |
|----------------|---------------------------------------------------------------------------------------------------|
| lat_field_name | Un campo o un'espressione che fa riferimento a un campo che rappresenta la latitudine del punto.  |
| lon_field_name | Un campo o un'espressione che fa riferimento a un campo che rappresenta la longitudine del punto. |



Non utilizzare la clausola **Group by** nell'editor caricamento dati con questa funzione e altre funzioni geospaziali non aggreganti, in quanto verrebbe generato un errore di caricamento.

### GeoProject

La funzione **GeoProject()** viene utilizzata negli script e nelle espressioni grafiche per applicare una proiezione a una geometria.

**Sintassi:**

```
GeoProject (type, field_name)
```

**Tipo di dati restituiti:** stringa

**Argomenti:**

Argomenti

| Argomento  | Descrizione                                                                                                                                                                                                                                                       |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| type       | Il tipo di proiezione utilizzato nella trasformazione della geometria della mappa. Questo può assumere uno di due valori: "unit", (valore predefinito), che dà come risultato una proiezione 1:1, o "mercator", che utilizza la proiezione standard Mercator Web. |
| field_name | Un campo o un'espressione che fa riferimento a un campo contenente la geometria da rappresentare. Potrebbe trattarsi di un punto (o di una serie di punti) che fornisce la longitudine e la latitudine o un'area.                                                 |



Non utilizzare la clausola **Group by** nell'editor caricamento dati con questa funzione e altre funzioni geospaziali non aggreganti, in quanto verrebbe generato un errore di caricamento.

Esempio:

Esempi di script

| Esempio                                                                     | Risultato                                                                                                                                 |
|-----------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| In un'istruzione Load:<br>GeoProject<br>( 'mercator',Area) as<br>GetProject | La proiezione Mercator viene applicata alla geometria caricata come <b>Area</b> e il risultato viene memorizzato come <b>GetProject</b> . |

## GeoProjectGeometry

La funzione **GeoProjectGeometry()** viene utilizzata per aggregare una geometria in un'area e applicare una proiezione.

Sintassi:

```
GeoProjectGeometry (type, field_name)
```

Tipo di dati restituiti: stringa

Argomenti:

Argomenti

| Argomento  | Descrizione                                                                                                                                                                                                                                                       |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| type       | Il tipo di proiezione utilizzato nella trasformazione della geometria della mappa. Questo può assumere uno di due valori: "unit", (valore predefinito), che dà come risultato una proiezione 1:1, o "mercator", che utilizza la proiezione standard Mercator Web. |
| field_name | Un campo o un'espressione che fa riferimento a un campo contenente la geometria da rappresentare. Potrebbe trattarsi di un punto (o di una serie di punti) che fornisce la longitudine e la latitudine o un'area.                                                 |

Esempio:

| Esempio                                                                                         | Risultato                                                                                                                                                                                          |
|-------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| In un'istruzione Load:<br>GeoProjectGeometry<br>( 'mercator',AreaPolygon)<br>as ProjectGeometry | La geometria caricata come <b>AreaPolygon</b> viene trasformata mediante la proiezione Mercator, quindi viene memorizzata come <b>ProjectGeometry</b> per essere utilizzata nelle visualizzazioni. |

## GeoReduceGeometry

La funzione **GeoReduceGeometry()** viene utilizzata per ridurre il numero di vertici di una geometria e per aggregare un certo numero di aree in una singola area, continuando a visualizzare le linee di confine delle singole aree.

Sintassi:

```
GeoReduceGeometry (field_name[, value])
```

**Tipo di dati restituiti:** stringa

**Argomenti:**

| Argomenti  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Argomento  | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| field_name | Un campo o un'espressione che fa riferimento a un campo contenente la geometria da rappresentare. Potrebbe trattarsi di un punto (o di una serie di punti) che fornisce la longitudine e la latitudine o un'area.                                                                                                                                                                                                                                                 |
| value      | <p>La quantità di riduzione da applicare alla geometria. La scala va da 0 a 1, in cui 0 rappresenta nessuna riduzione e 1 rappresenta la riduzione massima dei vertici.</p> <div>  <p>Utilizzando un value di 0,9 o superiore con una serie di dati complessa è possibile ridurre il numero di vertici a un livello in cui la rappresentazione visiva non è accurata.</p> </div> |

**GeoReduceGeometry()** esegue inoltre una funzione simile a **GeoAggrGeometry()** in quanto aggrega un certo numero di aree in un'area. La differenza è che le singole linee dei limiti dei dati di preaggregazione vengono visualizzate sulla mappa se si utilizza **GeoReduceGeometry()**.

Poiché **GeoReduceGeometry()** è una funzione di aggregazione, se viene utilizzata nello script è necessario utilizzare un'istruzione **LOAD** con una clausola **Group by**.

**Esempi:**

In questo esempio viene caricato un file KML con dati di area e quindi una tabella con i dati di area ridotti e aggregati.

```
[MapSource]:
LOAD [world.Name],
    [world.Point],
    [world.Area]
FROM [lib://Downloads/world.kml]
(kml, Table is [world.shp/Features]);

Map:
LOAD world.Name,
    GeoReduceGeometry(world.Area,0.5) as [ReducedArea]
resident MapSource Group By world.Name;

Drop Table MapSource;
```

## 8.15 Funzioni di interpretazione

Le funzioni di interpretazione valutano i contenuti dei campi o delle espressioni di testo di input e impongono il formato dati specificato al valore numerico risultante. Queste funzioni consentono di specificare il formato del numero, in conformità con il relativo tipo di dati, includendo attributi come i separatori decimali, i separatori delle migliaia e il formato dati.

Tutte le funzioni di interpretazione restituiscono un valore duale che riporta sia la stringa che il valore numerico, ma possono essere interpretate come una conversione da stringa a numero. Queste funzioni generano un numero che rappresenta la stringa a partire dal valore di testo dell'espressione di input.

Le funzioni di formattazione invece si comportano nel modo opposto: le espressioni numeriche vengono valutate come stringhe specificando il formato di visualizzazione del testo risultante.

Se non vengono utilizzate le funzioni di interpretazione, Qlik Sense interpreta i dati come un insieme di numeri, date, ore, indicatori temporali e stringhe, utilizzando le impostazioni predefinite per il formato numerico, il formato della data e il formato dell'ora definite dalle variabili dello script e dal sistema operativo.

Tutte le funzioni di interpretazione possono essere utilizzate sia negli script di caricamento dei dati che nelle espressioni grafiche.



*Tutte le rappresentazioni numeriche vengono fornite con un punto decimale come separatore decimale.*

### Prospetto delle funzioni di interpretazione

Ciascuna funzione viene descritta ulteriormente dopo la panoramica. È inoltre possibile fare clic sul nome della funzione nella sintassi per accedere immediatamente ai dettagli per tale funzione specifica.

#### Date#

**Date#** valuta un'espressione come data nel formato specificato nel secondo argomento, se fornito. Se si omette il codice del formato, verrà utilizzato il formato della data predefinito impostato nel sistema operativo.

```
Date#(text[, format])
```

#### Interval#

**Interval#()** valuta un'espressione di testo come intervallo di tempo nel formato impostato nel sistema operativo per impostazione predefinita, oppure nel formato specificato nel secondo argomento, se disponibile.

```
Interval#(text[, format])
```

### Money#

**Money#()** converte una stringa di testo in un valore di valuta nel formato impostato nello script di caricamento o nel sistema operativo, a meno che non venga fornita una stringa di formattazione. I simboli dei separatori decimali e delle migliaia personalizzati sono parametri opzionali.

```
Money#(text[, format[, dec_sep[, thou_sep ] ] ])
```

### Num#

**Num#()** interpreta una stringa di testo come un valore numerico, ovvero converte la stringa di input in un numero usando il formato specificato nel secondo parametro. Se il secondo parametro viene omissso, utilizza i separatori decimali e delle migliaia impostati nello script di caricamento dei dati. I simboli dei separatori decimali e delle migliaia personalizzati sono parametri opzionali.

```
Num#(text[, format[, dec_sep[, thou_sep]]])
```

### Text

**Text()** obbliga a considerare l'espressione come testo, anche nel caso in cui sia possibile un'interpretazione numerica.

```
Text(expr)
```

### Time#

**Time#()** valuta un'espressione come valore ora nel formato dell'ora impostato nello script di caricamento dei dati o nel sistema operativo, a meno che non venga fornita una stringa di formattazione..

```
Time#(text[, format])
```

### Timestamp#

**Timestamp#()** valuta un'espressione come valore data e ora nel formato dell'indicatore temporale impostato nello script di caricamento dei dati o nel sistema operativo, a meno che non venga fornita una stringa di formattazione.

```
Timestamp#(text[, format])
```

---

### Vedere anche:

p [Funzioni di formattazione \(page 1375\)](#)

### Date#

**Date#** valuta un'espressione come data nel formato specificato nel secondo argomento, se fornito.

### Sintassi:

```
Date#(text[, format])
```

Tipo di dati restituiti: duale

| Argomenti     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>text</b>   | La stringa di testo da valutare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>format</b> | <p>Stringa che descrive il formato da utilizzare per interpretare la stringa di testo. Se omessa, verrà utilizzato il formato della data impostato nelle variabili di sistema nello script di caricamento dei dati o nel sistema operativo.</p> <p>La funzione <code>Date#</code> supporta molte opzioni di formato per diverse strutture della data, ad esempio:</p> <p><code>DD/MM/YYYY</code>: Giorno/Mese/Anno (ad esempio, 15/02/2022)</p> <p><code>MM-DD-YYYY</code>: Mese-Giorno-Anno separati da trattini (ad esempio, 02-15-2022).</p> <p><code>YYYY-MM-DD</code>: Anno-Mese-Giorno in formato ISO (per esempio, 2022-02-15)</p> <p><code>D-MMM-YYYY</code>: Giorno, nome breve del Mese e Anno separati da trattini (ad esempio, 15-Feb-2022)</p> <p><code>MMMM YYYY</code>: nome completo del Mese e Anno (ad esempio, February 2022)</p> <p><code>MM/YYYY</code>: Mese e Anno (ad esempio, 02/2022)</p> <p><code>DD MMMM YYYY</code>: Giorno, nome completo del Mese e Anno (ad esempio, 15 February 2022)</p> |

### Esempio

`Date#('8/7/97')`

`Date#('21.10.2023',  
'DD.MM.YYYY' )`

`Date#('21.10.2023')`

Esempio: espressioni dei grafici

### Risultato

Restituisce una data valida per 8/7/97 quando l'impostazione `DateFormat` è `M/D/YY`.

Restituisce una data valida per 21.10.2023 quando l'impostazione è `DD.MM.YYYY`.

Restituisce `NULL` quando l'impostazione `DateFormat` è `M/D/YY`.

## Esempio: principi fondamentali della funzione `Date#`

Script di caricamento

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Il seguente campo in una tabella dati:
  - `StringDate`
- La variabile `DateFormat`, che utilizza il formato `M/D/YY`.
- Un caricamento precedente che crea un nuovo campo denominato `Date`. Le funzioni `Date#` e `Num` vengono utilizzate per convertire la stringa in una data e quindi restituire il valore come numero.

### Script di caricamento

```
SET DateFormat='M/D/YY';
Example:
Load *,
Num(Date#(StringDate, 'D/M/YYYY')) as Date
;
LOAD * INLINE [
StringDate
13/12/2024
31/1/23
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `StringDate`
- `Date`

Creare le seguenti misure:

- `=Date(Date)`, per visualizzare il valore come una data formattata secondo la variabile del formato data fornito nello script di caricamento.
- `=IsNum(StringDate)`, per verificare che il valore sia una stringa e non sia valutato come una data, perché il formato della data dei valori di campo non corrisponde all'impostazione del sistema.

Results table

| StringDate | Date  | Date(Date) | IsNum(StringDate) |
|------------|-------|------------|-------------------|
| 13/12/2024 | 45639 | 12/13/24   | 0                 |
| 31/1/23    | 44957 | 1/31/23    | 0                 |

I risultati mostrano che il campo `Date` nella colonna 2, creato dal caricamento precedente, ha convertito i valori `StringDate` in una data espressa come numero. La funzione di formato `Date` nella colonna 3 riconosce i valori di `Date` come date e li formatta in base alla variabile `DateFormat` nello script di caricamento.

### Esempio: scenario per l'applicazione della funzione Date

Script di caricamento

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata Example.
- I seguenti campi nella tabella dati:
  - Customer
  - BirthDate
- Un caricamento precedente che crea un nuovo campo denominato ParsedBirthDate.

#### Script di caricamento

```
Example:
Load *,
Date#(BirthDate, 'MM.DD.YYYY') as ParsedBirthDate;
Load * inline [
Customer, BirthDate
Customer A, 08.25.2022
Customer B, 08.25.2022
Customer C, 08.25.2022
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- Customer
- BirthDate
- ParsedBirthDate

Creare la seguente misura:

- =Date(ParsedBirthDate, 'DD/MM/YYYY'), per visualizzare il valore come data formattata.

Results table

| Customer   | BirthDate  | ParsedBirthDate | Date(ParsedBirthDate, 'DD/MM/YYYY') |
|------------|------------|-----------------|-------------------------------------|
| Customer A | 08.25.2022 | 08.25.2022      | 25/08/2022                          |
| Customer B | 06.29.2022 | 06.29.2022      | 29/06/2022                          |
| Customer C | 05.31.2022 | 05.31.2022      | 31/05/2022                          |

In questo esempio possiamo vedere che nella colonna 2, il campo `BirthDate` viene trattato come testo dal sistema. La funzione `Date#`, utilizzata nel caricamento precedente, converte `BirthDate` in una data riconoscibile nella colonna 3, `ParsedBirthDate`. La funzione di formato `Date` nella colonna 4 può quindi visualizzare i valori di `ParsedBirthDate` nel formato `DD/MM/YYYY`.

### Interval#

**Interval#()** valuta un'espressione di testo come intervallo di tempo nel formato impostato nel sistema operativo per impostazione predefinita, oppure nel formato specificato nel secondo argomento, se disponibile.

Sintassi:

```
Interval#(text[, format])
```

Tipo di dati restituiti: duale

#### Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>text</b>   | La stringa di testo da valutare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>format</b> | <p>Stringa che descrive il formato di input previsto da utilizzare per la conversione della stringa in un intervallo numerico.</p> <p>Se viene omessa, si utilizzerà il formato dell'ora e il separatore decimale impostati nel sistema operativo.</p> <p>La funzione <code>Interval#</code> supporta molte opzioni di formato per diverse strutture per l'ora, ad esempio:</p> <p><code>hh:mm</code>: solo ore e minuti</p> <p><code>hh:mm:ss</code>: ore, minuti e secondi</p> <p><code>d hh:mm</code>: giorni, ore e minuti</p> |

La funzione **Interval#** converte un intervallo di tempo in formato testuale in un equivalente numerico.

#### Esempio: espressioni dei grafici

| Esempio                                    | Risultato                                                                                            |
|--------------------------------------------|------------------------------------------------------------------------------------------------------|
| <code>Interval#( '12:30', 'hh:mm' )</code> | Restituisce il valore numerico 0.5208, che è 12.5 ore quando è convertito in una frazione di giorno. |
| <code>Interval#( '1 05', 'd hh' )</code>   | Restituisce una rappresentazione numerica equivalente a 1.2083 giorni.                               |

### Esempio: principi fondamentali della funzione Interval

Script di caricamento

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Il seguente campo in una tabella dati:
  - `IntervalValue`
- Un caricamento precedente che utilizza la funzione `Interval#` per convertire il testo del campo `IntervalValue` in un numero, che può essere riconosciuto come orario.

#### Script di caricamento

Example:

```
Load *, Interval#(IntervalValue, 'D hh') as FormattedIntervalValue;
Load * inline [
IntervalValue
1 05
2 15
1 09
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `IntervalValue`

Creare le seguenti misure:

- `=Num(FormattedIntervalValue)`, per visualizzare il valore come numero dopo che è stato elaborato nello script di caricamento utilizzando la funzione `Interval#`.
- `=Interval(FormattedIntervalValue, 'dd:hh:mm')`, per visualizzare il valore come un formato intervallo che utilizza il formato fornito nella funzione.

Tabella dei risultati

| IntervalValue | Num(FormattedIntervalValue) | Interval(FormattedIntervalValue, 'dd:hh:mm') |
|---------------|-----------------------------|----------------------------------------------|
| 1 05          | 1.20833333333333            | 01:05:00                                     |
| 1 09          | 1.375                       | 01:09:00                                     |
| 2 15          | 2.625                       | 02:15:00                                     |

I risultati dimostrano come la funzione `Interval#` converte i valori in un numero, che può poi essere visualizzato utilizzando un parametro di formato specificato. Per esempio, nella colonna `IntervalValue`, il valore `2 15` viene convertito in un numero nel caricamento precedente (campo `FormattedIntervalValue`). La seconda colonna utilizza la funzione `Num` per mostrare questo valore convertito come numero, `2.625`. La terza colonna utilizza la funzione `Interval` per formattare questo numero in giorni, ore e minuti, visualizzandolo come `02:15:00`, o `2 giorni 15 ore`.

### Esempio: scenario per l'applicazione della funzione `Interval`

#### Script di caricamento

##### Panoramica

Un set di dati di ticket di assistenza contiene un campo che mescola dati testuali e temporali, ad esempio `2 days 04 hours`. Bisogna convertire questi valori in valori temporali normali.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `TicketID`
  - `ResolutionTime`
- Un caricamento precedente che utilizza la funzione `Interval#` per convertire il testo nel campo `ResolutionTime` in un numero, che può essere riconosciuto come giorni e ore.

#### Script di caricamento

Example:

```
Load *, Interval#(ResolutionTime, 'd "days" hh "hours"') as FormattedResolutionTime;
Load * inline [
TicketID, ResolutionTime
Ticket A, 2 days 04 hours
Ticket B, 2 days 06 hours
Ticket C, 3 days 10 hours
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `TicketID`
- `ResolutionTime`

Creare le seguenti misure:

- `=Interval(FormattedResolutionTime, 'dd:hh')`, per convertire i valori nel formato desiderato con un valore doppio.

- `=Num(FormattedResolutionTime)`, per mostrare i valori numerici del campo.

Tabella dei risultati

| TicketID | ResolutionTime  | Interval<br>(FormattedResolutionTime, 'dd<br>hh') | Num<br>(FormattedResolutionTime) |
|----------|-----------------|---------------------------------------------------|----------------------------------|
| Ticket A | 2 giorni 04 ore | 02 04                                             | 2.16666666666667                 |
| Ticket B | 2 giorni 06 ore | 02 06                                             | 2.25                             |
| Ticket C | 3 giorni 10 ore | 03 10                                             | 3.41666666666667                 |

I risultati mostrano come è possibile utilizzare la funzione `Interval#` per convertire i valori in un numero, e quindi visualizzarli in un formato orario più tipico. Ad esempio, `Ticket B` ha un tempo di risoluzione di 2 days 06 hours. Il valore viene convertito in un numero nel caricamento precedente (campo `FormattedResolutionTime`). Il nuovo valore viene visualizzato come 02 06 (2 giorni e 6 ore) quando viene formattato utilizzando la funzione `Interval` nella colonna 3. Nella colonna 4, la funzione `Num` visualizza il proprio valore numerico come 2.25.

### Money#

**Money#()** converte una stringa di testo in un valore di valuta nel formato impostato nello script di caricamento o nel sistema operativo, a meno che non venga fornita una stringa di formattazione. I simboli dei separatori decimali e delle migliaia personalizzati sono parametri opzionali.

Sintassi:

```
Money#(text[, format[, dec_sep [, thou_sep ] ] ])
```

Tipo di dati restituiti: duale

Argomenti

| Argomento   | Descrizione                      |
|-------------|----------------------------------|
| <b>text</b> | La stringa di testo da valutare. |

| Argomento       | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>format</b>   | <p>Stringa che descrive il formato di input previsto da utilizzare per la conversione della stringa in un intervallo numerico.</p> <p>Se viene omessa, verrà utilizzato il valore impostato nello script di caricamento per MoneyFormat.</p> <p>La funzione Money# supporta molte opzioni di formato per diverse strutture per la valuta, ad esempio:</p> <p>\$#,##0.00: aggiunge il simbolo del dollaro e due cifre decimali.</p> <p>€ #,##0: aggiunge il simbolo dell'euro e nessun decimale, con uno spazio dopo il simbolo dell'euro.</p> <p>¥#,##0: aggiunge il simbolo dello yen giapponese senza cifre decimali.</p> <p>\$#,##0; (\$#,##0): formatta i valori negativi tra parentesi.</p> <p>F #.##0,00: aggiunge il simbolo del franco svizzero con punti come separatori delle migliaia e una virgola per i decimali (ad esempio, F 1.000,00).</p> <p>R #,##0.00: aggiunge il simbolo del rand sudafricano con due cifre decimali e uno spazio dopo il simbolo del rand.</p> <p>\$ #,##0.000: aggiunge tre cifre decimali al formato del dollaro.</p> |
| <b>dec_sep</b>  | <p>Stringa che specifica il separatore decimale dei numeri. Se viene omessa, verrà utilizzato il valore MoneyDecimalSep impostato nello script di caricamento dei dati.</p> <p>La funzione Money# supporta molti formati per il separatore decimale, ad esempio:</p> <p>Virgola ', ': ad esempio, \$1,000,00 per mille dollari.</p> <p>Punto '. ': ad esempio, € 1,000.00.</p> <p>Trattino '- ': se utilizzato al posto del punto o della virgola, utilizzati di norma per il separatore decimale, mostra valori come ¥1,000-00.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>thou_sep</b> | <p>Stringa che specifica il separatore delle migliaia dei numeri. Se viene omessa, verrà utilizzato il valore MoneyThousandSep impostato nello script di caricamento dei dati.</p> <p>La funzione Money# supporta molte opzioni per il separatore delle migliaia:</p> <p>Punto ' . ': utilizza un punto per il separatore di migliaia (comune in alcuni formati europei), visualizzandolo come \$1.000,00 se combinato con una virgola come separatore decimale.</p> <p>Spazio ' _ ': utilizza uno spazio per il separatore delle migliaia, che è comune in alcuni Paesi europei. Viene visualizzato come € 1 000.00.</p> <p>Trattino basso ' _ ': utilizza un trattino basso per il separatore delle migliaia, che a volte viene utilizzato in formati tecnici o regionali specifici. Visualizza valori come \$1_000.00.</p>                                                                                                                                                                                                                                  |

In genere, la funzione `Money#` si comporta in modo analogo alla funzione `Num#`, ma acquisisce i propri valori predefiniti per il separatore decimale e delle migliaia dalle variabili dello script per il formato valuta oppure dalle impostazioni di valuta del sistema.

Esempio: espressioni dei grafici

| Esempio                                                      | Risultati                                                                                                                                                                                                                                                                                                                                       |
|--------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>Money#('35 648,37 kr', '# ##0,00 kr', ',', ' ')</code> | Restituisce 35 648,37 kr quando l'impostazione <code>MoneyFormat</code> è # ##0,00 kr. Per una valutazione corretta come numero, è necessario impostare i seguenti valori nello script di caricamento: <ul style="list-style-type: none"><li>• <code>SET MoneyDecimalSep=',';</code></li><li>• <code>SET MoneyThousandSep=' ';</code></li></ul> |
| <code>Money#('\$35,648.37', '\$#', '.', ' ', ' ')</code>     | Restituisce \$35,648.37 quando sono impostati i seguenti valori: <ul style="list-style-type: none"><li>• <code>MoneyFormat</code> è \$#</li><li>• <code>SET MoneyDecimalSep=',';</code></li><li>• <code>SET MoneyThousandSep=' ';</code></li></ul>                                                                                              |

### Esempio: principi fondamentali della funzione Money#

Script di caricamento

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Un campo della tabella dati denominata `MoneyValue`.
- Un caricamento precedente che utilizza la funzione `Money#` per convertire i valori di testo nei dati in un valore che Qlik Sense riconoscerà come valore valido.

#### Script di caricamento

```
SET MoneyFormat='$ #,##0.00';
SET MoneyThousandSep=' ';
Example:
Load *, Money#(MoneyValue, '# ##0,00 kr', ',', ' ') as FormattedMoneyValue;
Load * inline [
MoneyValue
35 648,37 kr
56 735,56 kr
] (delimiter is ';');
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- MoneyValue
- FormattedMoneyValue

Creare le seguenti misure:

- =Money( FormattedMoneyValue, 'kr #,##0', '.', ',' ), per visualizzare il valore in valuta, formattato utilizzando gli argomenti impostati nella funzione Money.
- =Num(FormattedMoneyValue), per restituire il valore numerico dopo che è stato elaborato dalla funzione Money#.

| Results table |                     |                                                         |                              |  |
|---------------|---------------------|---------------------------------------------------------|------------------------------|--|
| MoneyValue    | FormattedMoneyValue | Money(<br>FormattedMoneyValue,<br>'kr #,##0', '.', ',') | Num<br>(FormattedMoneyValue) |  |
| 56 735,56 kr  | 56 735,56 kr        | kr 56,736                                               | 56735.56                     |  |
| 35 648,37 kr  | 35 648,37 kr        | kr 35,648                                               | 35648.37                     |  |

La tabella dei risultati mostra come i valori possono essere visualizzati utilizzando le funzioni e il formato seguenti: Money#Money e Num. La seconda colonna mostra la dimensione FormattedMoneyValue, che è stata calcolata utilizzando la funzione Money# per cambiare il valore in valuta con il formato '# ##0,00 kr', ',', '.'. La terza colonna utilizza la funzione Money per cambiare il formato della valuta da FormattedMoneyValue a 'kr #,##0', '.', ','. La quarta colonna utilizza la funzione Num per mostrare il valore numerico della dimensione FormattedMoneyValue utilizzando il formato nella variabile di sistema.

### Esempio: scenario per la funzione Money#

Script di caricamento

#### Panoramica

Un set di dati di ordini di vendita contiene campi che utilizzano valute diverse. Questo esempio mostra come separare e sommare i valori delle vendite in base alla valuta.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata Example.
- I seguenti campi nella tabella dati:
  - TransactionID
  - Currency
  - Amount
- Le variabili di sistema MoneyDecimalSep, MoneyFormat e MoneyThousandSep sono impostate per utilizzare un formato diverso da quello utilizzato nei dati a scopo dimostrativo.

- Un caricamento precedente che carica i campi TransactionID, Currency e Amount, e crea il seguente campo:
  - AmountValue, questo campo utilizza la funzione Money# per convertire il testo ricevuto in un valore comprensibile dal sistema.

### Script di caricamento

```
SET MoneyDecimalSep='-';
SET MoneyFormat='###0-00';
SET MoneyThousandSep='';
Example:
LOAD
TransactionID,
Currency,
Amount,
IF(Currency = 'USD', Money#(Amount, '$#,##0.00','.',',',''),
IF(Currency = 'EUR', Money#(Amount, '€ #.##0,00','.',',','.'))) AS AmountValue
INLINE [
TransactionID, Currency, Amount
"TXN001", "USD", "$ 1,500.00"
"TXN002", "EUR", "€ 1.200,75"
"TXN003", "USD", "$ 1,750.50"
"TXN004", "EUR", "€ 950,00"
"TXN005", "USD", "$ 2,600.25"
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- TransactionID
- Currency
- Amount
- AmountValue

Creare la seguente misura:

- Num(AmountValue)

Tabella dei risultati

| TransactionID | Currency | Amount      | AmountValue | Num(AmountValue) |
|---------------|----------|-------------|-------------|------------------|
| TXN001        | USD      | \$1,500.00  | \$ 1,500.00 | 1500             |
| TXN002        | EUR      | € 1.200,75  | € 1.200,75  | 1200.75          |
| TXN003        | USD      | \$ 1,750.50 | \$ 1,750.50 | 1750.5           |
| TXN004        | EUR      | € 950,00    | € 950,00    | 950              |
| TXN005        | USD      | \$ 2,600.25 | \$ 2,600.25 | 2600.25          |

I risultati mostrano come la funzione `Money#` converte i valori testuali in valori numerici per la valuta che possono poi essere analizzati. Ad esempio, i valori del campo `Amount` sono testuali e appaiono allineati a sinistra. Tuttavia, una volta elaborati dalla funzione `Money#` per creare il nuovo campo, `AmountValue`, il sistema li riconosce come valori numerici monetari. La funzione `Num`, utilizzata nella colonna più a destra, può ora valutare i valori monetari in `AmountValue` e cambiare il formato da denaro a numero.

### Num#

**Num#()** interpreta una stringa di testo come un valore numerico, ovvero converte la stringa di input in un numero usando il formato specificato nel secondo parametro. Se il secondo parametro viene omissso, utilizza i separatori decimali e delle migliaia impostati nello script di caricamento dei dati. I simboli dei separatori decimali e delle migliaia personalizzati sono parametri opzionali.

#### Sintassi:

```
Num# (text[, format[, dec_sep [, thou_sep ] ] ])
```

**Tipo di dati restituiti:** duale

La funzione **Num#()** restituisce un valore duale contenente sia la stringa che il valore numerico. La funzione prende la rappresentazione testuale dell'espressione di input e genera una stringa che rappresenta il numero. Non modifica il formato del numero; l'output viene formattato nello stesso modo dell'input.

#### Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>text</b>   | La stringa di testo da valutare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>format</b> | <p>Stringa che specifica il formato numerico utilizzato nel primo parametro. Se omissi, verranno utilizzati i separatori di decimali e migliaia impostati nello script di caricamento dei dati.</p> <p>La funzione <code>Num#</code> supporta molte opzioni di formato per diverse strutture numeriche, ad esempio:</p> <p><code>#,##0</code>: visualizza un numero senza cifre decimali.</p> <p><code>#,##0.00</code>: visualizza un numero con due cifre decimali.</p> <p><code>##0.00%</code>: visualizza un numero come una percentuale con due cifre decimali.</p> <p><code>#,##0;(#,##0)</code>: formatta i numeri negativi in modo da essere visualizzati tra parentesi.</p> <p><code>\$#,##0.00</code>: formatta i numeri come valuta con il simbolo \$.</p> <p><code>0.00E+0</code>: visualizza i numeri in una notazione scientifica.</p> |

| Argomento       | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>dec_sep</b>  | <p>Stringa che specifica il separatore decimale dei numeri. Se omesso, viene utilizzato il valore della variabile <code>DecimalSep</code> impostato nello script di caricamento dei dati. La funzione <code>Num#</code> supporta molti formati per il separatore decimale, ad esempio:</p> <p>Virgola ', ': ad esempio, \$1,000,00 per mille dollari.</p> <p>Punto '.': ad esempio, € 1,000.00.</p> <p>Trattino '-': se utilizzato al posto del punto o della virgola, utilizzati di norma per il separatore decimale, mostra valori come ¥1,000-00.</p> <p>Per ulteriori informazioni, vedere <a href="#">DecimalSep (page 239)</a>.</p>                                                                                                                                                                                                                                                                                           |
| <b>thou_sep</b> | <p>Stringa che specifica il separatore delle migliaia dei numeri. Se omesso, viene utilizzato il valore della variabile <code>ThousandSep</code> impostato nello script di caricamento dei dati. La funzione <code>Num#</code> supporta molte opzioni per il separatore delle migliaia:</p> <p>Punto ' . ': utilizza un punto per il separatore di migliaia (comune in alcuni formati europei), visualizzandolo come \$1.000,00 se combinato con una virgola come separatore decimale.</p> <p>Spazio ' ': utilizza uno spazio per il separatore delle migliaia, che è comune in alcuni Paesi europei. Viene visualizzato come € 1 000.00.</p> <p>Trattino basso ' _ ': utilizza un trattino basso per il separatore delle migliaia, che a volte viene utilizzato in formati tecnici o regionali specifici. Visualizza valori come \$1_000.00.</p> <p>Per ulteriori informazioni, vedere <a href="#">ThousandSep (page 277)</a>.</p> |

Esempio: espressioni dei grafici

### Esempio

### Risultati (valore stringa restituito)

|                                                     |                                                                    |
|-----------------------------------------------------|--------------------------------------------------------------------|
| <code>Num#( '35,648.31', '#', '.', ',') )</code>    | Restituisce la stringa 35,648.31 e il valore numerico 35648.31.    |
| <code>Num#( '35 648.312', '#', '.', ',') )</code>   | Restituisce la stringa 35 648.312 e il valore numerico 35648.312.  |
| <code>Num#( '35.648,3123', '#', '.', ',') )</code>  | Restituisce la stringa 35.648,3123 e il valore numerico NULL (-).  |
| <code>Num#( '35 648,31234', '#', '.', ',') )</code> | Restituisce la stringa 35 648,31234 e il valore numerico NULL (-). |
| <code>Num#('1,234.56', '#,##0.00')</code>           | Restituisce la stringa 1,234.56 e il valore numerico 1230.45.      |
| <code>Num#('\$1,230.45', '\$#,##0.00')</code>       | Restituisce la stringa \$1,230.45 e il valore numerico 1234.56.    |
| <code>Num#('20220315', 'YYYYMMDD')</code>           | Restituisce la stringa 20220315 e il valore numerico 20220315.     |

### Esempio: principi fondamentali della funzione Num#

Script di caricamento

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `ProductID`
  - `ProductName`
  - `SalesAmount`
- Le impostazioni standard per `ThousandSep` e `DecimalSep` illustrano questo esempio.
- Un caricamento precedente che utilizza la funzione `Num#`. Quando viene eseguito, crea un campo aggiuntivo denominato `FormattedSalesAmount`.

#### Script di caricamento

```
SET ThousandSep=',';
SET DecimalSep='.';
Example:
LOAD
ProductID,
ProductName,
SalesAmount,
Num#(SalesAmount, '#.##0,00',' ','.') AS FormattedSalesAmount
INLINE [
ProductID, ProductName, SalesAmount
1, Product A, '1.234,56'
2, Product B, '2.456,78'
3, Product C, '345,67'
4, Product D, '987,12'
5, Product E, '1.200,45'
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `ProductID`
- `ProductName`

- SalesAmount
- FormattedSalesAmount

Creare la seguente misura:

- =Num(FormattedSalesAmount), per visualizzare il valore numerico del campo in base alle impostazioni di sistema per il formato numerico.

Results table

| ProductID | ProductName | SalesAmount | FormattedSalesAmount | Num<br>(FormattedSalesAmount) |
|-----------|-------------|-------------|----------------------|-------------------------------|
| 1         | Product A   | 1.234,56    | 1.234,56             | 1234.56                       |
| 2         | Product B   | 2.456,78    | 2.456,78             | 2456.78                       |
| 3         | Product C   | 345,67      | 345,67               | 345.67                        |
| 4         | Product D   | 987,12      | 987,12               | 987.12                        |
| 5         | Product E   | 1.200,45    | 1.200,45             | 1200.45                       |

In questo esempio è possibile osservare come la funzione Num# converte i dati SalesAmount dal formato testuale originale in un valore numerico.

### Esempio: scenario per la funzione Num#

Script di caricamento

#### Panoramica

Un set di dati contiene i dati sulle vendite di diversi Paesi in diversi formati di testo. Bisogna standardizzare i dati e visualizzarli in una tabella.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata Example.
- I seguenti campi nella tabella dati:
  - Country
  - Product
  - Amount
- Un caricamento precedente che utilizza la funzione Num# per convertire i valori del campo Amount.

#### Script di caricamento

Example:

```
LOAD  
Country,
```

```
Product,
IF(Country = 'Germany', Num#(Amount, '#.##0,00', ',', '.'), Num#(Amount, '#,##0.00')) AS
Amount
INLINE [
Country, Product, Amount
Germany, Product A, '1.234,56'
USA, Product B, '2,345.78'
Germany, Product C, '987,12'
USA, Product D, '1,200.45'
Germany, Product E, '345,67'
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- Product
- Country
- Amount

Creare la seguente misura:

- Num(Amount), per visualizzare il valore numerico del campo Amount come set nelle impostazioni di sistema per il formato numerico.

Results table

| Product    | Country | Amount   | Num(Amount) |
|------------|---------|----------|-------------|
| Customer A | Germany | 1.234,56 | 1234.56     |
| Customer B | USA     | 2,345.78 | 2345.78     |
| Customer C | Germany | 987,12   | 987.12      |
| Customer D | USA     | 1,200.45 | 1200.45     |
| Customer E | Germany | 345,67   | 345.67      |

Confrontando la colonna Amount con la colonna Num(Amount), si nota che i dati presentati utilizzando la funzione Num# modificano i valori numerici sottostanti, ma mantengono il formato della stringa. Ad esempio, tutti i valori per Germany nella colonna Amount utilizzano una virgola come separatore decimale, ma il formato sottostante in Num(Amount) utilizza un punto per tutti i valori.

## Text

**Text()** obbliga a considerare l'espressione come testo, anche nel caso in cui sia possibile un'interpretazione numerica.

### Sintassi:

**Text** (expr)

Tipo di dati restituiti: stringa

### Argomenti

| Argomento         | Descrizione                            |
|-------------------|----------------------------------------|
| <code>expr</code> | Un'espressione da trattare come testo. |

### Esempio: espressioni del grafico

| Esempio                    | Risultati                                                                 |
|----------------------------|---------------------------------------------------------------------------|
| <code>Text( 1234 )</code>  | Restituisce 1234 come una stringa. Non c'è un valore numerico.            |
| <code>Text( pi( ) )</code> | Restituisce 3.1415926535898 come una stringa. Non c'è un valore numerico. |

## Esempio: principi fondamentali della funzione Text

Script di caricamento

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `ID`
  - `PostalCode`
- Un caricamento precedente che utilizza la funzione `Text` per creare un nuovo campo denominato `PostalCode_Text` per garantire che i codici postali rimangano come testo e non vengano convertiti in numeri.

### Script di caricamento

Example:

```
LOAD
ID,
PostalCode,
Text(PostalCode) AS PostalCode_Text;
LOAD * INLINE [
ID, PostalCode
1, 12345
2, 54321
3, 67890
4, 11223
5, 33445
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- ID
- PostalCode
- PostalCode\_Text

| Results table |            |                 |
|---------------|------------|-----------------|
| ID            | PostalCode | PostalCode_Text |
| 1             | 12345      | 12345           |
| 2             | 54321      | 54321           |
| 3             | 67890      | 67890           |
| 4             | 11223      | 11223           |
| 5             | 33445      | 33445           |

L'output del caricamento precedente utilizza la funzione `Text` per garantire che qualsiasi valore numerico sia trattato come testo. È possibile vedere che i valori della colonna `PostalCode_Text` sono elaborati come testo e appaiono allineati a sinistra per impostazione predefinita, mentre i valori della colonna `PostalCode` sono numeri e appaiono allineati a destra per impostazione predefinita.

### Esempio: scenario per la funzione Text

Script di caricamento

#### Panoramica

Un set di dati contiene i dati relativi alle fatture dei clienti che sono stati inseriti come valori numerici, ma si desidera conservare questi valori come testo.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `InvoiceNumber`
  - `CustomerName`
- Un caricamento precedente che utilizza la funzione `Text` per creare un nuovo campo denominato `InvoiceNumber_Text` e garantire che i numeri delle fatture vengano trattati come testo.

### Script di caricamento

Example:

```
LOAD
InvoiceNumber,
CustomerName,
Text(InvoiceNumber) AS InvoiceNumber_Text;
LOAD * INLINE [
InvoiceNumber, CustomerName
10001, John Doe
10002, Jane Smith
10003, Bob Johnson
10004, Alice Brown
10005, Michael Green
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- InvoiceNumber
- CustomerName
- InvoiceNumber\_Text

| Results table |               |                    |
|---------------|---------------|--------------------|
| InvoiceNumber | CustomerName  | InvoiceNumber_Text |
| 10001         | John Doe      | 10001              |
| 10002         | Jane Smith    | 10002              |
| 10003         | Bob Johnson   | 10003              |
| 10004         | Alice Brown   | 10004              |
| 10005         | Michael Green | 10005              |

I risultati mostrano che ora i valori numerici nella colonna `InvoiceNumber` sono stati convertiti in valori di testo nella colonna `InvoiceNumber_Text` e appaiono allineati a sinistra.

## Time#

**Time#()** valuta un'espressione come valore ora nel formato dell'ora impostato nello script di caricamento dei dati o nel sistema operativo, a meno che non venga fornita una stringa di formattazione.

Sintassi:

```
time#(text[, format])
```

Tipo di dati restituiti: duale

#### Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>text</b>   | La stringa di testo da valutare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>format</b> | <p>Stringa che descrive il formato da utilizzare per interpretare la stringa di testo. Se viene omessa, si utilizza il formato dell'ora e il separatore decimale impostati nel sistema operativo.</p> <p>La funzione <code>Time#</code> supporta molte opzioni di formato per diverse strutture per l'ora, ad esempio:</p> <p><code>hh:mm:ss</code>: formatta un valore orario in un formato orario standard e visualizza ore, minuti e secondi con zeri iniziali.</p> <p><code>hh:mm TT</code>: utilizza un formato di 12 ore con il suffisso AM/PM.</p> <p><code>hh:mm</code>: mostra solo le ore e i minuti, omettendo i secondi.</p> <p><code>hh.mm.ss</code>: formato orario con separatori diversi, come ad esempio un punto (.) anziché i due punti.</p> <p><code>hh TT</code>: include solo l'ora e AM/PM.</p> <p><code>hh "Hrs" mm "Mins"</code>: mostra l'ora con un testo personalizzato, ad esempio Hrs.</p> |

Esempio: espressioni dei grafici

#### Esempio

#### Risultato

`Time#('09:00:00')`

Restituisce 09:00:00 quando l'impostazione `TimeFormat` è 'hh:mm:ss'.

`Time#('09:00:00', 'hh:mm:ss')`

Restituisce 09:00:00 quando il formato fornito è 'hh:mm:ss'.

`Time#('03:25 PM', 'hh:mm TT')`

Restituisce 03:25 PM quando il formato fornito è 'hh:mm TT'.

## Esempio: principi fondamentali della funzione Time#

Script di caricamento

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Il seguente campo in una tabella dati:
  - `InputTime`
- La variabile `TimeFormat`, che utilizza il formato `h:mm:ss TT`.
- Un caricamento precedente che utilizza la funzione `Time#` per convertire il valore testuale `InputTime` in un campo orario/numerico denominato `ParsedTime` che il sistema può utilizzare.

### Script di caricamento

```
SET TimeFormat='h:mm:ss TT';
Example:
Load
InputTime,
Time#(InputTime, 'hh-mm-ss') as ParsedTime
Inline [
InputTime
12-30-45
05-15-00
22-10-55
08-20-35
19-45-15
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `InputTime`
- `ParsedTime`

Creare la seguente misura:

- `=Time(ParsedTime)`, per visualizzare l'ora utilizzando il formato fornito nello script di caricamento come impostazione predefinita.

Tabella dei risultati

| InputTime | ParsedTime | Time(ParsedTime) |
|-----------|------------|------------------|
| 05-15-00  | 05-15-00   | 5:15:00 AM       |
| 08-20-35  | 08-20-35   | 8:20:35 AM       |
| 12-30-45  | 12-30-45   | 12:30:45 PM      |
| 19-45-15  | 19-45-15   | 7:45:15 PM       |
| 22-10-55  | 22-10-55   | 10:10:55 PM      |

I valori del campo `InputTime` vengono visualizzati come testo e sono allineati a sinistra. Una volta che i valori in `InputTime` vengono analizzati attraverso la funzione `Time#`, viene creato il campo `ParsedTime`

risultante. I valori nel nuovo campo sono allineati a destra, in quanto ora vengono trattati come un valore temporale valido. La terza colonna utilizza la funzione `Time` per valutare i valori di `ParsedTime` e li formatta in base al valore `TimeFormat` nello script di caricamento.

### Esempio: scenario per l'applicazione della funzione `Time#`

Script di caricamento

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Il seguente campo in una tabella dati:
  - `NoDelimiterTime`
- La variabile `TimeFormat`, che utilizza il formato `hh.mm.ss`.
- Un caricamento precedente che utilizza la funzione `Time#` per convertire il valore testuale `NoDelimiterTime` in un campo orario/numerico denominato `ParsedNoDelimiterTime`.

#### Script di caricamento

```
SET TimeFormat='hh.mm.ss';
Example:
Load
NoDelimiterTime,
Time#(NoDelimiterTime, 'hhmmss') as ParsedNoDelimiterTime
Inline [
NoDelimiterTime
143015
091045
235959
060530
184010
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `NoDelimiterTime`

Creare la seguente misura:

- `=Time(ParsedNoDelimiterTime, 'hh:mm:ss')`, per visualizzare l'ora utilizzando il formato fornito nello script di caricamento come impostazione predefinita.

Tabella dei risultati

| NoDelimiterTime | Time(ParsedNoDelimiterTime,'hh:mm:ss') |
|-----------------|----------------------------------------|
| 060530          | 06:05:30                               |
| 091045          | 09:10:45                               |
| 143015          | 14:30:15                               |
| 184010          | 18:40:10                               |
| 235959          | 23:59:59                               |

I risultati mostrano come è possibile utilizzare la funzione `Time#` per analizzare e convertire i valori da testo a un valore temporale/numerico.

### Timestamp#

**Timestamp#()** valuta un'espressione come valore data e ora nel formato dell'indicatore temporale impostato nello script di caricamento dei dati o nel sistema operativo, a meno che non venga fornita una stringa di formattazione.

**Sintassi:**

```
timestamp#(text[, format])
```

**Tipo di dati restituiti:** duale

Argomenti

| Argomento   | Descrizione                      |
|-------------|----------------------------------|
| <b>text</b> | La stringa di testo da valutare. |

| Argomento | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| format    | <p>Stringa che descrive il formato da utilizzare per interpretare la stringa di testo. Se viene omessa, si utilizzerà il formato della data breve, il formato dell'ora e il separatore decimale impostati nel sistema operativo. ISO 8601 è supportato per gli indicatori temporali.</p> <p>La funzione <code>Timestamp#</code> supporta molte opzioni di formato per diverse strutture per l'ora, ad esempio:</p> <p><code>DD/MM/YYYY</code>: si concentra solo sulla data (ad esempio: 15/11/2024).</p> <p><code>YYYY-MM-DD hh:mm:ss</code>: visualizza il timestamp completo (ad esempio: 2024-11-15 13:30:45).</p> <p><code>MMM DD, YYYY</code>: visualizza una versione breve del mese e l'anno completo (ad esempio: Nov 15, 2024).</p> <p><code>hh:mm:ss TT</code>: mostra solo l'ora nel formato di 12 ore (per esempio: 01:30:45 PM)</p> <p><code>HH:mm:ss</code>: visualizza l'ora nel formato a 24 ore (ad esempio: 13:30:45).</p> <p><code>www, MMM DD YYYY</code>: mostra il giorno completo della settimana e la data (ad esempio: Friday, Nov 15 2024).</p> <p><code>YYYY-MM-DD hh:mm:ss.fff</code>: include i millisecondi (ad esempio: 2024-11-15 13:30:45.123).</p> |

Esempio: espressioni dei grafici

| Esempio                                                               | Risultato                       |
|-----------------------------------------------------------------------|---------------------------------|
| <code>Timestamp#('12-03-2023 10:25:30', 'DD-MM-YYYY hh:mm:ss')</code> | Restituisce 12-03-2023 10:25:30 |
| <code>Timestamp#('2023/03/15 13:45', 'YYYY/MM/DD hh:mm')</code>       | Restituisce 2023/03/15 13:45    |

Esempio: principi fondamentali della funzione Timestamp#

Script di caricamento

Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Il seguente campo in una tabella dati:
  - `InputString`

- Un caricamento precedente che utilizza la funzione `Timestamp#` per trattare l'espressione come un valore di data e ora. Quando viene eseguito, crea un campo aggiuntivo denominato `Timestamp_InputStream`.

### Script di caricamento

```
SET DateFormat='M/D/YYYY'  
Example:  
Load *,  
Timestamp(Timestamp#(InputStream)) as Timestamp_InputStream;  
LOAD * INLINE [  
InputStream  
2015-09-15T12:13:14  
1952-10-16T13:14:00+0200  
1109-03-01T14:15  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `InputStream`
- `Timestamp_InputStream`

Tabella dei risultati

| InputStream              | Timestamp_InputStream  |
|--------------------------|------------------------|
| 1109-03-01T14:15         | 3/1/1109 2:15:00 PM    |
| 1952-10-16T13:14:00+0200 | 10/16/1952 11:14:00 AM |
| 2015-09-15T12:13:14      | 9/15/2015 12:13:14 PM  |

In questo esempio, è possibile vedere come i valori delle stringhe possono essere valutati come timestamp utilizzando la funzione `Timestamp#`.

### Esempio: scenario per l'applicazione della funzione `Timestamp#`

#### Script di caricamento

#### Panoramica

I dati di un registro degli eventi contengono un formato di timestamp che si desidera utilizzare.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:

- Event
- EventDateTime
- Un caricamento precedente che utilizza la funzione `Timestamp#` per convertire i valori del campo `EventDateTime` in formato data e ora che il sistema può comprendere. Quando viene eseguito, crea un campo aggiuntivo denominato `EventTimestamp`.

### Script di caricamento

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
Example:
LOAD
Event,
EventDateTime,
Timestamp#(EventDateTime, 'DD-MM-YYYY hh:mm:ss') AS EventTimestamp
INLINE [
Event, EventDateTime
"Login", "12-03-2023 10:25:30"
"Logout", "12-03-2023 12:45:10"
"Login", "13-03-2023 09:10:15"
"Logout", "13-03-2023 17:30:20"
"Login", "14-03-2023 08:00:00"
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- Event
- EventDateTime
- EventTimestamp

Creare la seguente misura:

- `=Timestamp(EventTimestamp)`, per visualizzare il valore timestamp utilizzando il formato fornito nello script di caricamento come impostazione predefinita.

Tabella dei risultati

| Evento      | EventDateTime       | EventTimestamp      | Timestamp(EventTimestamp) |
|-------------|---------------------|---------------------|---------------------------|
| Accedi      | 12-03-2023 10:25:30 | 12-03-2023 10:25:30 | 3/12/2023 10:25:30 AM     |
| Accedi      | 13-03-2023 09:10:15 | 13-03-2023 09:10:15 | 3/13/2023 9:10:15 AM      |
| Accedi      | 14-03-2023 08:00:00 | 14-03-2023 08:00:00 | 3/14/2023 8:00:00 AM      |
| Disconnetti | 12-03-2023 12:45:10 | 12-03-2023 12:45:10 | 3/12/2023 12:45:10 PM     |
| Disconnetti | 13-03-2023 17:30:20 | 13-03-2023 17:30:20 | 3/13/2023 5:30:20 PM      |

In questo esempio, è possibile vedere come i valori possono essere convertiti utilizzando la funzione `Timestamp#`.

## 8.16 Funzioni intra-record

Le funzioni intra-record sono utilizzate:

- Nello script di caricamento dei dati quando è necessario il valore di un record di dati precedentemente caricato per valutare il record attuale.
- Nelle espressioni grafiche quando è necessario un altro valore della serie di dati di una visualizzazione.



*L'ordinamento sui valori y nei grafici o l'ordinamento per colonne di espressione nelle tabelle non è consentito quando si utilizza una funzione di grafico intra-record in una qualsiasi delle espressioni del grafico. Queste alternative di ordinamento verranno perciò disattivate automaticamente. Quando si utilizza una funzione di grafico intra-record in una visualizzazione o in una tabella, l'ordinamento della visualizzazione torna all'ordinamento inserito nella funzione intra-record. Questa limitazione non si applica alla funzione di script equivalente, se esiste.*



*Le definizioni di espressioni autoreferenziali possono essere create in modo affidabile solo in tabelle con meno di 100 righe, anche se ciò può variare in base all'hardware su cui Qlik Engine è in esecuzione.*

## Funzioni di riga

Queste funzioni possono essere utilizzate solo nelle espressioni grafiche.

### Above

**Above()** valuta un'espressione in una riga sopra la riga attuale all'interno di un segmento di colonna in una tabella. La riga per la quale viene calcolata dipende dal valore dell'**offset**, se presente, e l'impostazione predefinita è la riga direttamente sopra. Per i grafici diversi dalle tabelle, **Above()** restituisce un valore per la riga sopra la riga attuale nell'equivalente di tabella lineare del grafico.

**Above - funzione per grafici** ([TOTAL [<fld{,fld}>]] expr [ , offset [,count]])

### Below

**Below()** valuta un'espressione in una riga sotto la riga attuale all'interno di un segmento di colonna in una tabella. La riga per la quale viene calcolata dipende dall'**offset**, se presente, e l'impostazione predefinita è la riga direttamente sotto. Per i grafici diversi dalle tabelle, **Below()** restituisce un valore per la riga sotto la colonna attuale nell'equivalente di tabella lineare del grafico.

**Below - funzione per grafici** ([TOTAL [<fld{,fld}>]] expression [ , offset [,count ]])

### Bottom

**Bottom()** valuta un'espressione nell'ultima riga (inferiore) di un segmento di colonna in una tabella. La riga per la quale viene calcolata dipende dal valore dell'**offset**, se presente, e l'impostazione predefinita è la riga direttamente nella parte inferiore. Per i grafici diversi dalle tabelle, la valutazione viene effettuata sull'ultima riga della colonna attuale nell'equivalente di tabella lineare del grafico.

```
Bottom - funzione per grafici ([TOTAL [<fld{,fld}>]] expr [ , offset [,count ]])
```

### Top

**Top()** valuta un'espressione nella prima riga (superiore) di un segmento colonna in una tabella. La riga per la quale viene calcolata dipende dal valore dell'**offset**, se presente, e l'impostazione predefinita è la riga superiore. Per i grafici diversi dalle tabelle, la valutazione di **Top()** viene effettuata sulla prima riga della colonna attuale nell'equivalente di tabella lineare del grafico.

```
Top - funzione per grafici ([TOTAL [<fld{,fld}>]] expr [ , offset [,count ]])
```

### NoOfRows

**NoOfRows()** restituisce il numero di righe nel segmento di colonna attuale in una tabella. Per i grafici bitmap, **NoOfRows()** restituisce il numero di righe nell'equivalente della tabella lineare del grafico.

```
NoOfRows ([TOTAL])
```

## Funzioni di colonna

Queste funzioni possono essere utilizzate solo nelle espressioni grafiche.

### Column

**Column()** restituisce il valore trovato nella colonna corrispondente a **ColumnNo** in una tabella lineare, ignorando le dimensioni. Ad esempio **Column(2)** restituisce il valore della seconda colonna della misura.

```
Column - funzione per grafici (ColumnNo)
```

### Dimensionality

**Dimensionality()** restituisce il numero di dimensioni per la riga attuale. Nel caso delle tabelle pivot la funzione restituisce il numero totale di colonne di dimensione senza contenuti di aggregazione, ossia senza somme parziali o aggregati compressi.

```
Dimensionality - funzione per grafici ( )
```

### Secondarydimensionality

**SecondaryDimensionality()** restituisce il numero di righe di dimensione di una tabella pivot senza contenuti di aggregazione; ad esempio, senza somme parziali o aggregati compressi. Questa funzione è equivalente alla funzione **dimensionality()** per le dimensioni orizzontali delle tabelle pivot.

```
SecondaryDimensionality - funzione per grafici ( )
```

## Funzioni di campo

### FieldIndex

**FieldIndex()** restituisce la posizione del valore **value** nel campo **field\_name** (in ordine di caricamento).

```
FieldIndex (field_name , value)
```

FieldValue

**FieldValue()** restituisce il valore trovato nella posizione **elem\_no** nel campo **field\_name** (in ordine di caricamento).

```
FieldValue (field_name , elem_no)
```

FieldValueCount

**FieldValueCount()** è una funzione **integer** che restituisce il numero di valori distinti di un campo.

```
FieldValueCount (field_name)
```

## Funzioni tabella pivot

Queste funzioni possono essere utilizzate solo nelle espressioni grafiche.

After

**After()** restituisce il valore di un'espressione valutata con i valori di dimensione di una tabella pivot così come appaiono nella colonna successiva a quella attuale all'interno di un segmento di riga nella tabella pivot.

```
After - funzione per grafici ([TOTAL] expression [ , offset [,n]])
```

Before

**Before()** restituisce il valore di un'espressione valutata con i valori di dimensione di una tabella pivot così come appaiono nella colonna precedente a quella attuale all'interno di un segmento di riga della tabella pivot.

```
Before - funzione per grafici ([TOTAL] expression [ , offset [,n]])
```

First

**First()** restituisce il valore di un'espressione valutata con i valori di dimensione di una tabella pivot così come appaiono nella prima colonna del segmento di riga attuale della tabella pivot. Questa funzione restituisce NULL in tutti i tipi di grafico, ad eccezione delle tabelle pivot.

```
First - funzione per grafici ([TOTAL] expression [ , offset [,n]])
```

Last

**Last()** restituisce il valore di un'espressione valutata con i valori di dimensione di una tabella pivot così come appaiono nell'ultima colonna del segmento di riga attuale della tabella pivot. Questa funzione restituisce NULL in tutti i tipi di grafico, ad eccezione delle tabelle pivot.

```
Last - funzione per grafici ([TOTAL] expression [ , offset [,n]])
```

ColumnNo

**ColumnNo()** restituisce il numero della colonna attuale all'interno del segmento di riga attuale in una tabella pivot. La prima colonna è la numero 1.

```
ColumnNo - funzione per grafici ([TOTAL])
```

NoOfColumns

**NoOfColumns()** restituisce il numero di colonne nel segmento di riga attuale in una tabella pivot.

**NoOfColumns** - *funzione per grafici* ([TOTAL])

## Funzioni intra-record nello script di caricamento dei dati

### Exists

**Exists()** determina se un valore di campo specifico è già stato caricato nel campo nello script di caricamento dei dati. La funzione restituisce TRUE o FALSE, quindi può essere utilizzata nella clausola **where** di un'istruzione **LOAD** o un'istruzione **IF**.

**Exists** (field\_name [, expr])

### LookUp

**LookUp()** cerca in una tabella già caricata e restituisce il valore di **field\_name** corrispondente alla prima occorrenza del valore **match\_field\_value** nel campo **match\_field\_name**. La tabella può essere la tabella attuale o un'altra tabella caricata in precedenza.

**LookUp** (field\_name, match\_field\_name, match\_field\_value [, table\_name])

### Peek

**Peek()** restituisce il valore di un campo in una tabella per una riga che è già stata caricata. È possibile specificare il numero di riga così come la tabella. Se non viene specificato alcun numero di riga, verrà utilizzato l'ultimo record precedentemente caricato.

**Peek** (field\_name[, row\_no[, table\_name ] ])

### Previous

**Previous()** restituisce il valore dell'espressione **expr** utilizzando i dati del record di input precedente che non è stato eliminato a causa di una clausola **where**. Nel primo record di una tabella interna, la funzione restituirà NULL.

**Previous** (expr)

---

### Vedere anche:

p [Funzioni di scala \(page 1584\)](#)

## Above - funzione per grafici

**Above()** valuta un'espressione in una riga sopra la riga attuale all'interno di un segmento di colonna in una tabella. La riga per la quale viene calcolata dipende dal valore dell'**offset**, se presente, e l'impostazione predefinita è la riga direttamente sopra. Per i grafici diversi dalle tabelle, **Above()** restituisce un valore per la riga sopra la riga attuale nell'equivalente di tabella lineare del grafico.

### Sintassi:

**Above** ([TOTAL] expr [ , offset [,count]])

Tipo di dati restituiti: duale

#### Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>expr</b>   | L'espressione o il campo contenente i dati da misurare.                                                                                                                                                                                                                                                                                                                                                                |
| <b>offset</b> | <p>Specificando un offset <b>n</b>, maggiore di 0, è possibile spostare la valutazione dell'espressione <b>n</b> righe più in alto della riga attuale.</p> <p>Specificando un offset uguale a 0 verrà valutata l'espressione nella riga attuale.</p> <p>Specificando un numero di offset negativo, la funzione Above diventa equivalente alla funzione Below con il numero di offset positivo corrispondente.</p>      |
| <b>count</b>  | <p>Specificando un terzo argomento <b>count</b> maggiore di 1, la funzione restituirà una scala di valori <b>count</b>, uno per ciascuna delle righe della tabella <b>count</b> contando verso l'alto a partire dalla cella originaria.</p> <p>In questo modulo la funzione può essere utilizzata come argomento per una qualsiasi delle funzioni di scala speciali. <a href="#">Funzioni di scala (page 1584)</a></p> |
| <b>TOTAL</b>  | Se la tabella è unidimensionale o se è utilizzato il qualificatore <b>TOTAL</b> come argomento, il segmento colonna attuale sarà sempre uguale all'intera colonna.                                                                                                                                                                                                                                                     |

In corrispondenza della prima riga di un segmento di colonna verrà restituito un valore NULL perché non vi sono righe che la precedono.



*Un segmento di colonna viene definito come un sottogruppo consecutivo di celle con gli stessi valori per le dimensioni nell'ordine attuale. Le funzioni grafiche intra-record vengono calcolate nel segmento colonna escludendo la dimensione più a destra nel grafico della tabella lineare equivalente. Se nel grafico è presente una sola dimensione oppure è specificato il qualificatore TOTAL, l'espressione viene valutata nell'intera tabella.*



*Se la tabella o l'equivalente di tabella include più dimensioni verticali, il segmento colonna attuale includerà solo righe contenenti gli stessi valori della riga attuale in tutte le colonne di dimensione, eccetto la colonna che mostra l'ultima dimensione nell'ordinamento tra campi.*

#### Limiti:

- Le chiamate ricorrenti restituiranno NULL.
- L'ordinamento sui valori **y** nei grafici o l'ordinamento per colonne di espressione nelle tabelle non è consentito quando si utilizza questa funzione di grafico in una qualsiasi delle espressioni del grafico. Queste alternative di ordinamento verranno perciò disattivate automaticamente. Quando si utilizza questa funzione di grafico in una visualizzazione o in una tabella, l'ordinamento della visualizzazione torna all'ordinamento inserito in questa funzione.

### Esempio: espressioni dei grafici

| Esempio             | Risultato                                                          |
|---------------------|--------------------------------------------------------------------|
| Above( Sum(Sales) ) | Restituisce il valore di sales per la riga sopra la riga corrente. |

### Esempio: principi fondamentali della funzione Above

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Year`
  - `Sales`

#### Script di caricamento

Example:

```
LOAD * INLINE [  
Year, Sales  
2019, 1000  
2020, 1500  
2021, 1200  
2022, 1800  
2023, 1700  
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `Year`

Creare le seguenti misure:

- `=Sum(Sales)`, per individuare le vendite annuali.
- `=Above(Sum(Sales))`, per individuare il valore delle vendite dell'anno precedente.
- `=Sum(Sales) - Above(Sum(Sales))`, per calcolare la varianza tra il valore `sales` dell'anno in corso e il valore dell'anno precedente.

- `=RangeSum(Above(Sales, 1, 2))`, per dimostrare un'aggregazione biennale dei due anni di vendita precedenti.
- `=(Sales - Above(Sales)) / Above(Sales)`, per restituire la variazione di sales come rapporto/percentuale. Per mostrare questo valore come una percentuale, sotto la voce **Formattazione numero**, selezionare **Numero > Formattazione semplice > 12,34%**.

Tabella dei risultati

| Year   | Sum (Sales) | Above (Sum (Sales)) | Sum(Sales) - Above(Sum (Sales)) | RangeSum (Above(Sales, 1, 2)) | (Sales - Above (Sales)) / Above (Sales) |
|--------|-------------|---------------------|---------------------------------|-------------------------------|-----------------------------------------|
| Totals | 7200        | -                   | -                               | 0                             | -                                       |
| 2019   | 1000        | -                   | -                               | 0                             | -                                       |
| 2020   | 1500        | 1000                | 500                             | 1000                          | 50.00%                                  |
| 2021   | 1200        | 1500                | -300                            | 2500                          | -20.00%                                 |
| 2022   | 1800        | 1200                | 600                             | 2700                          | 50.00%                                  |
| 2023   | 1700        | 1800                | -100                            | 3000                          | -5.56%                                  |

Osservando i risultati, è possibile confrontare i valori per sales nel periodo corrente con il periodo precedente.

- La misura `Sum(Sales) - Above(Sum(Sales))` restituisce la variazione nei valori di sales, confrontando il periodo corrente con quello precedente.
- La misura `RangeSum(Above(Sales, 1, 2))` restituisce la somma dei due valori di sales precedenti.
- La misura `(Sales - Above(Sales)) / Above(Sales)` restituisce la variazione di sales in confronto al periodo precedente come una percentuale.



*Questa funzione può essere utilizzata nei grafici diversi dalle tabelle, ad esempio nei grafici a barre.*

**Procedere come indicato di seguito:**

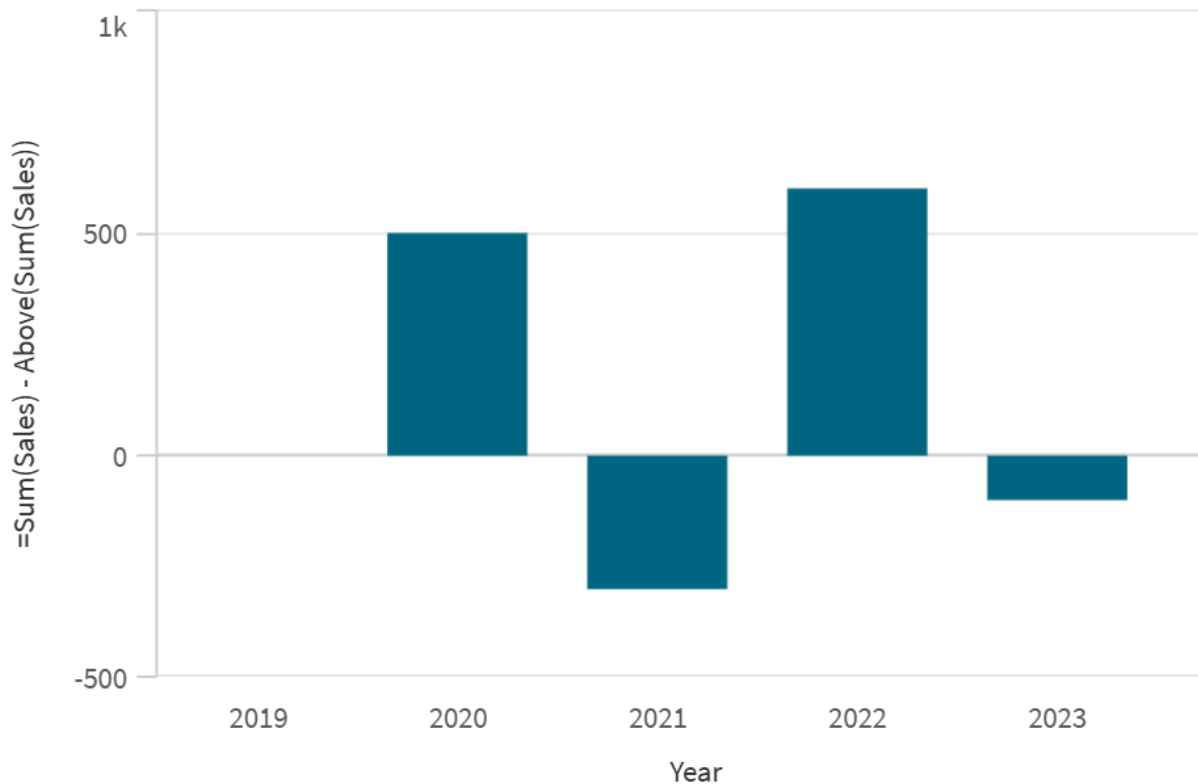
Creare un grafico a barre, quindi aggiungere questo campo come dimensione:

- Year

Creare la seguente misura:

- `=Sum(Sales) - Above(Sum(Sales))`, per restituire la varianza su base annua.

Grafico a barre che mostra i risultati della funzione Above



L'utilizzo della funzione Above in un grafico a barre fornisce un confronto visivo della varianza tra le vendite del periodo corrente rispetto a quelle del periodo precedente.

### Esempio: scenario per l'applicazione della funzione Above

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata Example.
- I seguenti campi nella tabella dati:
  - SalesYear
  - SalesAmount

#### Script di caricamento

Example:

```
LOAD * INLINE [  
SalesYear, SalesAmount
```

```
2019, 500
2020, 800
2021, 400
2022, 700
2023, 600
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- SalesYear

Creare le seguenti misure:

- =Sum(SalesAmount), per aggregare le vendite.
- =RangeSum(Above(SalesAmount, 0, RowNo())), per calcolare un totale parziale dei valori per SalesAmount.

Tabella dei risultati

| SalesYear | Sum(SalesAmount) | RangeSum(Above(SalesAmount, 0, RowNo())) |
|-----------|------------------|------------------------------------------|
| Totals    | 3000             | 0                                        |
| 2019      | 500              | 500                                      |
| 2020      | 800              | 1300                                     |
| 2021      | 400              | 1700                                     |
| 2022      | 700              | 2400                                     |
| 2023      | 600              | 3000                                     |

I risultati mostrano come è possibile utilizzare una combinazione della funzione RangeSum e della funzione Above per creare un totale parziale cumulativo.



*Questa funzione può essere utilizzata nei grafici diversi dalle tabelle, ad esempio nei grafici a barre.*

### Procedere come indicato di seguito:

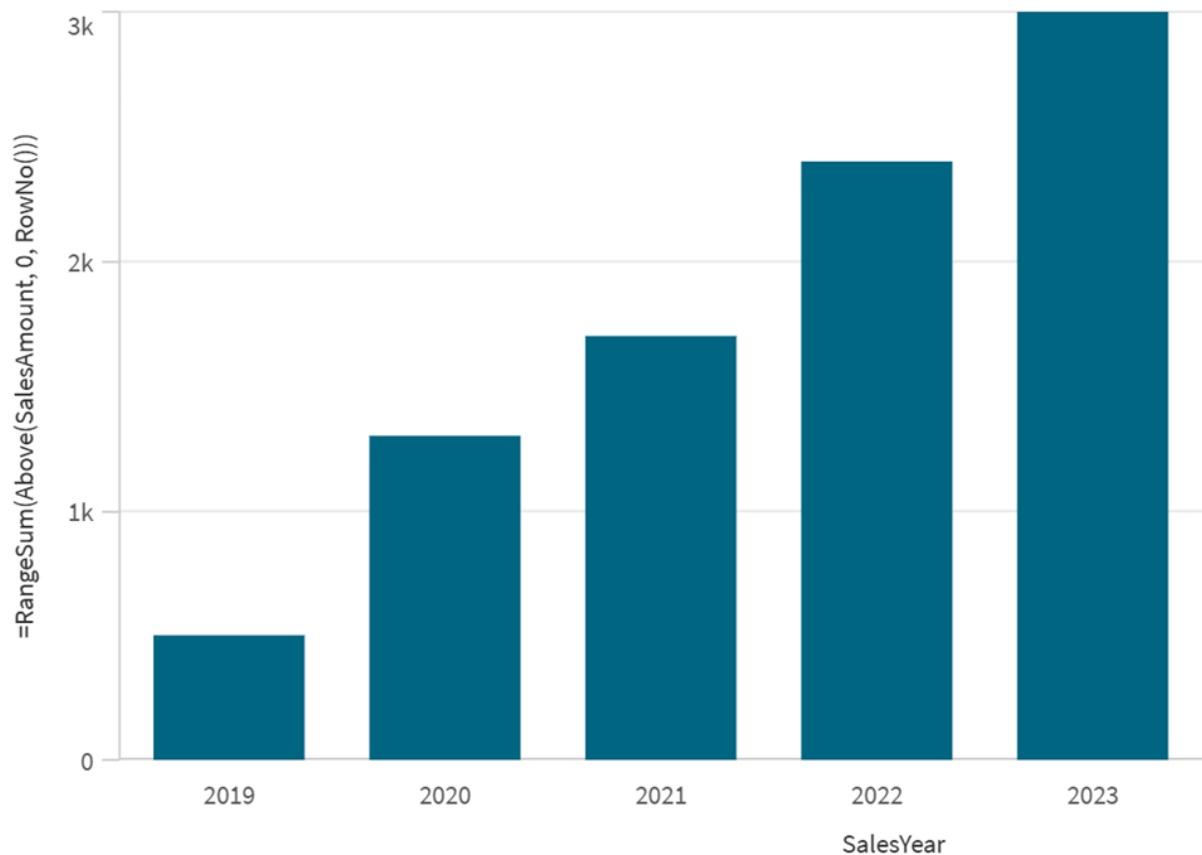
Creare un grafico a barre, quindi aggiungere questo campo come dimensione:

- SalesYear

Creare la seguente misura:

- =RangeSum(Above(SalesAmount, 0, RowNo())), per restituire il valore cumulativo delle vendite.

Grafico a barre che mostra il valore cumulativo delle vendite su base annua



### Esempio: espressione del grafico che usa Above

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Due tabelle dati chiamate monthnames e sales2013.

#### Script di caricamento

Monthnames:

```
LOAD *, Dual(MonthText,MonthNumber) as Month INLINE [  
MonthText, MonthNumber  
Jan, 1  
Feb, 2  
Mar, 3  
Apr, 4  
May, 5  
Jun, 6
```

```

Jul, 7
Aug, 8
Sep, 9
Oct, 10
Nov, 11
Dec, 12
];
Sales2013:
Crosstable (MonthText, Sales) LOAD * inline [
Customer|Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|Sep|Oct|Nov|Dec
Astrida|46|60|70|13|78|20|45|65|78|12|78|22
Betacab|65|56|22|79|12|56|45|24|32|78|55|15
Canutility|77|68|34|91|24|68|57|36|44|90|67|27
Divadip|57|36|44|90|67|27|57|68|47|90|80|94
] (delimiter is '|');

```

## Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- Customer

Creare le seguenti misure:

- =Sum(Sales), per calcolare la somma delle vendite.
- =Above(Sum(Sales)), per mostrare la somma dei valori di vendita nella riga precedente.
- =Sum(Sales)+Above(Sum(Sales)), per sommare il valore della somma delle vendite con la somma delle vendite per la riga del cliente sopra la riga corrente.
- =Sum(Sales)+Above(Sum(Sales), 3), per sommare il valore sum(Sales) per il Customer corrente al valore di customer, tre righe sopra.
- =IF(Sum(Sales)>Above(Sum(Sales)), 'Higher'), per aggiungere l'etichetta "Higher" ai clienti i cui risultati sum(Sales) sono superiori al valore di sum(Sales) per il cliente nella riga precedente.

Tabella dei risultati

| Customer      | Sum<br>(Sales) | Above<br>(Sum<br>(Sales)) | Sum<br>(Sales)+Above<br>(Sum(Sales)) | Sum<br>(Sales)+Above<br>(Sum(Sales), 3) | IF(Sum<br>(Sales)>Above<br>(Sum(Sales)),<br>'Higher') |
|---------------|----------------|---------------------------|--------------------------------------|-----------------------------------------|-------------------------------------------------------|
| <b>Totals</b> | <b>2566</b>    | -                         | -                                    | -                                       | -                                                     |
| Astrida       | 587            | -                         | -                                    | -                                       | -                                                     |
| Betacab       | 539            | 587                       | 1126                                 | -                                       | -                                                     |
| Canutility    | 683            | 539                       | 1222                                 | -                                       | Higher                                                |
| Divadip       | 757            | 683                       | 1440                                 | 1344                                    | Higher                                                |

La misura Above(Sum(Sales)) restituisce NULL per il cliente Astrida, perché non ci sono righe che la precedono. Il risultato per il cliente Betacab mostra il valore di sum(Sales) per Astrida. Il risultato per canutility mostra il valore di sum(Sales) per Betacab e così via.

Per la misura `sum(Sales)+Above(Sum(Sales))`, la riga di Betacab restituisce il risultato per l'aggiunta dei valori di `sum(Sales)` per le righe Betacab + Astrida (539+587). Il risultato per la riga Canutility mostra il risultato dell'aggiunta dei valori `sum(Sales)` per Canutility + Betacab (683+539).

La misura `sum(Sales)+Above(Sum(Sales), 3)` ha l'argomento `offset`, impostato su 3. Prende il valore della riga che si trova 3 righe sopra la riga corrente. Aggiunge il valore `sum(Sales)` per il valore customer corrente al valore per le tre righe customer precedenti. Ad esempio, il valore `sum(Sales)` per Divadip, viene aggiunto a `sum(Sales)` per Astrida, il cliente incluso nelle tre righe sopra Divadip. I valori restituiti per le prime 3 righe Customer sono null perché non sono disponibili dati.

La misura `IF(Sum(Sales)>Above(Sum(Sales)), 'Higher')` restituisce il valore `Higher` per i clienti Canutility e Divadip perché i loro risultati di vendita sono entrambi superiori alla somma delle vendite dei clienti nella riga precedente, rispettivamente Betacab e Canutility.



*Questa funzione può essere utilizzata nei grafici diversi dalle tabelle, ad esempio nei grafici a barre.*



*Per gli altri tipi di grafici, convertire il grafico nell'equivalente di tabella lineare per individuare facilmente la riga a cui si riferisce la funzione.*

### Esempio: scenario per l'applicazione della funzione Above in base all'ordinamento

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Tre tabelle dati chiamate Monthnames, Sales2013 e Product.

#### Script di caricamento

Monthnames:

```
LOAD *, Dual(MonthText,MonthNumber) as Month INLINE [
MonthText, MonthNumber
Jan, 1
Feb, 2
Mar, 3
Apr, 4
May, 5
Jun, 6
Jul, 7
Aug, 8
Sep, 9
```

```
Oct, 10
Nov, 11
Dec, 12
];

Sales2013:
Crosstable (MonthText, Sales) LOAD * inline [
Customer|Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|Sep|Oct|Nov|Dec
Astrida|46|60|70|13|78|20|45|65|78|12|78|22
Betacab|65|56|22|79|12|56|45|24|32|78|55|15
Canutility|77|68|34|91|24|68|57|36|44|90|67|27
Divadip|57|36|44|90|67|27|57|68|47|90|80|94
] (delimiter is '|');
```

```
Product:
LOAD * inline [
Customer|Product|OrderNumber|UnitSales|UnitPrice
Astrida|AA|1|4|16
Astrida|AA|7|10|15
Astrida|BB|4|9|9
Betacab|CC|6|5|10
Betacab|AA|5|2|20
Betacab|BB||| 25
Canutility|AA|||15
Canutility|CC| ||19
Divadip|CC|2|4|16
Divadip|DD|3|1|25
] (delimiter is '|');
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- Customer
- Product
- Month

Creare le seguenti misure:

- =Sum(Sales), per calcolare la somma delle vendite.
- =Above(Sum(Sales)), per mostrare la somma dei valori di vendita nella riga precedente.

Tabella dei risultati

| Customer      | Product | Month | Sum(Sales)  | Above(Sum(Sales)) |
|---------------|---------|-------|-------------|-------------------|
| <b>Totals</b> | -       | -     | <b>2566</b> | -                 |
| Astrida       | AA      | Jan   | 46          | -                 |
| Astrida       | AA      | Feb   | 60          | 46                |
| Astrida       | AA      | Mar   | 70          | 60                |

| Customer | Product | Month | Sum(Sales) | Above(Sum(Sales)) |
|----------|---------|-------|------------|-------------------|
| Totals   | -       | -     | 2566       | -                 |
| Astrida  | AA      | Apr   | 13         | 70                |
| Astrida  | AA      | May   | 78         | 13                |
| Astrida  | AA      | Jun   | 20         | 78                |
| Astrida  | AA      | Jul   | 45         | 20                |
| Astrida  | AA      | Aug   | 65         | 45                |
| Astrida  | AA      | Sep   | 78         | 65                |
| Astrida  | AA      | Oct   | 12         | 78                |
| Astrida  | AA      | Nov   | 78         | 12                |
| Astrida  | AA      | Dec   | 22         | 78                |
| Astrida  | BB      | Jan   | 46         | -                 |
| Astrida  | BB      | Feb   | 60         | 46                |

Per i grafici con più dimensioni, i risultati delle espressioni contenenti le funzioni Above, Below, Top e Bottom dipendono dall'ordine con cui le dimensioni di colonna vengono ordinate da Qlik Sense. Qlik Sense valuta le funzioni in base ai segmenti di colonna che derivano dalla dimensione ordinata per ultima.

L'ordinamento delle colonne viene controllato dal pannello delle proprietà in **Ordinamento** e non è necessariamente l'ordine in cui le colonne vengono visualizzate in una tabella.

Nella seguente tabella dei risultati, l'ultima dimensione di ordinamento è Month, in modo che la funzione Above completi la valutazione in base ai mesi. Esiste una serie di risultati per ciascun valore Product per ciascun mese (da Jan a Dec): un segmento colonna. Viene seguita da una serie per il segmento di colonna successivo per ogni Month per il valore Product successivo. Esisterà un segmento di colonna per ciascun valore Customer per ciascun valore Product.

## Esempio: scenario per l'applicazione della funzione Above con un ordinamento differente

Espressione del grafico

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Tre tabelle dati chiamate Monthnames, Sales2013 e Product.

### Script di caricamento

Monthnames:

```
LOAD *, Dual(MonthText,MonthNumber) as Month INLINE [
MonthText, MonthNumber
Jan, 1
Feb, 2
Mar, 3
Apr, 4
May, 5
Jun, 6
Jul, 7
Aug, 8
Sep, 9
Oct, 10
Nov, 11
Dec, 12
];
```

Sales2013:

```
Crosstable (MonthText, Sales) LOAD * inline [
Customer|Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|Sep|Oct|Nov|Dec
Astrida|46|60|70|13|78|20|45|65|78|12|78|22
Betacab|65|56|22|79|12|56|45|24|32|78|55|15
Canutility|77|68|34|91|24|68|57|36|44|90|67|27
Divadip|57|36|44|90|67|27|57|68|47|90|80|94
] (delimiter is '|');
```

Product:

```
LOAD * inline [
Customer|Product|OrderNumber|UnitSales|UnitPrice
Astrida|AA|1|4|16
Astrida|AA|7|10|15
Astrida|BB|4|9|9
Betacab|CC|6|5|10
Betacab|AA|5|2|20
Betacab|BB||| 25
Canutility|AA|||15
Canutility|CC| |19
Divadip|CC|2|4|16
Divadip|DD|3|1|25
] (delimiter is '|');
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- Customer
- Product
- Month

Creare le seguenti misure:

- `=Sum(Sales)`, per calcolare la somma delle vendite.
- `=Above(Sum(Sales))`, per mostrare la somma dei valori di vendita nella riga precedente.

Modificare il criterio di ordinamento della tabella nel pannello delle proprietà. Selezionare **Ordinamento**, quindi trascinare la dimensione **Prodotto** nella terza posizione dell'ordinamento.

Tabella dei risultati

| Customer      | Product | Month | Sum(Sales)  | Above(Sum(Sales)) |
|---------------|---------|-------|-------------|-------------------|
| <b>Totals</b> | -       | -     | <b>2566</b> | -                 |
| Astrida       | AA      | Jan   | 46          | -                 |
| Astrida       | BB      | Jan   | 46          | 46                |
| Astrida       | AA      | Feb   | 60          | -                 |
| Astrida       | BB      | Feb   | 60          | 60                |
| Astrida       | AA      | Mar   | 70          | -                 |
| Astrida       | BB      | Mar   | 70          | 70                |
| Astrida       | AA      | Apr   | 13          | -                 |
| Astrida       | BB      | Apr   | 13          | 13                |
| Astrida       | AA      | May   | 78          | -                 |
| Astrida       | BB      | May   | 78          | 78                |
| Astrida       | AA      | Jun   | 20          | -                 |
| Astrida       | BB      | Jun   | 20          | 20                |
| Astrida       | AA      | Jul   | 45          | -                 |
| Astrida       | BB      | Jul   | 45          | 45                |

Nella tabella dei risultati, l'ultima dimensione ordinata è `Product`. La funzione `Above` viene valutata per ciascun valore `Product` e, poiché esistono solo due prodotti, AA e BB, esiste solo un risultato non null in ciascuna serie. Nella riga BB per il mese Jan, il valore per `Above(Sum(Sales))` è 46. Per la riga AA, il valore è null. Il valore in ciascuna riga AA per ciascun mese sarà sempre null perché non esiste alcun valore `Product` sopra AA. La seconda serie viene valutata in AA e BB per il mese Feb, per il valore `Customer`, Astrida. Quando tutti i mesi sono stati valutati per Astrida, la sequenza viene ripetuta per il secondo cliente Betacab e così via.



*I risultati nella tabella mostrano solo un elenco parziale dei dati. Per visualizzare i risultati completi, provare l'esempio in un'app.*

### Esempio: utilizzo della funzione Above con RangeAvg

Espressione del grafico

#### Panoramica

La funzione `Above` può essere utilizzata come input per le funzioni di scala. Questo esempio utilizza la funzione `RangeAvg`.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Tre tabelle dati chiamate `Monthnames`, `Sales2013` e `Product`.

#### Script di caricamento

`Monthnames:`

```
LOAD *, Dual(MonthText,MonthNumber) as Month INLINE [
MonthText, MonthNumber
Jan, 1
Feb, 2
Mar, 3
Apr, 4
May, 5
Jun, 6
Jul, 7
Aug, 8
Sep, 9
Oct, 10
Nov, 11
Dec, 12
];
```

`Sales2013:`

```
Crosstable (MonthText, Sales) LOAD * inline [
Customer|Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|Sep|Oct|Nov|Dec
Astrida|46|60|70|13|78|20|45|65|78|12|78|22
Betacab|65|56|22|79|12|56|45|24|32|78|55|15
Canutility|77|68|34|91|24|68|57|36|44|90|67|27
Divadip|57|36|44|90|67|27|57|68|47|90|80|94
] (delimiter is '|');
```

`Product:`

```
LOAD * inline [
Customer|Product|OrderNumber|UnitSales|UnitPrice
Astrida|AA|1|4|16
Astrida|AA|7|10|15
Astrida|BB|4|9|9
Betacab|CC|6|5|10
Betacab|AA|5|2|20
Betacab|BB||| 25
Canutility|AA|||15
```

```
Canutility|CC| ||19
Divadip|CC|2|4|16
Divadip|DD|3|1|25
] (delimiter is '|');
```

## Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- Customer

Creare la seguente misura:

- =RangeAvg (Above(Sum(Sales),1,3))

Tabella dei risultati\*

| Customer   | RangeAvg (Above(Sum(Sales),1,3)) |
|------------|----------------------------------|
| Totals     | -                                |
| Astrida    | -                                |
| Betacab    | 587                              |
| Canutility | 563                              |
| Divadip    | 603                              |

Negli argomenti per la funzione Above, offset viene impostato su 1, mentre count viene impostato su 3. La funzione trova i risultati dell'espressione sum(sales) nelle tre righe immediatamente sopra la riga attuale nel segmento di colonna (dove si trova una riga). Questi tre valori vengono utilizzati come input per la funzione RangeAvg, che individua la media dei valori nella scala di numeri fornita.

## Vedere anche:

- p [Below - funzione per grafici \(page 1478\)](#)
- p [Bottom - funzione per grafici \(page 1488\)](#)
- p [Top - funzione per grafici \(page 1526\)](#)
- p [RangeAvg \(page 1587\)](#)

## Below - funzione per grafici

**Below()** valuta un'espressione in una riga sotto la riga attuale all'interno di un segmento di colonna in una tabella. La riga per la quale viene calcolata dipende dall'**offset**, se presente, e l'impostazione predefinita è la riga direttamente sotto. Per i grafici diversi dalle tabelle, **Below()** restituisce un valore per la riga sotto la colonna attuale nell'equivalente di tabella lineare del grafico.

## Sintassi:

```
Below([TOTAL] expr [ , offset [,count ]])
```

Tipo di dati restituiti: duale

### Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                   |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>expr</b>   | L'espressione o il campo contenente i dati da misurare.                                                                                                                                                                                                                                                                                                                                                                       |
| <b>offset</b> | <p>Specificando un <b>offset</b> n maggiore di 1, la valutazione dell'espressione viene spostata di n righe più in basso rispetto alla riga attuale.</p> <p>Specificando un offset uguale a 0 verrà valutata l'espressione nella riga attuale.</p> <p>Specificando un numero di offset negativo, la funzione <b>Below</b> diventa equivalente alla funzione <b>Above</b> con il numero di offset positivo corrispondente.</p> |
| <b>count</b>  | Specificando un terzo parametro <b>count</b> maggiore di 1, la funzione restituirà una scala di valori <b>count</b> , uno per ciascuna delle righe della tabella <b>count</b> contando verso il basso a partire dalla cella originaria. In questo modulo la funzione può essere utilizzata come argomento per una qualsiasi delle funzioni di scala speciali. <a href="#">Funzioni di scala (page 1584)</a>                   |
| <b>TOTAL</b>  | Se la tabella è unidimensionale o se è utilizzato il qualificatore <b>TOTAL</b> come argomento, il segmento colonna attuale sarà sempre uguale all'intera colonna.                                                                                                                                                                                                                                                            |

Sull'ultima riga di un segmento di colonna viene restituito un valore NULL, perché non vi sono righe che la seguono.



*Un segmento di colonna viene definito come un sottogruppo consecutivo di celle con gli stessi valori per le dimensioni nell'ordine attuale. Le funzioni grafiche intra-record vengono calcolate nel segmento colonna escludendo la dimensione più a destra nel grafico della tabella lineare equivalente. Se nel grafico è presente una sola dimensione oppure è specificato il qualificatore TOTAL, l'espressione viene valutata nell'intera tabella.*



*Se la tabella o l'equivalente di tabella include più dimensioni verticali, il segmento colonna attuale includerà solo righe contenenti gli stessi valori della riga attuale in tutte le colonne di dimensione, eccetto la colonna che mostra l'ultima dimensione nell'ordinamento tra campi.*

### Limiti:

- Le chiamate ricorrenti restituiranno NULL.
- L'ordinamento sui valori y nei grafici o l'ordinamento per colonne di espressione nelle tabelle non è consentito quando si utilizza questa funzione di grafico in una qualsiasi delle espressioni del grafico. Queste alternative di ordinamento verranno perciò disattivate automaticamente. Quando si utilizza questa funzione di grafico in una visualizzazione o in una tabella, l'ordinamento della visualizzazione torna all'ordinamento inserito in questa funzione.

### Esempio: espressioni dei grafici

| Esempio                          | Risultato                                                         |
|----------------------------------|-------------------------------------------------------------------|
| <code>Below( Sum(Sales) )</code> | Restituisce il valore di sales per la riga sotto quella corrente. |

### Esempio: principi fondamentali della funzione Below

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Year`
  - `Sales`

#### Script di caricamento

Example:

```
LOAD * INLINE [  
Year, Sales  
2019, 1000  
2020, 1500  
2021, 1200  
2022, 1800  
2023, 1700  
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `Year`

Creare le seguenti misure:

- `=Sum(Sales)`, per individuare le vendite annuali.
- `=Below(Sum(Sales))`, per individuare il valore delle vendite dell'anno successivo.

- $\text{=Sum(Sales) - Below(Sum(Sales))}$ , per calcolare la varianza tra il valore sales dell'anno in corso e il valore dell'anno successivo.
- $\text{=RangeSum(Below(Sales, 1, 2))}$ , per dimostrare un'aggregazione biennale dei due anni di vendita successivi.
- $\text{=(Sales - Below(Sales)) / Below(Sales)}$ , per restituire la variazione di sales come rapporto/percentuale. Per mostrare questo valore come una percentuale, sotto la voce **Formattazione numero**, selezionare **Numero > Formattazione semplice > 12,34%**.

Tabella dei risultati

| Year   | Sum (Sales) | Below (Sum (Sales)) | Sum(Sales) - Below(Sum (Sales)) | RangeSum (Below(Sales, 1, 2)) | (Sales - Below (Sales)) / Below (Sales) |
|--------|-------------|---------------------|---------------------------------|-------------------------------|-----------------------------------------|
| Totals | 7200        | -                   | -                               | 0                             | -                                       |
| 2019   | 1000        | 1500                | -500                            | 2700                          | -33.33%                                 |
| 2020   | 1500        | 1200                | 300                             | 3000                          | 25.00%                                  |
| 2021   | 1200        | 1800                | -600                            | 3500                          | -33.00%                                 |
| 2022   | 1800        | 1700                | 100                             | 1700                          | 5.88%                                   |
| 2023   | 1700        | -                   | -                               | 0                             | -                                       |

Osservando i risultati, è possibile confrontare i valori per sales nel periodo corrente con il periodo successivo.

- La misura  $\text{Sum(Sales) - Below(Sum(Sales))}$  restituisce la variazione nei valori di sales, confrontando il periodo corrente con quello successivo.
- La misura  $\text{RangeSum(Below(Sales, 1, 2))}$  restituisce la somma dei due valori di sales successivi.
- La misura  $\text{(Sales - Below(Sales)) / Below(Sales)}$  restituisce la variazione di sales in confronto al periodo successivo come una percentuale.



*Questa funzione può essere utilizzata nei grafici diversi dalle tabelle, ad esempio nei grafici a barre.*

**Procedere come indicato di seguito:**

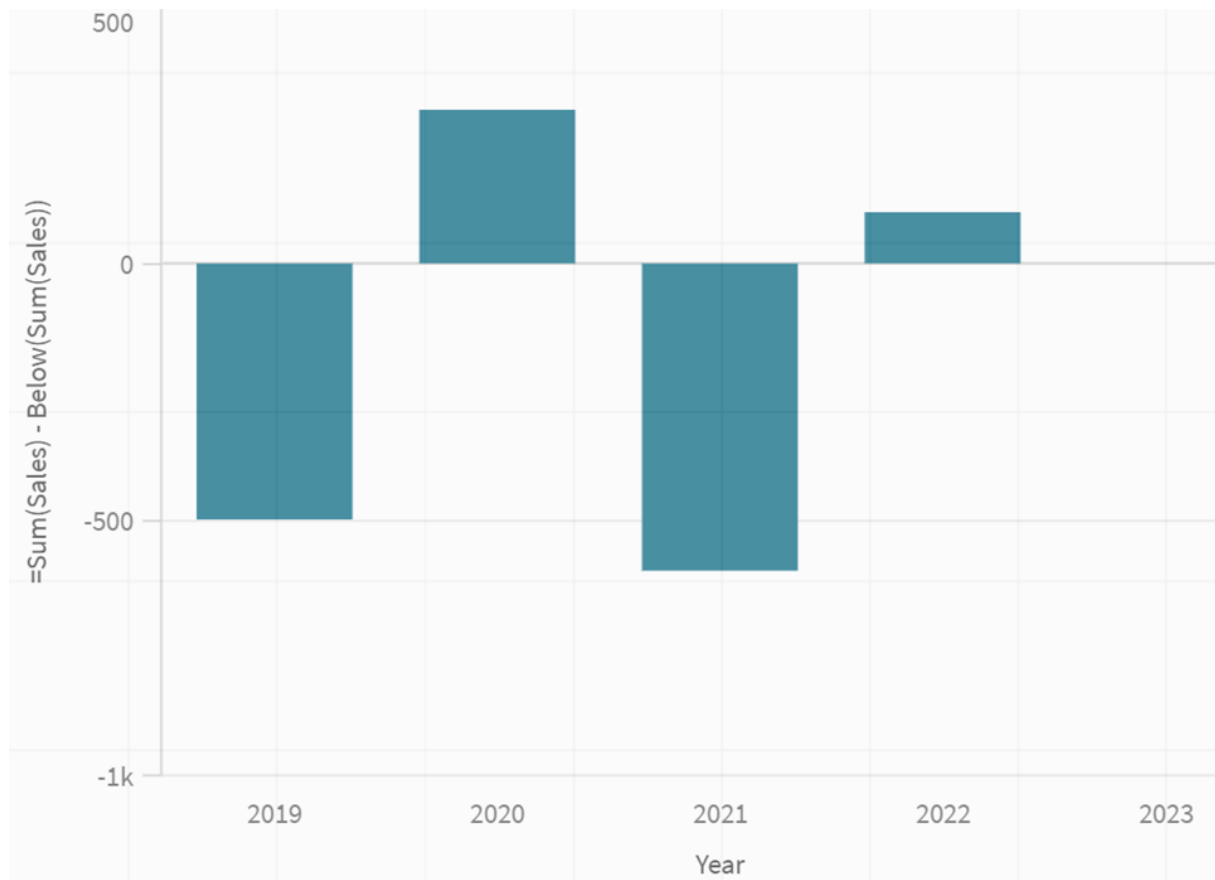
Creare un grafico a barre, quindi aggiungere questo campo come dimensione:

- Year

Creare la seguente misura:

- $\text{=Sum(Sales) - Below(Sum(Sales))}$ , per restituire la varianza su base annua.

Grafico a barre che mostra i risultati della funzione *Below*



L'utilizzo della funzione *Below* in un grafico a barre fornisce un confronto visivo della varianza tra le vendite del periodo corrente rispetto a quelle del periodo successivo.

### Esempio: scenario per l'applicazione della funzione *Below*

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `SalesYear`
  - `SalesAmount`

#### Script di caricamento

Example:

```
LOAD * INLINE [
```

```
SalesYear, SalesAmount
2019, 500
2020, 800
2021, 400
2022, 700
2023, 600
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- SalesYear

Creare le seguenti misure:

- =Sum(SalesAmount), per aggregare le vendite.
- =RangeSum(Below(SalesAmount, 0, NoOfRows() - RowNo()))), per calcolare un resoconto del totale per i valori di SalesAmount.

Tabella dei risultati

| SalesYear | Sum(SalesAmount) | RangeSum(Below(SalesAmount, 0, NoOfRows() - RowNo())) |
|-----------|------------------|-------------------------------------------------------|
| Totals    | 3000             | 0                                                     |
| 2019      | 500              | 2400                                                  |
| 2020      | 800              | 1900                                                  |
| 2021      | 400              | 1100                                                  |
| 2022      | 700              | 700                                                   |
| 2023      | 600              | 0                                                     |

I risultati mostrano come è possibile utilizzare una combinazione della funzione RangeSum e della funzione Below per creare un resoconto del totale cumulativo.



*Questa funzione può essere utilizzata nei grafici diversi dalle tabelle, ad esempio nei grafici a barre.*

### Procedere come indicato di seguito:

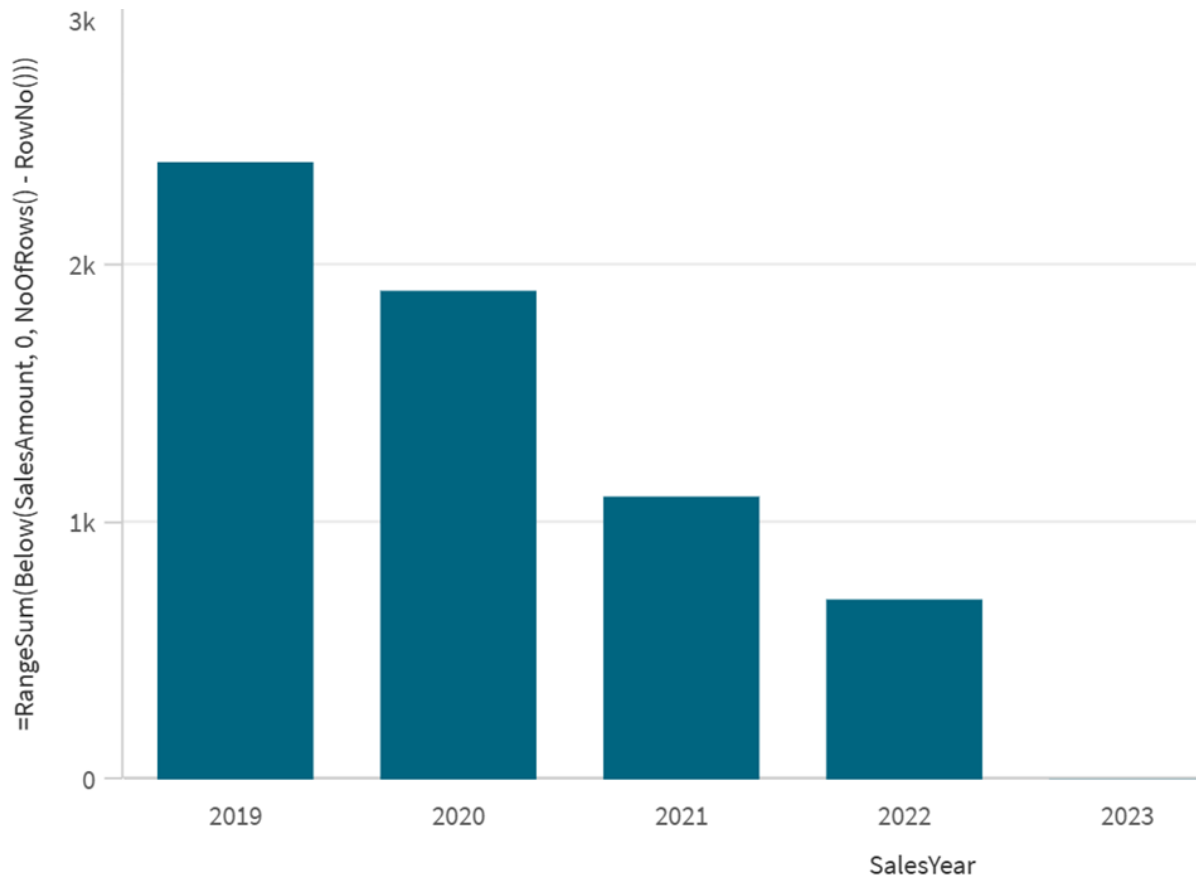
Creare un grafico a barre, quindi aggiungere questo campo come dimensione:

- SalesYear

Creare la seguente misura:

- =RangeSum(Below(SalesAmount, 0, NoOfRows() - RowNo()))), per restituire un resoconto del valore cumulativo delle vendite.

Grafico a barre che mostra un resoconto del valore cumulativo delle vendite su base annua



L'utilizzo della funzione `Below` in combinazione con `RangeSum` in un grafico a barre fornisce una visualizzazione con un riepilogo dei valori cumulativi delle vendite.

### Esempio: espressione del grafico che usa `Below`

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Due tabelle dati chiamate `Monthnames` e `Sales2013`.

#### Script di caricamento

`Monthnames:`

```
LOAD *, Dual(MonthText,MonthNumber) as Month INLINE [  
MonthText, MonthNumber  
Jan, 1  
Feb, 2  
Mar, 3
```

```
Apr, 4
May, 5
Jun, 6
Jul, 7
Aug, 8
Sep, 9
Oct, 10
Nov, 11
Dec, 12
];
```

```
Sales2013:
Crosstable (MonthText, Sales) LOAD * inline [
Customer|Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|Sep|Oct|Nov|Dec
Astrida|46|60|70|13|78|20|45|65|78|12|78|22
Betacab|65|56|22|79|12|56|45|24|32|78|55|15
Canutility|77|68|34|91|24|68|57|36|44|90|67|27
Divadip|57|36|44|90|67|27|57|68|47|90|80|94
] (delimiter is '|');
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- Customer

Creare le seguenti misure:

- =Sum(Sales), per calcolare la somma delle vendite.
- =Below(Sum(Sales)), per mostrare la somma dei valori di vendita nella riga precedente.
- =Sum(Sales)+Below(Sum(Sales)), per sommare il valore della somma delle vendite con la somma delle vendite per la riga del cliente sotto la riga corrente.
- =Sum(Sales)+Below(Sum(Sales), 3), per sommare il valore sum(Sales) per il customer corrente al valore di customer, tre righe sotto.
- =If(Sum(Sales)>Below(Sum(Sales)), 'Higher'), per aggiungere l'etichetta "Higher" ai clienti i cui risultati sum(Sales) sono superiori al valore di sum(Sales) per il cliente nella riga sotto.

Tabella dei risultati

| Customer      | Sum<br>(Sales) | Below<br>(Sum<br>(Sales)) | Sum<br>(Sales)+Below<br>(Sum(Sales)) | Sum<br>(Sales)+Below<br>(Sum(Sales), 3) | If(Sum<br>(Sales)>Below(Sum<br>(Sales)), 'Higher') |
|---------------|----------------|---------------------------|--------------------------------------|-----------------------------------------|----------------------------------------------------|
| <b>Totals</b> | <b>2566</b>    | -                         | -                                    | -                                       | -                                                  |
| Astrida       | 587            | 539                       | 1126                                 | 1334                                    | Higher                                             |
| Betacab       | 539            | 683                       | 1222                                 | -                                       | -                                                  |
| Canutility    | 683            | 757                       | 1440                                 | -                                       | -                                                  |
| Divadip       | 757            | -                         | -                                    | -                                       | -                                                  |

Per i grafici con più dimensioni, i risultati delle espressioni contenenti le funzioni Above, Below, Top e Bottom dipendono dall'ordine con cui le dimensioni di colonna vengono ordinate da Qlik Sense. Qlik Sense valuta le funzioni in base ai segmenti di colonna che derivano dalla dimensione ordinata per ultima.

L'ordinamento delle colonne viene controllato dal pannello delle proprietà in **Ordinamento** e non è necessariamente l'ordine in cui le colonne vengono visualizzate in una tabella. Per un esempio che dimostra l'uso dell'ordinamento, consultare [Esempio: scenario per l'applicazione della funzione Above in base all'ordinamento \(page 1472\)](#).

La misura `Below(Sum(Sales))` restituisce NULL per il cliente Divadip, perché non ci sono righe successive. Il risultato per la riga Canutility mostra il valore di `Sum(Sales)` per Divadip. Il risultato per Betacab mostra il valore di `Sum(Sales)` per Canutility e così via.

Per la misura `Sum(Sales)+Below(Sum(Sales))`, la riga di Astrida restituisce il risultato per l'aggiunta dei valori di `Sum(Sales)` per le righe Betacab + Astrida (539+587). Il risultato per la riga Betacab mostra il risultato dell'aggiunta dei valori `Sum(Sales)` per Canutility + Betacab (539+683).

La misura `Sum(Sales)+Below(Sum(Sales), 3)` ha l'argomento offset, impostato su 3, e ha l'effetto di portare il valore della riga tre righe sotto la riga attuale. Aggiunge il valore `Sum(Sales)` per il valore Customer attuale dal valore Customer tre righe successive. I valori per le prime tre righe Customer più in basso sono null.

La misura `IF(Sum(Sales)>Below(Sum(Sales)), 'Higher')` restituisce il valore Higher per il cliente Astrida perché i suoi risultati di vendita sono maggiori della somma delle vendite del cliente nella riga sottostante, Betacab.



*Questa funzione può essere utilizzata nei grafici diversi dalle tabelle, ad esempio nei grafici a barre.*



*Per gli altri tipi di grafici, convertire il grafico nell'equivalente di tabella lineare per individuare facilmente la riga a cui si riferisce la funzione.*

### Esempio: utilizzo della funzione Below con RangeAvg

Espressione del grafico

#### Panoramica

La funzione `Below` può essere utilizzata come input per le funzioni di scala. Questo esempio utilizza la funzione `RangeAvg`.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Due tabelle dati chiamate `Monthnames` e `Sales2013`.

### Script di caricamento

Monthnames:

```
LOAD *, Dual(MonthText,MonthNumber) as Month INLINE [
MonthText, MonthNumber
Jan, 1
Feb, 2
Mar, 3
Apr, 4
May, 5
Jun, 6
Jul, 7
Aug, 8
Sep, 9
Oct, 10
Nov, 11
Dec, 12
];
```

Sales2013:

```
Crosstable (MonthText, Sales) LOAD * inline [
Customer|Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|Sep|Oct|Nov|Dec
Astrida|46|60|70|13|78|20|45|65|78|12|78|22
Betacab|65|56|22|79|12|56|45|24|32|78|55|15
Canutility|77|68|34|91|24|68|57|36|44|90|67|27
Divadip|57|36|44|90|67|27|57|68|47|90|80|94
] (delimiter is '|');
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- Customer

Creare la seguente misura:

- =RangeAvg (Below(Sum(Sales),1,3))

Tabella dei risultati

| Customer   | RangeAvg (Below(Sum(Sales),1,3)) |
|------------|----------------------------------|
| Totals     | -                                |
| Astrida    | 660                              |
| Betacab    | 720                              |
| Canutility | 757                              |
| Divadip    | -                                |

Negli argomenti per la funzione `Below`, `offset` viene impostato su 1, mentre `count` viene impostato su 3. La funzione trova i risultati dell'espressione `sum(sales)` nelle tre righe immediatamente sotto la riga attuale nel segmento di colonna (dove si trova una riga). Questi tre valori vengono utilizzati come input per la funzione `RangeAvg`, che individua la media dei valori nella scala di numeri fornita.

### Vedere anche:

- p [Above - funzione per grafici \(page 1464\)](#)
- p [Bottom - funzione per grafici \(page 1488\)](#)
- p [Top - funzione per grafici \(page 1526\)](#)
- p [RangeAvg \(page 1587\)](#)

## Bottom - funzione per grafici

**Bottom()** valuta un'espressione nell'ultima riga (inferiore) di un segmento di colonna in una tabella. La riga per la quale viene calcolata dipende dal valore dell'**offset**, se presente, e l'impostazione predefinita è la riga direttamente nella parte inferiore. Per i grafici diversi dalle tabelle, la valutazione viene effettuata sull'ultima riga della colonna attuale nell'equivalente di tabella lineare del grafico.

### Sintassi:

```
Bottom([TOTAL] expr [ , offset [,count ]])
```

Tipo di dati restituiti: duale

### Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                        |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>expr</b>   | L'espressione o il campo contenente i dati da misurare.                                                                                                                                                                                                                                                                                                                                            |
| <b>offset</b> | Specificando un <b>offset</b> <code>n</code> maggiore di 1, la valutazione dell'espressione viene spostata di <code>n</code> righe sopra rispetto alla riga inferiore.<br>Specificando un numero di offset negativo, la funzione <b>Bottom</b> diventa equivalente alla funzione <b>Top</b> con il numero di offset positivo corrispondente.                                                       |
| <b>count</b>  | Specificando un terzo parametro <b>count</b> maggiore di 1, la funzione restituirà non un solo valore, ma una scala di valori <b>count</b> , uno per ciascuna delle ultime righe <b>count</b> del segmento di colonna attuale. In questo modulo la funzione può essere utilizzata come argomento per una qualsiasi delle funzioni di scala speciali. <a href="#">Funzioni di scala (page 1584)</a> |
| <b>TOTAL</b>  | Se la tabella è unidimensionale o se è utilizzato il qualificatore <b>TOTAL</b> come argomento, il segmento colonna attuale sarà sempre uguale all'intera colonna.                                                                                                                                                                                                                                 |



Un segmento di colonna viene definito come un sottogruppo consecutivo di celle con gli stessi valori per le dimensioni nell'ordine attuale. Le funzioni grafiche intra-record vengono calcolate nel segmento colonna escludendo la dimensione più a destra nel grafico della tabella lineare equivalente. Se nel grafico è presente una sola dimensione oppure è specificato il qualificatore TOTAL, l'espressione viene valutata nell'intera tabella.



Se la tabella o l'equivalente di tabella include più dimensioni verticali, il segmento colonna attuale includerà solo righe contenenti gli stessi valori della riga attuale in tutte le colonne di dimensione, eccetto la colonna che mostra l'ultima dimensione nell'ordinamento tra campi.

### Limiti:

- Le chiamate ricorrenti restituiranno NULL.
- L'ordinamento sui valori y nei grafici o l'ordinamento per colonne di espressione nelle tabelle non è consentito quando si utilizza questa funzione di grafico in una qualsiasi delle espressioni del grafico. Queste alternative di ordinamento verranno perciò disattivate automaticamente. Quando si utilizza questa funzione di grafico in una visualizzazione o in una tabella, l'ordinamento della visualizzazione torna all'ordinamento inserito in questa funzione.

### Esempio: espressioni dei grafici

| Esempio              | Risultato                                                                                             |
|----------------------|-------------------------------------------------------------------------------------------------------|
| Bottom( Sum(Sales) ) | Restituisce il valore di sales per l'ultima riga (in basso) di un segmento di colonna in una tabella. |

### Esempio: principi fondamentali della funzione Bottom

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata Example.
- I seguenti campi nella tabella dati:

- Year
- Sales

### Script di caricamento

Example:

```
LOAD * INLINE [  
Year, Sales  
2019, 1000  
2020, 1500  
2021, 1200  
2022, 1800  
2023, 1700  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- Year

Creare le seguenti misure:

- $=\text{Sum}(\text{Sales})$ , per individuare le vendite annuali.
- $=\text{Bottom}(\text{Sum}(\text{Sales}))$ , per individuare il valore delle vendite per l'ultimo anno.
- $=\text{Sum}(\text{Sales}) - \text{Bottom}(\text{Sum}(\text{Sales}))$ , per calcolare la varianza tra il valore sales dell'anno in corso e il valore per l'ultimo anno.
- $=\text{Sum}(\text{Sales}) + \text{Bottom}(\text{Sum}(\text{Sales}), 3)$ , per calcolare la somma dei valori di vendita dell'anno corrente e il valore del terzultimo anno.
- $=(\text{Sales} - \text{Bottom}(\text{Sales})) / \text{Bottom}(\text{Sales})$ , per restituire la variazione di sales come rapporto/percentuale in confronto all'ultimi anno. Per mostrare questo valore come una percentuale, sotto la voce **Formattazione numero**, selezionare **Numero > Formattazione semplice > 12,34%**.

Ordinare la tabella per la dimensione year in ordine crescente.

Tabella dei risultati

| Year   | Sum<br>(Sales) | Bottom<br>(Sum<br>(Sales)) | Sum(Sales) -<br>Bottom(Sum<br>(Sales)) | Sum<br>(Sales)+Bottom<br>(Sum(Sales), 3) | (Sales - Bottom<br>(Sales)) / Bottom<br>(Sales) |
|--------|----------------|----------------------------|----------------------------------------|------------------------------------------|-------------------------------------------------|
| Totals | 7200           | 1700                       | 5500                                   | 8400                                     | -                                               |
| 2019   | 1000           | 1700                       | -700                                   | 2200                                     | -41.18%                                         |
| 2020   | 1500           | 1700                       | -200                                   | 2700                                     | -11.76%                                         |
| 2021   | 1200           | 1700                       | -500                                   | 2400                                     | -29.41%                                         |
| 2022   | 1800           | 1700                       | 100                                    | 3000                                     | 5.88%                                           |
| 2023   | 1700           | 1700                       | 0                                      | 2900                                     | 0.00%                                           |

Osservando i risultati, è possibile confrontare i valori per sales nel periodo corrente con l'ultimo periodo, ossia quello più recente.

- La misura  $\text{sum}(\text{Sales}) - \text{Bottom}(\text{Sum}(\text{Sales}))$  restituisce la variazione nei valori di sales, confrontando il periodo corrente con quello più recente.
- La misura  $\text{sum}(\text{Sales}) + \text{Bottom}(\text{Sum}(\text{Sales}), 3)$  restituisce la somma di sales per la riga corrente e il terzultimo periodo.
- La misura  $(\text{Sales} - \text{Bottom}(\text{Sales})) / \text{Bottom}(\text{Sales})$  restituisce la variazione di sales in confronto al periodo più recente come una percentuale.



*Questa funzione può essere utilizzata nei grafici diversi dalle tabelle, ad esempio nei grafici a barre.*

**Procedere come indicato di seguito:**

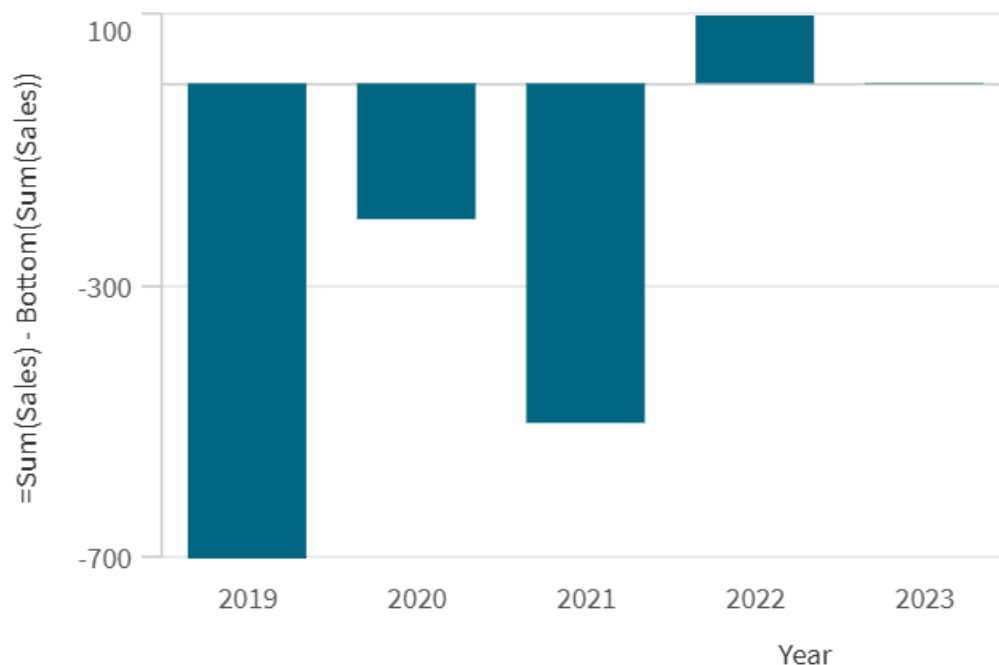
Creare un grafico a barre, quindi aggiungere questo campo come dimensione:

- Year

Creare la seguente misura:

- $=\text{Sum}(\text{Sales}) - \text{Bottom}(\text{Sum}(\text{Sales}))$ , per restituire la varianza delle vendite tra il periodo corrente e il periodo più recente.

*Grafico a barre che mostra i risultati della funzione Bottom*



L'utilizzo della funzione `Bottom` in un grafico a barre fornisce un confronto visivo della varianza tra le vendite del periodo corrente rispetto a quelle del periodo più recente.

### Esempio: scenario per l'applicazione della funzione `Bottom`

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `OrderID`
  - `Product`
  - `SalesValue`
  - `OrderDate`

#### Script di caricamento

Example:

```
LOAD * INLINE [
OrderID, Product, SalesValue, OrderDate
1, "Product A", 500, '2023-01-10'
2, "Product B", 300, '2023-01-15'
3, "Product C", 450, '2023-02-01'
4, "Product D", 700, '2023-02-20'
5, "Product E", 350, '2023-03-05'
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `Product`

Creare le seguenti misure:

- `=RangeAvg (Bottom(Sum(SalesValue),1,3))`, per restituire la media delle 3 righe più recenti.
- `=Sum(SalesValue) - RangeAvg (Bottom(Sum(SalesValue),1,3))`, per calcolare la varianza tra la riga corrente e la media delle 3 righe più recenti.
- `=Sum(SalesValue)` per calcolare il totale di `SalesValue` per la riga corrente.

Tabella dei risultati

| Product | RangeAvg (Bottom(Sum (SalesValue),1,3)) | Sum(SalesValue) - RangeAvg (Bottom (Sum(SalesValue),1,3)) | Sum (SalesValue) |
|---------|-----------------------------------------|-----------------------------------------------------------|------------------|
| Totals  | 500                                     | 1800                                                      | 2300             |
| 2019    | 500                                     | 0                                                         | 500              |
| 2020    | 500                                     | -200                                                      | 300              |
| 2021    | 500                                     | -50                                                       | 450              |
| 2022    | 500                                     | 200                                                       | 700              |
| 2023    | 500                                     | -150                                                      | 350              |

I risultati mostrano come è possibile utilizzare una combinazione della funzione RangeSum e della funzione Bottom per analizzare e confrontare i valori di vendita.

### Esempio: espressione del grafico che usa Bottom

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Due tabelle dati chiamate Monthnames e sales2013.

#### Script di caricamento

Monthnames:

```
LOAD *, Dual(MonthText,MonthNumber) as Month INLINE [
MonthText, MonthNumber
Jan, 1
Feb, 2
Mar, 3
Apr, 4
May, 5
Jun, 6
Jul, 7
Aug, 8
Sep, 9
Oct, 10
Nov, 11
Dec, 12
];
```

sales2013:

```
Crosstable (MonthText, Sales) LOAD * inline [
Customer|Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|Sep|Oct|Nov|Dec
Astrida|46|60|70|13|78|20|45|65|78|12|78|22
```

```
Betacab|65|56|22|79|12|56|45|24|32|78|55|15
Canutility|77|68|34|91|24|68|57|36|44|90|67|27
Divadip|57|36|44|90|67|27|57|68|47|90|80|94
] (delimiter is '|');
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- Customer

Creare le seguenti misure:

- =Sum(Sales), per calcolare la somma delle vendite.
- =Bottom(Sum(Sales)), per mostrare la somma dei valori di vendita per Customer nella riga sottostante.
- =Sum(Sales)+Bottom(Sum(Sales)), per sommare il valore sum(Sales) dell'attuale Customer a quello di Customer nella riga inferiore.
- =Sum(Sales)+Bottom(Sum(Sales), 3), per sommare il valore sum(Sales) dell'attuale Customer a quello di Customer, che è terzo a partire dalla riga inferiore.

Tabella dei risultati

| Customer      | Sum<br>(Sales) | Bottom(Sum<br>(Sales)) | Sum(Sales)+Bottom<br>(Sum(Sales)) | Sum(Sales)+Bottom(Sum<br>(Sales), 3) |
|---------------|----------------|------------------------|-----------------------------------|--------------------------------------|
| <b>Totals</b> | <b>2566</b>    | <b>757</b>             | <b>3323</b>                       | <b>3105</b>                          |
| Astrida       | 587            | 757                    | 1344                              | 1126                                 |
| Betacab       | 539            | 757                    | 1296                              | 1078                                 |
| Canutility    | 683            | 757                    | 1440                              | 1222                                 |
| Divadip       | 757            | 757                    | 1514                              | 1296                                 |

La misura Bottom(Sum(Sales)) restituisce 757 per tutte le righe perché questo è il valore dell'ultima riga: Divadip.

Per la misura sum(Sales)+Bottom(Sum(Sales)), la riga Astrida restituisce 1344, che è il risultato per aggiungere il valore sum(Sales) a Astrida e Divadip, Customer nella riga inferiore.

Per la misura sum(Sales)+Bottom(Sum(Sales), 3), l'argomento offset è impostato su 3. L'espressione aggiunge il valore sum(Sales) per la riga corrente e la terza riga dal basso, Betacab. Ad esempio, il valore 1126 viene restituito per il cliente Astrida. Questo è il risultato dell'aggiunta del valore sum(Sales) per Astrida e del valore sumSales per Betacab (587+539).

### Vedere anche:

p [Top - funzione per grafici \(page 1526\)](#)

## Column - funzione per grafici

**Column()** restituisce il valore trovato nella colonna corrispondente a **ColumnNo** in una tabella lineare, ignorando le dimensioni. Ad esempio **Column(2)** restituisce il valore della seconda colonna della misura.

### Sintassi:

**Column** (ColumnNo)

Tipo di dati restituiti: duale

### Argomenti

| Argomento | Descrizione                                                                                                                                                                                                               |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ColumnNo  | Numero di una colonna nella tabella contenente una misura.<br><br><div>  La funzione Column() ignora le colonne delle dimensioni. </div> |

### Limiti:

- Le chiamate ricorrenti restituiranno NULL.
- Se **ColumnNo** fa riferimento a una colonna per la quale non esiste una misura, viene restituito un valore NULL.
- L'ordinamento sui valori y nei grafici o l'ordinamento per colonne di espressione nelle tabelle non è consentito quando si utilizza questa funzione di grafico in una qualsiasi delle espressioni del grafico. Queste alternative di ordinamento verranno perciò disattivate automaticamente. Quando si utilizza questa funzione di grafico in una visualizzazione o in una tabella, l'ordinamento della visualizzazione torna all'ordinamento inserito in questa funzione.

### Esempio: espressioni dei grafici

| Esempio             | Risultato                                                                                       |
|---------------------|-------------------------------------------------------------------------------------------------|
| Column(1)/Column(2) | Restituisce il valore della prima misura diviso per il valore della seconda misura nel grafico. |

### Esempio: principi fondamentali della funzione Column

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata Example.
- I seguenti campi nella tabella dati:
  - Customer
  - Product
  - UnitsSales
  - UnitPrice

#### Script di caricamento

Example:

```
LOAD * inline [
Customer|Product|UnitsSales|UnitPrice
Astrida|AA|4|16
Astrida|AA|10|15
Astrida|BB|9|9
Betacab|BB|5|10
Betacab|CC|2|20
Betacab|DD||25
Canutility|AA|8|15
Canutility|CC||19
] (delimiter is '|');
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- Customer
- Product
- UnitsSales
- UnitPrice

Creare le seguenti misure:

- =Sum(UnitPrice\*Unitsales), per individuare le vendite annuali.
- =Sum(TOTAL UnitPrice\*Unitsales), per individuare il valore totale delle vendite.
- =100\*Column(1)/Column(2), per calcolare la percentuale (%) delle vendite totali.

Tabella dei risultati

| Customer   | Product | UnitPrice | UnitSales | Sum (UnitPrice*UnitSales) | Sum(TOTAL UnitPrice*UnitSales) | 100*Column (1)/Column (2) |
|------------|---------|-----------|-----------|---------------------------|--------------------------------|---------------------------|
| Totals     |         |           |           | 505                       | 505                            | 100.00                    |
| Astrida    | AA      | 15        | 10        | 150                       | 505                            | 29.70                     |
| Astrida    | AA      | 16        | 4         | 64                        | 505                            | 12.67                     |
| Astrida    | BB      | 9         | 9         | 81                        | 505                            | 16.04                     |
| Betacab    | BB      | 10        | 5         | 50                        | 505                            | 9.90                      |
| Betacab    | CC      | 20        | 2         | 40                        | 505                            | 7.92                      |
| Betacab    | DD      | 25        |           | 0                         | 505                            | 0.00                      |
| Canutility | AA      | 15        | 8         | 120                       | 505                            | 23.76                     |
| Canutility | CC      | 19        |           | 0                         | 505                            | 0.00                      |

I risultati mostrano che è possibile analizzare i dati facendo riferimento alla posizione di una colonna e riutilizzando quella logica in un'espressione di misura. Per esempio, la misura  $100 * \text{column}(1) / \text{column}(2)$  prende il valore della prima misura,  $\text{Sum}(\text{UnitPrice} * \text{UnitSales})$ , lo divide per il valore della seconda misura,  $\text{Sum}(\text{TOTAL UnitPrice} * \text{UnitSales})$ , e poi moltiplica il risultato per 100.

### Esempio: scenario per l'applicazione della funzione Column

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Month`
  - `Sales`
  - `Expenses`
  - `Profit`

#### Script di caricamento

Example:

```
LOAD * INLINE [
Month, Sales, Expenses, Profit
Jan, 10000, 4000, 6000
```

```
Feb, 12000, 4500, 7500
Mar, 15000, 5000, 10000
Apr, 11000, 4200, 6800
May, 13000, 4800, 8200
Jun, 14000, 4700, 9300
Jul, 16000, 5100, 10900
Aug, 12500, 4300, 8200
Sep, 15500, 4900, 10600
Oct, 13500, 4600, 8900
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- Product

Creare le seguenti misure:

- =Sum(Sales), per aggregare le vendite.
- =Sum(Expenses), per aggregare le vendite.
- =Column(1) - Column(2), per calcolare la differenza tra le vendite e la colonna delle spese.

Tabella dei risultati

| Month         | Sum(Sales)    | Sum(Expenses) | Column(1) - Column(2) |
|---------------|---------------|---------------|-----------------------|
| <b>Totals</b> | <b>132500</b> | <b>46100</b>  | <b>86400</b>          |
| Jan           | 10000         | 4000          | 6000                  |
| Feb           | 12000         | 4500          | 7500                  |
| Mar           | 15000         | 5000          | 10000                 |
| Apr           | 11000         | 4200          | 6800                  |
| May           | 13000         | 4800          | 8200                  |
| Jun           | 14000         | 4700          | 9300                  |
| Jul           | 16000         | 5100          | 10900                 |
| Aug           | 12500         | 4300          | 8200                  |
| Sep           | 15500         | 4900          | 8200                  |
| Oct           | 13500         | 4600          | 8900                  |

I risultati della tabella mostrano come utilizzare la funzione `column` per eseguire l'analisi dei dati. La misura `column(1) - column(2)` restituisce l'importo delle vendite delle giacenze dopo aver sottratto le spese (la seconda colonna della misura).

### Esempio: scenario per la funzione Column che analizza i dati di vendita

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata Example.
- I seguenti campi nella tabella dati:
  - ProductName
  - Q1\_Sales
  - Q2\_Sales
  - Q3\_Sales
  - Q4\_Sales

#### Script di caricamento

Example:

```
LOAD * INLINE [  
  ProductName, Q1_Sales, Q2_Sales, Q3_Sales, Q4_Sales  
  Product A, 1200, 1500, 1100, 1300  
  Product B, 1400, 1600, 1200, 1450  
  Product C, 1350, 1700, 1150, 1550  
  Product D, 1250, 1550, 1300, 1500  
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- ProductName

Creare le seguenti misure:

- =Sum(Q1\_Sales), per aggregare le vendite del primo trimestre.
- =Sum(Q2\_Sales), per aggregare le vendite del secondo trimestre.
- =Sum(Q3\_Sales), per aggregare le vendite del terzo trimestre.
- =Sum(Q4\_Sales), per aggregare le vendite del quarto trimestre.
- =Column(1) - Column(2), per calcolare il rapporto tra Q1\_Sales e Q2\_Sales.
- =(Column(2) - Column(1)) / Column(1) per calcolare i tassi di crescita tra due periodi.

Tabella dei risultati

| ProductName | Sum (Q1_Sales) | Sum (Q2_Sales) | Sum (Q3_Sales) | Sum (Q4_Sales) | Column(1) - Column(2) | (Column(2) - Column(1)) / Column(1) |
|-------------|----------------|----------------|----------------|----------------|-----------------------|-------------------------------------|
| Totals      | 5200           | 6350           | 4750           | 5800           | 0.81889763779528      | 0.22115384615385                    |
| Product A   | 1200           | 1500           | 1100           | 1300           | 0.8                   | 0.25                                |
| Product B   | 1400           | 1600           | 1200           | 1450           | 0.875                 | 0.14285714285714                    |
| Product C   | 1350           | 1700           | 1150           | 1550           | 0.79411764705882      | 0.25925925925926                    |
| Product D   | 1250           | 1550           | 1300           | 1500           | 0.80645161290323      | 0.24                                |

I risultati mostrano come è possibile utilizzare la funzione `column` per analizzare i dati.

## Dimensionality - funzione per grafici

**Dimensionality()** restituisce il numero di dimensioni per la riga attuale. Nel caso delle tabelle pivot la funzione restituisce il numero totale di colonne di dimensione senza contenuti di aggregazione, ossia senza somme parziali o aggregati compressi.

### Sintassi:

```
Dimensionality ( )
```

**Tipo di dati restituiti:** numero intero

### Limiti:

Questa funzione è disponibile solo nei grafici. Per tutti gli altri tipi di grafici, eccetto le tabelle pivot, restituirà il numero di dimensioni in tutte le righe eccetto il totale, che sarà 0.

L'ordinamento sui valori y nei grafici o l'ordinamento per colonne di espressione nelle tabelle non è consentito quando si utilizza questa funzione di grafico in una qualsiasi delle espressioni del grafico. Queste alternative di ordinamento verranno perciò disattivate automaticamente. Quando si utilizza questa funzione di grafico in una visualizzazione o in una tabella, l'ordinamento della visualizzazione torna all'ordinamento inserito in questa funzione.

## Esempio: espressione del grafico che utilizza la Dimensionalità

Espressione del grafico

### Panoramica

Usare la funzione `Dimensionality` in una tabella pivot come espressione del grafico dove si vuole applicare una formattazione diversa delle celle a seconda del numero di dimensioni in una riga che ha dati non aggregati. Questo esempio applica un colore di sfondo alle celle della tabella che corrispondono a una data condizione.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Country`
  - `Product`
  - `Sales`
  - `Budget`

### Script di caricamento

Example:

```
Load * inline [  
Country, Product, Sales, Budget  
Sweden, AA, 100000, 50000  
Germany, AA, 125000, 175000  
Canada, AA, 105000, 98000  
Norway, AA, 74850, 68500  
Ireland, AA, 49000, 48000  
Sweden, BB, 98000, 99000  
Germany, BB, 115000, 175000  
Norway, BB, 71850, 68500  
Ireland, BB, 31000, 48000  
] (delimiter is ',');
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella pivot e aggiungere questi campi come dimensioni della riga:

- `Country`
- `Product`

Creare le seguenti misure:

- `=Sum(Sales)`, per calcolare la somma dei valori per `Sales`.
- `=Sum(Budget)`, per calcolare la somma dei valori per `Budget`.
- `=Dimensionality()`, per restituire il numero di dimensioni per la riga attuale.

Nel pannello **Proprietà** della misura `sum(Sales)`, inserire l'espressione seguente come **Espressione colore di sfondo**.

```
=If(Dimensionality()=1 and Sum(Sales)<Sum(Budget),RGB(255,156,156),  
If(Dimensionality()=2 and Sum(Sales)<Sum(Budget),RGB(178,29,29)  
)  
)
```

L'output dell'espressione del colore di sfondo applica uno sfondo rosso chiaro o rosso scuro alle celle della tabella quando vengono soddisfatte le condizioni applicabili.

Tabella pivot che mostra i risultati per le dimensioni di  
riga: Country, Product

| Country     |            |             |                  |
|-------------|------------|-------------|------------------|
| Product     |            |             |                  |
|             | Sum(Sales) | Sum(Budget) | Dimensionality() |
| (+) Canada  | 105000     | 98000       | 1                |
| (-) Germany | 240000     | 350000      | 1                |
| AA          | 125000     | 175000      | 2                |
| BB          | 115000     | 175000      | 2                |
| (+) Ireland | 80000      | 96000       | 1                |
| (+) Norway  | 146700     | 137000      | 1                |
| (+) Sweden  | 198000     | 149000      | 1                |

Quando viene visualizzata la prima dimensione della riga (Country), la funzione `Dimensionality` restituisce 1. Quando si espande una dimensione di riga, come Germany, per visualizzare la seconda dimensione di riga (Product), la funzione restituisce 2.

Le espressioni del colore di sfondo per la misura `sum(Sales)` eseguono calcoli basati sulla dimensionalità e sulle vendite per evidenziare i diversi risultati. Quando la dimensionalità è 1, qualsiasi valore Country con `sum(Sales)` inferiore a `sum(Budget)` appare in rosso chiaro, in questo esempio, Germany e Ireland. Quando la dimensionalità è 2, qualsiasi valore Product con `sum(Sales)` inferiore a `sum(Budget)` appare in rosso scuro, in questo esempio, AA e BB per Germany.

## Esempio: scenario per l'applicazione della funzione Dimensionality

Espressione del grafico

### Panoramica

Questo esempio utilizza la funzione `Dimensionality` per applicare i dettagli a un report ed evidenziare le celle della tabella per alcune dimensioni.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Region`
  - `SalesProduct`
  - `Year`
  - `SalesValue`

- Quantity
- Product

### Script di caricamento

Example:

```
LOAD * INLINE [  
Region, SalesProduct, Year, SalesValue, Quantity, Profit  
North, Laptop, 2023, 20000, 15, 4000  
North, Mobile, 2023, 15000, 25, 3000  
North, Laptop, 2022, 18000, 20, 3500  
North, Mobile, 2022, 14000, 22, 2800  
South, Laptop, 2023, 22000, 18, 4500  
South, Mobile, 2023, 16000, 27, 3200  
South, Laptop, 2022, 19000, 19, 3800  
South, Mobile, 2022, 15000, 23, 3100  
East, Laptop, 2023, 21000, 16, 4200  
East, Mobile, 2023, 15500, 24, 3100  
East, Laptop, 2022, 18500, 18, 3900  
East, Mobile, 2022, 14500, 21, 2950  
West, Laptop, 2023, 23000, 17, 4700  
West, Mobile, 2023, 16500, 26, 3300  
West, Laptop, 2022, 19500, 20, 4000  
West, Mobile, 2022, 15500, 24, 3100  
North, Tablet, 2023, 12000, 10, 2500  
South, Tablet, 2023, 13000, 12, 2700  
East, Tablet, 2023, 12500, 11, 2600  
West, Tablet, 2023, 13500, 13, 2800  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella pivot e aggiungere questi campi come dimensioni della riga:

- Year
- SalesProduct
- Region

Creare le seguenti misure:

- =If(Dimensionality()=1, 'Total', 'Detail'), per indicare se si tratta di una riga Total o Detail.
- =If(Dimensionality()=3, 'Region-Level', If(Dimensionality()=2, 'Product-Level', 'Summary-Level')), per indicare il livello di dimensione che viene visualizzato.
- =Sum(SalesValue), per calcolare la somma delle vendite.

Nel pannello **Proprietà** della misura sum(SalesValue), inserire l'espressione seguente come **Espressione colore di sfondo**.

```
=If(Dimensionality()=1, LightGray())
```

L'output dell'espressione del colore di sfondo applica uno sfondo grigio chiaro alle celle della tabella quando il valore per la dimensionalità è 1.

Tabella pivot che mostra i risultati per le dimensioni di riga: Country, Product

| Anno                      |                                                  |                                                                                                                 |                     |
|---------------------------|--------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|---------------------|
| SalesProduct              |                                                  |                                                                                                                 |                     |
| Area geografica           |                                                  |                                                                                                                 |                     |
|                           | If(Dimensionality()<br>=1, 'Total',<br>'Detail') | If(Dimensionality()<br>=3, 'Region-Level', If<br>(Dimensionality()<br>=2, 'Product-Level',<br>'Summary-Level')) | Sum<br>(SalesValue) |
| (+) 2022                  | Totale                                           | Livello di riepilogo                                                                                            | 134000              |
| (-) 2023                  | Totale                                           | Livello di riepilogo                                                                                            | 200000              |
| (-)Laptop                 | Dettaglio                                        | A livello di prodotto                                                                                           | 86000               |
| Est                       | Dettaglio                                        | Livello di regione                                                                                              | 21000               |
| Nord                      | Dettaglio                                        | Livello di regione                                                                                              | 20000               |
| Sud                       | Dettaglio                                        | Livello di regione                                                                                              | 22000               |
| Ovest                     | Dettaglio                                        | Livello di regione                                                                                              | 23000               |
| (-) Dispositivo<br>mobile | Dettaglio                                        | A livello di prodotto                                                                                           | 63000               |
| (-) Tablet                | Dettaglio                                        | A livello di prodotto                                                                                           | 51000               |

È possibile vedere come la funzione `Dimensionality` può essere utilizzata per migliorare i dettagli del report ed evidenziare tutte le dimensioni delle righe di livello superiore.

### Exists

**Exists()** determina se un valore di campo specifico è già stato caricato nel campo nello script di caricamento dei dati. La funzione restituisce TRUE o FALSE, quindi può essere utilizzata nella clausola **where** di un'istruzione **LOAD** o un'istruzione **IF**.



È possibile utilizzare **Not Exists()** per determinare se un valore di campo non è stato caricato, ma occorre prestare attenzione se si usa **Not Exists()** in una clausola **Where**. La funzione **Exists()** verifica sia le tabelle caricate in precedenza che i valori caricati in precedenza nella tabella corrente. Verrà quindi caricata solo la prima occorrenza. Quando viene rilevata la seconda occorrenza, il valore sarà già caricato. Vedere gli esempi per ulteriori informazioni.

#### Sintassi:

```
Exists(field_name [, expr])
```

Tipo di dati restituiti: Boolean

Argomenti:

Argomenti

| Argomento  | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| field_name | <p>Nome del campo in cui si desidera cercare un valore. È possibile utilizzare un nome di campo esplicito senza virgolette.</p> <p>Il campo deve già essere caricato dallo script. Ciò significa che non è possibile fare riferimento a un campo caricato in una clausola che si trova più in basso nello script.</p>                                                                                                                                                                                                                    |
| expr       | <p>Valore del quale si vuole verificare l'esistenza. È possibile utilizzare un valore esplicito o un'espressione che fa riferimento a uno o più campi nell'istruzione LOAD corrente.</p> <div> <i>Non è possibile fare riferimento a campi non inclusi nell'istruzione LOAD corrente.</i></div> <p>Questo argomento è facoltativo. Se viene omissso, la funzione verificherà se il valore di <b>field_name</b> nel record corrente è già esistente.</p> |

Esempi e risultati:

### Esempio 1

Exists (Employee)

Restituisce -1 (True) se il valore di campo **Employee** nel record attuale esiste già in uno qualsiasi dei record letti in precedenza contenenti tale campo.

Le istruzioni Exists (Employee, Employee) e Exists (Employee) sono equivalenti.

### Esempio 2

Exists(Employee, 'Bill')

Restituisce -1 (True) se il valore di campo **'Bill'** viene trovato nel contenuto attuale del campo **Employee**.

### Esempio 3

```
Employees:
LOAD * inline [
Employee|ID|Salary
Bill|001|20000
John|002|30000
Steve|003|35000
] (delimiter is '|');
```

```
Citizens:
Load * inline [
Employee|Address
Bill|New York
Mary|London
Steve|Chicago
Lucy|Madrid
Lucy|Paris
John|Miami
] (delimiter is '|') where Exists (Employee);
```

```
Drop Tables Employees;
```

Viene generata una tabella che può essere usata nella visualizzazione di una tabella con le dimensioni Employee e Address.

La clausola where, where Exists (Employee), significa che nella nuova tabella vengono caricati solo i nomi della tabella Citizens che sono presenti anche in Employees. L'istruzione Drop rimuove la tabella Employees per evitare confusione.

Risultati

| Employee | Address  |
|----------|----------|
| Bill     | New York |
| John     | Miami    |
| Steve    | Chicago  |

### Esempio 4

```
Employees:
Load * inline [
Employee|ID|Salary
Bill|001|20000
John|002|30000
Steve|003|35000
] (delimiter is '|');

Citizens:
Load * inline [
Employee|Address
Bill|New York
Mary|London
Steve|Chicago
Lucy|Madrid
Lucy|Paris
John|Miami
] (delimiter is '|') where not Exists (Employee);

Drop Tables Employees;
```

La clausola `where include not: where not Exists (Employee)`.

Ciò significa che nella nuova tabella vengono caricati solo i nomi della tabella `Citizens` che non sono presenti in `Employees`.

Si noti che sono presenti due valori per Lucy nella tabella `Citizens`, ma solo uno è incluso nella tabella risultante. Quando si carica la prima riga con il valore Lucy, esso viene incluso nel campo `Employee`. Quindi, quando viene controllata la seconda riga, il valore esiste già.

Risultati

| Dipendente | Indirizzo |
|------------|-----------|
| Mary       | London    |
| Lucy       | Madrid    |

### Esempio 5

Questo esempio mostra come caricare tutti i valori.

`Employees:`

```
Load Employee As Name;  
LOAD * inline [  
Employee|ID|Salary  
Bill|001|20000  
John|002|30000  
Steve|003|35000  
] (delimiter is '|');
```

`Citizens:`

```
Load * inline [  
Employee|Address  
Bill|New York  
Mary|London  
Steve|Chicago  
Lucy|Madrid  
Lucy|Paris  
John|Miami  
] (delimiter is '|') where not Exists (Name, Employee);
```

```
Drop Tables Employees;
```

Per ottenere tutti i valori per Lucy è stato necessario effettuare due modifiche:

- È stato inserito un caricamento precedente per la tabella `Employees` in cui `Employee` è stato rinominato `Name`.  
`Load Employee As Name;`
- La condizione `Where` in `Citizens` è stata modificata in:  
`not Exists (Name, Employee)`.

Ciò crea campi per Name e Employee. Quando viene controllata la seconda riga con Lucy, il valore è ancora inesistente in Name.

Risultati

| Dipendente | Indirizzo |
|------------|-----------|
| Mary       | London    |
| Lucy       | Madrid    |
| Lucy       | Paris     |

### FieldIndex

**FieldIndex()** restituisce la posizione del valore **value** nel campo **field\_name** (in ordine di caricamento).

**Sintassi:**

```
FieldIndex(field_name , value)
```

**Tipo di dati restituiti:** numero intero

**Argomenti:**

Argomenti

| Argomento  | Descrizione                                                                                                                                                                                                             |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| field_name | Nome del campo per cui è richiesto l'indice. Ad esempio, la colonna in una tabella. Deve essere specificato come valore di stringa. Questo significa che il nome di campo deve essere racchiuso tra virgolette singole. |
| value      | Il valore del campo <b>field_name</b> .                                                                                                                                                                                 |

**Limiti:**

- Se non è possibile trovare **value** tra i valori del campo **field\_name**, verrà restituito 0.
- L'ordinamento sui valori y nei grafici o l'ordinamento per colonne di espressione nelle tabelle non è consentito quando si utilizza questa funzione di grafico in una qualsiasi delle espressioni del grafico. Queste alternative di ordinamento verranno perciò disattivate automaticamente. Quando si utilizza questa funzione di grafico in una visualizzazione o in una tabella, l'ordinamento della visualizzazione torna all'ordinamento inserito in questa funzione. Questa limitazione non si applica alla funzione di script equivalente.

**Esempi e risultati:**

Gli esempi seguenti utilizzano il campo: **First name** dalla tabella **Names**.

## Esempi e risultati

| Esempi                                                                                                                      | Risultati                                                                                                                                                                                                                                                                                                             |
|-----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Aggiungere i dati campione all'app ed eseguirli.                                                                            | La tabella <b>Names</b> viene caricata come nei seguenti dati campione.                                                                                                                                                                                                                                               |
| Funzione grafico: in una tabella contenente la dimensione First name, aggiungere come misura i valori riportati di seguito. |                                                                                                                                                                                                                                                                                                                       |
| FieldIndex ('First name', 'John')                                                                                           | 1, in quanto 'John' è visualizzato per primo nell'ordine di caricamento del campo <b>First name</b> . Tenere presente che in una casella di filtro <b>John</b> sarebbe stato visualizzato come numero 2 a partire dall'alto, in quanto è ordinato in modo alfabetico e non secondo l'ordine di caricamento.           |
| FieldIndex ('First name', 'Peter')                                                                                          | 4, in quanto <b>FieldIndex()</b> restituisce solo un valore, che corrisponde alla prima occorrenza nell'ordine di caricamento.                                                                                                                                                                                        |
| Funzione di script: dato che la tabella <b>Names</b> è stata caricata, come nei dati campione:                              |                                                                                                                                                                                                                                                                                                                       |
| John1:<br><br>Load FieldIndex('First name', 'John') as MyJohnPos<br><br>Resident Names;                                     | MyJohnPos=1, in quanto 'John' è visualizzato per primo nell'ordine di caricamento del campo <b>First name</b> . Tenere presente che in una casella di filtro <b>John</b> sarebbe stato visualizzato come numero 2 a partire dall'alto, in quanto è ordinato in modo alfabetico e non secondo l'ordine di caricamento. |
| Peter1:<br><br>Load FieldIndex('First name', 'Peter') as MyPeterPos<br><br>Resident Names;                                  | MyPeterPos=4, in quanto <b>FieldIndex()</b> restituisce un solo valore, che corrisponde alla prima occorrenza nell'ordine di caricamento.                                                                                                                                                                             |

Dati utilizzati nell'esempio:

```
Names:
LOAD * inline [
First name|Last name|Initials|Has cellphone
John|Anderson|JA|Yes
Sue|Brown|SB|Yes
Mark|Carr|MC|No
Peter|Devonshire|PD|No
Jane|Elliot|JE|Yes
Peter|Franc|PF|Yes ] (delimiter is '|');
```

```
John1:
Load FieldIndex('First name', 'John') as MyJohnPos
Resident Names;
```

```
Peter1:
```

```
Load FieldIndex('First name','Peter') as MyPeterPos  
Resident Names;
```

## FieldValue

**FieldValue()** restituisce il valore trovato nella posizione **elem\_no** nel campo **field\_name** (in ordine di caricamento).

### Sintassi:

```
FieldValue(field_name , elem_no)
```

**Tipo di dati restituiti:** duale

### Argomenti:

#### Argomenti

| Argomento  | Descrizione                                                                                                                                                                                                                                     |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| field_name | Nome del campo per cui è richiesto il valore. Ad esempio, la colonna in una tabella. Deve essere specificato come valore di stringa. Questo significa che il nome di campo deve essere racchiuso tra virgolette singole.                        |
| elem_no    | Il numero della posizione (elemento) del campo, che segue l'ordine di caricamento, per cui viene restituito il valore. Ciò potrebbe corrispondere alla riga in una tabella, anche se dipende dall'ordine di caricamento degli elementi (righe). |

### Limiti:

- Se **elem\_no** è maggiore del numero di valori del campo, viene restituito NULL .
- L'ordinamento sui valori y nei grafici o l'ordinamento per colonne di espressione nelle tabelle non è consentito quando si utilizza questa funzione di grafico in una qualsiasi delle espressioni del grafico. Queste alternative di ordinamento verranno perciò disattivate automaticamente. Quando si utilizza questa funzione di grafico in una visualizzazione o in una tabella, l'ordinamento della visualizzazione torna all'ordinamento inserito in questa funzione. Questa limitazione non si applica alla funzione di script equivalente.

### Esempio

#### Script di caricamento

Caricare i dati seguenti come un caricamento inline nell'editor di caricamento dati per creare l'esempio in basso.

Names :

```
LOAD * inline [  
First name|Last name|Initials|Has cellphone  
John|Anderson|JA|Yes  
Sue|Brown|SB|Yes  
Mark|Carr|MC |No  
Peter|Devonshire|PD|No
```

```
Jane|Elliot|JE|Yes  
Peter|Franc|PF|Yes ] (delimiter is '|');
```

John1:

```
Load FieldValue('First name',1) as MyPos1  
Resident Names;
```

Peter1:

```
Load FieldValue('First name',5) as MyPos2  
Resident Names;
```

### Creazione di una visualizzazione

Creare una visualizzazione di una tabella in un foglio di Qlik Sense. Aggiungere i campi **First name**, **MyPos1** e **MyPos2** alla tabella.

### Risultato

| First name | MyPos1 | MyPos2 |
|------------|--------|--------|
| Jane       | John   | Jane   |
| John       | John   | Jane   |
| Mark       | John   | Jane   |
| Peter      | John   | Jane   |
| Sue        | John   | Jane   |

### Spiegazione

**FieldValue('First name','1')** restituisce John come valore per **MyPos1** per tutti i nomi perché John appare prima nell'ordine di caricamento del campo **Nome**. Tenere presente che in una casella di filtro John sarebbe stato visualizzato come numero 2 a partire dall'alto, dopo Jane, in quanto è ordinato in modo alfabetico e non secondo l'ordine di caricamento.

**FieldValue('First name','5')** restituisce Jane come valore per **MyPos2** per tutti i nomi, dato che Jane appare quinta nell'ordine di caricamento del campo **First name**.

### FieldValueCount

**FieldValueCount()** è una funzione **integer** che restituisce il numero di valori distinti di un campo.

Un ricaricamento parziale può rimuovere valori dai dati, che non si rifletteranno nel numero restituito. Il numero restituito corrisponderà a tutti i valori distinti che sono stati caricati nel ricaricamento iniziale o in qualsiasi ricaricamento parziale successivo.



*L'ordinamento sui valori y nei grafici o l'ordinamento per colonne di espressione nelle tabelle non è consentito quando si utilizza questa funzione di grafico in una qualsiasi delle espressioni del grafico. Queste alternative di ordinamento verranno perciò disattivate automaticamente. Quando si utilizza questa funzione di grafico in una visualizzazione o in una tabella, l'ordinamento della visualizzazione torna all'ordinamento inserito in questa funzione. Questa limitazione non si applica alla funzione di script equivalente.*

#### Sintassi:

```
FieldValueCount(field_name)
```

**Tipo di dati restituiti:** numero intero

#### Argomenti:

##### Argomenti

| Argomento               | Descrizione                                                                                                                                                                                                              |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>field_name</code> | Nome del campo per cui è richiesto il valore. Ad esempio, la colonna in una tabella. Deve essere specificato come valore di stringa. Questo significa che il nome di campo deve essere racchiuso tra virgolette singole. |

#### Esempi e risultati:

Gli esempi seguenti utilizzano il campo **First name** dalla tabella **Names**.

##### Esempi e risultati

| Esempi                                                                                                                      | Risultati                                                                  |
|-----------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------|
| Aggiungere i dati campione all'app ed eseguirli.                                                                            | La tabella <b>Names</b> viene caricata come nei seguenti dati campione.    |
| Funzione grafico: in una tabella contenente la dimensione First name, aggiungere come misura i valori riportati di seguito. |                                                                            |
| <code>FieldValueCount('First name')</code>                                                                                  | 5 in quanto <b>Peter</b> è visualizzato due volte.                         |
| <code>FieldValueCount('Initials')</code>                                                                                    | 6 in quanto <b>Initials</b> presenta solo due valori distinti.             |
| Funzione di script: dato che la tabella <b>Names</b> è stata caricata, come nei dati campione:                              |                                                                            |
| <pre>FieldCount1: Load FieldValueCount('First name') as MyFieldCount1 Resident Names;</pre>                                 | <code>MyFieldCount1=5</code> , in quanto 'Peter' è visualizzato due volte. |

| Esempi                                                                                         | Risultati                                                                |
|------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------|
| <pre>FieldCount2:  Load FieldValueCount('Initials') as MyInitialsCount1  Resident Names;</pre> | MyFieldCount1=6, in quanto 'Initials' presenta solo due valori distinti. |

Dati utilizzati negli esempi:

Names:

```
LOAD * inline [
First name|Last name|Initials|Has cellphone
John|Anderson|JA|Yes
Sue|Brown|SB|Yes
Mark|Carr|MC|No
Peter|Devonshire|PD|No
Jane|Elliot|JE|Yes
Peter|Franc|PF|Yes ] (delimiter is '|');
```

FieldCount1:

```
Load FieldValueCount('First name') as MyFieldCount1
Resident Names;
```

FieldCount2:

```
Load FieldValueCount('Initials') as MyInitialsCount1
Resident Names;
```

## LookUp

**LookUp()** cerca in una tabella già caricata e restituisce il valore di **field\_name** corrispondente alla prima occorrenza del valore **match\_field\_value** nel campo **match\_field\_name**. La tabella può essere la tabella attuale o un'altra tabella caricata in precedenza.

**Sintassi:**

```
LookUp (field_name, match_field_name, match_field_value [, table_name])
```

**Tipo di dati restituiti:** duale

**Argomenti:**

Argomenti

| Argomento                | Descrizione                                                                                                                                                          |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>field_name</b>        | Nome del campo per cui è richiesto il valore restituito. Il valore di input deve essere specificato come stringa (ad esempio, un valore letterale tra virgolette).   |
| <b>match_field_name</b>  | Nome del campo in cui ricercare <b>match_field_value</b> . Il valore di input deve essere specificato come stringa (ad esempio, un valore letterale tra virgolette). |
| <b>match_field_value</b> | Valore da ricercare nel campo <b>match_field_name</b> .                                                                                                              |

| Argomento         | Descrizione                                                                                                                                                                                                                                       |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>table_name</b> | Nome della tabella in cui ricercare il valore. Il valore di input deve essere specificato come stringa (ad esempio un valore letterale tra virgolette).<br><br>Se il valore <b>table_name</b> viene omissso, verrà utilizzata la tabella attuale. |



*Gli argomenti non racchiusi tra virgolette fanno riferimento alla tabella attuale. Per fare riferimento ad altre tabelle, racchiudere l'argomento tra virgolette singole.*

### Limiti:

L'ordine in cui viene eseguita la ricerca è l'ordine di caricamento, a meno che la tabella non sia il risultato di operazioni complesse come unioni, nel qual caso l'ordine non sarà definito in modo preciso. Sia **field\_name** che **match\_field\_name** devono essere campi della stessa tabella, specificata da **table\_name**.

Se non viene trovata una corrispondenza, viene restituito il valore NULL.

### Esempio

#### Script di caricamento

Caricare i dati seguenti come un caricamento inline nell'editor di caricamento dati per creare l'esempio in basso.

```
ProductList:
Load * Inline [
ProductID|Product|Category|Price
1|AA|1|1
2|BB|1|3
3|CC|2|8
4|DD|3|2
] (delimiter is '|');

OrderData:
Load *, Lookup('Category', 'ProductID', ProductID, 'ProductList') as CategoryID
Inline [
InvoiceID|CustomerID|ProductID|Units
1|Astrida|1|8
1|Astrida|2|6
2|Betacab|3|10
3|Divadip|3|5
4|Divadip|4|10
] (delimiter is '|');

Drop Table ProductList;
```

#### Creazione di una visualizzazione

Creare una visualizzazione di una tabella in un foglio di Qlik Sense. Aggiungere i campi **ProductID**, **InvoiceID**, **CustomerID**, **Units** e **CategoryID** alla tabella.

### Risultato

Tabella risultante

| ProductID | InvoiceID | CustomerID | Units | CategoryID |
|-----------|-----------|------------|-------|------------|
| 1         | 1         | Astrida    | 8     | 1          |
| 2         | 1         | Astrida    | 6     | 1          |
| 3         | 2         | Betacab    | 10    | 2          |
| 3         | 3         | Divadip    | 5     | 2          |
| 4         | 4         | Divadip    | 10    | 3          |

### Spiegazione

I dati campione utilizzano la funzione **Lookup()** con il seguente formato:

```
Lookup('Category', 'ProductID', ProductID, 'ProductList')
```

La tabella **ProductList** viene caricata per prima.

La funzione **Lookup()** viene utilizzata per creare la tabella **OrderData**. Specifica il terzo argomento come **ProductID**. Si tratta del campo per cui il valore deve essere ricercato nel secondo argomento **'ProductID'** in **ProductList**, come indicato dalle virgolette singole che lo racchiudono.

La funzione restituisce il valore per **'Category'** (nella tabella **ProductList**), caricata come **CategoryID**.

L'istruzione **drop** elimina la tabella **ProductList** dal modello dati poiché non è richiesta, il che lascia la tabella **OrderData** risultante.



*La funzione **Lookup()** è flessibile e può accedere a qualsiasi tabella caricata in precedenza. Tuttavia, risulta più lenta rispetto alla funzione **Applymap()**.*

### Vedere anche:

p [ApplyMap \(page 1567\)](#)

## NoOfRows

**NoOfRows()** restituisce il numero di righe nel segmento di colonna attuale in una tabella. Per i grafici bitmap, **NoOfRows()** restituisce il numero di righe nell'equivalente della tabella lineare del grafico.

Se la tabella o l'equivalente di tabella include più dimensioni verticali, il segmento colonna attuale includerà solo righe contenenti gli stessi valori della riga attuale in tutte le colonne di dimensione, eccetto la colonna che mostra l'ultima dimensione nell'ordinamento tra campi.



*L'ordinamento sui valori y nei grafici o l'ordinamento per colonne di espressione nelle tabelle non è consentito quando si utilizza questa funzione di grafico in una qualsiasi delle espressioni del grafico. Queste alternative di ordinamento verranno perciò disattivate automaticamente. Quando si utilizza questa funzione di grafico in una visualizzazione o in una tabella, l'ordinamento della visualizzazione torna all'ordinamento inserito in questa funzione.*

#### Sintassi:

**NoOfRows ( [TOTAL] )**

**Tipo di dati restituiti:** numero intero

#### Argomenti:

##### Argomenti

| Argomento    | Descrizione                                                                                                                                                        |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>TOTAL</b> | Se la tabella è unidimensionale o se è utilizzato il qualificatore <b>TOTAL</b> come argomento, il segmento colonna attuale sarà sempre uguale all'intera colonna. |

### Esempio: espressione del grafico che utilizza NoOfRows

Esempio - Espressione del grafico

#### Script di caricamento

Caricare i dati seguenti come un caricamento inline nell'editor caricamento dati per creare gli esempi di espressione del grafico in basso.

```
Temp:
LOAD * inline [
Region|SubRegion|RowNo()|NoOfRows()
Africa|Eastern
Africa|Western
Americas|Central
Americas|Northern
Asia|Eastern
Europe|Eastern
Europe|Northern
Europe|Western
Oceania|Australia
] (delimiter is '|');
```

#### Espressione del grafico

Creare la visualizzazione di una tabella in un foglio di Qlik Sense con **Region** e **SubRegion** come dimensioni. Aggiungere **RowNo()**, **NoOfRows()** e **NoOfRows(Total)** come misure.

## Risultato

| Region   | SubRegion | RowNo() | NoOfRows() | NoOfRows (Total) |
|----------|-----------|---------|------------|------------------|
| Africa   | Eastern   | 1       | 2          | 9                |
| Africa   | Western   | 2       | 2          | 9                |
| Americas | Central   | 1       | 2          | 9                |
| Americas | Northern  | 2       | 2          | 9                |
| Asia     | Eastern   | 1       | 1          | 9                |
| Europe   | Eastern   | 1       | 3          | 9                |
| Europe   | Northern  | 2       | 3          | 9                |
| Europe   | Western   | 3       | 3          | 9                |
| Oceania  | Australia | 1       | 1          | 9                |

## Spiegazione

In questo esempio, il criterio di ordinamento è per la prima dimensione, Regione. Di conseguenza, ciascun segmento colonna è formato da un gruppo di regioni che presenta lo stesso valore, ad esempio, Africa.

La colonna **RowNo()** mostra i numeri di riga per ciascun segmento colonna, ad esempio, sono presenti due righe per la regione Africa. La numerazione delle righe riparte da 1 per il segmento di colonna successivo, vale a dire Americas.

La colonna **NoOfRows()** conteggia il numero di righe in ciascun segmento colonna, ad esempio, Europa ha tre righe nel segmento colonna.

La colonna **NoOfRows(Total)** ignora le dimensioni a causa dell'argomento **TOTAL** per **NoOfRows()** e conteggia le righe nella tabella.

Se la tabella è stata ordinata in base alla seconda dimensione, SubRegion, i segmenti colonna si baserebbero su tale dimensione, pertanto la numerazione delle righe cambierebbe per ciascun SubRegion.

## Vedere anche:

per [RowNo - funzione per grafici \(page 696\)](#)

## Peek

**Peek()** restituisce il valore di un campo in una tabella per una riga che è già stata caricata. È possibile specificare il numero di riga così come la tabella. Se non viene specificato alcun numero di riga, verrà utilizzato l'ultimo record precedentemente caricato.

La funzione `peek()` viene principalmente utilizzata per trovare i limiti pertinenti in una tabella precedentemente caricata, ovvero, il primo valore o l'ultimo valore di un campo specifico. Nella maggior parte dei casi, questo valore viene archiviato in una variabile per l'utilizzo futuro, ad esempio, come una condizione in un loop `do-while`.

### Sintassi:

**Peek (**

`field_name`

**[, row\_no[, table\_name ] ])**

**Tipo di dati restituiti:** duale

### Argomenti:

#### Argomenti

| Argomento               | Descrizione                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>field_name</code> | Nome del campo per cui è richiesto il valore restituito. Il valore di input deve essere specificato come stringa (ad esempio, un valore letterale tra virgolette).                                                                                                                                                                                                              |
| <code>row_no</code>     | La riga nella tabella che specifica il campo richiesto. Può essere un'espressione, tuttavia deve restituire un numero intero. 0 indica il primo record, 1 indica il secondo e così via. I numeri negativi indicano l'ordine dalla fine della tabella. -1 indica l'ultimo record letto.<br><br>Se non viene dichiarato alcun valore <b>row_no</b> verrà utilizzato il valore -1. |
| <code>table_name</code> | Un'etichetta di tabella senza due punti finali. Se non è dichiarato <b>table_name</b> , viene utilizzata la tabella attuale. Se utilizzato al di fuori dell'istruzione <b>LOAD</b> o se fa riferimento a un'altra tabella, è necessario includere <b>table_name</b> .                                                                                                           |

### Limiti:

La funzione può solo restituire valori da record già caricati. Questo significa che nel primo record di una tabella, una chiamata che usa -1 come `row_no` restituirà NULL.

### Esempi e risultati:

#### Esempio 1

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

```
EmployeeDates:
Load * Inline [
EmployeeCode|StartDate|EndDate
101|02/11/2010|23/06/2012
102|01/11/2011|30/11/2013
103|02/01/2012|
```

```
104|02/01/2012|31/03/2012
105|01/04/2012|31/01/2013
106|02/11/2013|
] (delimiter is '|');
```

```
First_Last_Employee:
Load
EmployeeCode,
Peek('EmployeeCode',0,'EmployeeDates') As FirstCode,
Peek('EmployeeCode',-1,'EmployeeDates') As LastCode
Resident EmployeeDates;
```

Tabella risultante

| Employee code | StartDate  | EndDate    | FirstCode | LastCode |
|---------------|------------|------------|-----------|----------|
| 101           | 02/11/2010 | 23/06/2012 | 101       | 106      |
| 102           | 01/11/2011 | 30/11/2013 | 101       | 106      |
| 103           | 02/01/2012 |            | 101       | 106      |
| 104           | 02/01/2012 | 31/03/2012 | 101       | 106      |
| 105           | 01/04/2012 | 31/01/2013 | 101       | 106      |
| 106           | 02/11/2013 |            | 101       | 106      |

FirstCode = 101 poiché `Peek('EmployeeCode',0, 'EmployeeDates')` restituisce il primo valore di EmployeeCode nella tabella EmployeeDates.

LastCode = 106 perché `Peek('EmployeeCode',-1, 'EmployeeDates')` restituisce l'ultimo valore di EmployeeCode nella tabella EmployeeDates.

La sostituzione del valore dell'argomento **row\_no** restituisce i valori delle altre righe nella tabella, nel modo seguente:

`Peek('EmployeeCode',2, 'EmployeeDates')` restituisce il terzo valore, 103, nella tabella come FirstCode.

Tuttavia, tenere presente che se non si specifica la tabella come il terzo argomento **table\_name** in questi esempi, la funzione farà riferimento alla tabella attuale (in questo caso, interna).

## Esempio 2

Se si desidera accedere ai dati più in profondità in una tabella, è possibile farlo in due passaggi: prima, caricare l'intera tabella in una tabella temporanea, quindi riordinarla quando si usa **Peek()**.

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

```
T1:
LOAD * inline [
ID|value
1|3
1|4
1|6
3|7
```

```
3|8
2|1
2|11
5|2
5|78
5|13
] (delimiter is '|');
```

T2:

```
LOAD *,
IF(ID=Peek('ID'), Peek('List')&','&Value,Value) AS List
RESIDENT T1
ORDER BY ID ASC;
DROP TABLE T1;
```

Create a table in a sheet in your app with **ID**, **List**, and **Value** as the dimensions.

Tabella risultante

| ID | Elenco  | Valore |
|----|---------|--------|
| 1  | 3,4     | 4      |
| 1  | 3,4,6   | 6      |
| 1  | 3       | 3      |
| 2  | 1,11    | 11     |
| 2  | 1       | 1      |
| 3  | 7,8     | 8      |
| 3  | 7       | 7      |
| 5  | 2,78    | 78     |
| 5  | 2,78,13 | 13     |
| 5  | 2       | 2      |

L'istruzione **IF()** viene creata a partire dalla tabella temporanea T1.

`Peek('ID')` fa riferimento al campo ID nella riga precedente nella tabella corrente T2.

`Peek('List')` fa riferimento al campo List nella riga precedente nella tabella T2, attualmente in fase di creazione mentre l'espressione viene valutata.

L'istruzione viene valutata nel seguente modo:

Se il valore attuale di ID è identico al valore precedente di ID, scrivere il valore di `Peek('List')` concatenandolo con il valore corrente di Value. In alternativa, scrivere solo il valore attuale di Value.

Se `Peek('List')` contiene già un risultato concatenato, il nuovo risultato di `Peek('List')` verrà concatenato a tale risultato.



Prendere nota della clausola **Order by**. Questa clausola specifica il metodo di ordinamento della tabella (in base all'ID in ordine crescente). Senza questa clausola, la funzione Peek() utilizzerebbe qualsiasi ordinamento arbitrario presente nella tabella interna, portando a risultati potenzialmente imprevedibili.

### Esempio 3

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

```
Amounts:
Load
Date#(Month,'YYYY-MM') as Month,
Amount,
Peek(Amount) as AmountMonthBefore
Inline
[Month,Amount
2022-01,2
2022-02,3
2022-03,7
2022-04,9
2022-05,4
2022-06,1];
```

Tabella risultante

| Amount | AmountMonthBefore | Month   |
|--------|-------------------|---------|
| 1      | 4                 | 2022-06 |
| 2      | -                 | 2022-01 |
| 3      | 2                 | 2022-02 |
| 4      | 9                 | 2022-05 |
| 7      | 3                 | 2022-03 |
| 9      | 7                 | 2022-04 |

Il campo AmountMonthBefore conterrà l'importo dal mese precedente.

Qui i parametri row\_no e table\_name vengono omessi, pertanto vengono utilizzati i valori predefiniti. In questo esempio, le seguenti tre chiamate di funzione sono equivalenti:

- Peek(Amount)
- Peek(Amount,-1)
- Peek(Amount,-1,'Amounts')

L'utilizzo di -1 come row\_no indica che verrà utilizzato il valore dalla riga precedente. Sostituendo tale valore, sarà possibile recuperare i valori delle altre righe nella tabella:

Peek(Amount,2) restituisce il terzo valore nella tabella: 7.

### Esempio 4

I dati devono essere ordinati correttamente per ottenere i risultati corretti ma, sfortunatamente, questo non è sempre il caso. Inoltre, la funzione Peek() non può essere utilizzata per fare riferimento a dati che non sono ancora stati caricati. Utilizzando tabelle temporanee ed eseguendo più passaggi attraverso i dati, questi problemi possono essere evitati.

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

```
tmp1Amounts:
Load * Inline
[Month,Product,Amount
2022-01,B,3
2022-01,A,8
2022-02,B,4
2022-02,A,6
2022-03,B,1
2022-03,A,6
2022-04,A,5
2022-04,B,5
2022-05,B,6
2022-05,A,7
2022-06,A,4
2022-06,B,8];

tmp2Amounts:
Load *,
If(Product=Peek(Product),Peek(Amount)) as AmountMonthBefore
Resident tmp1Amounts
Order By Product, Month Asc;
Drop Table tmp1Amounts;

Amounts:
Load *,
If(Product=Peek(Product),Peek(Amount)) as AmountMonthAfter
Resident tmp2Amounts
Order By Product, Month Desc;
Drop Table tmp2Amounts;
```

### Spiegazione

La tabella iniziale è ordinata in base al mese, il che significa che la funzione peek() restituirebbe in molti casi l'importo del prodotto sbagliato. Quindi, questa tabella deve essere riordinata. Ciò avviene eseguendo un secondo passaggio attraverso i dati creando una nuova tabella tmp2Amounts. Prendere nota della clausola Order by. Ordina i record prima per prodotto, poi per mese in ordine crescente.

La funzione If() è necessaria poiché AmountMonthBefore dovrebbe essere calcolato solo se la riga precedente contiene i dati per lo stesso prodotto ma per il mese precedente. Confrontando il prodotto sulla riga corrente con il prodotto sulla riga precedente, questa condizione può essere convalidata.

Quando viene creata la seconda tabella, la prima tabella tmp1Amounts viene eliminata utilizzando un'istruzione Drop Table.

Infine, viene fatto un terzo passaggio attraverso i dati, ma ora con i mesi ordinati in ordine inverso. In questo modo, può essere calcolato anche AmountMonthAfter.



*Le clausole Ordina per specificano come è ordinata la tabella; senza di loro, la funzione Peek() userà qualsiasi ordine arbitrario della tabella interna, il che può portare a risultati imprevedibili.*

### Risultato

Tabella risultante

| Month   | Product | Amount | AmountMonthBefore | AmountMonthAfter |
|---------|---------|--------|-------------------|------------------|
| 2022-01 | A       | 8      | -                 | 6                |
| 2022-02 | B       | 3      | -                 | 4                |
| 2022-03 | A       | 6      | 8                 | 6                |
| 2022-04 | B       | 4      | 3                 | 1                |
| 2022-05 | A       | 6      | 6                 | 5                |
| 2022-06 | B       | 1      | 4                 | 5                |
| 2022-01 | A       | 5      | 6                 | 7                |
| 2022-02 | B       | 5      | 1                 | 6                |
| 2022-03 | A       | 7      | 5                 | 4                |
| 2022-04 | B       | 6      | 5                 | 8                |
| 2022-05 | A       | 4      | 7                 | -                |
| 2022-06 | B       | 8      | 6                 | -                |

### Esempio 5

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

T1:

```
Load * inline [
Quarter, value
2003q1, 10000
2003q1, 25000
2003q1, 30000
2003q2, 1250
2003q2, 55000
2003q2, 76200
2003q3, 9240
2003q3, 33150
2003q3, 89450
```

```
2003q4, 1000
2003q4, 3000
2003q4, 5000
2004q1, 1000
2004q1, 1250
2004q1, 3000
2004q2, 5000
2004q2, 9240
2004q2, 10000
2004q3, 25000
2004q3, 30000
2004q3, 33150
2004q4, 55000
2004q4, 76200
2004q4, 89450 ];
```

T2:

```
Load *, rangesum(SumVal,peek('AccSumVal')) as AccSumVal;
Load Quarter, sum(Value) as SumVal resident T1 group by Quarter;
```

### Risultato

Tabella risultante

| Quarter | SumVal | AccSumVal |
|---------|--------|-----------|
| 2003q1  | 65000  | 65000     |
| 2003q2  | 132450 | 197450    |
| 2003q3  | 131840 | 329290    |
| 2003q4  | 9000   | 338290    |
| 2004q1  | 5250   | 343540    |
| 2004q2  | 24240  | 367780    |
| 2004q3  | 88150  | 455930    |
| 2004q4  | 220650 | 676580    |

### Spiegazione

L'istruzione **LOAD Load \*, rangesum(SumVal,peek('AccSumVal')) as AccSumVal** include una chiamata ricorsiva dove i valori precedenti vengono aggiunti al valore corrente. Questa operazione è utilizzata per calcolare un accumulo di valori nello script.

---

### Vedere anche:

## Previous

**Previous()** restituisce il valore dell'espressione **expr** utilizzando i dati del record di input precedente che non è stato eliminato a causa di una clausola **where**. Nel primo record di una tabella interna, la funzione restituirà NULL.

### Sintassi:

**Previous** (*expr*)

Tipo di dati restituiti: duale

### Argomenti:

| Argomenti   |                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Argomento   | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>expr</b> | L'espressione o il campo contenente i dati da misurare.<br>L'espressione può contenere funzioni <b>Previous()</b> nidificate per accedere ai record precedenti. I dati vengono caricati direttamente dalla sorgente di input, rendendo possibile fare riferimento anche a campi che non sono stati caricati in Qlik Sense, vale a dire persino se non sono stati memorizzati nel relativo database associativo. |

### Limiti:

Nel primo record di una tabella interna, la funzione restituisce NULL.

### Esempio:

Inserire quanto riportato di seguito nello script di caricamento:

sales2013:

```
Load *, (Sales - Previous(Sales) )as Increase Inline [
```

```
Month|Sales
```

```
1|12
```

```
2|13
```

```
3|15
```

```
4|17
```

```
5|21
```

```
6|21
```

```
7|22
```

8|23

9|32

10|35

11|40

12|41

```
] (delimiter is '|');
```

Utilizzando la funzione **Previous()** nell'istruzione **Load**, è possibile confrontare il valore attuale di Sales con il valore precedente e utilizzarlo in un terzo campo, Increase.

Tabella risultante

| Mese | Vendite | Incremento |
|------|---------|------------|
| 1    | 12      | -          |
| 2    | 13      | 1          |
| 3    | 15      | 2          |
| 4    | 17      | 2          |
| 5    | 21      | 4          |
| 6    | 21      | 0          |
| 7    | 22      | 1          |
| 8    | 23      | 1          |
| 9    | 32      | 9          |
| 10   | 35      | 3          |
| 11   | 40      | 5          |
| 12   | 41      | 1          |

### Top - funzione per grafici

**Top()** valuta un'espressione nella prima riga (superiore) di un segmento colonna in una tabella. La riga per la quale viene calcolata dipende dal valore dell'**offset**, se presente, e l'impostazione predefinita è la riga superiore. Per i grafici diversi dalle tabelle, la valutazione di **Top()** viene effettuata sulla prima riga della colonna attuale nell'equivalente di tabella lineare del grafico.

#### Sintassi:

```
Top ([TOTAL] expr [ , offset [,count ]])
```

Tipo di dati restituiti: duale

#### Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                 |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>expr</b>   | L'espressione o il campo contenente i dati da misurare.                                                                                                                                                                                                                                                                                                                     |
| <b>offset</b> | Se si specifica un <b>offset</b> di n maggiore di 1, la valutazione dell'espressione viene spostata in basso di n righe rispetto alla riga superiore.<br><br>Specificando un numero di offset negativo, la funzione <b>Top</b> diventa equivalente alla funzione <b>Bottom</b> con il numero di offset positivo corrispondente.                                             |
| <b>count</b>  | Specificando un terzo parametro <b>count</b> maggiore di 1, la funzione restituirà una scala di valori <b>count</b> , uno per ciascuna delle ultime righe <b>count</b> del segmento di colonna attuale. In questo modulo la funzione può essere utilizzata come argomento per una qualsiasi delle funzioni di scala speciali. <a href="#">Funzioni di scala (page 1584)</a> |
| <b>TOTAL</b>  | Se la tabella è unidimensionale o se è utilizzato il qualificatore <b>TOTAL</b> come argomento, il segmento colonna attuale sarà sempre uguale all'intera colonna.                                                                                                                                                                                                          |



Un segmento di colonna viene definito come un sottogruppo consecutivo di celle con gli stessi valori per le dimensioni nell'ordine attuale. Le funzioni grafiche intra-record vengono calcolate nel segmento colonna escludendo la dimensione più a destra nel grafico della tabella lineare equivalente. Se nel grafico è presente una sola dimensione oppure è specificato il qualificatore **TOTAL**, l'espressione viene valutata nell'intera tabella.



Se la tabella o l'equivalente di tabella include più dimensioni verticali, il segmento colonna attuale includerà solo righe contenenti gli stessi valori della riga attuale in tutte le colonne di dimensione, eccetto la colonna che mostra l'ultima dimensione nell'ordinamento tra campi.

#### Limiti:

- Le chiamate ricorrenti restituiranno NULL.
- L'ordinamento sui valori y nei grafici o l'ordinamento per colonne di espressione nelle tabelle non è consentito quando si utilizza questa funzione di grafico in una qualsiasi delle espressioni del grafico. Queste alternative di ordinamento verranno perciò disattivate automaticamente. Quando si utilizza questa funzione di grafico in una visualizzazione o in una tabella, l'ordinamento della visualizzazione torna all'ordinamento inserito in questa funzione.

### Esempio: espressioni dei grafici

| Esempio           | Risultato                                                                                            |
|-------------------|------------------------------------------------------------------------------------------------------|
| Top( Sum(Sales) ) | Restituisce il valore di sales per la prima riga (in alto) di un segmento di colonna in una tabella. |

### Esempio: principi fondamentali della funzione Top

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata Example.
- I seguenti campi nella tabella dati:
  - Year
  - Sales

#### Script di caricamento

```
Example:  
LOAD * INLINE [  
Year, Sales  
2019, 1000  
2020, 1500  
2021, 1200  
2022, 1800  
2023, 1700  
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- Year

Creare le seguenti misure:

- $\text{=Sum(Sales)}$ , per individuare le vendite annuali.
- $\text{=Top(Sum(Sales))}$ , per individuare il valore delle vendite per il primo anno, ovvero la prima riga (in alto) di una colonna in una tabella.
- $\text{=Sum(Sales) - Top(Sum(Sales))}$ , per calcolare la varianza tra il valore sales dell'anno in corso e il valore per la prima riga (in alto) di una colonna.
- $\text{=Sum(Sales) - Top(Sum(Sales), 3)}$ , per calcolare la varianza tra il valore delle vendite della riga corrente e il valore della terza riga dall'alto.
- $\text{=(Sales - Top(Sales)) / Top(Sales)}$ , per restituire la variazione di sales come rapporto/percentuale in confronto alla prima riga. Per mostrare questo valore come una percentuale, sotto la voce **Formattazione numero**, selezionare **Numero > Formattazione semplice > 12,34%**.

Ordinare la tabella per la dimensione year in ordine crescente.

Tabella dei risultati

| Year   | Sum (Sales) | Top(Sum (Sales)) | Sum(Sales) - Top (Sum(Sales)) | Sum(Sales)-Top (Sum(Sales), 3) | (Sales - Top (Sales)) / Top (Sales) |
|--------|-------------|------------------|-------------------------------|--------------------------------|-------------------------------------|
| Totals | 7200        | 1000             | 6200                          | 6000                           | -                                   |
| 2019   | 1000        | 1000             | 0                             | -200                           | 0.00%                               |
| 2020   | 1500        | 1000             | 500                           | 300                            | 50.00%                              |
| 2021   | 1200        | 1000             | 200                           | 0                              | 20.00%                              |
| 2022   | 1800        | 1000             | 800                           | 600                            | 80.00%                              |
| 2023   | 1700        | 1000             | 700                           | 500                            | 70.00%                              |

Osservando i risultati, è possibile confrontare i valori per sales nel periodo corrente con il periodo superiore (o primo).

- La misura  $\text{sum(Sales) - Top(Sum(Sales))}$  restituisce la variazione nei valori di sales, confrontando il periodo corrente con il primo periodo.
- La misura  $\text{sum(Sales) - Top(Sum(Sales), 3)}$  restituisce la somma di sales per la riga corrente meno il terzo periodo dall'alto.
- La misura  $\text{=(Sales - Top(Sales)) / Top(Sales)}$  restituisce la variazione di sales in confronto al primo periodo come una percentuale.



*Questa funzione può essere utilizzata nei grafici diversi dalle tabelle, ad esempio nei grafici a barre.*

**Procedere come indicato di seguito:**

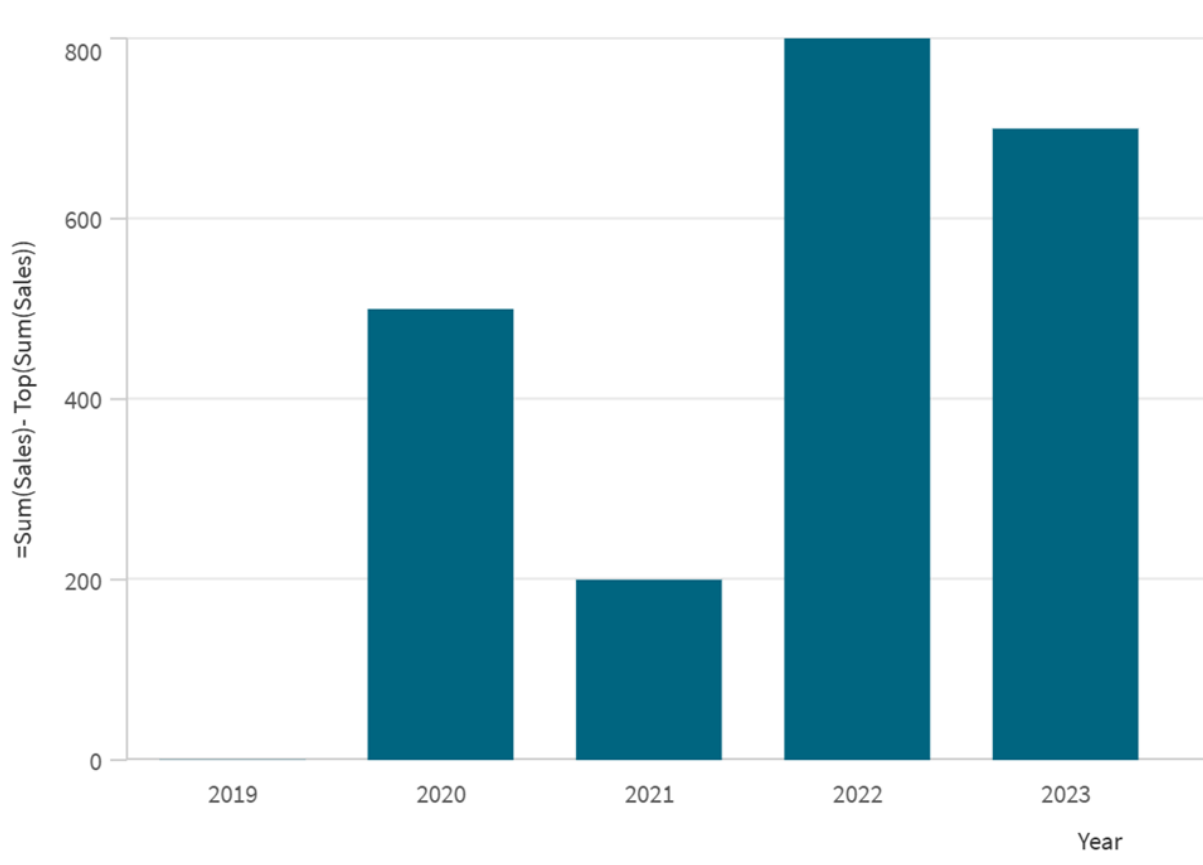
Creare un grafico a barre, quindi aggiungere questo campo come dimensione:

- Year

Creare la seguente misura:

- $\text{=Sum(Sales) - Top(Sum(Sales))}$ , per restituire la varianza delle vendite tra il periodo corrente e il periodo superiore.

*Grafico a barre che mostra i risultati della funzione Top*



L'utilizzo della funzione `Top` in un grafico a barre fornisce un confronto visivo della varianza tra le vendite del periodo corrente rispetto al periodo superiore (o il primo).

### Esempio: scenario per l'applicazione della funzione Top

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:

- OrderID
- Product
- SalesValue
- OrderDate

### Script di caricamento

Example:

```
LOAD * INLINE [
OrderID, Product, SalesValue, OrderDate
1, "Product A", 500, '2023-01-10'
2, "Product B", 300, '2023-01-15'
3, "Product C", 450, '2023-02-01'
4, "Product D", 700, '2023-02-20'
5, "Product E", 350, '2023-03-05'
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- Product

Creare le seguenti misure:

- $\text{=RangeAvg}(\text{Top}(\text{Sum}(\text{SalesValue}), 1, 3))$  per restituire la media delle tre righe superiori.
- $\text{=Sum}(\text{SalesValue}) - \text{RangeAvg}(\text{Top}(\text{Sum}(\text{SalesValue}), 1, 3))$ , per calcolare la varianza tra la riga corrente e la media delle tre righe più superiori.
- $\text{=Sum}(\text{SalesValue})$  per calcolare il totale di salesValue per la riga corrente.

Tabella dei risultati

| Product       | RangeAvg(Top(Sum(SalesValue),1,3)) | Sum(SalesValue) - RangeAvg (Top(Sum(SalesValue),1,3)) | Sum (SalesValue) |
|---------------|------------------------------------|-------------------------------------------------------|------------------|
| <b>Totals</b> | <b>416.67</b>                      | <b>1883.33</b>                                        | <b>2300</b>      |
| 2019          | 416.67                             | 83.33                                                 | 500              |
| 2020          | 416.67                             | --116.67                                              | 300              |
| 2021          | 416.67                             | 33.33                                                 | 450              |
| 2022          | 416.67                             | 283.33                                                | 700              |
| 2023          | 416.67                             | -66.67                                                | 350              |

I risultati mostrano come è possibile utilizzare una combinazione della funzione RangeSum e della funzione Top per analizzare e confrontare i valori di vendita.

### Esempio: espressione del grafico che usa Top

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Due tabelle dati chiamate `Monthnames` e `Sales2013`.

#### Script di caricamento

`Monthnames:`

```
LOAD *, Dual(MonthText,MonthNumber) as Month INLINE [
MonthText, MonthNumber
Jan, 1
Feb, 2
Mar, 3
Apr, 4
May, 5
Jun, 6
Jul, 7
Aug, 8
Sep, 9
Oct, 10
Nov, 11
Dec, 12
];
```

`Sales2013:`

```
Crosstable (MonthText, Sales) LOAD * inline [
Customer|Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|Sep|Oct|Nov|Dec
Astrida|46|60|70|13|78|20|45|65|78|12|78|22
Betacab|65|56|22|79|12|56|45|24|32|78|55|15
Canutility|77|68|34|91|24|68|57|36|44|90|67|27
Divadip|57|36|44|90|67|27|57|68|47|90|80|94
] (delimiter is '|');
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `Customer`

Creare le seguenti misure:

- `=Sum(Sales)`, per calcolare la somma delle vendite.
- `=Top(Sum(Sales))`, per mostrare la somma dei valori di vendita per customer nella riga superiore.

- `=Sum(Sales)+Top(Sum(Sales))`, per sommare il valore `sum(Sales)` dell'attuale Customer a quello di Customer nella riga superiore.
- `=Sum(Sales)+Top(Sum(Sales), 3)`, per sommare il valore `sum(Sales)` dell'attuale Customer a quello di Customer, che è terzo a partire dalla riga superiore.

Tabella dei risultati

| Customer  | Sum (Sales) | Top(Sum (Sales)) | Sum(Sales)+Top(Sum (Sales)) | Sum(Sales)+Top(Sum (Sales), 3) |
|-----------|-------------|------------------|-----------------------------|--------------------------------|
| Totals    | 2566        | 587              | 3153                        | 3249                           |
| Astrida   | 587         | 587              | 1174                        | 1270                           |
| Betacab   | 539         | 587              | 1126                        | 1222                           |
| Canutlity | 683         | 587              | 1270                        | 1366                           |
| Divadip   | 757         | 587              | 1344                        | 1440                           |

La misura `Top(Sum(Sales))` restituisce 587 per tutte le righe perché questo è il valore della riga superiore: Astrida.

Per la misura `sum(Sales)+Top(Sum(Sales))`, la riga Betacab restituisce 1126, che è il risultato per aggiungere il valore `sum(Sales)` a Betacab e Astrida, Customer nella riga superiore.

Per la misura `sum(Sales)+Top(Sum(Sales), 3)`, l'argomento `offset` è impostato su 3. L'espressione aggiunge il valore `sum(Sales)` per la riga corrente e la terza riga dall'alto, `canutlity`. Ad esempio, il valore 1440 viene restituito per il cliente `Divadip`. Questo è il risultato dell'aggiunta del valore `sum(Sales)` per `Divadip` e del valore `sumSales` per `canutlity` (757+683).

### Vedere anche:

- p [Bottom - funzione per grafici \(page 1488\)](#)
- p [Above - funzione per grafici \(page 1464\)](#)
- p [Sum - funzione per grafici \(page 389\)](#)
- p [RangeAvg \(page 1587\)](#)
- p [Funzioni di scala \(page 1584\)](#)

## SecondaryDimensionality - funzione per grafici

**SecondaryDimensionality()** restituisce il numero di righe di dimensione di una tabella pivot senza contenuti di aggregazione; ad esempio, senza somme parziali o aggregati compressi. Questa funzione è equivalente alla funzione **dimensionality()** per le dimensioni orizzontali delle tabelle pivot.

### Sintassi:

```
SecondaryDimensionality( )
```

**Tipo di dati restituiti:** numero intero

**Limiti:**

- A meno che non venga utilizzata nelle tabelle pivot, la funzione **SecondaryDimensionality** restituisce sempre 0.
- L'ordinamento sui valori y nei grafici o l'ordinamento per colonne di espressione nelle tabelle non è consentito quando si utilizza questa funzione di grafico in una qualsiasi delle espressioni del grafico. Queste alternative di ordinamento verranno perciò disattivate automaticamente. Quando si utilizza questa funzione di grafico in una visualizzazione o in una tabella, l'ordinamento della visualizzazione torna all'ordinamento inserito in questa funzione.

### Esempio: principi fondamentali della funzione SecondaryDimensionality

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Date`
  - `Product`
  - `Category`
  - `Sales`

#### Script di caricamento

Example:

```
Load * inline [
Date, Product, Category, Sales
2024-01-01, Apple, Fruit, 100
2024-01-01, Banana, Fruit, 80
2024-02-01, Apple, Fruit, 120
2024-02-01, Banana, Fruit, 90
2024-03-01, Apple, Fruit, 110
2024-03-01, Banana, Fruit, 85
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella pivot e aggiungere questo campo come dimensione di riga:

- Date

Aggiungere questi campi come dimensioni di colonna:

- Category
- Product

Creare le seguenti misure:

- =Sum(Sales), per calcolare le vendite totali.
- =SecondaryDimensionality(), per restituire il numero della dimensione per dimostrare l'uso di questa funzione.

Pivot table showing results for the dimension: Category

|            | (+ ) Fruit |                           |
|------------|------------|---------------------------|
| Date       | Sum(Sales) | SecondaryDimensionality() |
| 2024-01-01 | 180        | 1                         |
| 2024-02-01 | 120        | 1                         |
| 2024-03-01 | 110        | 1                         |

I risultati della tabella pivot mostrano solo la dimensione Category visualizzata come colonna. Pertanto, la funzione SecondaryDimensionality restituisce il valore 1.

Se si espande la dimensione Category, Fruit, i risultati della tabella includeranno ora i valori della dimensione Product, Apple e Banana.

Pivot table showing results for two dimensions: Category, Product

| Date       | (-) Fruit  |                           |             |                           |
|------------|------------|---------------------------|-------------|---------------------------|
|            | Apple      |                           | Banana      |                           |
|            | Sum(Sales) | SecondaryDimensionality() | Sum (Sales) | SecondaryDimensionality() |
| 2024-01-01 | 100        | 2                         | 80          | 2                         |
| 2024-02-01 | 120        | 2                         | 90          | 2                         |
| 2024-03-01 | 110        | 2                         | 85          | 2                         |

La funzione SecondaryDimensionality ora restituisce il valore 2, poiché nella tabella pivot sono presenti valori per due dimensioni: Category e Product.

**Esempio: utilizzo di SecondaryDimensionality per applicare la formattazione condizionale in un'espressione del grafico.**

Espressione del grafico

### Panoramica

Questo esempio utilizza la funzione in un'espressione del grafico in una tabella pivot per applicare la formattazione condizionale delle celle a seconda del livello di dimensione visualizzato.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - Year
  - Quarter
  - Product Category
  - Sales

### Script di caricamento

Example:

```
Load * inline [  
Year, Quarter, Product Category, Sales  
2023, Q1, Electronics, 5000  
2023, Q1, Furniture, 3000  
2023, Q2, Electronics, 4000  
2023, Q2, Furniture, 3500  
2023, Q3, Electronics, 6000  
2023, Q3, Furniture, 4500  
2023, Q4, Electronics, 7000  
2023, Q4, Furniture, 5000  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella pivot e aggiungere questo campo come dimensione di riga:

- Product Category

Aggiungere questi campi come dimensioni di colonna:

- Year
- Quarter

Creare la seguente misura:

- `=Sum(Sales)`, per calcolare le vendite totali.

Nel pannello **Proprietà** della misura `sum(Sales)`, inserire l'espressione seguente come **Espressione colore di sfondo**.

```
IF(SecondaryDimensionality() = 1,
  RGB(144, 238, 144), // Green for Year-level
  RGB(173, 216, 230), // Blue for Quarter-level
)
```

L'espressione applicherà uno sfondo verde alle celle della tabella quando la funzione `SecondaryDimensionality` restituisce 1 e uno sfondo blu quando la funzione restituisce qualsiasi altro risultato.

Tabella pivot che mostra i risultati per una dimensione: **Anno** (con le celle con sfondo verde

| Categoria di prodotti | (+) 2023 |
|-----------------------|----------|
| Elettronica           | 22000    |
| Arredamento           | 16000    |

I risultati mostrano che quando viene visualizzata solo la dimensione `Year`, quindi il valore di `SecondaryDimensionality` è 1, le celle `sum(Sales)` appaiono con uno sfondo verde. Quando si espande la tabella per mostrare il valore `sum(Sales)` per `quarter`, lo sfondo appare ora in blu, perché sono presenti due dimensioni, `year` e `quarter`, e il valore `SecondaryDimensionality` è ora 2.

Pivot table showing results for two dimensions: **Year** and **Quarter** with blue cell background

| Product Category | (-) 2023 |      |      |      |
|------------------|----------|------|------|------|
|                  | Q1       | Q2   | Q3   | Q4   |
| Electronics      | 5000     | 4000 | 6000 | 7000 |
| Furniture        | 3000     | 3500 | 4500 | 5000 |

### After - funzione per grafici

**After()** restituisce il valore di un'espressione valutata con i valori di dimensione di una tabella pivot così come appaiono nella colonna successiva a quella attuale all'interno di un segmento di riga nella tabella pivot.

**Sintassi:**

```
after([TOTAL] expr [, offset [, count ]])
```



*L'ordinamento sui valori y nei grafici o l'ordinamento per colonne di espressione nelle tabelle non è consentito quando si utilizza questa funzione di grafico in una qualsiasi delle espressioni del grafico. Queste alternative di ordinamento verranno perciò disattivate automaticamente. Quando si utilizza questa funzione di grafico in una visualizzazione o in una tabella, l'ordinamento della visualizzazione torna all'ordinamento inserito in questa funzione.*



Questa funzione restituisce NULL in tutti i tipi di grafico, ad eccezione delle tabelle pivot.

#### Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>expr</b>   | L'espressione o il campo contenente i dati da misurare.                                                                                                                                                                                                                                                                                                                                                                                |
| <b>offset</b> | <p>Specificando un <b>offset</b> n maggiore di 1, la valutazione dell'espressione viene spostata di n colonne verso destra rispetto alla colonna attuale.</p> <p>Specificando un offset uguale a 0 verrà valutata l'espressione nella colonna attuale.</p> <p>Specificando un numero di offset negativo, la funzione <b>After</b> diventa equivalente alla funzione <b>Before</b> con il numero di offset positivo corrispondente.</p> |
| <b>count</b>  | Specificando un terzo parametro <b>count</b> maggiore di 1, la funzione restituirà una scala di valori, uno per ciascuna delle colonne della tabella fino al valore di <b>count</b> , contando verso destra a partire dalla cella originaria.                                                                                                                                                                                          |
| <b>TOTAL</b>  | Se la tabella è unidimensionale o se è utilizzato il qualificatore <b>TOTAL</b> come argomento, il segmento colonna attuale sarà sempre uguale all'intera colonna.                                                                                                                                                                                                                                                                     |

In corrispondenza dell'ultima colonna di un segmento di riga verrà restituito un valore NULL, perché non vi sono colonne che la seguono.

Se una tabella pivot include più dimensioni orizzontali, il segmento della riga attuale includerà solo le colonne con gli stessi valori della colonna attuale in tutte le righe della dimensione, tranne per la riga che visualizza l'ultima dimensione orizzontale dell'ordinamento tra campi. La sequenza di ordinamento tra campi per le dimensioni orizzontali nelle tabelle pivot è definita semplicemente dall'ordine delle dimensioni dall'alto verso il basso.

#### Esempio: espressioni del grafico

| Esempio                                    | Risultato                                                                                                                                               |
|--------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>After( Sum(Sales ))</code>           | Restituisce la somma per sales dalla colonna successiva immediatamente a destra rispetto alla colonna corrente.                                         |
| <code>After( Sum(Sales ), 2)</code>        | Restituisce la somma per sales dalla colonna che si trova due colonne a destra della colonna corrente.                                                  |
| <code>After( Total Sum ( sales ))</code>   | Restituisce la somma totale per sales dalla colonna successiva immediatamente a destra rispetto alla colonna corrente.                                  |
| <code>RangeAvg (After (Sum(x),1,3))</code> | Restituisce la media dei tre risultati della funzione <code>sum(x)</code> calcolata in base alle tre colonne immediatamente a destra di quella attuale. |

### Esempio: principi fondamentali della funzione After

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Year`
  - `Quarter`
  - `Sales`

#### Script di caricamento

Example:

```
Load * inline [  
Year, Quarter, Sales  
2023, Q1, 5000  
2023, Q2, 4000  
2023, Q3, 6000  
2023, Q4, 7000  
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella pivot e aggiungere questo campo come dimensione di riga:

- `Year`

Aggiungere questo campo come dimensione di colonna:

- `Quarter`

Creare le seguenti misure:

- `=Sum(Sales)`, per calcolare la somma delle vendite.
- `=After(Sum(Sales))`, per mostrare la somma delle vendite nel trimestre successivo.

Tabella dei risultati

| Anno | Q1              |                        | Q2              |                        | Q3              |                        | Q4              |                        |
|------|-----------------|------------------------|-----------------|------------------------|-----------------|------------------------|-----------------|------------------------|
|      | Somma (Vendite) | Dopo (Somma (Vendite)) | Somma (Vendite) | Dopo (Somma (Vendite)) | Somma (Vendite) | Dopo (Somma (Vendite)) | Somma (Vendite) | Dopo (Somma (Vendite)) |
| 2023 | 5000            | 4000                   | 4000            | 6000                   | 6000            | 7000                   | 7000            | -                      |

Analizzando i risultati, si può notare come la funzione `After` restituisce il valore della colonna successiva che si trova immediatamente a destra della colonna `sum(Sales)` per ogni quarter. Ad esempio, in Q1 il valore `sum(Sales)` per quel periodo è 5000. L'output della funzione `After` per Q1 è 4000 perché questo è il valore della colonna immediatamente a destra di `sum(Sales)` per Q1, cioè `sum(Sales)` per Q2.

### Esempio: scenario per la funzione `After`

Espressione del grafico

#### Panoramica

Un set di dati contiene i valori di più anni di vendita per le unità vendute presso un'azienda di elettronica. L'azienda vuole presentare le vendite totali per ogni anno e confrontare le vendite unitarie da un anno all'altro.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Year`
  - `Sales`

#### Script di caricamento

```
Example:
Load * inline [
Year, Sales
2021, 1350
2022, 1840
2023, 2250
2024, 1750
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella pivot e aggiungere questo campo come dimensione di colonna:

- Year

Creare le seguenti misure:

- `=Sum(Sales)`, per calcolare la somma delle vendite.
- `=Sum(Sales) - After(Sum(Sales))`, per calcolare la differenza delle vendite da un anno all'altro.

Tabella dei risultati

| Anno | 2021            |                                        | 2022            |                                        | 2023            |                                        | 2024            |                                        |
|------|-----------------|----------------------------------------|-----------------|----------------------------------------|-----------------|----------------------------------------|-----------------|----------------------------------------|
|      | Somma (Vendite) | Somma (Vendite)-Dopo (Somma (Vendite)) | Somma (Vendite) | Somma (Vendite)-Dopo (Somma (Vendite)) | Somma (Vendite) | Somma (Vendite)-Dopo (Somma (Vendite)) | Somma (Vendite) | Somma (Vendite)-Dopo (Somma (Vendite)) |
| 2023 | 1350            | -490                                   | 1840            | -410                                   | 2250            | 500                                    | 1750            | -                                      |

I risultati mostrano la somma delle vendite per ogni anno e la differenza tra le cifre delle vendite da un anno all'altro. Per esempio, in 2022, il valore `sum(Sales)` era 1840. L'output della funzione `After` indica che questo valore era inferiore di 410 unità rispetto alla somma delle vendite per il 2023.

### Esempio: scenario avanzatato per la funzione After

Espressione del grafico

#### Panoramica

Un set di dati contiene i dati sulle vendite per diversi anni. L'azienda vuole presentare le vendite totali per l'anno in corso e un totale mobile delle vendite dei successivi 3 anni.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `SalesYear`
  - `Sales`

#### Script di caricamento

Example:

```
Load * inline [
SalesYear, Sales
2021, 1300
2022, 1800
2023, 2000
```

```
2024, 2150
];
```

## Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella pivot e aggiungere questo campo come dimensione di colonna:

- SalesYear

Creare le seguenti misure:

- =Sum(Sales), per calcolare la somma delle vendite.
- =RangeSum(After(Sum(Sales), 1, 3)), per calcolare la somma delle vendite per i 3 anni successivi.

Tabella dei risultati

| 2021               |                                                        | 2022               |                                                        | 2023               |                                                        | 2024               |                                                        |
|--------------------|--------------------------------------------------------|--------------------|--------------------------------------------------------|--------------------|--------------------------------------------------------|--------------------|--------------------------------------------------------|
| Somma<br>(Vendite) | SommaIntervallo(Dopo<br>(Somma<br>(Vendite), 1,<br>3)) | Somma<br>(Vendite) | SommaIntervallo(Dopo<br>(Somma<br>(Vendite), 1,<br>3)) | Somma<br>(Vendite) | SommaIntervallo(Dopo<br>(Somma<br>(Vendite), 1,<br>3)) | Somma<br>(Vendite) | SommaIntervallo(Dopo<br>(Somma<br>(Vendite), 1,<br>3)) |
| 1300               | 5950                                                   | 1800               | 4150                                                   | 2000               | 2150                                                   | 2150               | 0                                                      |

I risultati dimostrano come utilizzare la funzione `After` per restituire la somma dei valori delle vendite per i tre anni successivi. Ad esempio, in 2021, il valore `sum(Sales)` annuale è 1300 e la somma delle vendite per i 3 anni successivi (2022, 2023, 2024) è 5950. Il valore 0 viene restituito per 2024 perché i dati per gli anni futuri non sono disponibili.

## Before - funzione per grafici

**Before()** restituisce il valore di un'espressione valutata con i valori di dimensione di una tabella pivot così come appaiono nella colonna precedente a quella attuale all'interno di un segmento di riga della tabella pivot.

### Sintassi:

```
Before([TOTAL] expr [, offset [, count]])
```



Questa funzione restituisce NULL in tutti i tipi di grafico, ad eccezione delle tabelle pivot.



L'ordinamento sui valori y nei grafici o l'ordinamento per colonne di espressione nelle tabelle non è consentito quando si utilizza questa funzione di grafico in una qualsiasi delle espressioni del grafico. Queste alternative di ordinamento verranno perciò disattivate automaticamente. Quando si utilizza questa funzione di grafico in una visualizzazione o in una tabella, l'ordinamento della visualizzazione torna all'ordinamento inserito in questa funzione.

## Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>expr</b>   | L'espressione o il campo contenente i dati da misurare.                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>offset</b> | <p>Se si specifica un <b>offset</b> n, maggiore di 1, la valutazione dell'espressione viene spostata di n colonne verso sinistra rispetto alla colonna attuale.</p> <p>Specificando un offset uguale a 0 verrà valutata l'espressione nella colonna attuale.</p> <p>Specificando un numero di offset negativo, la funzione <b>Before</b> diventa equivalente alla funzione <b>After</b> con il numero di offset positivo corrispondente.</p> |
| <b>count</b>  | Specificando un terzo parametro <b>count</b> maggiore di 1, la funzione restituirà una scala di valori, uno per ciascuna delle colonne della tabella fino al valore di <b>count</b> , contando verso sinistra a partire dalla cella originaria.                                                                                                                                                                                              |
| <b>TOTAL</b>  | Se la tabella è unidimensionale o se è utilizzato il qualificatore <b>TOTAL</b> come argomento, il segmento colonna attuale sarà sempre uguale all'intera colonna.                                                                                                                                                                                                                                                                           |

In corrispondenza della prima colonna di un segmento di riga verrà restituito un valore NULL, perché non vi sono colonne che la precedono.

Se una tabella pivot include più dimensioni orizzontali, il segmento della riga attuale includerà solo le colonne con gli stessi valori della colonna attuale in tutte le righe della dimensione, tranne per la riga che visualizza l'ultima dimensione orizzontale dell'ordinamento tra campi. La sequenza di ordinamento tra campi per le dimensioni orizzontali nelle tabelle pivot è definita semplicemente dall'ordine delle dimensioni dall'alto verso il basso.

## Esempio: espressioni del grafico

| Esempio                                         | Risultato                                                                                                                                                 |
|-------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>Before( Sum(Sales ))</code>               | Restituisce la somma per sales dalla colonna precedente immediatamente a sinistra rispetto alla colonna corrente.                                         |
| <code>Before( Sum(Sales ), 2)</code>            | Restituisce la somma per sales dalla colonna che si trova due colonne a sinistra della colonna corrente.                                                  |
| <code>Before( Total Sum ( Sales ))</code>       | Restituisce la somma totale per sales dalla colonna precedente immediatamente a sinistra rispetto alla colonna corrente.                                  |
| <code>RangeAvg ( Before (Sum(x) , 1, 3))</code> | Restituisce la media dei tre risultati della funzione <code>sum(x)</code> calcolata in base alle tre colonne immediatamente a sinistra di quella attuale. |

### Esempio: principi fondamentali della funzione Before

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - Year
  - Quarter
  - Sales

#### Script di caricamento

Example:

```
Load * inline [  
Year, Quarter, Sales  
2023, Q1, 5000  
2023, Q2, 4000  
2023, Q3, 6000  
2023, Q4, 7000  
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella pivot e aggiungere questo campo come dimensione di riga:

- Year

Aggiungere questo campo come dimensione di colonna:

- Quarter

Creare le seguenti misure:

- `=Sum(Sales)`, per calcolare la somma delle vendite.
- `=Before(Sum(Sales))`, per mostrare la somma dei valori di vendita nel trimestre precedente.

Tabella dei risultati

| Anno | Q1             |                            | Q2             |                            | Q3             |                            | Q4             |                            |
|------|----------------|----------------------------|----------------|----------------------------|----------------|----------------------------|----------------|----------------------------|
|      | Sum<br>(Sales) | Before<br>(Sum<br>(Sales)) | Sum<br>(Sales) | Before<br>(Sum<br>(Sales)) | Sum<br>(Sales) | Before<br>(Sum<br>(Sales)) | Sum<br>(Sales) | Before<br>(Sum<br>(Sales)) |
| 2023 | 5000           | -                          | 4000           | 5000                       | 6000           | 4000                       | 7000           | 6000                       |

Analizzando i risultati, si può notare come la funzione `Before` restituisce il valore della colonna precedente che si trova immediatamente a sinistra della colonna `sum(sales)` per ogni `quarter`. Per esempio, in T4 il valore `sum(sales)` per quel periodo è 7000. L'output della funzione `Before` per Q4 è 6000, che è il valore della colonna immediatamente a sinistra di `sum(sales)` per Q4, cioè `sum(sales)` per Q3.

### Esempio: utilizzo della funzione `Before` per calcolare la crescita delle vendite anno dopo anno

Espressione del grafico

#### Panoramica

Un set di dati contiene i dati sulle vendite per molti anni. L'azienda vuole presentare le vendite totali per ogni anno e l'aumento delle vendite in base al valore dell'anno precedente.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Year`
  - `Sales`

#### Script di caricamento

```
Example:
Load * inline [
Year, Sales
2021, 1300
2022, 1800
2023, 2000
2024, 2100
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella pivot e aggiungere questo campo come dimensione di colonna:

- `Year`

Creare le seguenti misure:

- `=Sum(Sales)`, per calcolare la somma delle vendite.
- `=(Sum(Sales) - Before(Sum(Sales))) / Before(Sum(Sales))`, per calcolare la crescita annuale delle vendite. Per mostrare questo valore come una percentuale, sotto la voce **Formattazione numero**, selezionare **Numero > Formattazione semplice > 12,34%**.

Tabella dei risultati

| 2021        |                                                      | 2022        |                                                      | 2023        |                                                      | 2024        |                                                      |
|-------------|------------------------------------------------------|-------------|------------------------------------------------------|-------------|------------------------------------------------------|-------------|------------------------------------------------------|
| Sum (Sales) | (Sum(Sales) - Before(Sum(Sales)))/Before(Sum(Sales)) | Sum (Sales) | (Sum(Sales) - Before(Sum(Sales)))/Before(Sum(Sales)) | Sum (Sales) | (Sum(Sales) - Before(Sum(Sales)))/Before(Sum(Sales)) | Sum (Sales) | (Sum(Sales) - Before(Sum(Sales)))/Before(Sum(Sales)) |
| 1300        | -                                                    | 1800        | 38.46%                                               | 2000        | 11.11%                                               | 2100        | 5.00%                                                |

L'output della funzione `Before` restituisce la crescita annuale in base ai dati di vendita dell'anno precedente. Per esempio, la somma delle vendite per 2022 era superiore del 38.46% rispetto al valore `sum(Sales)` per 2021.



È possibile anche utilizzare la funzione `After` per eseguire lo stesso calcolo dell'aumento delle vendite in questo scenario. Quando si specifica un numero di offset negativo nei parametri della funzione, la funzione `After` diventa equivalente alla funzione `Before` con il numero di offset positivo corrispondente. Ad esempio, la seguente espressione che utilizza la funzione `After` (con un valore di offset negativo di -1) restituisce gli stessi risultati di crescita annuale delle vendite della funzione `Before`:

$$=(\text{Sum}(\text{Sales}) - \text{After}(\text{Sum}(\text{Sales}), -1)) / (\text{After}(\text{Sum}(\text{Sales}), -1))$$

## First - funzione per grafici

**First()** restituisce il valore di un'espressione valutata con i valori di dimensione di una tabella pivot così come appaiono nella prima colonna del segmento di riga attuale della tabella pivot. Questa funzione restituisce NULL in tutti i tipi di grafico, ad eccezione delle tabelle pivot.



L'ordinamento sui valori *y* nei grafici o l'ordinamento per colonne di espressione nelle tabelle non è consentito quando si utilizza questa funzione di grafico in una qualsiasi delle espressioni del grafico. Queste alternative di ordinamento verranno perciò disattivate automaticamente. Quando si utilizza questa funzione di grafico in una visualizzazione o in una tabella, l'ordinamento della visualizzazione torna all'ordinamento inserito in questa funzione.

### Sintassi:

```
First([TOTAL] expr [, offset [, count]])
```

### Argomenti

| Argomento         | Descrizione                                             |
|-------------------|---------------------------------------------------------|
| <b>expression</b> | L'espressione o il campo contenente i dati da misurare. |

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>offset</b> | <p>Se si specifica un <b>offset</b> n maggiore di 1, la valutazione dell'espressione viene spostata di n colonne verso destra rispetto alla prima colonna.</p> <p>Se si specifica un offset uguale a 0 verrà valutata l'espressione nella prima colonna del segmento della riga attuale.</p> <p>Specificando un offset uguale a 0 verrà valutata l'espressione nella riga attuale.</p> <p>Specificando un numero di offset negativo, la funzione <b>First</b> diventa equivalente alla funzione <b>Last</b> con il numero di offset positivo corrispondente.</p> |
| <b>count</b>  | Specificando un terzo parametro <b>count</b> maggiore di 1, la funzione restituirà una scala di valori, uno per ciascuna delle colonne della tabella fino al valore di <b>count</b> , contando verso destra a partire dalla cella originaria.                                                                                                                                                                                                                                                                                                                    |
| <b>TOTAL</b>  | Se la tabella è unidimensionale o se è utilizzato il qualificatore <b>TOTAL</b> come argomento, il segmento colonna attuale sarà sempre uguale all'intera colonna.                                                                                                                                                                                                                                                                                                                                                                                               |

Se una tabella pivot include più dimensioni orizzontali, il segmento della riga attuale includerà solo le colonne con gli stessi valori della colonna attuale in tutte le righe della dimensione, tranne per la riga che visualizza l'ultima dimensione orizzontale dell'ordinamento tra campi. La sequenza di ordinamento tra campi per le dimensioni orizzontali nelle tabelle pivot è definita semplicemente dall'ordine delle dimensioni dall'alto verso il basso.

#### Esempio: espressioni del grafico

| Esempio                                    | Risultato                                                                                                                                                 |
|--------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>First( Sum(Sales ))</code>           | Restituisce la somma per sales dalla prima colonna del segmento di riga corrente.                                                                         |
| <code>First( Sum(Sales ), 2)</code>        | Restituisce la somma per sales che si trova due colonne a destra della prima colonna nel segmento della riga corrente.                                    |
| <code>First( Total Sum ( sales ))</code>   | Restituisce la somma totale per sales dalla prima colonna del segmento della riga corrente.                                                               |
| <code>RangeAvg (First (Sum(x),1,5))</code> | Restituisce una media dei risultati della funzione <code>sum(x)</code> calcolata in base alle cinque colonne più a sinistra del segmento di riga attuale. |

### Esempio: principi fondamentali della funzione First

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Year`
  - `Quarter`
  - `Sales`

### Script di caricamento

Example:

```
Load * inline [
Year, Quarter, Sales
2023, Q1, 5000
2023, Q2, 4000
2023, Q3, 6000
2023, Q4, 7000
2024, Q1, 4500
2024, Q2, 3250
2024, Q3, 6500
2024, Q4, 7250
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella pivot e aggiungere questo campo come dimensione di riga:

- `Year`

Aggiungere questo campo come dimensione di colonna:

- `Quarter`

Creare le seguenti misure:

- `=Sum(Sales)`, per calcolare la somma delle vendite.
- `=First(Sum(Sales))`, per calcolare la somma delle vendite della prima colonna per il segmento di riga corrente.

Tabella dei risultati

| Anno | Q1             |                           | Q2             |                           | Q3             |                           | Q4             |                           |
|------|----------------|---------------------------|----------------|---------------------------|----------------|---------------------------|----------------|---------------------------|
|      | Sum<br>(Sales) | First<br>(Sum<br>(Sales)) | Sum<br>(Sales) | First<br>(Sum<br>(Sales)) | Sum<br>(Sales) | First<br>(Sum<br>(Sales)) | Sum<br>(Sales) | First<br>(Sum<br>(Sales)) |
| 2023 | 5000           | 5000                      | 4000           | 5000                      | 6000           | 5000                      | 7000           | 5000                      |
| 2024 | 4500           | 4500                      | 3250           | 4500                      | 6500           | 4500                      | 7250           | 4500                      |

Osservando i risultati, si può notare come la funzione `First` restituisce la somma del valore delle vendite per la prima colonna di ogni segmento di riga per ogni trimestre. Ad esempio, in Q4 il valore `sum(Sales)` per quel periodo è 7000. Il valore di `First(sum(Sales))` per Q4 è 5000, che è il valore di `sum(Sales)` in Q1, il valore della prima colonna della riga corrente, 2023.

### Esempio: scenario per la funzione `First`

Espressione del grafico

#### Panoramica

Un set di dati contiene i dati di vendita in base all'anno per diversi prodotti in un periodo di tre anni. L'azienda vuole presentare le vendite totali per l'anno in corso e la crescita delle vendite dal primo anno di questo periodo.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Date`
  - `Product`
  - `Sales`

#### Script di caricamento

Example:

```
Load * inline [
Date, Product, Sales
2021, Apple, 100
2021, Banana, 150
2021, Carrot, 120
2022, Carrot, 80
2022, Banana, 200
2022, Apple, 110
2023, Carrot, 160
2023, Banana, 300
2023, Apple, 150
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella pivot e aggiungere questo campo come dimensione di riga:

- `Product`

Aggiungere questo campo come dimensione di colonna:

- Date

Creare le seguenti misure:

- $\text{=Sum(Sales)}$ , per calcolare le vendite annuali.
- $\text{=(Sum(Sales) - First(Sales)) / First(Sales)}$ , per calcolare la crescita delle vendite dal primo anno nel set di dati. Per mostrare questo valore come una percentuale, sotto la voce **Formattazione numero**, selezionare **Numero > Formattazione semplice > 12,34%**.

Tabella dei risultati

| Prodotto | 2021        |                                            | 2022        |                                            | 2023        |                                            |
|----------|-------------|--------------------------------------------|-------------|--------------------------------------------|-------------|--------------------------------------------|
|          | Sum (Sales) | (Sum(Sales) - First(Sales)) / First(Sales) | Sum (Sales) | (Sum(Sales) - First(Sales)) / First(Sales) | Sum (Sales) | (Sum(Sales) - First(Sales)) / First(Sales) |
| Apple    | 100         | 0.00%                                      | 110         | 10.00%                                     | 150         | 50.00%                                     |
| Banana   | 150         | 0.00%                                      | 200         | 33.33%                                     | 300         | 100.00%                                    |
| Carrot   | 120         | 0.00%                                      | 80          | -33.33%                                    | 160         | 33.33%                                     |

I risultati mostrano i dati di vendita annuali per ogni anno e il tasso di crescita annuale in percentuale rispetto ai valori di vendita del primo anno. Per esempio, le vendite di Apple sono aumentate da un valore iniziale di 100 nel 2021 a un valore di 150 nel 2023, con un aumento del 50%. La vendita di Carrot è diminuita da 120 nel 2021 a 80 nel 2022, con una riduzione del 33.33%.

### Esempio - Primo scenario che utilizza una somma di intervalli per presentare le cifre del budget

Espressione del grafico

#### Panoramica

Un set di dati contiene le cifre del budget per trimestre, per un periodo di due anni, per categoria di prodotto. L'azienda vuole presentare il budget per il trimestre in corso e il totale corrente del budget per ogni anno.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `BudgetYear`
  - `BudgetQuarter`
  - `QuarterNo`

- BudgetAmount
- Category

### Script di caricamento

Example:

```
Load * inline [
BudgetYear, BudgetQuarter, QuarterNo, BudgetAmount, Category
2023, Q1, 1, 2100, Online
2023, Q2, 2, 2500, Online
2023, Q3, 3, 3100, Online
2023, Q4, 4, 4100, Online
2023, Q1, 1, 5100, Store
2023, Q2, 2, 5500, Store
2023, Q3, 3, 6100, Store
2023, Q4, 4, 7100, Store
2024, Q1, 1, 2300, Online
2024, Q2, 2, 2800, Online
2024, Q3, 3, 3300, Online
2024, Q4, 4, 4400, Online
2024, Q1, 1, 5300, Store
2024, Q2, 2, 5800, Store
2024, Q3, 3, 6300, Store
2024, Q4, 4, 7400, Store
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella pivot e aggiungere questi campi come dimensioni della riga:

- BudgetYear
- Category

Aggiungere questo campo come dimensione di colonna:

- BudgetQuarter

Creare le seguenti misure:

- =Sum(BudgetAmount), per calcolare i dati di vendita del budget.
- =RangeSum(First(Sum(BudgetAmount), 1, QuarterNo)), per calcolare il totale parziale delle vendite in base al budget per ogni trimestre, sommando l'intervallo dal primo trimestre a QuarterNo, come valutato per ogni trimestre.

Tabella dei risultati

| BudgetYear<br>Categoria | Q1                        |                                                                            | Q2                        |                                                                            | Q3                        |                                                                            | Q4                        |                                                                            |
|-------------------------|---------------------------|----------------------------------------------------------------------------|---------------------------|----------------------------------------------------------------------------|---------------------------|----------------------------------------------------------------------------|---------------------------|----------------------------------------------------------------------------|
|                         | Sum<br>(Budget<br>Amount) | RangeSum<br>(First<br>(Sum<br>(Budget<br>Amount),<br>1,<br>Quarter<br>No)) | Sum<br>(Budget<br>Amount) | RangeSum<br>(First<br>(Sum<br>(Budget<br>Amount),<br>1,<br>Quarter<br>No)) | Sum<br>(Budget<br>Amount) | RangeSum<br>(First<br>(Sum<br>(Budget<br>Amount),<br>1,<br>Quarter<br>No)) | Sum<br>(Budget<br>Amount) | RangeSum<br>(First<br>(Sum<br>(Budget<br>Amount),<br>1,<br>Quarter<br>No)) |
| (-) 2023                | 7200                      | 7200                                                                       | 8000                      | 15200                                                                      | 9200                      | 24400                                                                      | 11200                     | 35600                                                                      |
| Online                  | 2100                      | 2100                                                                       | 2500                      | 4600                                                                       | 3100                      | 7700                                                                       | 4100                      | 11800                                                                      |
| Archivia                | 5100                      | 5100                                                                       | 5500                      | 10600                                                                      | 6100                      | 16700                                                                      | 7100                      | 23800                                                                      |
| (-) 2024                | 7600                      | 7600                                                                       | 8600                      | 16200                                                                      | 9600                      | 25800                                                                      | 11800                     | 37600                                                                      |
| Online                  | 2300                      | 2300                                                                       | 2800                      | 5100                                                                       | 3300                      | 8400                                                                       | 4400                      | 12800                                                                      |
| Archivia                | 5300                      | 5300                                                                       | 5800                      | 11100                                                                      | 6300                      | 17400                                                                      | 7400                      | 24800                                                                      |

I risultati mostrano le cifre del budget per ogni Year e Category, e il totale corrente per quarter. Per esempio, per il Q2 del 2023, il valore di quarterNo è 2. Pertanto, l'espressione della funzione viene valutata come `RangeSum(First(Sum(BudgetAmount), 1, 2))`, che somma le colonne `BudgetAmount` per Q1 (7200) e Q2 (8000) e restituisce il valore 15200. Il risultato restituisce il totale delle colonne `Sum(BudgetAmount)` per Q1 e Q2. Nel 2024, per Category online, `Sum(BudgetAmount)` per Q1 è 2300 e per Q2, è 2800. Pertanto, l'espressione `RangeSum` restituisce 5100.

## Last - funzione per grafici

**Last()** restituisce il valore di un'espressione valutata con i valori di dimensione di una tabella pivot così come appaiono nell'ultima colonna del segmento di riga attuale della tabella pivot. Questa funzione restituisce NULL in tutti i tipi di grafico, ad eccezione delle tabelle pivot.



*L'ordinamento sui valori y nei grafici o l'ordinamento per colonne di espressione nelle tabelle non è consentito quando si utilizza questa funzione di grafico in una qualsiasi delle espressioni del grafico. Queste alternative di ordinamento verranno perciò disattivate automaticamente. Quando si utilizza questa funzione di grafico in una visualizzazione o in una tabella, l'ordinamento della visualizzazione torna all'ordinamento inserito in questa funzione.*

### Sintassi:

```
Last([TOTAL] expr [, offset [, count]])
```

## Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>expr</b>   | L'espressione o il campo contenente i dati da misurare.                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>offset</b> | <p>Se si specifica un <b>offset</b> n, maggiore di 1, la valutazione dell'espressione viene spostata di n colonne a sinistra rispetto all'ultima colonna.</p> <p>Se si specifica un offset uguale a 0 verrà valutata l'espressione nell'ultima colonna del segmento della riga attuale.</p> <p>Specificando un numero di offset negativo, la funzione <b>Last</b> diventa equivalente alla funzione <b>First</b> con il numero di offset positivo corrispondente.</p> |
| <b>count</b>  | Specificando un terzo parametro <b>count</b> maggiore di 1, la funzione restituirà una scala di valori, uno per ciascuna delle colonne della tabella fino al valore di <b>count</b> , contando verso sinistra a partire dalla cella originaria.                                                                                                                                                                                                                       |
| <b>TOTAL</b>  | Se la tabella è unidimensionale o se è utilizzato il qualificatore <b>TOTAL</b> come argomento, il segmento colonna attuale sarà sempre uguale all'intera colonna.                                                                                                                                                                                                                                                                                                    |

Se una tabella pivot include più dimensioni orizzontali, il segmento della riga attuale includerà solo le colonne con gli stessi valori della colonna attuale in tutte le righe della dimensione, tranne per la riga che visualizza l'ultima dimensione orizzontale dell'ordinamento tra campi. La sequenza di ordinamento tra campi per le dimensioni orizzontali nelle tabelle pivot è definita semplicemente dall'ordine delle dimensioni dall'alto verso il basso.

## Esempio: espressioni del grafico

| Esempio                                     | Risultato                                                                                                                                                  |
|---------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>Last( Sum(Sales ))</code>             | Restituisce la somma per sales dall'ultima colonna del segmento di riga corrente.                                                                          |
| <code>Last( Sum(Sales ), 2)</code>          | Restituisce la somma per sales dalla colonna che si trova due colonne a sinistra dell'ultima colonna nel segmento della riga corrente.                     |
| <code>Last( Total Sum ( Sales ))</code>     | Restituisce la somma per sales dall'ultima colonna del segmento della riga corrente.                                                                       |
| <code>RangeAvg (Last (Sum(x), 1, 5))</code> | Restituisce una media dei risultati della funzione <code>sum(x)</code> calcolata in base alle cinque colonne più a destra del segmento della riga attuale. |

## Esempio: principi fondamentali della funzione Last

Espressione del grafico

## Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Year`
  - `Quarter`
  - `Sales`

### Script di caricamento

Example:

```
Load * inline [
Year, Quarter, Sales
2023, Q1, 5000
2023, Q2, 4000
2023, Q3, 6000
2023, Q4, 7000
2024, Q1, 4500
2024, Q2, 3250
2024, Q3, 6500
2024, Q4, 7250
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella pivot e aggiungere questo campo come dimensione di riga:

- `Year`

Aggiungere questo campo come dimensione di colonna:

- `Quarter`

Creare le seguenti misure:

- `=Sum(Sales)`, per calcolare la somma delle vendite.
- `=Last(Sum(Sales))`, per calcolare la somma delle vendite dell'ultima colonna per il segmento della riga corrente.

Tabella dei risultati

| Anno | Q1             |                          | Q2             |                          | Q3             |                          | Q4             |                          |
|------|----------------|--------------------------|----------------|--------------------------|----------------|--------------------------|----------------|--------------------------|
|      | Sum<br>(Sales) | Last<br>(Sum<br>(Sales)) | Sum<br>(Sales) | Last<br>(Sum<br>(Sales)) | Sum<br>(Sales) | Last<br>(Sum<br>(Sales)) | Sum<br>(Sales) | Last<br>(Sum<br>(Sales)) |
| 2023 | 5000           | 7000                     | 4000           | 7000                     | 6000           | 7000                     | 7000           | 7000                     |
| 2024 | 4500           | 7250                     | 3250           | 7250                     | 6500           | 7250                     | 7250           | 7250                     |

Osservando i risultati, si può notare che la funzione `Last` restituisce la somma del valore delle vendite per l'ultima colonna di ogni segmento di riga per ogni trimestre. Ad esempio, in Q1 il valore `sum(Sales)` per quel periodo è 5000. Il valore di `Last(sum(Sales))` per Q1 è 7000, che è il valore di `sum(Sales)` in Q4, il valore dell'ultima colonna della riga corrente, 2023.

### Esempio: scenario per la funzione Last

Espressione del grafico

#### Panoramica

Un set di dati contiene i dati di vendita per diversi prodotti in un periodo di tre mesi. L'azienda vuole presentare le vendite totali per l'anno in corso e la crescita delle vendite rispetto agli ultimi (o i più recenti) dati di vendita per questo periodo.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Date`
  - `Product`
  - `Sales`

#### Script di caricamento

Example:

```
Load * inline [
Date, Product, Sales
2024-01-01, Apple, 100
2024-01-01, Banana, 150
2024-01-01, Carrot, 120
2024-02-01, Carrot, 80
2024-02-01, Banana, 200
2024-02-01, Apple, 110
2024-03-01, Carrot, 160
2024-03-01, Banana, 300
2024-03-01, Apple, 150
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella pivot e aggiungere questo campo come dimensione di riga:

- `Product`

Aggiungere questo campo come dimensione di colonna:

- Date

Creare le seguenti misure:

- $\text{=Sum(Sales)}$ , per calcolare le vendite annuali.
- $\text{=(Last(Sum(Sales)) - Sum(Sales)) / Sum(Sales)}$ , per calcolare la crescita delle vendite dal periodo corrente all'ultimo periodo. Per mostrare questo valore come una percentuale, sotto la voce **Formattazione numero**, selezionare **Numero > Formattazione semplice > 12,34%**.

Tabella dei risultati

| Prodotto | 2024-01-01  |                                                 | 2024-02-01  |                                                 | 2024-03-01  |                                                 |
|----------|-------------|-------------------------------------------------|-------------|-------------------------------------------------|-------------|-------------------------------------------------|
|          | Sum (Sales) | (Last(Sum (Sales)) - Sum (Sales)) / Sum (Sales) | Sum (Sales) | (Last(Sum (Sales)) - Sum (Sales)) / Sum (Sales) | Sum (Sales) | (Last(Sum (Sales)) - Sum (Sales)) / Sum (Sales) |
| Apple    | 100         | 50.00%                                          | 110         | 36.36%                                          | 150         | 0.00%                                           |
| Banana   | 150         | 100.00%                                         | 200         | 50.00%                                          | 300         | 0.00%                                           |
| Carrot   | 120         | 33.33%                                          | 80          | 100.00%                                         | 160         | 0.00%                                           |

In questo esempio, è possibile vedere l'andamento delle vendite dall'inizio del periodo di vendita all'ultimo periodo. Ad esempio, nel primo periodo 2024-01-01, le vendite di Apple sono state 100 rispetto agli ultimi dati di vendita di 150 per il periodo 2024-03-01. Questo indica un aumento del 50%.

### Esempio: utilizzo della funzione Last per calcolare la varianza del budget

Espressione del grafico

#### Panoramica

Un set di dati contiene le cifre del budget per trimestre, per un periodo di due anni. L'azienda vuole presentare l'importo totale del budget per ogni trimestre e la varianza tra l'importo del budget per il periodo corrente rispetto al budget trimestrale medio dell'anno.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - Year
  - Quarter
  - Budget

### Script di caricamento

Example:

```
Load * inline [
Year, Quarter, Budget
2023, Q1, 1200
2023, Q2, 1100
2023, Q3, 1300
2023, Q4, 1400
2024, Q1, 1550
2024, Q2, 1230
2024, Q3, 1400
2024, Q4, 1600
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella pivot e aggiungere questo campo come dimensione di riga:

- Year

Aggiungere questo campo come dimensione di colonna:

- Quarter

Creare le seguenti misure:

- =Sum(Budget), per calcolare il budget per il trimestre.
- =Sum(Budget) - RangeAvg>Last(Budget, 1, 4)), per calcolare la varianza tra l'importo di Budget per il periodo corrente rispetto al budget medio trimestrale per l'anno.

Tabella dei risultati

| Anno | Q1              |                                                                 | Q2              |                                                                 | Q3              |                                                                 | Q4              |                                                                 |
|------|-----------------|-----------------------------------------------------------------|-----------------|-----------------------------------------------------------------|-----------------|-----------------------------------------------------------------|-----------------|-----------------------------------------------------------------|
|      | Sum<br>(Budget) | Sum<br>(Budget)<br>-<br>RangeAvg<br>(Last<br>(Budget,<br>1, 4)) | Sum<br>(Budget) | Sum<br>(Budget)<br>-<br>RangeAvg<br>(Last<br>(Budget,<br>1, 4)) | Sum<br>(Budget) | Sum<br>(Budget)<br>-<br>RangeAvg<br>(Last<br>(Budget,<br>1, 4)) | Sum<br>(Budget) | Sum<br>(Budget)<br>-<br>RangeAvg<br>(Last<br>(Budget,<br>1, 4)) |
| 2023 | 1200            | -50                                                             | 1100            | -150                                                            | 1300            | 50                                                              | 1400            | 150                                                             |
| 2024 | 1550            | 105                                                             | 1230            | -215                                                            | 1400            | -45                                                             | 1600            | 155                                                             |

Utilizzando la funzione Last, è possibile vedere come varia l'importo del budget per trimestre rispetto al budget medio trimestrale. Ad esempio, nel 2023, il budget totale per l'anno è 5000 (1200 + 1100 + 1300 + 1400), quindi il budget medio trimestrale è 1250. Poiché l'importo del budget per q1 è 1200, ha un valore inferiore di 50 rispetto al budget medio trimestrale di 1250, mentre la varianza del budget per q4 è superiore di 150 rispetto al budget medio trimestrale.

## ColumnNo - funzione per grafici

**ColumnNo()** restituisce il numero della colonna attuale all'interno del segmento di riga attuale in una tabella pivot. La prima colonna è la numero 1.

### Sintassi:

**ColumnNo** ( [total] )

### Argomenti

| Argomento | Descrizione                                                                                                                                                        |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TOTAL     | Se la tabella è unidimensionale o se è utilizzato il qualificatore <b>TOTAL</b> come argomento, il segmento colonna attuale sarà sempre uguale all'intera colonna. |

Se una tabella pivot include più dimensioni orizzontali, il segmento della riga attuale includerà solo le colonne con gli stessi valori della colonna attuale in tutte le righe della dimensione, tranne per la riga che visualizza l'ultima dimensione orizzontale dell'ordinamento tra campi. La sequenza di ordinamento tra campi per le dimensioni orizzontali nelle tabelle pivot è definita semplicemente dall'ordine delle dimensioni dall'alto verso il basso.



*L'ordinamento sui valori y nei grafici o l'ordinamento per colonne di espressione nelle tabelle non è consentito quando si utilizza questa funzione di grafico in una qualsiasi delle espressioni del grafico. Queste alternative di ordinamento verranno perciò disattivate automaticamente. Quando si utilizza questa funzione di grafico in una visualizzazione o in una tabella, l'ordinamento della visualizzazione torna all'ordinamento inserito in questa funzione.*

### Esempio: espressione del grafico

| Esempio                                               | Risultato                                                                                                                                                                                                                                 |
|-------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IF( ColumnNo()=1, 0, Sum(Sales) / Before(Sum(Sales))) | Se ColumnNo è uguale a 1, il grafico restituirà zero. Altrimenti, restituirà il risultato Sum(Sales) / Before(Sum(Sales)). In questo esempio, il grafico restituirà i valori dalla colonna 2 in poi, mentre la colonna 1 restituirà zero. |

## Esempio: principi fondamentali della funzione ColumnNo

### Espressione del grafico

### Panoramica

Un set di dati contiene i dati di vendita trimestrali. Questo esempio moltiplica il valore delle vendite per ogni periodo per il numero di colonna di quel periodo.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Period`
  - `Sales`

### Script di caricamento

```
Example:
Load * inline [
Period, Sales
Q1, 1000
Q2, 2000
Q3, 3000
Q4, 4000
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella pivot e aggiungere questo campo come dimensione di colonna:

- `Period`

Creare la seguente misura:

- `=Sum(Sales) * ColumnNo()`, per calcolare la somma dei valori di vendita moltiplicati per il numero della colonna, che in questo esempio è il trimestre.

Tabella dei risultati

| Q1   | Q2   | Q3   | Q4    |
|------|------|------|-------|
| 1000 | 4000 | 9000 | 16000 |

La somma dei valori di vendita per ogni trimestre viene moltiplicata per l'output della funzione `ColumnNo`. Ad esempio, il valore delle vendite in Q4 è 4000, e l'output della funzione `ColumnNo` restituisce 4 perché Q4 è la quarta colonna. Pertanto, l'espressione di misura moltiplica 4000 per 4 e restituisce 16000.

### Esempio: scenario per la funzione `ColumnNo`

Espressione del grafico

### Panoramica

Un set di dati contiene i dati di vendita trimestrali. Un'azienda vuole aumentare i dati di vendita per T1 e T3 (le colonne dispari) del 5% e aumentare le vendite delle colonne rimanenti (T2 e T4) del 10%.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Period`
  - `Sales`

### Script di caricamento

```
Example:
Load * inline [
Period, Sales
Q1, 1000
Q2, 2000
Q3, 3000
Q4, 4000
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella pivot e aggiungere questo campo come dimensione di colonna:

- `Period`

Creare le seguenti misure:

- `=Sum(Sales)`, per calcolare la somma delle vendite.
- `=If(Odd(ColumnNo()), Amount * 1.05, Amount * 1.10)`, per aumentare gli importi delle vendite per le colonne dispari del 5% e per le colonne rimanenti del 10%.

Tabella dei risultati

| Q1             |                                                                    | Q2             |                                                                    | Q3             |                                                                    | Q4             |                                                                    |
|----------------|--------------------------------------------------------------------|----------------|--------------------------------------------------------------------|----------------|--------------------------------------------------------------------|----------------|--------------------------------------------------------------------|
| Sum<br>(Sales) | If(Odd<br>(ColumnNo<br>()), Amount<br>* 1.05,<br>Amount *<br>1.10) | Sum<br>(Sales) | If(Odd<br>(ColumnNo<br>()), Amount<br>* 1.05,<br>Amount *<br>1.10) | Sum<br>(Sales) | If(Odd<br>(ColumnNo<br>()), Amount<br>* 1.05,<br>Amount *<br>1.10) | Sum<br>(Sales) | If(Odd<br>(ColumnNo<br>()), Amount<br>* 1.05,<br>Amount *<br>1.10) |
| 1000           | 1050                                                               | 2000           | 2200                                                               | 3000           | 3150                                                               | 4000           | 4400                                                               |

I risultati mostrano che i valori di vendita delle colonne dispari, 1 e 3 (Q1 e Q3), sono aumentati del 5%, mentre le restanti colonne pari, 2 e 4 (Q2 e Q4), sono aumentate del 10%.

### Esempio: scenario avanzato per la funzione ColumnNo

Espressione del grafico

#### Panoramica

Questo esempio utilizza lo stesso set di dati dello scenario precedente per presentare il totale cumulativo per i valori in ogni periodo.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Period`
  - `Sales`

#### Script di caricamento

Example:

```
Load * inline [  
Period, Sales  
Q1, 1000  
Q2, 2000  
Q3, 3000  
Q4, 4000  
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella pivot e aggiungere questo campo come dimensione di colonna:

- `Period`

Creare la seguente misura:

- `=RangeSum(Before(Sum(Sales), 0, ColumnNo()))`, per calcolare il totale cumulativo.

Tabella dei risultati

| Q1   | Q2   | Q3   | Q4    |
|------|------|------|-------|
| 1000 | 3000 | 6000 | 10000 |

Questo esempio mostra come utilizzare la funzione `ColumnNo` per generare un totale parziale. La spiegazione che segue fornisce maggiori dettagli sull'espressione di misura: `RangeSum(Before(Sum(Sales), 0, ColumnNo()))`.

`sum(Sales)`: calcola la somma delle vendite per la cella corrente nella tabella pivot.

`Before(Sum(Sales), 0, ColumnNo())`: la funzione `Before` recupera i valori dalle colonne precedenti nella tabella pivot.

- L'argomento `sum(Sales)` assicura che la funzione recuperi la somma di `sales` per le celle precedenti.
- Il valore di offset di 0 specifica l'offset della colonna (rimanente nel segmento di riga corrente della colonna corrente).
- `ColumnNo()` determina il numero di colonne da recuperare quando si recuperano i dati. Rappresenta in modo dinamico la posizione della colonna corrente (ad esempio, per la colonna 3, `ColumnNo()` equivale a 3, quindi la funzione valuta le tre colonne precedenti).

`RangeSum()`: la funzione `RangeSum` calcola la somma dell'intervallo di valori forniti. Gestisce i valori null trattandoli come zero. In questa espressione, `RangeSum` genera effettivamente un totale cumulativo sommando tutti i valori `sales` precedenti recuperati dalla funzione `Before()`, oltre alle vendite della cella corrente.

### NoOfColumns - funzione per grafici

`NoOfColumns()` restituisce il numero di colonne nel segmento di riga attuale in una tabella pivot.



*L'ordinamento sui valori y nei grafici o l'ordinamento per colonne di espressione nelle tabelle non è consentito quando si utilizza questa funzione di grafico in una qualsiasi delle espressioni del grafico. Queste alternative di ordinamento verranno perciò disattivate automaticamente. Quando si utilizza questa funzione di grafico in una visualizzazione o in una tabella, l'ordinamento della visualizzazione torna all'ordinamento inserito in questa funzione.*

#### Sintassi:

`NoOfColumns([total])`

#### Argomenti

| Argomento    | Descrizione                                                                                                                                                        |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>TOTAL</b> | Se la tabella è unidimensionale o se è utilizzato il qualificatore <b>TOTAL</b> come argomento, il segmento colonna attuale sarà sempre uguale all'intera colonna. |

In una tabella pivot con più dimensioni orizzontali, il segmento di riga corrente include solo le colonne con lo stesso valore della colonna attuale in tutte le righe di dimensione, ad eccezione della riga che mostra l'ultima dimensione nell'ordinamento tra campi. La sequenza di ordinamento tra campi per le dimensioni orizzontali nelle tabelle pivot è definita semplicemente dall'ordine delle dimensioni dall'alto verso il basso.

Esempio: espressione del grafico

| Esempio                                                                | Risultato                                                                                                                                                                                                                                                                                                                                                               |
|------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>IF(ColumnNo() &gt; NoOfColumns() - 2, Sum(Sales), Null())</code> | Se <code>ColumnNo</code> è maggiore di <code>NoOfColumns</code> meno 2, il grafico restituirà il valore <code>Sum(Sales)</code> . Altrimenti, se <code>ColumnNo</code> è inferiore a questo valore, restituirà <code>NULL</code> . In questo esempio, dalla terza colonna in poi, il grafico restituirà i valori, mentre le colonne 1 e 2 non restituiranno alcun zero. |

### Esempio: principi fondamentali della funzione NoOfColumns

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Period`
  - `Sales`

#### Script di caricamento

```
Example:
Load * inline [
Period, Sales
Q1, 1000
Q2, 2000
Q3, 3000
Q4, 4000
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella pivot e aggiungere questo campo come dimensione di colonna:

- `Period`

Creare le seguenti misure:

- `=Sum(Sales)`, per calcolare la somma delle vendite.
- `=NoOfColumns()`, per calcolare il numero di colonne.

Results table

| Q1          |               | Q2          |               | Q3          |               | Q4          |               |
|-------------|---------------|-------------|---------------|-------------|---------------|-------------|---------------|
| Sum (Sales) | NoOfColumns() | Sum (Sales) | NoOfColumns() | Sum (Sales) | NoOfColumns() | Sum (Sales) | NoOfColumns() |
| 1000        | 4             | 2000        | 4             | 3000        | 4             | 4000        | 4             |

I risultati mostrano come la somma delle vendite per i periodi dal Q1 al Q4. La funzione `NoOfColumns` restituisce il valore 4, perché sono presenti 4 colonne per il segmento di riga corrente.

### Esempio: scenario per la funzione `NoOfColumns`

Espressione del grafico

#### Panoramica

Un set di dati contiene i dati di vendita trimestrali. Bisogna mostrare i valori delle vendite solo per gli ultimi due trimestri (T3 e T4) nella tabella pivot.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Product`
  - `Quarter`
  - `Sales`

#### Script di caricamento

Example:

```
Load * inline [
Product, Quarter, Sales
Apple, Q1, 100
Apple, Q2, 120
Apple, Q3, 140
Apple, Q4, 160
Banana, Q1, 150
Banana, Q2, 180
Banana, Q3, 200
Banana, Q4, 220
Cherry, Q1, 80
Cherry, Q2, 90
Cherry, Q3, 110
Cherry, Q4, 150
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella pivot e aggiungere questo campo come dimensione di riga:

- Product

Aggiungere questo campo come dimensione di colonna:

- Quarter

Creare la seguente misura:

- `=IF(ColumnNo() > NoOfColumns() - 2, Sum(Sales), Null())`, per visualizzare i valori solo delle ultime due colonne della tabella pivot (Q3 e Q4).

Results table

| Product | Q1 | Q2 | Q3  | Q4  |
|---------|----|----|-----|-----|
| Apple   | -  | -  | 140 | 160 |
| Banana  | -  | -  | 200 | 220 |
| Cherry  | -  | -  | 110 | 150 |

I risultati mostrano i valori delle vendite solo per le colonne 3 e 4, gli ultimi due trimestri dell'anno.

## 8.17 Funzioni logiche

In questa sezione vengono descritte le funzioni di gestione delle operazioni logiche. Tutte le funzioni possono essere utilizzate sia nello script di caricamento dei dati che nelle espressioni grafiche.

### IsNum

Restituisce -1 (True) se l'espressione può essere interpretata come numero, altrimenti restituisce 0 (False).

```
IsNum( expr )
```

### IsText

Restituisce -1 (True) se l'espressione presenta una rappresentazione di testo, altrimenti restituisce 0 (False).

```
IsText( expr )
```



Entrambe **IsNum** e **IsText** restituiscono 0 se l'espressione è NULL.

### Esempio:

Nel seguente esempio viene caricata una tabella inline contenente testo e valori numerici misti e vengono aggiunti due campi per verificare se il valore è, rispettivamente, un valore numerico o un valore di testo.

```
Load *, IsNum(Value), IsText(Value)
Inline [
Value
23
Green
Blue
12
33Red];
```

La tabella risultante avrà l'aspetto seguente:

Resulting table

| Value | IsNum(Value) | IsText(Value) |
|-------|--------------|---------------|
| 23    | -1           | 0             |
| Green | 0            | -1            |
| Blue  | 0            | -1            |
| 12    | -1           | 0             |
| 33Red | 0            | -1            |

## 8.18 Funzioni di mapping

In questa sezione vengono descritte le funzioni di gestione delle tabelle di mapping. È possibile utilizzare una tabella di mapping per sostituire valori o nomi di campo durante l'esecuzione dello script.

È possibile utilizzare le funzioni di mapping solo nello script di caricamento dei dati.

### Panoramica sulle funzioni di mapping

Ciascuna funzione viene descritta ulteriormente dopo la panoramica. È inoltre possibile fare clic sul nome della funzione nella sintassi per accedere immediatamente ai dettagli per tale funzione specifica.

#### ApplyMap

La funzione di script **ApplyMap** viene utilizzata per eseguire il mapping dell'output di un'espressione a una tabella di mapping caricata in precedenza.

```
ApplyMap ('mapname', expr [ , defaultexpr ] )
```

### MapSubstring

La funzione di script **MapSubstring** consente di eseguire il mapping di parti di un'espressione a una tabella di mapping caricata in precedenza. Il mapping rispetta la distinzione tra maiuscole e minuscole e non è iterativo, mentre il mapping delle sottostringhe viene eseguito da sinistra a destra.

```
MapSubstring ('mapname', expr)
```

### ApplyMap

La funzione di script **ApplyMap** viene utilizzata per eseguire il mapping dell'output di un'espressione a una tabella di mapping caricata in precedenza.

#### Sintassi:

```
ApplyMap('map_name', expression [ , default_mapping ] )
```

**Tipo di dati restituiti:** duale

#### Argomenti:

##### Argomenti

| Argomento       | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| map_name        | Il nome di una tabella di mapping che è stata creata in precedenza mediante l'istruzione <b>mapping load</b> o <b>mapping select</b> . Il nome deve essere incluso in virgolette singole diritte.<br><br><div>  <p><i>Se si utilizza questa funzione in una variabile con espansione macro e si fa riferimento a una tabella di mapping non esistente, la chiamata alla funzione ha esito negativo e non viene creato un campo.</i></p> </div> |
| expression      | L'espressione, il cui risultato verrà mappato.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| default_mapping | Se definito, questo valore verrà utilizzato come un valore predefinito nel caso in cui la tabella di mapping non contenga un valore corrispondente per expression. Se non definito, il valore di expression verrà restituito così com'è.                                                                                                                                                                                                                                                                                          |



*Il campo di output di ApplyMap non deve avere lo stesso nome di uno dei suoi campi di input. In caso contrario, possono essere generati risultati imprevisti. Esempio da non utilizzare:*  
*ApplyMap('Map', A) as A.*

#### Esempio:

In questo esempio viene caricato un elenco del personale addetto alle vendite con un codice paese che ne identifica il paese di residenza. Viene utilizzata una tabella per il mapping di un codice paese a un paese al fine di per sostituire il codice paese con il relativo nome. Nella tabella di mapping vengono definiti solo tre paesi, mentre gli altri codici paese vengono mappati a 'Rest of the world'.

```
// Load mapping table of country codes:
map1:
mapping LOAD *
Inline [
CCode, Country
Sw, Sweden
Dk, Denmark
No, Norway
] ;

// Load list of salesmen, mapping country code to country
// If the country code is not in the mapping table, put Rest of the world
Salespersons:
LOAD *,
ApplyMap('map1', CCode,'Rest of the world') As Country
Inline [
CCode, Salesperson
Sw, John
Sw, Mary
Sw, Per
Dk, Preben
Dk, Olle
No, Ole
Sf, Risttu
] ;

// We don't need the CCode anymore
Drop Field 'CCode';
La tabella risultante (Salespersons) avrà l'aspetto seguente:
```

Resulting table

| Salesperson | Country           |
|-------------|-------------------|
| John        | Sweden            |
| Mary        | Sweden            |
| Per         | Sweden            |
| Preben      | Denmark           |
| Olle        | Denmark           |
| Ole         | Norway            |
| Risttu      | Rest of the world |

## MapSubstring

La funzione di script **MapSubstring** consente di eseguire il mapping di parti di un'espressione a una tabella di mapping caricata in precedenza. Il mapping rispetta la distinzione tra maiuscole e minuscole e non è iterativo, mentre il mapping delle sottostringhe viene eseguito da sinistra a destra.

### Sintassi:

```
MapSubstring('map_name', expression)
```

**Tipo di dati restituiti:** stringa

### Argomenti:

#### Argomenti

| Argomento  | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| map_name   | <p>Il nome di un tabella di mapping letta in precedenza da un'istruzione <b>mapping load</b> o <b>mapping select</b>. Il nome deve essere racchiuso tra virgolette singole diritte.</p> <div><i>Se si utilizza questa funzione in una variabile con espansione macro e si fa riferimento a una tabella di mapping non esistente, la chiamata alla funzione ha esito negativo e non viene creato un campo.</i></div> |
| expression | L'espressione il cui risultato deve essere mappato dalle sottostringhe.                                                                                                                                                                                                                                                                                                                                                                                                                              |

### Esempio:

In questo esempio verrà caricato un elenco di modelli del prodotto. Ogni modello presenta una serie di attributi che sono descritti da un codice composto. Utilizzando la tabella di mapping con MapSubstring, è possibile espandere i codici degli attributi per ottenere una descrizione.

```
map2:
mapping LOAD *
inline [
AttCode, Attribute
R, Red
Y, Yellow
B, Blue
C, Cotton
P, Polyester
S, Small
M, Medium
L, Large
] ;

Productmodels:
LOAD *,
MapSubString('map2', AttCode) as Description
inline [
Model, AttCode
Twixie, R C S
Boomer, B P L
Raven, Y P M
Seedling, R C L
SeedlingPlus, R C L with hood
```

```
Younger, B C with patch  
MultiStripe, R Y B C S/M/L  
] ;  
// We don't need the AttCode anymore  
Drop Field 'AttCode';
```

La tabella risultante avrà l'aspetto seguente:

Resulting table

| Model        | Description                               |
|--------------|-------------------------------------------|
| Twixie       | Red Cotton Small                          |
| Boomer       | Blue Polyester Large                      |
| Raven        | Yellow Polyester Medium                   |
| Seedling     | Red Cotton Large                          |
| SeedlingPlus | Red Cotton Large with hood                |
| Younger      | Blue Cotton with patch                    |
| MultiStripe  | Red Yellow Blue Cotton Small/Medium/Large |

### 8.19 Funzioni matematiche

In questa sezione vengono descritte le funzioni per le costanti matematiche e i valori booleani. Queste funzioni non presentano parametri, tuttavia le parentesi sono obbligatorie.

Tutte le funzioni possono essere utilizzate sia nello script di caricamento dei dati che nelle espressioni grafiche.

#### **e**

La funzione restituisce la base degli algoritmi naturali, **e** (2.71828...).

```
e ( )
```

#### **false**

La funzione restituisce un valore duale con il valore di testo 'False' e il valore numerico 0, che può essere utilizzato come valore logico false nelle espressioni.

```
false ( )
```

#### **pi**

La funzione restituisce il valore di  $\pi$  (3.14159...).

```
pi ( )
```

#### **rand**

La funzione restituisce un numero casuale tra 0 e 1. Può essere utilizzata per creare dati campione.

```
rand ( )
```

### Esempio:

Questo script di esempio crea una tabella di 1000 record con caratteri maiuscoli selezionati casualmente, vale a dire caratteri compresi nell'intervallo tra 65 e 91 (65+26).

```
Load
    Chr( Floor(rand() * 26) + 65) as UCaseChar,
    RecNo() as ID
Autogenerate 1000;
```

### true

La funzione restituisce un valore duale con il valore di testo 'True' e il valore numerico -1, che può essere utilizzato come valore logico true nelle espressioni.

```
true( )
```

## 8.20 Funzioni di NULL

In questa sezione vengono descritte le funzioni per la restituzione o il rilevamento dei valori NULL.

Tutte le funzioni possono essere utilizzate sia nello script di caricamento dei dati che nelle espressioni grafiche.

### Panoramica sulle funzioni di NULL

Ciascuna funzione viene descritta ulteriormente dopo la panoramica. È inoltre possibile fare clic sul nome della funzione nella sintassi per accedere immediatamente ai dettagli per tale funzione specifica.

#### EmptyIsNull

La funzione **EmptyIsNull** converte stringhe vuote in NULL. Pertanto, restituisce NULL se il parametro è una stringa vuota, altrimenti restituisce il parametro.

```
EmptyIsNull (expr )
```

#### IsNull

La funzione **IsNull** verifica se il valore di un'espressione sia NULL restituendo -1 (True) in caso affermativo e 0 (False) in caso contrario.

```
IsNull (expr )
```

#### Null

La funzione **Null** restituisce un valore null.

```
NULL ( )
```

### EmptyIsNull

La funzione **EmptyIsNull** converte stringhe vuote in NULL. Pertanto, restituisce NULL se il parametro è una stringa vuota, altrimenti restituisce il parametro.

### Sintassi:

```
EmptyIsNull (exp )
```

**Tipo di dati restituiti:** NULL se il parametro è una stringa vuota, altrimenti restituisce il parametro.

### Argomenti

| Argomento | Descrizione                                             |
|-----------|---------------------------------------------------------|
| expr      | L'espressione o il campo contenente i dati da misurare. |

### Esempio: espressioni del grafico

| Esempio                                     | Risultato                                                                                                                                                                                                                                               |
|---------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EmptyIsNull(AdditionalComments)             | Questa espressione restituirà NULL per qualsiasi valore di stringa vuoto per il campo AdditionalComments, al posto delle stringhe vuote. Vengono restituite le stringhe e i numeri non vuoti.                                                           |
| EmptyIsNull(PurgeChar(PhoneNumber, ' -()')) | Questa espressione rimuoverà qualsiasi trattino, spazio e parentesi dal campo PhoneNumber. Se non sono rimasti caratteri, la funzione EmptyIsNull restituisce la stringa vuota come NULL; un phone number corrisponde a un valore phone number assente. |

### Esempio: principi fondamentali della funzione EmptyIsNull

Espressione del grafico

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata Example.
- I seguenti campi nella tabella dati:
  - CustomerID
  - OrderID
  - Product

- Status
- PhoneNumber

### Script di caricamento

Example:

```
LOAD * inline [
CustomerID, OrderID, Product, Status, PhoneNumber
1, 1001, widget,, (0123) 456 789
2, 1002, Gizmo, Open,-
3, 1003, Gadget, Closed,()
4, 1004, widget,,678-9888
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- CustomerID
- OrderID
- Product
- Status

Creare le seguenti dimensioni calcolate:

- =EmptyIsNull(Status), per restituire come NULL qualsiasi valore di stringa vuoto nel campo Status.
- =EmptyIsNull(PurgeChar(PhoneNumber, ' -() ')), per rimuovere qualsiasi trattino, spazio e parentesi dal campo PhoneNumber. Se non sono rimasti caratteri, la funzione EmptyIsNull restituisce la stringa vuota come NULL; un numero di telefono vuoto corrisponde a un numero di telefono assente.



La tabella seguente contiene delle celle vuote per dimostrare l'uso della funzione EmptyIsNull.

Results table

| CustomerID | OrderID | Product | Status | EmptyIsNull<br>(Status) | EmptyIsNull(PurgeChar<br>(PhoneNumber, ' -() ')) |
|------------|---------|---------|--------|-------------------------|--------------------------------------------------|
| 1          | 1001    | Widget  |        | -                       | 0123456789                                       |
| 2          | 1002    | Gizmo   | Open   | Open                    | -                                                |
| 3          | 1003    | Gadget  | Closed | Closed                  | -                                                |
| 4          | 1004    | Widget  |        | -                       | 6789888                                          |

Nella colonna status, notare che i record con valori vuoti appaiono come celle vuote con un colore di sfondo trasparente predefinito. L'output della misura EmptyIsNull(Status) restituisce NULL per questi valori vuoti, come indicato dal carattere trattino (-) con sfondo grigio della cella.

La seconda misura utilizza la funzione `IsEmpty` con il campo [PurgeChar \(page 1817\)](#) per rimuovere i trattini (-), gli spazi e le parentesi ( ) dal campo `PhoneNumber` e quindi restituire le stringhe vuote rimanenti come `NULL`. Le celle con valori null ora appaiono con un trattino (-) e una formattazione con sfondo grigio.

### Esempio: sostituzione di valori nulli con testo e aggiunta di formattazione condizionale

Espressione del grafico

#### Panoramica

Un responsabile vendite vuole analizzare i dati delle vendite e identificare facilmente i record che contengono valori mancanti.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `ID`
  - `Name`
  - `Age`
  - `Email`
  - `Sales`
  - `Region`
  - `OrderStatus`

#### Script di caricamento

Example:

```
LOAD * inline [
ID,Name, Age, Email, Sales, Region, OrderStatus
1, John Smith, 30, john@email.com, 1000, North, Active
2, Jane Doe, '', jane@email.com, 1500, South, Active
3, Bob Johnson, 45, '', 800, East, Inactive
4, Alice Brown, 28, alice@email.com, '', West, Active
5, Charlie Lee, '', charlie@email.com, 1200, '', Active
6, Eva Green, 35, eva@email.com, 950, North, ''
7, David White, 50, '', '', South, Inactive
8, Fiona Black, 42, fiona@email.com, 1100, East, Active
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `ID`

Creare le seguenti dimensioni calcolate:

- `=If(IsNull(EmptyIsNull(Age)), 'NULL', EmptyIsNull(Age))`, per restituire come NULL qualsiasi valore di stringa vuoto nel campo Age, e quindi identificare e popolare tutti i valori null con il valore NULL anziché con un carattere trattino (-).
- `=If(IsNull(EmptyIsNull(Email)), 'NULL', EmptyIsNull(Email))`, per restituire come NULL qualsiasi valore di stringa vuoto nel campo Email, e quindi identificare e popolare tutti i valori null con il valore NULL anziché con un carattere trattino (-).
- `=If(IsNull(EmptyIsNull(Sales)), 'NULL', EmptyIsNull(Sales))`, per restituire come NULL qualsiasi valore di stringa vuoto nel campo Sales, e quindi identificare e popolare tutti i valori null con il valore NULL anziché con un carattere trattino (-).
- `=If(IsNull(EmptyIsNull(Region)), 'NULL', EmptyIsNull(Region))`, per restituire come NULL qualsiasi valore di stringa vuoto nel campo Region, e quindi identificare e popolare tutti i valori null con il valore NULL anziché con un carattere trattino (-).
- `=If(IsNull(EmptyIsNull(OrderStatus)), 'NULL', EmptyIsNull(OrderStatus))`, per restituire come NULL qualsiasi valore di stringa vuoto nel campo OrderStatus, e quindi identificare e popolare tutti i valori null con il valore NULL anziché con un carattere trattino (-).

Impostare la formattazione condizionale delle celle per cambiare il colore di sfondo in rosso per i valori NULL e in verde per tutti gli altri valori. Nel pannello **Proprietà** per ogni dimensione e misura, inserire le seguenti espressioni come **Espressione colore di sfondo**.

- ID (dimensione): `=RGB(200,255,200)`
- Age (misura): `=If(IsNull(EmptyIsNull(Age)), RGB(255,200,200), RGB(200,255,200))`
- Email (misura): `=If(IsNull(EmptyIsNull(Email)), RGB(255,200,200), RGB(200,255,200))`
- Sales (misura): `=If(IsNull(EmptyIsNull(Sales)), RGB(255,200,200), RGB(200,255,200))`
- Region (misura): `=If(IsNull(EmptyIsNull(Region)), RGB(255,200,200), RGB(200,255,200))`
- OrderStatus (misura): `=If(IsNull(EmptyIsNull(OrderStatus)), RGB(255,200,200), RGB(200,255,200))`

Results table

| ID | If(IsNull<br>(EmptyIsNull<br>(Age)), 'NULL',<br>EmptyIsNull<br>(Age)) | If(IsNull<br>(EmptyIsNull<br>(Email)), 'NULL',<br>EmptyIsNull(Email)) | If(IsNull<br>(EmptyIsNull<br>(Sales)),<br>'NULL',<br>EmptyIsNull<br>(Sales)) | If(IsNull<br>(EmptyIsNull<br>(Region)),<br>'NULL',<br>EmptyIsNull<br>(Region)) | If(IsNull<br>(EmptyIsNull<br>(OrderStatus)),<br>'NULL',<br>EmptyIsNull<br>(OrderStatus)) |
|----|-----------------------------------------------------------------------|-----------------------------------------------------------------------|------------------------------------------------------------------------------|--------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
| 1  | 30                                                                    | john@email.com                                                        | 1000                                                                         | North                                                                          | Active                                                                                   |
| 2  | NULL                                                                  | jane@email.com                                                        | 1500                                                                         | South                                                                          | Active                                                                                   |
| 3  | 45                                                                    | NULL                                                                  | 800                                                                          | East                                                                           | Inactive                                                                                 |
| 4  | 28                                                                    | alice@email.com                                                       | NULL                                                                         | West                                                                           | Active                                                                                   |
| 5  | NULL                                                                  | charlie@email.com                                                     | 1200                                                                         | NULL                                                                           | Active                                                                                   |

| ID | If(IsNull<br>(EmptyIsNull<br>(Age)), 'NULL',<br>EmptyIsNull<br>(Age)) | If(IsNull<br>(EmptyIsNull<br>(Email)), 'NULL',<br>EmptyIsNull(Email)) | If(IsNull<br>(EmptyIsNull<br>(Sales)),<br>'NULL',<br>EmptyIsNull<br>(Sales)) | If(IsNull<br>(EmptyIsNull<br>(Region)),<br>'NULL',<br>EmptyIsNull<br>(Region)) | If(IsNull<br>(EmptyIsNull<br>(OrderStatus)),<br>'NULL',<br>EmptyIsNull<br>(OrderStatus)) |
|----|-----------------------------------------------------------------------|-----------------------------------------------------------------------|------------------------------------------------------------------------------|--------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
| 6  | 35                                                                    | eva@email.com                                                         | 950                                                                          | North                                                                          | NULL                                                                                     |
| 7  | 50                                                                    | NULL                                                                  | NULL                                                                         | South                                                                          | Inactive                                                                                 |
| 8  | 42                                                                    | fiona@email.com                                                       | 1100                                                                         | East                                                                           | Active                                                                                   |

I risultati mostrano che utilizzando la funzione `EmptyIsNull` e aggiungendo la formattazione condizionale, è possibile identificare facilmente i record con valori mancanti.

## IsNull

La funzione `IsNull` verifica se il valore di un'espressione sia NULL restituendo -1 (True) in caso affermativo e 0 (False) in caso contrario.

### Sintassi:

**IsNull** (expr )



Una stringa con lunghezza zero non è considerata NULL e farà sì che **IsNull** restituisca False.

**Tipo di dati restituiti:** Booleano

### Argomenti

| Argomento | Descrizione                                             |
|-----------|---------------------------------------------------------|
| expr      | L'espressione o il campo contenente i dati da misurare. |

### Esempio: espressione del grafico

| Esempio                          | Risultato                                                                                             |
|----------------------------------|-------------------------------------------------------------------------------------------------------|
| <code>IsNull(Productname)</code> | Restituisce -1 (Vero) se <code>ProductName</code> è un valore null, altrimenti restituisce 0 (Falso). |

## Esempio: principi fondamentali della funzione IsNull

Espressione del grafico

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `ID`
  - `Value`

### Script di caricamento

Example:

```
LOAD * Inline
[ID, Value
0,
1,NULL
2,-
3,378];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `ID`
- `Value`

Creare le seguenti dimensioni calcolate:

- `=If(len(trim(Value))= 0 or Value='NULL' or Value='-', Null(), Value )`, per restituire `NULL` se il campo `value` non ha alcun valore, un carattere trattino (-) o il valore `NULL`, altrimenti restituisce il valore.
- `=If(IsNull(If(len(trim(Value))= 0 or Value='NULL' or Value='-', Null(), Value )), 'T', 'F')`, per restituire il valore `T` (Vero) o `F` (Falso) a seconda che il campo `value` contenga o meno un valore `null`.

Results table

| ID | Value | If(len(trim(Value))= 0 or Value='NULL' or Value='-', Null(), Value ) | If(IsNull(If(len(trim(Value))= 0 or Value='NULL' or Value='-', Null(), Value )), 'T', 'F') |
|----|-------|----------------------------------------------------------------------|--------------------------------------------------------------------------------------------|
| 0  |       | -                                                                    | T                                                                                          |
| 1  | NULL  | -                                                                    | T                                                                                          |
| 2  | -     | -                                                                    | T                                                                                          |
| 3  | 378   | 378                                                                  | F                                                                                          |

La colonna `value` mostra i dati come sono stati inseriti nel set di dati. La prima misura (colonna 3) restituisce `NULL` per le prime 3 righe il cui valore ha soddisfatto le condizioni dell'istruzione `IF: len(trim(Value))= 0 or Value='NULL' or Value='-'`. L'ultima riga restituisce il valore inserito, 378.

La seconda misura (colonna 4) prende il valore dell'espressione nella colonna 3 e restituisce T se NULL o F se non NULL. I risultati mostrano che le prime 3 righe sono valori null.

### Script di caricamento

In questo esempio viene caricata una tabella inline con quattro righe, in cui le prime tre righe non contengono alcun valore oppure contengono - o NULL nella colonna `value`. Questi valori verranno convertiti in rappresentazioni di valori NULL true con l'istruzione centrale preceding **LOAD** mediante la funzione **Null**.

La prima istruzione **LOAD** aggiunge un campo che consente di verificare se il valore è NULL mediante la funzione **IsNull**.

```
NullsDetectedAndConverted:
```

```
LOAD *,
If(IsNull(ValueNullConv), 'T', 'F') as IsItNull;
```

```
LOAD *,
If(len(trim(value))= 0 or value='NULL' or value='-', Null(), value ) as valueNullConv;
```

```
LOAD * Inline
[ID, value
0,
1,NULL
2,-
3, 378];
```

Qui sotto è riportata la tabella risultante. Nella colonna `valueNullConv` i valori NULL vengono rappresentati da -.

Results table

| ID | Value | ValueNullConv | IsItNull |
|----|-------|---------------|----------|
| 0  |       | -             | T        |
| 1  | NULL  | -             | T        |
| 2  | -     | -             | T        |
| 3  | 378   | 378           | F        |

### Esempio: scenario per la funzione IsNull

Espressione del grafico

### Panoramica

Un set di dati sui dati delle vendite viene analizzato per verificare quali campi hanno valori null.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Una variabile `NULLINTERPRET` impostata su `NULL`.
- I seguenti campi nella tabella dati:
  - `SalesID`
  - `Name`
  - `Age`
  - `Email`
  - `Region`
  - `Status`

### Script di caricamento

```
Set NULLINTERPRET = NULL;
Example:
LOAD * Inline [
SalesID, Name, Age, Email, Region, Status
1, John Smith, 30, john@email.com, North, Active
2, Jane Doe, NULL, jane@email.com, South, Active
3, Bob Johnson, 45, NULL, East, Inactive
4, Alice Brown, 28, alice@email.com, West, Active
5, Charlie Lee, NULL, charlie@email.com, NULL, Active
6, Eva Green, 35, eva@email.com, North, NULL
7, David White, 50, NULL, South, Inactive
8, Fiona Black, 42, fiona@email.com, East, Active
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `SalesID`
- `Name`

Creare le seguenti dimensioni calcolate:

- `=If(IsNull(Age), 'Age unavailable', Age)`, per popolare il campo `Age` con il valore `Age` `unavailable` se è `NULL`.
- `=If(IsNull(Email), 'Email unavailable', Email)`, per popolare il campo `Email` con il valore `Email` `unavailable` se è `NULL`.
- `=If(IsNull(Region), 'Region unavailable', Region)`, per popolare il campo `Region` con il valore `Region` `unavailable` se è `NULL`.

Results table

| SalesID | Name        | If(IsNull(Age), 'Age unavailable', Age) | If(IsNull(Email), 'Email unavailable', Email) | If(IsNull(Region), 'Region unavailable', Region) |
|---------|-------------|-----------------------------------------|-----------------------------------------------|--------------------------------------------------|
| 1       | John Smith  | 30                                      | john@email.com                                | North                                            |
| 2       | Jane Doe    | Age unavailable                         | jane@email.com                                | South                                            |
| 3       | Bob Johnson | 45                                      | Email unavailable                             | East                                             |
| 4       | Alice Brown | 28                                      | alice@email.com                               | West                                             |
| 5       | Charlie Lee | Age unavailable                         | charlie@email.com                             | Region unavailable                               |
| 6       | Eva Green   | 35                                      | eva@email.com                                 | North                                            |
| 7       | David White | 50                                      | Email unavailable                             | South                                            |
| 8       | Fiona Black | 42                                      | fiona@email.com                               | East                                             |

I risultati della funzione `IsNull`, che verifica e restituisce vero (true) se vengono trovati valori null, indicano che diversi campi del set di dati contengono valori null. Ad esempio, è stato identificato un valore null nella colonna Age per Charlie Lee, ed è stato restituito l'output Age unavailable.

### NULL

La funzione **Null** restituisce un valore null.

#### Sintassi:

```
Null ( )
```

**Tipo di dati restituiti:** NULL

Esempio: espressioni dei grafici

| Esempio                                            | Risultato                                                                                                         |
|----------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| <code>If(Product = 'A', Null(), Sum(Sales))</code> | Questa espressione restituisce NULL per Product A, altrimenti restituisce la somma delle vendite per il prodotto. |

### Esempio: principi fondamentali della funzione Null

Espressione del grafico

#### Panoramica

Per dimostrare i diversi usi della funzione `Null`, viene utilizzato un semplice set di dati.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `ID`
  - `Value`

#### Script di caricamento

Example:

```
LOAD * Inline
[ID, Value
0,
1, NULL
2, -
3, 378];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `ID`
- `Value`

Creare le seguenti dimensioni calcolate:

- `=If(Length(trim(Value))= 0 or Value='NULL' or Value='-', Null(), Value )`, per restituire `NULL` se il campo `Value` ha uno dei valori seguenti: nessun valore, un carattere trattino (-) o il valore `NULL`, altrimenti restituisce il valore.
- `=If(IsNull(If(Length(trim(Value))= 0 or Value='NULL' or Value='-', Null(), Value )), 'T', 'F')`, per restituire il valore `T` (Vero) o `F` (Falso) a seconda che il campo `Value` contenga o meno un valore `null`.

Results table

| ID | Value | If(len(trim(Value))= 0 or Value='NULL' or Value='- ', Null(), Value ) | If(IsNull(If(len(trim(Value))= 0 or Value='NULL' or Value='- ', Null(), Value )), 'T', 'F') |
|----|-------|-----------------------------------------------------------------------|---------------------------------------------------------------------------------------------|
| 0  |       | -                                                                     | T                                                                                           |
| 1  | NULL  | -                                                                     | T                                                                                           |
| 2  | -     | -                                                                     | T                                                                                           |
| 3  | 378   | 378                                                                   | F                                                                                           |

La colonna `value` mostra i dati come sono stati inseriti nel set di dati. La prima misura (colonna 3) usa la funzione `Null` per restituire `NULL` per quei valori che hanno soddisfatto le condizioni dell'istruzione `IF`: `len(trim(Value))= 0 or Value='NULL' or Value='- '`. Ad esempio, le prime 3 righe soddisfano i requisiti e restituiscono `NULL`. L'ultima riga restituisce il valore inserito, 378.

La seconda misura (colonna 4) prende il valore nella colonna 3 e restituisce `T` se `NULL` o `F` se non `NULL`. I risultati mostrano che le prime 3 righe sono valori null.

### Script di caricamento

In questo esempio viene caricata una tabella inline con quattro righe, in cui le prime tre righe non contengono alcun valore oppure contengono - o `NULL` nella colonna `value`. Bisogna convertire questi valori in rappresentazioni di valore `NULL` true (vere).

L'istruzione centrale preceding **LOAD** esegue la conversione utilizzando la funzione `Null`.

La prima istruzione **LOAD** precedente aggiunge un campo, per verificare se il valore corrisponde a `NULL`, solo a scopo illustrativo in questo esempio.

`NullsDetectedAndConverted:`

```
LOAD *,
If(IsNull(ValueNullConv), 'T', 'F') as IsItNull;
```

```
LOAD *,
If(len(trim(Value))= 0 or Value='NULL' or Value='- ', Null(), value ) as ValueNullConv;
```

`LOAD * Inline`

`[ID, value`

`0,`

`1,NULL`

`2,-`

`3, 386];`

Qui sotto è riportata la tabella risultante. Nella colonna `ValueNullConv` i valori `NULL` vengono rappresentati da -.

Results table

| ID | Value | ValueNullConv | IsItNull |
|----|-------|---------------|----------|
| 0  |       | -             | T        |

| ID | Value | ValueNullConv | IsNull |
|----|-------|---------------|--------|
| 1  | NULL  | -             | T      |
| 2  | -     | -             | T      |
| 3  | 386   | 386           | F      |

### Esempio: esclusione dei valori utilizzando la funzione Null

Espressione del grafico

#### Panoramica

I dati sulle vendite vengono analizzati per determinare le vendite totali e quelle che non sono correlate a Product A.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata Example.
- I seguenti campi nella tabella dati:
  - Product
  - Sales

#### Script di caricamento

```
Example:  
LOAD * Inline [  
Product, Sales  
A, 100  
B, 150  
C, 200  
D, 250  
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- Product

Creare le seguenti misure:

- =Sum(Sales), per calcolare le vendite totali.
- =If(Product = 'A', Null(), Sum(Sales)), per calcolare le vendite totali in cui Product non è A.

Results table

| Product | Sum(Sales) | If(Product = 'A', Null(), Sum(Sales)) |
|---------|------------|---------------------------------------|
| A       | 100        | -                                     |
| B       | 150        | 150                                   |
| C       | 200        | 200                                   |
| D       | 250        | 250                                   |

Questo esempio mostra un modo semplice per calcolare le vendite totali e poi escludere le vendite di un prodotto specifico, utilizzando la funzione `Null` per impostare il valore su `NULL`. Ad esempio, nella seconda misura, tutti i valori di `Product` A hanno restituito `NULL` e sono stati esclusi.

## 8.21 Funzioni di scala

Le funzioni scala sono funzioni che elaborano una matrice di valori e restituiscono un valore unico. Tutte le funzioni scala possono essere utilizzate sia nello script di caricamento dei dati che nelle espressioni grafiche.

Ad esempio, in una visualizzazione, una funzione scala è in grado di calcolare un valore singolo da una matrice intra-record. Nello script di caricamento dei dati, una funzione scala è in grado di calcolare un valore singolo dalla matrice di valori in una tabella interna.



*Le funzioni scala sostituiscono le seguenti funzioni numeriche generiche: **numsum**, **numavg**, **numcount**, **nummin** e **nummax**, che devono essere ritenute obsolete.*

### Funzioni di scala di base

#### RangeMax

**RangeMax()** restituisce i valori numerici più alti trovati nel campo o nell'espressione.

```
RangeMax (first_expr[, Expression])
```

#### RangeMaxString

**RangeMaxString()** restituisce l'ultimo valore nell'ordine del testo che trova nel campo o nell'espressione.

```
RangeMaxString (first_expr[, Expression])
```

#### RangeMin

**RangeMin()** restituisce i valori numerici più bassi trovati nel campo o nell'espressione.

```
RangeMin (first_expr[, Expression])
```

#### RangeMinString

**RangeMinString()** restituisce il primo valore nell'ordine del testo che trova nel campo o nell'espressione.

```
RangeMinString (first_expr[, Expression])
```

### RangeMode

**RangeMode()** trova il valore più comune (valore della modalità) nel campo o nell'espressione.

```
RangeMode (first_expr[, Expression])
```

### RangeOnly

**RangeOnly()** è una funzione duale che restituisce un valore se l'espressione viene valutata in base a un valore univoco. In caso contrario, viene restituito **NULL**.

```
RangeOnly (first_expr[, Expression])
```

### RangeSum

**RangeSum()** restituisce la somma di una scala di valori. Tutti i valori non numerici vengono trattati come 0.

```
RangeSum (first_expr[, Expression])
```

## Funzioni di scala di conteggio

### RangeCount

**RangeCount()** restituisce il numero di valori, sia testuali che numerici, nell'espressione o nel campo.

```
RangeCount (first_expr[, Expression])
```

### RangeMissingCount

**RangeMissingCount()** restituisce il numero di valori non numerici (incluso NULL) nell'espressione o nel campo.

```
RangeMissingCount (first_expr[, Expression])
```

### RangeNullCount

**RangeNullCount()** trova il numero di valori NULL nel campo o nell'espressione.

```
RangeNullCount (first_expr[, Expression])
```

### RangeNumericCount

**RangeNumericCount()** trova il numero di valori numerici nel campo o nell'espressione.

```
RangeNumericCount (first_expr[, Expression])
```

### RangeTextCount

**RangeTextCount()** restituisce il numero di valori di testo nel campo o nell'espressione.

```
RangeTextCount (first_expr[, Expression])
```

## Funzioni di scala statistiche

### RangeAvg

**RangeAvg()** restituisce la media di una scala. L'input per la funzione può essere una scala di valori o un'espressione.

```
RangeAvg (first_expr[, Expression])
```

### RangeCorrel

**RangeCorrel()** restituisce il coefficiente di correlazione per due serie di dati. Il coefficiente di correlazione è una misura della relazione tra le serie di dati.

```
RangeCorrel (x_values , y_values[, Expression])
```

### RangeFractile

**RangeFractile()** restituisce il valore che corrisponde al **fractile** n-esimo (quantile) di un intervallo di numeri.

```
RangeFractile (fractile, first_expr[, Expression])
```

### RangeKurtosis

**RangeKurtosis()** restituisce il valore che corrisponde al kurtosis di una scala di numeri.

```
RangeKurtosis (first_expr[, Expression])
```

### RangeSkew

**RangeSkew()** restituisce il valore che corrisponde all'asimmetria di una scala di numeri.

```
RangeSkew (first_expr[, Expression])
```

### RangeStdev

**RangeStdev()** trova la deviazione standard di una scala di numeri.

```
RangeStdev (expr1[, Expression])
```

## Funzioni di scala finanziarie

### RangeIRR

**RangeIRR()** restituisce il tasso di rendimento interno per una serie di flussi di cassa rappresentati dai valori di input.

```
RangeIRR (value[, value][, Expression])
```

### RangeNPV

**RangeNPV()** restituisce il valore attuale netto di un investimento basato su un tasso di sconto e una serie di futuri pagamenti periodici (valori negativi) ed entrate (valori positivi). Il risultato presenta un formato numerico predefinito di **money**.

```
RangeNPV (discount_rate, value[, value][, Expression])
```

### RangeXIRR

**RangeXIRR()** restituisce il tasso di rendimento interno (annuale) per una programmazione di flussi di cassa che non è necessariamente su base periodica. Per calcolare il tasso di rendimento interno di una serie di flussi di cassa periodici, utilizzare la funzione **RangeIRR**.

```
RangeXIRR (values, dates[, Expression])
```

**RangeXNPV**

**XNPV()** restituisce il valore attuale netto per una programmazione di flussi di cassa (non necessariamente periodici) rappresentati da coppie di numeri nelle espressioni date da **pmt** e **date**. Tutti i pagamenti sono scontati in base ad un anno composto da 365 giorni.

```
RangeXNPV (discount_rate, values, dates[, Expression])
```

**Vedere anche:**

p [Funzioni intra-record \(page 1461\)](#)

**RangeAvg**

**RangeAvg()** restituisce la media di una scala. L'input per la funzione può essere una scala di valori o un'espressione.

**Sintassi:**

```
RangeAvg (first_expr[, Expression])
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

Gli argomenti di questa funzione possono contenere funzioni intra-record, che a loro volta restituiscono un elenco di valori.

## Argomenti

| Argomento  | Descrizione                                                               |
|------------|---------------------------------------------------------------------------|
| first_expr | L'espressione o il campo contenente i dati da misurare.                   |
| Expression | Le espressioni o campi opzionali contenenti la scala di dati da misurare. |

**Limiti:**

Se non viene trovato nessun valore numerico, viene restituito NULL.

**Esempi e risultati:**

## Esempi di script

| Esempi                    | Risultati              |
|---------------------------|------------------------|
| RangeAvg (1,2,4)          | Restituisce 2,33333333 |
| RangeAvg (1,'xyz')        | Restituisce 1          |
| RangeAvg (null( ), 'abc') | Restituisce NULL       |

### Esempio:

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

```
RangeTab3:
LOAD recno() as RangeID, RangeAvg(Field1,Field2,Field3) as MyRangeAvg INLINE [
Field1, Field2, Field3
10,5,6
2,3,7
8,2,8
18,11,9
5,5,9
9,4,2
];
```

La tabella risultante mostra i valori restituiti in MyRangeAvg per ciascun record della tabella.

Tabella risultante

| RangeID | MyRangeAvg |
|---------|------------|
| 1       | 7          |
| 2       | 4          |
| 3       | 6          |
| 4       | 12.666     |
| 5       | 6.333      |
| 6       | 5          |

### Esempio con espressione:

```
RangeAvg (Above(MyField),0,3))
```

Restituisce la media mobile del risultato della scala di tre valori di **MyField** calcolati in base alla riga attuale e alle due righe che la precedono. Specificando 3 per il terzo argomento, la funzione **Above()** restituisce tre valori, se il numero di righe che precedono la riga attuale è sufficiente, che vengono considerati come input per la funzione **RangeAvg()**.

Dati utilizzati negli esempi:



Disattivare l'ordinamento di **MyField** per garantire il funzionamento corretto dell'esempio.

Dati campione

| MyField | RangeAvg (Above (MyField,0,3)) | Comments                                                            |
|---------|--------------------------------|---------------------------------------------------------------------|
| 10      | 10                             | Dato che questa è la prima riga, la scala presenta un unico valore. |
| 2       | 6                              | Questa riga è preceduta da un'unica riga, quindi la scala è: 10,2.  |
| 8       | 6.6666666667                   | L'equivalente di RangeAvg(10,2,8)                                   |
| 18      | 9.3333333333                   | -                                                                   |
| 5       | 10.3333333333                  | -                                                                   |
| 9       | 10.6666666667                  | -                                                                   |

```
RangeTab:
LOAD * INLINE [
MyField
10
2
8
18
5
9
] ;
```

**Vedere anche:**

- p [Avg - funzione per grafici \(page 457\)](#)
- p [Count - funzione per grafici \(page 396\)](#)

## RangeCorrel

**RangeCorrel()** restituisce il coefficiente di correlazione per due serie di dati. Il coefficiente di correlazione è una misura della relazione tra le serie di dati.

**Sintassi:**

```
RangeCorrel (x_value , y_value[, Expression])
```

**Tipo di dati restituiti:** numerico

Le serie di dati devono essere immesse come coppie (x,y). Ad esempio, per valutare due serie di dati, l'insieme 1 e l'insieme 2, in cui l'insieme 1 = 2,6,9 e l'insieme 2 = 3,8,4, occorrerà scrivere `RangeCorrel (2,3,6,8,9,4)`, che restituisce 0,269.

### Argomenti:

Argomenti

| Argomento        | Descrizione                                                                                                                                                                                                                                            |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| x-value, y-value | Ogni valore rappresenta un valore singolo o una scala di valori restituiti da una funzione intra-record con un terzo parametro opzionale. Ogni valore o scala di valori deve corrispondere a un <b>x-value</b> oppure a una scala di <b>y-values</b> . |
| Expression       | Le espressioni o campi opzionali contenenti la scala di dati da misurare.                                                                                                                                                                              |

### Limiti:

Per essere calcolata, la funzione necessita almeno di due coppie di coordinate.

I valori di testo, i valori NULL e i valori mancanti restituiscono NULL.

### Esempi e risultati:

Esempi di funzioni

| Esempi                           | Risultati                                                                                                                            |
|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| RangeCorrel<br>(2,3,6,8,9,4,8,5) | Restituisce 0,2492. Questa funzione può essere caricata nello script o aggiunta a una visualizzazione nell'editor delle espressioni. |

### Esempio:

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

```
RangeList:
Load * Inline [
ID1|x1|y1|x2|y2|x3|y3|x4|y4|x5|y5|x6|y6
01|46|60|70|13|78|20|45|65|78|12|78|22
02|65|56|22|79|12|56|45|24|32|78|55|15
03|77|68|34|91|24|68|57|36|44|90|67|27
04|57|36|44|90|67|27|57|68|47|90|80|94
](delimiter is '|');
```

```
XY:
LOAD recno() as RangeID, * Inline [
X|Y
2|3
6|8
9|4
8|5
](delimiter is '|');
```

In una tabella con ID1 come dimensione e la misura `RangeCorrel(x1,y1,x2,y2,x3,y3,x4,y4,x5,y5,x6,y6)`, la funzione **RangeCorrel()** è in grado di trovare il valore di **Correl** su un intervallo di sei coppie x,y per ciascuno dei valori ID1.

Tabella risultante

| ID1 | MyRangeCorrel |
|-----|---------------|
| 01  | -0.9517       |
| 02  | -0.5209       |
| 03  | -0.5209       |
| 04  | -0.1599       |

### Esempio:

```
XY:
LOAD recno() as RangeID, * Inline [
X|Y
2|3
6|8
9|4
8|5
](delimiter is '|');
```

In una tabella RangeID come dimensione e la misura: `RangeCorrel(Below(X,0,4,BelowY,0,4))`, la funzione **RangeCorrel()** utilizza i risultati delle funzioni **Below()**, che, in virtù del terzo argomento (count) impostato su 4, producono un intervallo di quattro valori x-y dalla tabella XY caricata.

Tabella risultante

| RangeID | MyRangeCorrel2 |
|---------|----------------|
| 01      | 0.2492         |
| 02      | -0.9959        |
| 03      | -1.0000        |
| 04      | -              |

Il valore di RangeID 01 è uguale a inserire manualmente `RangeCorrel(2,3,6,8,9,4,8,5)`. Per gli altri valori di RangeID, le serie prodotte dalla funzione `Below()` sono: (6,8,9,4,8,5), (9,4,8,5) e (8,5), l'ultima delle quali produce un risultato null.

### Vedere anche:

p [Correl - funzione per grafici \(page 460\)](#)

## RangeCount

**RangeCount()** restituisce il numero di valori, sia testuali che numerici, nell'espressione o nel campo.

### Sintassi:

```
RangeCount (first_expr[, Expression])
```

**Tipo di dati restituiti:** numero intero

### Argomenti:

Gli argomenti di questa funzione possono contenere funzioni intra-record, che a loro volta restituiscono un elenco di valori.

#### Argomenti

| Argomento  | Descrizione                                                                    |
|------------|--------------------------------------------------------------------------------|
| first_expr | L'espressione o il campo contenente i dati da conteggiare.                     |
| Expression | Le espressioni o i campi opzionali contenenti la scala di dati da conteggiare. |

### Limiti:

I valori NULL non vengono considerati.

### Esempi e risultati:

#### Esempi di funzioni

| Esempi                       | Risultati     |
|------------------------------|---------------|
| RangeCount (1,2,4)           | Restituisce 3 |
| RangeCount (2,'xyz')         | Restituisce 2 |
| RangeCount (null( ))         | Restituisce 0 |
| RangeCount (2,'xyz', null()) | Restituisce 2 |

### Esempio:

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

RangeTab3:

```
LOAD recno() as RangeID, RangeCount(Field1,Field2,Field3) as MyRangeCount INLINE [
```

```
Field1, Field2, Field3
```

```
10,5,6
```

```
2,3,7
```

```
8,2,8
```

```
18,11,9
```

5,5,9

9,4,2

];

La tabella risultante mostra i valori restituiti in **MyRangeCount** per ciascun record della tabella.

Tabella dei risultati

| RangeID | MyRangeCount |
|---------|--------------|
| 1       | 3            |
| 2       | 3            |
| 3       | 3            |
| 4       | 3            |
| 5       | 3            |
| 6       | 3            |

Esempio con espressione:

`RangeCount (Above(MyField,1,3))`

Restituisce il numero di valori contenuto nei tre risultati di **MyField**. Specificando il primo argomento della funzione **Above()** come 1 e il secondo argomento come 3, restituisce i valori dei primi tre campi sopra la riga corrente, in presenza di un numero di righe sufficiente prese come input della funzione **RangeCount()**.

Dati utilizzati negli esempi:

Dati campione

| MyField | RangeCount(Above(MyField,1,3)) |
|---------|--------------------------------|
| 10      | 0                              |
| 2       | 1                              |
| 8       | 2                              |
| 18      | 3                              |
| 5       | 3                              |
| 9       | 3                              |

Dati utilizzati negli esempi:

RangeTab:

```
LOAD * INLINE [  
MyField  
10  
2  
8  
18  
5  
9
```

] ;

**Vedere anche:**p [Count - funzione per grafici \(page 396\)](#)

## RangeFractile

**RangeFractile()** restituisce il valore che corrisponde al **fractile** n-esimo (quantile) di un intervallo di numeri.*Durante il calcolo del frattale, RangeFractile() utilizza l'interpolazione lineare tra le classificazioni più prossime.***Sintassi:****RangeFractile**(fractile, first\_expr[, Expression])**Tipo di dati restituiti:** numerico**Argomenti:**

Gli argomenti di questa funzione possono contenere funzioni intra-record, che a loro volta restituiscono un elenco di valori.

**Argomenti**

| Argomento  | Descrizione                                                                                             |
|------------|---------------------------------------------------------------------------------------------------------|
| fractile   | Un numero compreso tra 0 e 1 corrispondente al frattale (quantile espresso come frazione) da calcolare. |
| first_expr | L'espressione o il campo contenente i dati da misurare.                                                 |
| Expression | Le espressioni o campi opzionali contenenti la scala di dati da misurare.                               |

**Esempi e risultati:****Esempi di funzioni**

| Esempi                       | Risultati        |
|------------------------------|------------------|
| RangeFractile (0.24,1,2,4,6) | Restituisce 1,72 |
| RangeFractile(0.5,1,2,3,4,6) | Restituisce 3    |
| RangeFractile (0.5,1,2,5,6)  | Restituisce 3,5  |

**Esempio:**

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

RangeTab:

```
LOAD recno() as RangeID, RangeFractile(0.5,Field1,Field2,Field3) as MyRangeFrac INLINE [  
Field1, Field2, Field3  
10,5,6  
2,3,7  
8,2,8  
18,11,9  
5,5,9  
9,4,2  
];
```

La tabella risultante mostra i valori restituiti in MyRangeFrac per ciascun record della tabella.

Tabella risultante

| RangeID | MyRangeFrac |
|---------|-------------|
| 1       | 6           |
| 2       | 3           |
| 3       | 8           |
| 4       | 11          |
| 5       | 5           |
| 6       | 4           |

Esempio con espressione:

```
RangeFractile (0.5, Above(Sum(MyField),0,3))
```

In questo esempio, la funzione intra-record **Above()** contiene gli argomenti offset e count opzionali. Ciò restituisce una scala di risultati che può essere utilizzata come input per una qualsiasi delle funzioni scala. In questo caso, `Above(Sum(MyField),0,3)` restituisce il valore di `MyField` per la riga attuale e le due righe che la precedono. Questi valori forniscono l'input per la funzione **RangeFractile()**. Per la riga inferiore della tabella seguente, ciò equivale a `RangeFractile(0.5, 3,4,6)`, vale a dire il calcolo del frattale 0,5 per le serie 3, 4 e 6. Per le prime due righe della tabella seguente, il numero dei valori nella scala viene ridotto di conseguenza, se non vi sono righe che precedono la riga attuale. Vengono restituiti risultati simili per le altre funzioni intra-record.

Dati campione

| MyField | RangeFractile(0.5, Above(Sum(MyField),0,3)) |
|---------|---------------------------------------------|
| 1       | 1                                           |
| 2       | 1.5                                         |
| 3       | 2                                           |
| 4       | 3                                           |
| 5       | 4                                           |
| 6       | 5                                           |

Dati utilizzati negli esempi:

```
RangeTab:
LOAD * INLINE [
MyField
1
2
3
4
5
6
] ;
```

**Vedere anche:**

p [Above - funzione per grafici \(page 1464\)](#)

p [Fractile - funzione per grafici \(page 464\)](#)

### RangeIRR

**RangeIRR()** restituisce il tasso di rendimento interno per una serie di flussi di cassa rappresentati dai valori di input.

Il tasso di rendimento interno è il tasso di interesse ricevuto per un investimento che consiste in pagamenti (valori negativi) ed entrate (valori positivi) che ricorrono ad intervalli regolari.

Questa funzione utilizza una versione semplificata del metodo Newton per il calcolo del tasso di rendimento interno (IRR).

**Sintassi:**

```
RangeIRR (value[, value][, Expression])
```

**Tipo di dati restituiti:** numerico

#### Argomenti

| Argomento  | Descrizione                                                                                                                                                                                                              |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value      | Un valore singolo oppure una scala di valori restituita da una funzione intra-record con un terzo parametro opzionale. Per essere calcolata, la funzione necessita almeno di un valore positivo e di un valore negativo. |
| Expression | Le espressioni o campi opzionali contenenti la scala di dati da misurare.                                                                                                                                                |

**Limiti:**

I valori di testo, i valori NULL e i valori mancanti vengono ignorati.

Tabella di esempio

| Esempi                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Risultati                                                                                                                                                                                                                                                                                                                                                                                                          |         |          |   |        |   |        |   |   |   |        |   |        |   |         |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|----------|---|--------|---|--------|---|---|---|--------|---|--------|---|---------|
| RangeIRR(-70000,12000,15000,18000,21000,26000)                                                                                                                                                                                                                                                                                                                                                                                                         | Restituisce 0.0866                                                                                                                                                                                                                                                                                                                                                                                                 |         |          |   |        |   |        |   |   |   |        |   |        |   |         |
| <p>Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.</p> <p>RangeTab3:</p> <pre> LOAD *, recno() as RangeID, RangeIRR(Field1,Field2,Field3) as RangeIRR;  LOAD * INLINE [ Field1 Field2 Field3 -10000 5000 6000 -2000 NULL 7000 -8000 'abc' 8000 -1800 11000 9000 -5000 5000 9000 -9000 4000 2000 ] (delimiter is ' ');</pre> | <p>La tabella risultante mostra i valori restituiti in RangeIRR per ciascun record della tabella.</p> <table> <thead> <tr> <th>RangeID</th><th>RangeIRR</th></tr> </thead> <tbody> <tr><td>1</td><td>0.0639</td></tr> <tr><td>2</td><td>0.8708</td></tr> <tr><td>3</td><td>-</td></tr> <tr><td>4</td><td>5.8419</td></tr> <tr><td>5</td><td>0.9318</td></tr> <tr><td>6</td><td>-0.2566</td></tr> </tbody> </table> | RangeID | RangeIRR | 1 | 0.0639 | 2 | 0.8708 | 3 | - | 4 | 5.8419 | 5 | 0.9318 | 6 | -0.2566 |
| RangeID                                                                                                                                                                                                                                                                                                                                                                                                                                                | RangeIRR                                                                                                                                                                                                                                                                                                                                                                                                           |         |          |   |        |   |        |   |   |   |        |   |        |   |         |
| 1                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0.0639                                                                                                                                                                                                                                                                                                                                                                                                             |         |          |   |        |   |        |   |   |   |        |   |        |   |         |
| 2                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0.8708                                                                                                                                                                                                                                                                                                                                                                                                             |         |          |   |        |   |        |   |   |   |        |   |        |   |         |
| 3                                                                                                                                                                                                                                                                                                                                                                                                                                                      | -                                                                                                                                                                                                                                                                                                                                                                                                                  |         |          |   |        |   |        |   |   |   |        |   |        |   |         |
| 4                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 5.8419                                                                                                                                                                                                                                                                                                                                                                                                             |         |          |   |        |   |        |   |   |   |        |   |        |   |         |
| 5                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0.9318                                                                                                                                                                                                                                                                                                                                                                                                             |         |          |   |        |   |        |   |   |   |        |   |        |   |         |
| 6                                                                                                                                                                                                                                                                                                                                                                                                                                                      | -0.2566                                                                                                                                                                                                                                                                                                                                                                                                            |         |          |   |        |   |        |   |   |   |        |   |        |   |         |

**Vedere anche:**p [Funzioni intra-record \(page 1461\)](#)

## RangeKurtosis

**RangeKurtosis()** restituisce il valore che corrisponde al kurtosis di una scala di numeri.**Sintassi:****RangeKurtosis**(first\_expr[, Expression])

**Tipo di dati restituiti:** numerico

**Argomenti:**

Gli argomenti di questa funzione possono contenere funzioni intra-record, che a loro volta restituiscono un elenco di valori.

Argomenti

| Argomento  | Descrizione                                                               |
|------------|---------------------------------------------------------------------------|
| first_expr | L'espressione o il campo contenente i dati da misurare.                   |
| Expression | Le espressioni o campi opzionali contenenti la scala di dati da misurare. |

**Limiti:**

Se non viene trovato nessun valore numerico, viene restituito NULL.

**Esempi e risultati:**

Esempi di funzioni

| Esempi                  | Risultati                     |
|-------------------------|-------------------------------|
| RangeKurtosis (1,2,4,7) | Restituisce -0,28571428571429 |

**Vedere anche:**

p [Kurtosis - funzione per grafici \(page 472\)](#)

## RangeMax

**RangeMax()** restituisce i valori numerici più alti trovati nel campo o nell'espressione.

**Sintassi:**

```
RangeMax (first_expr[, Expression])
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

Argomenti

| Argomento  | Descrizione                                                               |
|------------|---------------------------------------------------------------------------|
| first_expr | L'espressione o il campo contenente i dati da misurare.                   |
| Expression | Le espressioni o campi opzionali contenenti la scala di dati da misurare. |

### Limiti:

Se non viene trovato nessun valore numerico, viene restituito NULL.

### Esempi e risultati:

Esempi di funzioni

| Esempi                    | Risultati        |
|---------------------------|------------------|
| RangeMax (1,2,4)          | Restituisce 4    |
| RangeMax (1,'xyz')        | Restituisce 1    |
| RangeMax (null( ), 'abc') | Restituisce NULL |

### Esempio:

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

```
RangeTab3:
LOAD recno() as RangeID, RangeMax(Field1,Field2,Field3) as MyRangeMax INLINE [
Field1, Field2, Field3
10,5,6
2,3,7
8,2,8
18,11,9
5,5,9
9,4,2
];
```

La tabella risultante mostra i valori restituiti in MyRangeMax per ciascun record della tabella.

Tabella risultante

| RangeID | MyRangeMax |
|---------|------------|
| 1       | 10         |
| 2       | 7          |
| 3       | 8          |
| 4       | 18         |
| 5       | 9          |
| 6       | 9          |

### Esempio con espressione:

```
RangeMax (Above(MyField,0,3))
```

Restituisce il valore massimo nella scala di tre valori di **MyField** calcolati in base alla riga attuale e alle due righe che la precedono. Specificando 3 per il terzo argomento, la funzione **Above()** restituisce tre valori, se il numero di righe che precedono la riga attuale è sufficiente, che vengono considerati come input per la funzione **RangeMax()**.

Dati utilizzati negli esempi:



*Disattivare l'ordinamento di **MyField** per garantire il funzionamento corretto dell'esempio.*

Dati campione

| MyField | RangeMax (Above(Sum(MyField),1,3)) |
|---------|------------------------------------|
| 10      | 10                                 |
| 2       | 10                                 |
| 8       | 10                                 |
| 18      | 18                                 |
| 5       | 18                                 |
| 9       | 18                                 |

Dati utilizzati negli esempi:

```
RangeTab:
LOAD * INLINE [
MyField
10
2
8
18
5
9
] ;
```

### RangeMaxString

**RangeMaxString()** restituisce l'ultimo valore nell'ordine del testo che trova nel campo o nell'espressione.

**Sintassi:**

```
RangeMaxString(first_expr[, Expression])
```

**Tipo di dati restituiti:** stringa

**Argomenti:**

Gli argomenti di questa funzione possono contenere funzioni intra-record, che a loro volta restituiscono un elenco di valori.

## Argomenti

| Argomento  | Descrizione                                                               |
|------------|---------------------------------------------------------------------------|
| first_expr | L'espressione o il campo contenente i dati da misurare.                   |
| Expression | Le espressioni o campi opzionali contenenti la scala di dati da misurare. |

## Esempi e risultati:

## Esempi di funzioni

| Esempi                       | Risultati         |
|------------------------------|-------------------|
| RangeMaxString (1,2,4)       | Restituisce 4     |
| RangeMaxString ('xyz','abc') | Restituisce 'xyz' |
| RangeMaxString (5,'abc')     | Restituisce 'abc' |
| RangeMaxString (null( ))     | Restituisce NULL  |

## Esempio con espressione:

RangeMaxString (Above(MaxString(MyField),0,3))

Restituisce l'ultimo (nell'ordine alfabetico del testo) dei tre risultati della funzione **MaxString(MyField)** valutati in base alla riga attuale e alle due righe che la precedono.

## Dati utilizzati negli esempi:



Disattivare l'ordinamento di **MyField** per garantire il funzionamento corretto dell'esempio.

## Dati campione

| MyField | RangeMaxString(Above(MaxString(MyField),0,3)) |
|---------|-----------------------------------------------|
| 10      | 10                                            |
| abc     | abc                                           |
| 8       | abc                                           |
| def     | def                                           |
| xyz     | xyz                                           |
| 9       | xyz                                           |

## Dati utilizzati negli esempi:

```
RangeTab:
LOAD * INLINE [
MyField
10
'abc'
8
'def'
```

```
'xyz'
9
] ;
```

**Vedere anche:**

p [MaxString - funzione per grafici \(page 598\)](#)

## RangeMin

**RangeMin()** restituisce i valori numerici più bassi trovati nel campo o nell'espressione.

**Sintassi:**

```
RangeMin (first_expr[, Expression])
```

**Tipo di dati restituiti:** numerico

**Argomenti:****Argomenti**

| Argomento  | Descrizione                                                               |
|------------|---------------------------------------------------------------------------|
| first_expr | L'espressione o il campo contenente i dati da misurare.                   |
| Expression | Le espressioni o campi opzionali contenenti la scala di dati da misurare. |

**Limiti:**

Se non viene trovato nessun valore numerico, viene restituito NULL.

**Esempi e risultati:****Esempi di funzioni**

| Esempi                    | Risultati        |
|---------------------------|------------------|
| RangeMin (1,2,4)          | Restituisce 1    |
| RangeMin (1,'xyz')        | Restituisce 1    |
| RangeMin (null( ), 'abc') | Restituisce NULL |

**Esempio:**

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

```
RangeTab3:
LOAD recno() as RangeID, RangeMin(Field1,Field2,Field3) as MyRangeMin INLINE [
Field1, Field2, Field3
10,5,6
2,3,7
8,2,8
18,11,9
```

```
5,5,9  
9,4,2  
];
```

La tabella risultante mostra i valori restituiti in MyRangeMin per ciascun record della tabella.

Tabella risultante

| RangeID | MyRangeMin |
|---------|------------|
| 1       | 5          |
| 2       | 2          |
| 3       | 2          |
| 4       | 9          |
| 5       | 5          |
| 6       | 2          |

Esempio con espressione:

```
RangeMin (Above(MyField,0,3))
```

Restituisce il valore minimo nella scala di tre valori di **MyField** calcolati in base alla riga attuale e alle due righe che la precedono. Specificando 3 per il terzo argomento, la funzione **Above()** restituisce tre valori, se il numero di righe che precedono la riga attuale è sufficiente, che vengono considerati come input per la funzione **RangeMin()**.

Dati utilizzati negli esempi:

Dati campione

| MyField | RangeMin(Above(MyField,0,3)) |
|---------|------------------------------|
| 10      | 10                           |
| 2       | 2                            |
| 8       | 2                            |
| 18      | 2                            |
| 5       | 5                            |
| 9       | 5                            |

Dati utilizzati negli esempi:

```
RangeTab:  
LOAD * INLINE [  
MyField  
10  
2  
8  
18  
5  
9
```

] ;

#### Vedere anche:

p [Min - funzione per grafici \(page 372\)](#)

## RangeMinString

**RangeMinString()** restituisce il primo valore nell'ordine del testo che trova nel campo o nell'espressione.

#### Sintassi:

```
RangeMinString(first_expr[, Expression])
```

**Tipo di dati restituiti:** stringa

#### Argomenti:

Gli argomenti di questa funzione possono contenere funzioni intra-record, che a loro volta restituiscono un elenco di valori.

#### Argomenti

| Argomento  | Descrizione                                                               |
|------------|---------------------------------------------------------------------------|
| first_expr | L'espressione o il campo contenente i dati da misurare.                   |
| Expression | Le espressioni o campi opzionali contenenti la scala di dati da misurare. |

#### Esempi e risultati:

#### Esempi di funzioni

| Esempi                       | Risultati         |
|------------------------------|-------------------|
| RangeMinString (1,2,4)       | Restituisce 1     |
| RangeMinString ('xyz','abc') | Restituisce 'abc' |
| RangeMinString (5,'abc')     | Restituisce 5     |
| RangeMinString (null( ))     | Restituisce NULL  |

#### Esempio con espressione:

```
RangeMinString (Above(MinString(MyField),0,3))
```

Restituisce il primo (nell'ordine alfabetico del testo) dei tre risultati della funzione **MinString(MyField)** valutati in base alla riga attuale e alle due righe che la precedono.

Dati utilizzati negli esempi:



Disattivare l'ordinamento di **MyField** per garantire il funzionamento corretto dell'esempio.

Dati campione

| MyField | RangeMinString(Above(MinString(MyField),0,3)) |
|---------|-----------------------------------------------|
| 10      | 10                                            |
| abc     | 10                                            |
| 8       | 8                                             |
| def     | 8                                             |
| xyz     | 8                                             |
| 9       | 9                                             |

Dati utilizzati negli esempi:

```
RangeTab:
LOAD * INLINE [
MyField
10
'abc'
8
'def'
'xyz'
9
] ;
```

**Vedere anche:**

p [MinString - funzione per grafici \(page 600\)](#)

## RangeMissingCount

**RangeMissingCount()** restituisce il numero di valori non numerici (incluso NULL) nell'espressione o nel campo.

**Sintassi:**

```
RangeMissingCount(first_expr[, Expression])
```

**Tipo di dati restituiti:** numero intero

**Argomenti:**

Gli argomenti di questa funzione possono contenere funzioni intra-record, che a loro volta restituiscono un elenco di valori.

Argomenti

| Argomento  | Descrizione                                                                    |
|------------|--------------------------------------------------------------------------------|
| first_expr | L'espressione o il campo contenente i dati da conteggiare.                     |
| Expression | Le espressioni o i campi opzionali contenenti la scala di dati da conteggiare. |

## Esempi e risultati:

## Esempi di funzioni

| Esempi                                   | Risultati     |
|------------------------------------------|---------------|
| <code>RangeMissingCount (1,2,4)</code>   | Restituisce 0 |
| <code>RangeMissingCount (5,'abc')</code> | Restituisce 1 |
| <code>RangeMissingCount (null( ))</code> | Restituisce 1 |

## Esempio con espressione:

```
RangeMissingCount (Above(MinString(MyField),0,3))
```

Restituisce il numero di valori non numerici trovati nei tre risultati della funzione **MinString(MyField)** valutati in base alla riga corrente e alle due righe che la precedono.



*Disattivare l'ordinamento di **MyField** per garantire il funzionamento corretto dell'esempio.*

## Dati campione

| MyField | RangeMissingCount<br>(Above(MinString<br>(MyField),0,3)) | Explanation                                                                                                 |
|---------|----------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| 10      | 2                                                        | Restituisce 2 perché non ci sono righe che precedono questa riga, quindi 2 dei 3 valori sono mancanti.      |
| abc     | 2                                                        | Restituisce 2 perché la riga corrente è preceduta da 1 sola riga e la riga corrente non è numerica ('abc'). |
| 8       | 1                                                        | Restituisce 1 perché 1 delle 3 righe include un valore non numerico ('abc').                                |
| def     | 2                                                        | Restituisce 2 perché 2 delle 3 righe includono valori non numerici ('def' e 'abc').                         |
| xyz     | 2                                                        | Restituisce 2 perché 2 delle 3 righe includono valori non numerici ('xyz' e 'def').                         |
| 9       | 2                                                        | Restituisce 2 perché 2 delle 3 righe includono valori non numerici ('xyz' e 'def').                         |

## Dati utilizzati negli esempi:

```
RangeTab:
LOAD * INLINE [
MyField
10
'abc'
8
```

```
'def'
'xyz'
9
] ;
```

**Vedere anche:**

p [MissingCount - funzione per grafici \(page 402\)](#)

## RangeMode

**RangeMode()** trova il valore più comune (valore della modalità) nel campo o nell'espressione.

**Sintassi:**

```
RangeMode (first_expr {, Expression})
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

Gli argomenti di questa funzione possono contenere funzioni intra-record, che a loro volta restituiscono un elenco di valori.

**Argomenti**

| Argomento  | Descrizione                                                               |
|------------|---------------------------------------------------------------------------|
| first_expr | L'espressione o il campo contenente i dati da misurare.                   |
| Expression | Le espressioni o campi opzionali contenenti la scala di dati da misurare. |

**Limiti:**

Se più di un valore condivide la frequenza più alta, viene restituito NULL.

**Esempi e risultati:****Esempi di funzioni**

| Esempi                  | Risultati        |
|-------------------------|------------------|
| RangeMode (1,2,9,2,4)   | Restituisce 2    |
| RangeMode ('a',4,'a',4) | Restituisce NULL |
| RangeMode (null( ))     | Restituisce NULL |

**Esempio:**

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

RangeTab3:

```
LOAD recno() as RangeID, RangeMode(Field1,Field2,Field3) as MyRangeMode INLINE [  
  
Field1, Field2, Field3  
  
10,5,6  
  
2,3,7  
  
8,2,8  
  
18,11,9  
  
5,5,9  
  
9,4,2  
  
];
```

La tabella risultante mostra i valori restituiti da **MyRangeMode** per ciascun record della tabella.

Tabella dei risultati

| RangeID | MyRangMode |
|---------|------------|
| 1       | -          |
| 2       | -          |
| 3       | 8          |
| 4       | -          |
| 5       | 5          |
| 6       | -          |

Esempio con espressione:

```
RangeMode (Above(MyField,0,3))
```

Restituisce il valore più ricorrente all'interno dei tre risultati di **MyField** valutati in base alla riga attuale e alle due righe che la precedono. Specificando 3 per il terzo argomento, la funzione **Above()** restituisce tre valori, se il numero di righe che precedono la riga attuale è sufficiente, che vengono considerati come input per la funzione **RangeMode()**.

Dati utilizzati nell'esempio:

```
RangeTab:  
LOAD * INLINE [  
MyField  
10  
2  
8  
18  
5  
9  
] ;
```



Disattivare l'ordinamento di **MyField** per garantire il funzionamento corretto dell'esempio.

Dati campione

| MyField | RangeMode(Above(MyField,0,3))                                                                                           |
|---------|-------------------------------------------------------------------------------------------------------------------------|
| 10      | Restituisce 10 perché non vi sono righe che precedono quella attuale, quindi il singolo valore è quello più ricorrente. |
| 2       | -                                                                                                                       |
| 8       | -                                                                                                                       |
| 18      | -                                                                                                                       |
| 5       | -                                                                                                                       |
| 9       | -                                                                                                                       |

**Vedere anche:**

p [Mode - funzione per grafici \(page 377\)](#)

## RangeNPV

**RangeNPV()** restituisce il valore attuale netto di un investimento basato su un tasso di sconto e una serie di futuri pagamenti periodici (valori negativi) ed entrate (valori positivi). Il risultato presenta un formato numerico predefinito di **money**.

Per i flussi di cassa che non sono necessariamente periodici, fare riferimento a [RangeXNPV \(page 1622\)](#).

**Sintassi:**

```
RangeNPV (discount_rate, value[,value][, Expression])
```

**Tipo di dati restituiti:** numerico

Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                      |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| discount_rate | Il tasso di interesse per periodo.                                                                                                                                                                               |
| value         | Un pagamento o un'entrata ricorrente alla fine di ogni periodo. Ciascun valore può essere un valore singolo oppure una scala di valori restituita da una funzione intra-record con un terzo parametro opzionale. |
| Expression    | Le espressioni o campi opzionali contenenti la scala di dati da misurare.                                                                                                                                        |

**Limiti:**

I valori di testo, i valori NULL e i valori mancanti vengono ignorati.

| Esempi                                                                                                                                                                                                                                                                                                                                                                                                              | Risultati                                                                                                                                                                                                                                                                                                                                                                                                                       |         |          |   |          |   |          |   |         |   |         |   |          |   |         |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|----------|---|----------|---|----------|---|---------|---|---------|---|----------|---|---------|
| RangeNPV(0.1,-10000,3000,4200,6800)                                                                                                                                                                                                                                                                                                                                                                                 | Rendimenti 1188.44                                                                                                                                                                                                                                                                                                                                                                                                              |         |          |   |          |   |          |   |         |   |         |   |          |   |         |
| <p>Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.</p> <p>RangeTab3:</p> <pre>LOAD *, recno() as RangeID, RangeNPV(Field1,Field2,Field3) as RangeNPV;  LOAD * INLINE [ Field1 Field2 Field3 10 5 -6000 2 NULL 7000 8 'abc' 8000 18 11 9000 5 5 9000 9 4 2000 ] (delimiter is ' ');</pre> | <p>La tabella risultante mostra i valori restituiti in RangeNPV per ciascun record della tabella.</p> <table> <thead> <tr> <th>RangeID</th><th>RangeNPV</th></tr> </thead> <tbody> <tr><td>1</td><td>\$-49.13</td></tr> <tr><td>2</td><td>\$777.78</td></tr> <tr><td>3</td><td>\$98.77</td></tr> <tr><td>4</td><td>\$25.51</td></tr> <tr><td>5</td><td>\$250.83</td></tr> <tr><td>6</td><td>\$20.40</td></tr> </tbody> </table> | RangeID | RangeNPV | 1 | \$-49.13 | 2 | \$777.78 | 3 | \$98.77 | 4 | \$25.51 | 5 | \$250.83 | 6 | \$20.40 |
| RangeID                                                                                                                                                                                                                                                                                                                                                                                                             | RangeNPV                                                                                                                                                                                                                                                                                                                                                                                                                        |         |          |   |          |   |          |   |         |   |         |   |          |   |         |
| 1                                                                                                                                                                                                                                                                                                                                                                                                                   | \$-49.13                                                                                                                                                                                                                                                                                                                                                                                                                        |         |          |   |          |   |          |   |         |   |         |   |          |   |         |
| 2                                                                                                                                                                                                                                                                                                                                                                                                                   | \$777.78                                                                                                                                                                                                                                                                                                                                                                                                                        |         |          |   |          |   |          |   |         |   |         |   |          |   |         |
| 3                                                                                                                                                                                                                                                                                                                                                                                                                   | \$98.77                                                                                                                                                                                                                                                                                                                                                                                                                         |         |          |   |          |   |          |   |         |   |         |   |          |   |         |
| 4                                                                                                                                                                                                                                                                                                                                                                                                                   | \$25.51                                                                                                                                                                                                                                                                                                                                                                                                                         |         |          |   |          |   |          |   |         |   |         |   |          |   |         |
| 5                                                                                                                                                                                                                                                                                                                                                                                                                   | \$250.83                                                                                                                                                                                                                                                                                                                                                                                                                        |         |          |   |          |   |          |   |         |   |         |   |          |   |         |
| 6                                                                                                                                                                                                                                                                                                                                                                                                                   | \$20.40                                                                                                                                                                                                                                                                                                                                                                                                                         |         |          |   |          |   |          |   |         |   |         |   |          |   |         |

**Vedere anche:**

p [Funzioni intra-record \(page 1461\)](#)

## RangeNullCount

**RangeNullCount()** trova il numero di valori NULL nel campo o nell'espressione.

**Sintassi:**

```
RangeNullCount(first_expr [, Expression])
```

**Tipo di dati restituiti:** numero intero

**Argomenti:**

Gli argomenti di questa funzione possono contenere funzioni intra-record, che a loro volta restituiscono un elenco di valori.

## Argomenti

| Argomento  | Descrizione                                                               |
|------------|---------------------------------------------------------------------------|
| first_expr | L'espressione o il campo contenente i dati da misurare.                   |
| Expression | Le espressioni o campi opzionali contenenti la scala di dati da misurare. |

## Esempi e risultati:

## Esempi di funzioni

| Esempi                            | Risultati     |
|-----------------------------------|---------------|
| RangeNullCount (1,2,4)            | Restituisce 0 |
| RangeNullCount (5,'abc')          | Restituisce 0 |
| RangeNullCount (null( ), null( )) | Restituisce 2 |

## Esempio con espressione:

RangeNullCount (Above(Sum(MyField),0,3))

Restituisce il numero di valori NULL trovati nei tre risultati della funzione **Sum(MyField)** valutati in base alla riga attuale e alle due righe che la precedono.



*Se si copia **MyField** nell'esempio seguente, non verrà restituito il valore NULL.*

## Dati campione

| MyField | RangeNullCount(Above(Sum(MyField),0,3))                                                                          |
|---------|------------------------------------------------------------------------------------------------------------------|
| 10      | Restituisce 2 perché non ci sono righe che precedono questa riga, quindi 2 dei 3 valori sono mancanti (=NULL).   |
| 'abc'   | Restituisce 1 perché la riga attuale è preceduta solo da una riga, quindi uno dei tre valori è mancante (=NULL). |
| 8       | Restituisce 0 perché nessuna delle tre righe corrisponde a un valore NULL.                                       |

## Dati utilizzati negli esempi:

```
RangeTab:
LOAD * INLINE [
MyField
10
'abc'
8
] ;
```

## Vedere anche:

p [NullCount - funzione per grafici \(page 407\)](#)

## RangeNumericCount

**RangeNumericCount()** trova il numero di valori numerici nel campo o nell'espressione.

**Sintassi:**

```
RangeNumericCount (first_expr[, Expression])
```

**Tipo di dati restituiti:** numero intero

**Argomenti:**

Gli argomenti di questa funzione possono contenere funzioni intra-record, che a loro volta restituiscono un elenco di valori.

Argomenti

| Argomento  | Descrizione                                                               |
|------------|---------------------------------------------------------------------------|
| first_expr | L'espressione o il campo contenente i dati da misurare.                   |
| Expression | Le espressioni o campi opzionali contenenti la scala di dati da misurare. |

**Esempi e risultati:**

Esempi di funzioni

| Esempi                      | Risultati     |
|-----------------------------|---------------|
| RangeNumericCount (1,2,4)   | Restituisce 3 |
| RangeNumericCount (5,'abc') | Restituisce 1 |
| RangeNumericCount (null( )) | Restituisce 0 |

**Esempio con espressione:**

```
RangeNumericCount (Above(MaxString(MyField),0,3))
```

Restituisce il numero di valori numerici trovati nei tre risultati della funzione **MaxString(MyField)** valutati in base alla riga attuale e alle due righe che la precedono.



*Disattivare l'ordinamento di **MyField** per garantire il funzionamento corretto dell'esempio.*

Dati campione

| MyField | RangeNumericCount(Above(MaxString(MyField),0,3)) |
|---------|--------------------------------------------------|
| 10      | 1                                                |
| abc     | 1                                                |
| 8       | 2                                                |

| MyField | RangeNumericCount(Above(MaxString(MyField),0,3)) |
|---------|--------------------------------------------------|
| def     | 1                                                |
| xyz     | 1                                                |
| 9       | 1                                                |

Dati utilizzati negli esempi:

```
RangeTab:
LOAD * INLINE [
MyField
10
'abc'
8
def
xyz
9
] ;
```

**Vedere anche:**

p [NumericCount - funzione per grafici \(page 412\)](#)

## RangeOnly

**RangeOnly()** è una funzione duale che restituisce un valore se l'espressione viene valutata in base a un valore univoco. In caso contrario, viene restituito **NULL**.

**Sintassi:**

```
RangeOnly (first_expr[, Expression])
```

**Tipo di dati restituiti:** duale

**Argomenti:**

Gli argomenti di questa funzione possono contenere funzioni intra-record, che a loro volta restituiscono un elenco di valori.

| Argomento  | Descrizione                                                               |
|------------|---------------------------------------------------------------------------|
| first_expr | L'espressione o il campo contenente i dati da misurare.                   |
| Expression | Le espressioni o campi opzionali contenenti la scala di dati da misurare. |

**Esempi e risultati:**

| Esempi            | Risultati        |
|-------------------|------------------|
| RangeOnly (1,2,4) | Restituisce NULL |

| Esempi                                  | Risultati         |
|-----------------------------------------|-------------------|
| <code>RangeOnly (5, 'abc')</code>       | Restituisce NULL  |
| <code>RangeOnly (null( ), 'abc')</code> | Restituisce 'abc' |
| <code>RangeOnly(10,10,10)</code>        | Restituisce 10    |

**Vedere anche:**

[p \*Only - funzione per grafici \(page 382\)\*](#)

## RangeSkew

**RangeSkew()** restituisce il valore che corrisponde all'asimmetria di una scala di numeri.

**Sintassi:**

```
RangeSkew (first_expr[, Expression])
```

**Tipo di dati restituiti:** numerico

**Argomenti:**

Gli argomenti di questa funzione possono contenere funzioni intra-record, che a loro volta restituiscono un elenco di valori.

**Argomenti**

| Argomento               | Descrizione                                                               |
|-------------------------|---------------------------------------------------------------------------|
| <code>first_expr</code> | L'espressione o il campo contenente i dati da misurare.                   |
| <code>Expression</code> | Le espressioni o campi opzionali contenenti la scala di dati da misurare. |

**Limiti:**

Se non viene trovato nessun valore numerico, viene restituito NULL.

**Esempi e risultati:****Esempi di funzioni**

| Esempi                                          | Risultati                                                                                                                                                                                  |
|-------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>rangeskew (1,2,4)</code>                  | Restituisce 0,93521952958283                                                                                                                                                               |
| <code>rangeskew (above (SalesValue,0,3))</code> | Restituisce il grado di asimmetria mobile della scala dei tre valori restituiti dalla funzione <code>above()</code> calcolata in base alla riga attuale e alle due righe che la precedono. |

Dati utilizzati nell'esempio:

Dati campione

| CustID | RangeSkew(Above(SalesValue,0,3))                                                                                           |
|--------|----------------------------------------------------------------------------------------------------------------------------|
| 1-20   | -, -, 0.5676, 0.8455, 1.0127, -0.8741, 1.7243, -1.7186, 1.5518, 1.4332, 0, 1.1066, 1.3458, 1.5636, 1.5439, 0.6952, -0.3766 |

```
SalesTable:
LOAD recno() as CustID, * inline [
SalesValue
101
163
126
139
167
86
83
22
32
70
108
124
176
113
95
32
42
92
61
21
] ;
```

### Vedere anche:

p [Skew - funzione per grafici \(page 505\)](#)

## RangeStdev

**RangeStdev()** trova la deviazione standard di una scala di numeri.

### Sintassi:

```
RangeStdev(first_expr[, Expression])
```

**Tipo di dati restituiti:** numerico

### Argomenti:

Gli argomenti di questa funzione possono contenere funzioni intra-record, che a loro volta restituiscono un elenco di valori.

### Argomenti

| Argomento  | Descrizione                                                               |
|------------|---------------------------------------------------------------------------|
| first_expr | L'espressione o il campo contenente i dati da misurare.                   |
| Expression | Le espressioni o campi opzionali contenenti la scala di dati da misurare. |

### Limiti:

Se non viene trovato nessun valore numerico, viene restituito NULL.

### Esempi e risultati:

#### Esempi di funzioni

| Esempi                               | Risultati                                                                                                                                                                     |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RangeStdev (1,2,4)                   | Restituisce 1,5275252316519                                                                                                                                                   |
| RangeStdev (null( ))                 | Restituisce NULL                                                                                                                                                              |
| RangeStdev (above (SalesValue),0,3)) | Restituisce la deviazione standard mobile della scala dei tre valori restituiti dalla funzione above() calcolata in base alla riga attuale e alle due righe che la precedono. |

Dati utilizzati nell'esempio:

#### Dati campione

| CustID | RangeStdev(SalesValue, 0,3))                                                                                                                     |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| 1-20   | -,43.841, 34.192, 18.771, 20.953, 41.138, 47.655, 36.116, 32.716, 25.325, 38,000, 27.737, 35.553, 33.650, 42.532, 33.858, 32.146, 25.239, 35.595 |

SalesTable:

```
LOAD recno() as CustID, * inline [
SalesValue
101
163
126
139
167
86
83
22
32
70
108
124
176
113
95
32
42
92
61
```

21  
] ;

#### Vedere anche:

► [Stdev - funzione per grafici \(page 508\)](#)

## RangeSum

**RangeSum()** restituisce la somma di una scala di valori. Tutti i valori non numerici vengono trattati come 0.

#### Sintassi:

```
RangeSum (first_expr[, Expression])
```

**Tipo di dati restituiti:** numerico

#### Argomenti:

Gli argomenti di questa funzione possono contenere funzioni intra-record, che a loro volta restituiscono un elenco di valori.

#### Argomenti

| Argomento  | Descrizione                                                               |
|------------|---------------------------------------------------------------------------|
| first_expr | L'espressione o il campo contenente i dati da misurare.                   |
| Expression | Le espressioni o campi opzionali contenenti la scala di dati da misurare. |

#### Limiti:

La funzione **RangeSum** tratta tutti i valori non numerici come 0.

#### Esempi e risultati:

#### Esempi

| Esempi              | Risultati     |
|---------------------|---------------|
| RangeSum (1,2,4)    | Restituisce 7 |
| RangeSum (5, 'abc') | Restituisce 5 |
| RangeSum (null( ))  | Restituisce 0 |

#### Esempio:

Aggiungere lo script di esempio all'app ed eseguirlo. Per visualizzare il risultato, aggiungere i campi elencati nella colonna risultati a un foglio nell'app dell'utente.

RangeTab3:

```
LOAD recno() as RangeID, Rangesum(Field1,Field2,Field3) as MyRangeSum INLINE [
```

Field1, Field2, Field3

10,5,6

2,3,7

8,2,8

18,11,9

5,5,9

9,4,2

];

La tabella risultante mostra i valori restituiti in MyRangeSum per ciascun record della tabella.

Tabella risultante

| RangeID | MyRangeSum |
|---------|------------|
| 1       | 21         |
| 2       | 12         |
| 3       | 18         |
| 4       | 38         |
| 5       | 19         |
| 6       | 15         |

Esempio con espressione:

RangeSum (Above(MyField,0,3))

Restituisce la somma dei tre valori di **MyField**: dalla riga attuale e dalle due righe che la precedono. Specificando 3 per il terzo argomento, la funzione **Above()** restituisce tre valori, se il numero di righe che precedono la riga attuale è sufficiente, che vengono considerati come input per la funzione **RangeSum()**.

Dati utilizzati negli esempi:



Disattivare l'ordinamento di **MyField** per garantire il funzionamento corretto dell'esempio.

Dati campione

| MyField | RangeSum(Above(MyField,0,3)) |
|---------|------------------------------|
| 10      | 10                           |
| 2       | 12                           |
| 8       | 20                           |

| MyField | RangeSum(Above(MyField,0,3)) |
|---------|------------------------------|
| 18      | 28                           |
| 5       | 31                           |
| 9       | 32                           |

Dati utilizzati negli esempi:

```
RangeTab:
LOAD * INLINE [
MyField
10
2
8
18
5
9
] ;
```

**Vedere anche:**

p [Sum - funzione per grafici \(page 389\)](#)

p [Above - funzione per grafici \(page 1464\)](#)

## RangeTextCount

**RangeTextCount()** restituisce il numero di valori di testo nel campo o nell'espressione.

**Sintassi:**

```
RangeTextCount (first_expr[, Expression])
```

**Tipo di dati restituiti:** numero intero

**Argomenti:**

Gli argomenti di questa funzione possono contenere funzioni intra-record, che a loro volta restituiscono un elenco di valori.

Argomento

| Argomento  | Descrizione                                                               |
|------------|---------------------------------------------------------------------------|
| first_expr | L'espressione o il campo contenente i dati da misurare.                   |
| Expression | Le espressioni o campi opzionali contenenti la scala di dati da misurare. |

## Esempi e risultati:

## Esempi di funzioni

| Esempi                    | Risultati     |
|---------------------------|---------------|
| RangeTextCount (1,2,4)    | Restituisce 0 |
| RangeTextCount (5, 'abc') | Restituisce 1 |
| RangeTextCount (null( ))  | Restituisce 0 |

## Esempio con espressione:

RangeTextCount (Above(MaxString(MyField),0,3))

Restituisce il numero di valori di testo all'interno dei tre risultati della funzione **MaxString(MyField)** valutati in base alla riga attuale e alle due righe che la precedono.

## Dati utilizzati negli esempi:



Disattivare l'ordinamento di **MyField** per garantire il funzionamento corretto dell'esempio.

## Dati di esempio

| MyField | MaxString(MyField) | RangeTextCount(Above(Sum(MyField),0,3)) |
|---------|--------------------|-----------------------------------------|
| 10      | 10                 | 0                                       |
| abc     | abc                | 1                                       |
| 8       | 8                  | 1                                       |
| def     | def                | 2                                       |
| xyz     | xyz                | 2                                       |
| 9       | 9                  | 2                                       |

## Dati utilizzati negli esempi:

RangeTab:

LOAD \* INLINE [

MyField

10

'abc'

8

null()

```
'xyz'
```

```
9
```

```
] ;
```

#### Vedere anche:

p [TextCount - funzione per grafici \(page 418\)](#)

## RangeXIRR

**RangeXIRR()** restituisce il tasso di rendimento interno (annuale) per una programmazione di flussi di cassa che non è necessariamente su base periodica. Per calcolare il tasso di rendimento interno di una serie di flussi di cassa periodici, utilizzare la funzione **RangeIRR**.

La funzionalità XIRR di Qlik (funzioni **XIRR()** e **RangeXIRR()**) utilizza la seguente equazione, risolvendo il valore Rate, per determinare il valore XIRR corretto:

$$XNPV(\text{Rate}, \text{pmt}, \text{date}) = 0$$

l'equazione viene risolta usando una versione semplificata del metodo Newton.

#### Sintassi:

```
RangeXIRR(value, date{, value, date})
```

**Tipo di dati restituiti:** numerico

#### Argomenti

| Argomento | Descrizione                                                                                                                                                                              |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value     | Un flusso di cassa o una serie di flussi di cassa che corrisponde a una programmazione di date di pagamento. La serie di valori deve contenere almeno un valore positivo e uno negativo. |
| date      | Una data di pagamento o una programmazione di date di pagamento che corrisponde ai pagamenti con flusso di cassa.                                                                        |

Quando si lavora con questa funzione, si applicano le seguenti limitazioni:

- I valori di testo, i valori NULL e i valori mancanti vengono ignorati.
- Tutti i pagamenti sono scontati in base ad un anno composto da 365 giorni.
- Questa funzione richiede almeno un pagamento negativo valido e un pagamento positivo valido (con date corrispondenti valide). Se questi pagamenti non vengono forniti, sarà restituito un valore NULL.

I seguenti argomenti possono risultare utili per lavorare con questa funzione:

- [RangeXNPV \(page 1622\)](#): usare questa funzione per calcolare il valore attuale netto per una programmazione di flussi di cassa non necessariamente periodica.

- [XIRR \(page 434\)](#): la funzione **XIRR()** calcola il tasso di rendimento interno aggregato (annuale) per una programmazione di flussi di cassa (che non è necessariamente periodica).



Nelle diverse versioni di Qlik Sense Client-Managed, sono presenti delle variazioni nell'algoritmo sottostante usato da questa funzione. Per maggiori informazioni sugli aggiornamenti recenti dell'algoritmo, vedere l'articolo del supporto [Correzione e aggiornamento della funzione XIRR](#).

#### Esempi e risultati:

##### Esempi e risultati

| Esempi                                                          | Risultati          |
|-----------------------------------------------------------------|--------------------|
| <code>RangeXIRR(-2500, '2008-01-01', 2750, '2008-09-01')</code> | Restituisce 0,1532 |

#### Vedere anche:

- p [RangeIRR \(page 1596\)](#)
- p [RangeXNPV \(page 1622\)](#)
- p [XIRR \(page 434\)](#)
- ≤ [Correzione e aggiornamento della funzione XIRR](#)

## RangeXNPV

**XNPV()** restituisce il valore attuale netto per una programmazione di flussi di cassa (non necessariamente periodici) rappresentati da coppie di numeri nelle espressioni date da **pmt** e **date**. Tutti i pagamenti sono scontati in base ad un anno composto da 365 giorni.

#### Sintassi:

```
RangeXNPV (discount_rate, value, date{, value, date})
```

Tipo di dati restituiti: numerico

##### Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                               |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| discount_rate | <b>discount_rate</b> è il tasso annuale in base al quale dovrebbero essere scontati i pagamenti.                                                                                                                                                                                                                                          |
| value         | Un flusso di cassa o una serie di flussi di cassa che corrisponde a una programmazione di date di pagamento. Ciascun valore può essere un valore singolo oppure una scala di valori restituita da una funzione intra-record con un terzo parametro opzionale. La serie di valori deve contenere almeno un valore positivo e uno negativo. |
| date          | Una data di pagamento o una programmazione di date di pagamento che corrisponde ai pagamenti con flusso di cassa.                                                                                                                                                                                                                         |

Quando si lavora con questa funzione, si applicano le seguenti limitazioni:

- I valori di testo, i valori NULL e i valori mancanti vengono ignorati.
- Tutti i pagamenti sono scontati in base ad un anno composto da 365 giorni.

### Esempio - script

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Dati finanziari contenuti in una tabella chiamata RangeTab3.
- L'utilizzo della funzione **RangeXNPV()** per calcolare il valore attuale netto.

#### Script di caricamento

```
RangeTab3:
LOAD *,
recno() as RangeID,
RangeXNPV(DiscountRate,Value1,Date1,Value2,Date2) as RangeXNPV;
LOAD * INLINE [
DiscountRate|Value1|Date1|Value2|Date2
0.1|-100|2021-01-01|100|2022-01-01|
0.1|-100|2021-01-01|110|2022-01-01|
0.1|-100|2021-01-01|125|2022-01-01|
] (delimiter is '|');
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- RangeID
- RangeXNPV

Tabella dei risultati

| ID intervallo | Intervallo XNPV |
|---------------|-----------------|
| 1             | -\$9.09         |
| 2             | -\$0.00         |
| 3             | \$13.64         |

### Esempio - Espressione del grafico

Script di caricamento ed espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Dati finanziari contenuti in una tabella chiamata RangeTab3.
- L'utilizzo della funzione **RangeXNPV()** per calcolare il valore attuale netto.

#### Script di caricamento

```
RangeTab3:
LOAD *,
recno() as RangeID,
RangeXNPV(DiscountRate,Value1,Date1,Value2,Date2) as RangeXNPV;
LOAD * INLINE [
DiscountRate|Value1|Date1|Value2|Date2
0.1|-100|2021-01-01|100|2022-01-01|
0.1|-100|2021-01-01|110|2022-01-01|
0.1|-100|2021-01-01|125|2022-01-01|
] (delimiter is '|');
```

#### Risultati

Procedere come indicato di seguito:

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere il seguente calcolo come misura.

=RangeXNPV(0.1, -2500, '2008-01-01', 2750, '2008-09-01')

Tabella dei risultati

|                       |
|-----------------------|
| =XIRR(Payments, Date) |
| \$80.25               |

Vedere anche:

p [XNPV \(page 440\)](#)

## 8.22 Funzioni relazionali

Si tratta di un gruppo di funzioni che calcolano le proprietà dei singoli valori dimensionali in un grafico, utilizzando numeri già aggregati.

Si tratta di un gruppo di funzioni che calcolano le proprietà dei singoli valori dimensionali in un grafico, utilizzando numeri già aggregati. Ad esempio, non è possibile calcolare una classificazione senza un confronto con altri valori dimensionali.

Queste funzioni possono essere utilizzate solo nelle espressioni grafiche. Non possono essere utilizzati nello script di caricamento.

Nel grafico è richiesta una dimensione, che definisce gli altri punti di dati necessari per il confronto. Di conseguenza, una funzione relazionale non è significativa in un grafico senza dimensioni (ad esempio, un oggetto KPI).

### Funzioni di classificazione



*Se si utilizzano queste funzioni, verrà automaticamente disabilitata la soppressione dei valori zero. I valori NULL vengono ignorati.*

#### Rank

**Rank()** valuta le righe del grafico nell'espressione, e per ciascuna riga, visualizza la posizione relativa del valore della dimensione valutata nell'espressione. Quando valuta l'espressione, la funzione confronta il risultato con quello delle altre righe contenenti il segmento di colonna attuale e restituisce la classificazione della riga attuale all'interno del segmento.

**Rank - funzione per grafici** ([TOTAL [<fld {, fld}>]] expr[, mode[, fmt]])

#### HRank

**HRank()** valuta l'espressione, confronta il risultato con il risultato delle altre colonne contenenti il segmento di riga attuale di una tabella pivot. La funzione quindi restituisce la classificazione della colonna attuale all'interno del segmento.

**HRank - funzione per grafici** ([TOTAL] expr[, mode[, fmt]])

### Funzioni di raggruppamento

#### KMeans2D

Il gruppo di proprietà **Licenza sito** contiene proprietà correlate alla licenza per il sistema Qlik Sense. Tutti i campi sono obbligatori e non devono essere vuoti.

Proprietà licenza sito

| Nome della proprietà        | Descrizione                                                                         |
|-----------------------------|-------------------------------------------------------------------------------------|
| Nome proprietario           | Il nome utente del proprietario del prodotto Qlik Sense.                            |
| Organizzazione proprietario | Il nome dell'organizzazione di cui il proprietario prodotto Qlik Sense è un membro. |
| Numero di serie             | Il numero di serie assegnato al software Qlik Sense.                                |
| Codice di controllo         | Il numero di controllo assegnato al software Qlik Sense.                            |
| Accesso LEF                 | Il License Enabler File (LEF) assegnato al software Qlik Sense.                     |

**KMeans2D()** valuta le righe del grafico applicando il clustering K-means, e per ciascuna riga del grafico visualizza l'id cluster del cluster a cui è stato assegnato questo punto dati. Le colonne utilizzate dall'algoritmo di clustering sono determinate rispettivamente dai parametri `coordinate_1` e `coordinate_2`. Sono entrambe aggregazioni. Il numero di cluster creati è determinato dal parametro `num_clusters`. I dati possono essere normalizzati in via opzionale dal parametro `norm`.

```
KMeans2D - funzione per grafici(num_clusters, coordinate_1, coordinate_2 [, norm])
```

### KMeansND

**KMeansND()** valuta le righe del grafico applicando il clustering K-means, e per ciascuna riga del grafico visualizza l'id cluster del cluster a cui è stato assegnato questo punto dati. Le colonne utilizzate dall'algoritmo di clustering sono determinate dai parametri `coordinate_1`, `coordinate_2`, ecc. fino a `n` colonne. Sono tutte aggregazioni. Il numero di cluster creati è determinato dal parametro `num_clusters`.

```
KMeansND - funzione per grafici(num_clusters, num_iter, coordinate_1, coordinate_2 [, coordinate_3 [, ...]])
```

### KMeansCentroid2D

**KMeansCentroid2D()** valuta le righe del grafico applicando il clustering K-means, e per ciascuna riga del grafico visualizza la coordinata desiderata del cluster a cui è stato assegnato questo punto dati. Le colonne utilizzate dall'algoritmo di clustering sono determinate rispettivamente dai parametri `coordinate_1` e `coordinate_2`. Sono entrambe aggregazioni. Il numero di cluster creati è determinato dal parametro `num_clusters`. I dati possono essere normalizzati in via opzionale dal parametro `norm`.

```
KMeansCentroid2D - funzione per grafici(num_clusters, coordinate_no, coordinate_1, coordinate_2 [, norm])
```

### KMeansCentroidND

**KMeansCentroidND()** valuta le righe del grafico applicando il clustering K-means, e per ciascuna riga del grafico visualizza la coordinata desiderata del cluster a cui è stato assegnato questo punto dati. Le colonne utilizzate dall'algoritmo di clustering sono determinate dai parametri `coordinate_1`, `coordinate_2`, ecc. fino a `n` colonne. Sono tutte aggregazioni. Il numero di cluster creati è determinato dal parametro `num_clusters`.

```
KMeansCentroidND - funzione per grafici(num_clusters, num_iter, coordinate_no, coordinate_1, coordinate_2 [, coordinate_3 [, ...]])
```

## Funzioni di scomposizione serie temporale

### STL\_Trend

**STL\_Trend** è una funzione di scomposizione delle serie temporali. Insieme a **STL\_Seasonal** e a **STL\_Residual**, questa funzione viene utilizzata per scomporre una serie temporale in componenti stagionali, di tendenza e residuali. Nel contesto dell'algoritmo STL, la scomposizione delle serie temporali viene utilizzata per identificare sia un modello stagionale ricorrente sia una tendenza generale, data una metrica di ingresso e altri parametri. La funzione **STL\_Trend** identifica una tendenza generale, indipendente da schemi o cicli stagionali, dai dati delle serie temporali.

```
STL_Trend - funzione per grafici(target_measure, period_int [,seasonal_smoother [,trend_smoother]])
```

### STL\_Seasonal

**STL\_Seasonal** è una funzione di scomposizione delle serie temporali. Insieme a **STL\_Trend** e a **STL\_Residual**, questa funzione viene utilizzata per scomporre una serie temporale in componenti stagionali, di tendenza e residuali. Nel contesto dell'algoritmo STL, la scomposizione delle serie temporali viene utilizzata per identificare sia un modello stagionale ricorrente sia una tendenza generale, data una metrica di ingresso e altri parametri. La funzione **STL\_Seasonal** è in grado di identificare un modello stagionale all'interno di una serie temporale, separandolo dalla tendenza generale mostrata dai dati.

```
STL_Seasonal - funzione per grafici(target_measure, period_int [,seasonal_smoother [,trend_smoother]])
```

### STL\_Residual

**STL\_Residual** è una funzione di scomposizione delle serie temporali. Insieme a **STL\_Seasonal** e a **STL\_Trend**, questa funzione viene utilizzata per scomporre una serie temporale in componenti stagionali, di tendenza e residuali. Nel contesto dell'algoritmo STL, la scomposizione delle serie temporali viene utilizzata per identificare sia un modello stagionale ricorrente sia una tendenza generale, data una metrica di ingresso e altri parametri. Nell'eseguire questa operazione, una parte della variazione della metrica di input non rientrerà né nella componente stagionale né in quella di tendenza e sarà definita come componente residua. La funzione grafico **STL\_Residual** acquisisce questa parte del calcolo.

```
STL_Residual - funzione per grafici(target_measure, period_int [,seasonal_smoother [,trend_smoother]])
```

## Rank - funzione per grafici

**Rank()** valuta le righe del grafico nell'espressione, e per ciascuna riga, visualizza la posizione relativa del valore della dimensione valutata nell'espressione. Quando valuta l'espressione, la funzione confronta il risultato con quello delle altre righe contenenti il segmento di colonna attuale e restituisce la classificazione della riga attuale all'interno del segmento.

| Rank | Product Group       | Region  | Sales        |
|------|---------------------|---------|--------------|
| -    | -                   | -       | 104852674.81 |
| 1    | Alcoholic Beverages | Japan   | 1342137      |
| 2    | Alcoholic Beverages | USA     | 713227.4     |
| 3    | Alcoholic Beverages | Nordic  | 177327.42    |
| 4    | Alcoholic Beverages | UK      | 121212.81    |
| 5    | Alcoholic Beverages | Spain   | 52294.66     |
| 6    | Alcoholic Beverages | Germany | 34235.98     |
| 1    | Baked Goods         | USA     | 129996.04    |
| 2    | Baked Goods         | UK      | 108020.84    |
| 3    | Baked Goods         | Japan   | 85282.02     |
| 4    | Baked Goods         | Nordic  | 69138.51     |
| 5    | Baked Goods         | Germany | 34071.04     |
| 6    | Baked Goods         | Spain   | 22792.64     |
| 1    | Baking Goods        | UK      | 481369.57    |
| 2    | Baking Goods        | USA     | 1179065.7    |
| 3    | Baking Goods        | Japan   | 504910.68    |
| 4    | Baking Goods        | Nordic  | 189728.55    |
| 5    | Baking Goods        | Germany | 127157.65    |
| 6    | Baking Goods        | Spain   | 118308.26    |
| 1    | Beverages           | USA     | 2528718.15   |
| 2    | Beverages           | UK      | 2195896.97   |
| 3    | Beverages           | Japan   | 785487.64    |
| 4    | Beverages           | Nordic  | 305995.76    |
| 5    | Beverages           | Spain   | 246216.7     |
| 6    | Beverages           | Germany | 229205.51    |
| 1    | Breakfast Foods     | UK      | 423952.21    |
| 2    | Breakfast Foods     | USA     | 142792.95    |
| 3    | Breakfast Foods     | Japan   | 36256.1      |
| 4    | Breakfast Foods     | Nordic  | 21278.18     |
| 5    | Breakfast Foods     | Germany | 15878.15     |
| 6    | Breakfast Foods     | Spain   | 6997.53      |
| 1    | Canned Products     | USA     | 15548195.8   |
| 2    | Canned Products     | UK      | 1603884.49   |
| 3    | Canned Products     | Japan   | 1505446.57   |
| 4    | Canned Products     | Nordic  | 1285244.35   |
| 5    | Canned Products     | Germany | 426928.58    |
| 6    | Canned Products     | Spain   | 307962.25    |
| 1    | Dairy               | USA     | 1015952.96   |

### Segmenti delle colonne

|                   | Region   | Country                  | Population  | Rank(Population) |
|-------------------|----------|--------------------------|-------------|------------------|
| Column segment #1 | Americas | Mexico                   | 128,932,753 | 2                |
|                   | Americas | Canada                   | 37,742,154  | 3                |
|                   | Americas | United States of America | 331,002,851 | 1                |
| Column segment #2 | Europe   | Sweden                   | 10,099,265  | 4                |
|                   | Europe   | United Kingdom           | 67,886,011  | 2                |
|                   | Europe   | France                   | 65,273,511  | 3                |
|                   | Europe   | Germany                  | 83,789,342  | 1                |

Per i grafici che non includono tabelle, il segmento di colonna attuale è definito come visualizzato nell'equivalente di tabella lineare del grafico.

### Sintassi:

**Rank** ([TOTAL] expr[, mode[, fmt]])

**Tipo di dati restituiti:** duale

**Argomenti:**

### Argomenti

| Argomento | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| expr      | L'espressione o il campo contenente i dati da misurare.                                                                                                                                                                                                                                                                                                                                                                                        |
| mode      | Specifica la rappresentazione numerica del risultato della funzione.                                                                                                                                                                                                                                                                                                                                                                           |
| fmt       | Specifica la rappresentazione testuale del risultato della funzione.                                                                                                                                                                                                                                                                                                                                                                           |
| TOTAL     | Se il grafico è unidimensionale o se l'espressione è preceduta dal qualificatore <b>TOTAL</b> , la funzione viene valutata lungo l'intera colonna. Se la tabella o l'equivalente della tabella presenta più dimensioni verticali, il segmento di colonna attuale includerà solo righe con lo stesso valore della riga attuale in tutte le colonne di dimensione, eccetto la colonna che mostra l'ultima dimensione nell'ordinamento tra campi. |

La classificazione viene restituita come valore duale che, nel caso in cui ogni riga presenti una classificazione univoca, sarà un numero intero compreso tra 1 e il numero di righe nel segmento di colonna attuale.

Nel caso in cui molte righe condividano la stessa classificazione, la rappresentazione testuale e numerica potrà essere controllata mediante i parametri **mode** e **fmt**.

### **mode**

Il secondo argomento, **mode**, può assumere i seguenti valori:

#### Esempi **mode**

| Valore          | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0 (predefinito) | Se tutte le classificazioni all'interno del gruppo condiviso rientrano nella parte bassa del valore medio dell'intera classificazione, tutte le righe all'interno di tale gruppo riceveranno la classificazione più bassa.<br><br>Se tutte le classificazioni all'interno del gruppo condiviso rientrano nella parte alta del valore medio dell'intera classificazione, tutte le righe all'interno di tale gruppo riceveranno la classificazione più alta.<br><br>Se tutte le classificazioni all'interno del gruppo condiviso si distribuiscono lungo il valore medio dell'intera classificazione, tutte le righe all'interno di tale gruppo riceveranno il valore corrispondente alla media calcolata in base alla classificazione massima e minima dell'intero segmento di colonna. |
| 1               | Classificazione minima su tutte le righe.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 2               | Classificazione media su tutte le righe.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 3               | Classificazione massima su tutte le righe.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 4               | Classificazione minima sulla prima riga, quindi incrementata di uno per ogni riga.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

### **fmt**

Il terzo argomento, **fmt**, può assumere i seguenti valori:

#### Esempi **fmt**

| Valore          | Descrizione                                                       |
|-----------------|-------------------------------------------------------------------|
| 0 (predefinito) | Valore basso - valore alto su tutte le righe (ad esempio, 3 - 4). |
| 1               | Valore basso su tutte le righe.                                   |
| 2               | Valore basso sulla prima riga, vuoto sulle righe successive.      |

L'ordine delle righe per **mode** 4 e **fmt** 2 è determinato dall'ordine delle dimensioni del grafico.

### Esempi e risultati:

Creare due visualizzazioni dalle dimensioni Product e Sales, e un'altra da Product e da UnitSales. Aggiungere le misure come mostrato nella seguente tabella.

Esempi di classificazione

| Esempi                                                                                            | Risultati                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|---------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Esempio 1. Creare una tabella con le dimensioni Customer e Sales e la misura Rank(Sales)          | <p>Il risultato dipende dall'ordinamento delle dimensioni. Se viene ordinata in base a Customer, la tabella elencherà tutti i valori di Sales per Astrida, quindi Betacab e così via. Il risultato per Rank(Sales) mostrerà 10 per il valore Sales 12, 9 per il valore Sales 13 e così via, con il valore di classificazione 1 restituito per il valore Sales 78. Il segmento di colonna successivo inizia con Betacab, per il quale, il primo valore di Sales nel segmento è 12. Il valore di classificazione di Rank(Sales) per questo è fornito come 11.</p> <p>Se la tabella viene ordinata in base a Sales, i segmenti colonna consisteranno nei valori di Sales e del Customer corrispondente. Poiché vi sono due valori Sales di 12 (per Astrida e Betacab), il valore di Rank(Sales) per tale segmento di colonna è 1-2, per ciascun valore di Customer. Questo perché sono presenti due valori di Customer per il valore Sales 12. Se fossero stati presenti 4 valori, il risultato sarebbe stato 1-4 per tutte le righe. Questo mostra l'aspetto del risultato per il valore predefinito (0) dell'argomento fmt.</p> |
| Esempio 2. Sostituire la dimensione Customer con Product e aggiungere la misura Rank(Sales, 1, 2) | Viene restituito 1 sulla prima riga di ciascun segmento di colonna, mentre tutte le altre righe vengono lasciate vuote perché gli argomenti <b>mode</b> e <b>fmt</b> sono impostati rispettivamente su 1 e 2.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

Risultati per l'esempio 1, con la tabella ordinata per Customer:

Tabella dei risultati

| Customer | Sales | Rank(Sales) |
|----------|-------|-------------|
| Astrida  | 12    | 10          |
| Astrida  | 13    | 9           |
| Astrida  | 20    | 8           |
| Astrida  | 22    | 7           |
| Astrida  | 45    | 6           |
| Astrida  | 46    | 5           |

| Customer | Sales | Rank(Sales) |
|----------|-------|-------------|
| Astrida  | 60    | 4           |
| Astrida  | 65    | 3           |
| Astrida  | 70    | 2           |
| Astrida  | 78    | 1           |
| Betcab   | 12    | 11          |

Risultati per l'esempio 1, con la tabella ordinata per Sales:

Tabella dei risultati

| Customer   | Sales | Rank(Sales) |
|------------|-------|-------------|
| Astrida    | 12    | 1-2         |
| Betacab    | 12    | 1-2         |
| Astrida    | 13    | 1           |
| Betacab    | 15    | 1           |
| Astrida    | 20    | 1           |
| Astrida    | 22    | 1-2         |
| Betacab    | 22    | 1-2         |
| Betacab    | 24    | 1-2         |
| Canutility | 24    | 1-2         |

Dati utilizzati negli esempi:

ProductData:

```
Load * inline [
```

```
Customer|Product|UnitsSales|UnitPrice
```

```
Astrida|AA|4|16
```

```
Astrida|AA|10|15
```

```
Astrida|BB|9|9
```

```
Betacab|BB|5|10
```

```
Betacab|CC|2|20
```

```
Betacab|DD|0|25
```

```
Canutility|AA|8|15
```

```
Canutility|CC|0|19
```

```
] (delimiter is '|');
```

```
Sales2013:
crosstable (Month, Sales) LOAD * inline [
Customer|Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|Sep|Oct|Nov|Dec
Astrida|46|60|70|13|78|20|45|65|78|12|78|22
Betacab|65|56|22|79|12|56|45|24|32|78|55|15
Canutility|77|68|34|91|24|68|57|36|44|90|67|27
Divadip|57|36|44|90|67|27|57|68|47|90|80|94
] (delimiter is '|');
```

#### Vedere anche:

p [Sum - funzione per grafici \(page 389\)](#)

## HRank - funzione per grafici

**HRank()** valuta l'espressione, confronta il risultato con il risultato delle altre colonne contenenti il segmento di riga attuale di una tabella pivot. La funzione quindi restituisce la classificazione della colonna attuale all'interno del segmento.

#### Sintassi:

```
HRank ([ TOTAL ] expr [ , mode [ , fmt ] ])
```

Tipo di dati restituiti: duale



*Questa funzione è valida solo per le tabelle pivot. In tutti gli altri tipi di grafici restituisce NULL.*

#### Argomenti:

##### Argomenti

| Argomento | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| expr      | L'espressione o il campo contenente i dati da misurare.                                                                                                                                                                                                                                                                                                                                                                                        |
| mode      | Specifica la rappresentazione numerica del risultato della funzione.                                                                                                                                                                                                                                                                                                                                                                           |
| fmt       | Specifica la rappresentazione testuale del risultato della funzione.                                                                                                                                                                                                                                                                                                                                                                           |
| TOTAL     | Se il grafico è unidimensionale o se l'espressione è preceduta dal qualificatore <b>TOTAL</b> , la funzione viene valutata lungo l'intera colonna. Se la tabella o l'equivalente della tabella presenta più dimensioni verticali, il segmento di colonna attuale includerà solo righe con lo stesso valore della riga attuale in tutte le colonne di dimensione, eccetto la colonna che mostra l'ultima dimensione nell'ordinamento tra campi. |

Se la tabella pivot è unidimensionale o se è preceduta dal qualificatore **total**, il segmento di riga attuale sarà sempre uguale all'intera riga. Se una tabella pivot include più dimensioni orizzontali, il segmento della riga attuale includerà solo le colonne con gli stessi valori della colonna attuale in tutte le righe della dimensione, tranne per la riga che visualizza l'ultima dimensione orizzontale dell'ordinamento tra campi.

La classificazione viene restituita come valore duale che, nel caso in cui ciascuna colonna presenti una classificazione univoca, avrà come valore un numero intero compreso tra 1 e il numero di colonne nel segmento della riga attuale.

Nel caso in cui molte righe condividano la stessa classificazione, la rappresentazione testuale e numerica potrà essere controllata mediante gli argomenti **mode** e **format**.

Il secondo argomento **mode** specifica la rappresentazione numerica del risultato della funzione:

### Esempi **mode**

| Valore          | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0 (predefinito) | <p>Se tutte le classificazioni all'interno del gruppo condiviso ricadono nella parte bassa del valore medio dell'intera classificazione, tutte le colonne all'interno di tale gruppo riceveranno la classificazione più bassa.</p> <p>Se tutte le classificazioni all'interno del gruppo condiviso ricadono nella parte alta del valore medio dell'intera classificazione, tutte le colonne all'interno di tale gruppo riceveranno la classificazione più alta.</p> <p>Se tutte le classificazioni all'interno del gruppo condiviso si distribuiscono lungo il valore medio dell'intera classificazione, tutte le righe all'interno di tale gruppo riceveranno il valore corrispondente alla media calcolata in base alla classificazione massima e minima dell'intero segmento di colonna.</p> |
| 1               | Classificazione minima su tutte le colonne nel gruppo.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 2               | Classificazione media su tutte le colonne nel gruppo.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 3               | Classificazione massima su tutte le colonne nel gruppo.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 4               | Classificazione minima sulla prima colonna, quindi incrementata di uno per ciascuna colonna nel gruppo.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

Il terzo argomento, **format**, specifica la rappresentazione testuale del risultato della funzione:

### Esempi **format**

| Valore          | Descrizione                                                                  |
|-----------------|------------------------------------------------------------------------------|
| 0 (predefinito) | Valore basso&' - '&valore alto su tutte le colonne (ad esempio 3 - 4).       |
| 1               | Valore basso su tutte le colonne nel gruppo.                                 |
| 2               | Valore basso nella prima colonna, vuoto nelle colonne successive nel gruppo. |

L'ordine delle colonne per **mode** 4 e **format** 2 è determinato dall'ordinamento delle dimensioni del grafico.

**Esempi:**

```
HRank( sum( Sales ))
```

```
HRank( sum( Sales ), 2 )
```

```
HRank( sum( Sales ), 0, 1 )
```

**"Ottimizzazione con k-means: un esempio del mondo reale"**

L'esempio seguente illustra un caso d'uso del mondo reale, in cui le funzioni di clustering KMeans e Centroid sono applicate a un set di dati. La funzione KMeans segrega i punti dati in cluster che condividono somiglianze. I cluster diventano più compatti e differenziati dato che l'algoritmo KMeans viene applicato su un numero configurabile di iterazioni.

KMeans è una funzione utilizzata in molti campi in un'ampia varietà di casi di utilizzo; alcuni esempi di casi di utilizzo del clustering includono la segmentazione dei clienti, il rilevamento di frodi, la previsione di attriti negli account, il targeting degli incentivi ai clienti, l'identificazione del cybercrimine e l'ottimizzazione dei percorsi di consegna. L'algoritmo di clustering KMeans risulta sempre più utilizzato laddove le aziende cercano di inferire schemi e ottimizzare le offerte di servizi.

**Funzioni KMeans e Centroid di Qlik Sense**

Qlik Sense fornisce due funzioni KMeans che raggruppano i punti dati in cluster in base alla somiglianza. Vedere [KMeans2D - funzione per grafici \(page 1643\)](#) e [KMeansND - funzione per grafici \(page 1658\)](#). La funzione **KMeans2D** accetta due dimensioni e funziona bene al momento di visualizzare risultati attraverso un **grafico a dispersione**. La funzione **KMeansND** accetta più di due dimensioni. Essendo semplice concettualizzare un esito 2D su grafici standard, la seguente dimostrazione applica KMeans a un **grafico a dispersione** usando due dimensioni. Il clustering KMeans può essere visualizzato attraverso il coloramento per espressione; oppure per dimensione come descritto in questo esempio.

Le funzioni centroide di Qlik Sense determinano la posizione aritmetica media di tutti i punti dati nel cluster e identificano un punto centrale, o centroide, per tale cluster. Per ciascuna riga del grafico (o record), la funzione centroide visualizza la coordinata del cluster a cui è stato assegnato questo punto dati. Vedere [KMeansCentroid2D - funzione per grafici \(page 1673\)](#) e [KMeansCentroidND - funzione per grafici \(page 1674\)](#).

**Panoramica sui casi di utilizzo ed esempio**

Il seguente esempio si riferisce a uno scenario simulato del mondo reale. Un'azienda tessile nello stato di New York, USA, interessata a ridurre le spese minimizzando i costi di consegna. Un modo per farlo è trasferire i magazzini più vicini ai loro distributori. L'azienda impiega 118 distributori in tutto lo stato di New York. La seguente dimostrazione simula come un operations manager potrebbe segmentare i distributori in cinque geografie raggruppate usando la funzione KMeans e poi identificare cinque posizioni di magazzino ottimali centrali rispetto a quei cluster usando la funzione centroide. L'obiettivo è individuare le coordinate di mappatura utilizzabili per identificare cinque posizioni centrali del magazzino.

## Il set di dati

Il set di dati è basato su nomi e indirizzi generati casualmente nello stato di New York con coordinate reali di latitudine e longitudine. Il set di dati contiene le seguenti dieci colonne: id, first\_name, last\_name, telefono, indirizzo, città, stato, codice postale, latitudine, longitudine. Il set di dati è disponibile in basso come file da scaricare localmente e poi caricare in Qlik Sense o inline per l'editor caricamento dati. L'app creata viene denominata *Distributori KMeans e Centroidi* e il primo foglio nell'app viene denominato *Analisi cluster di distribuzione*.

Selezionare il collegamento seguente per scaricare il file di dati campione: [DistributorData.csv](#)

[Set di dati Distributor: caricamento inline per l'editor caricamento dati in Qlik Sense \(page 1641\)](#)

Titolo: DistributorData

Numero totale di record: 118

## Applicazione della funzione KMeans2D

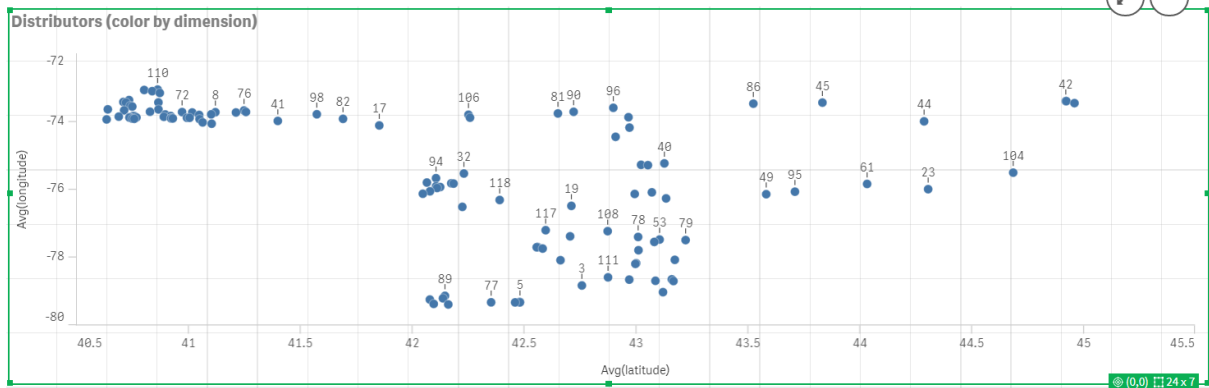
In questo esempio, la configurazione di un grafico a **dispersione** viene dimostrata utilizzando il set di dati *DistributorData*, viene applicata la funzione **KMeans2D**, mentre il grafico viene colorato in base alla dimensione.

Notare che le funzioni KMeans Qlik Sense supportano il clustering automatico usando un metodo definito differenza di profondità (DeD). Quando un utente imposta lo 0 per il numero di cluster, viene determinato il numero ottimale di cluster per tale set di dati. Tuttavia, per questo esempio viene creata una variabile per l'argomento **num\_clusters** (fare riferimento a [KMeans2D - funzione per grafici \(page 1643\)](#) per la sintassi). Pertanto, il numero desiderato di cluster (k=5) viene specificato mediante una variabile.

1. Un **Grafico a dispersione** è trascinato sul foglio e denominato *Distributori (per dimensione)*.
2. Viene creata una **variabile** per specificare il numero di cluster. La **variabile** viene denominata *vDistClusters*. Per la variabile **Definizione**, viene inserito 5.
3. Configurazione **Dati** per il grafico:
  - a. Sotto **Dimensioni**, il campo *id* viene selezionato per **Bolla**. *Id cluster* viene inserito per l'**Etichetta**.
  - b. Sotto **Misure**, *Avg([latitude])* rappresenta l'espressione per l'**asse X**.
  - c. Sotto **Misure**, *Avg([longitude])* rappresenta l'espressione per l'**asse Y**.
4. Configurazione dell'**Aspetto**:
  - a. Sotto **Colori e legenda**, viene scelto **Personalizzato** per **Colori**.
  - b. **Per dimensione** viene selezionato per colorare il grafico.
  - c. Viene inserita la seguente espressione: `=pick(aggr(KMeans2D(vDistClusters,only(latitude),only(longitude)),id)+1, 'Cluster 1', 'Cluster 2', 'Cluster 3', 'Cluster 4', 'Cluster 5')`
  - d. Viene selezionata la casella di controllo per **Colori persistenti**.

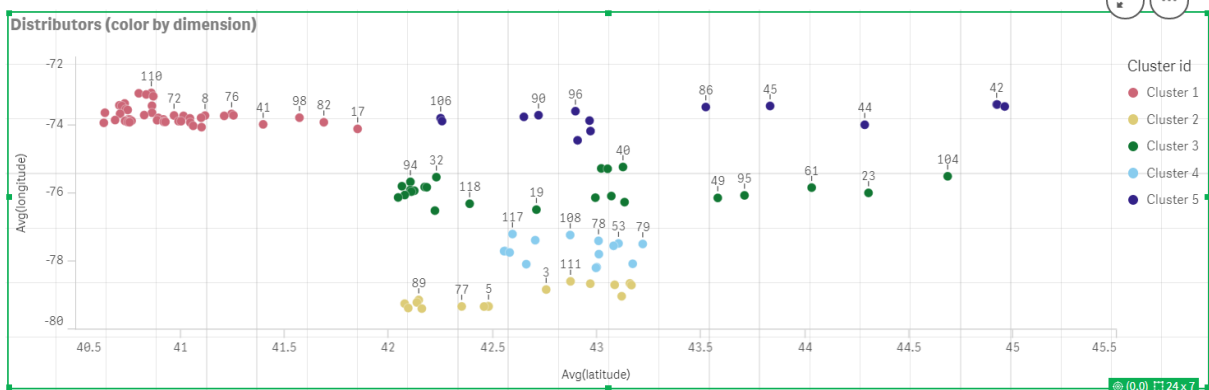
Viene applicato Grafico a dispersione prima del coloramento KMeans per dimensione

Distribution cluster analysis



Viene applicato Grafico a dispersione dopo il coloramento KMeans per dimensione

Distribution cluster analysis



### Aggiungere una tabella: *Distributori*

Può essere utile avere a disposizione una tabella per l'accesso rapido ai dati pertinenti. Il grafico a **dispersione** mostra gli *id* attraverso una tabella con i nomi dei distributori corrispondenti che viene aggiunta per riferimento.

1. Una **Tabella lineare** denominata *Distributori* viene trascinata sul foglio con le seguenti **Colonne** (Dimensioni) aggiunte: *id*, *first\_name* e *last\_name*.

Tabella: Nomi distributore

| Distributors |    |            |           |
|--------------|----|------------|-----------|
|              | id | first_name | last_name |
|              | 1  | Kaiya      | Snow      |
|              | 2  | Dean       | Roy       |
|              | 3  | Eden       | Paul      |
|              | 4  | Bryanna    | Higgins   |
|              | 5  | Elisabeth  | Lee       |
|              | 6  | Skylar     | Robinson  |
|              | 7  | Cody       | Bailey    |
|              | 8  | Dario      | Sims      |
|              | 9  | Deacon     | Hood      |

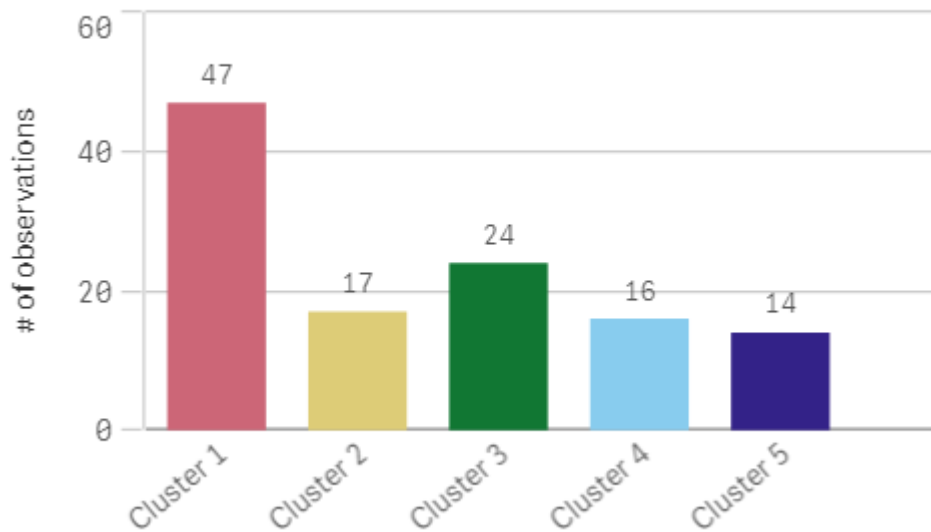
### Aggiunta di un grafico a barre: # osservazioni per cluster

Per uno scenario di distribuzione magazzino, è utile conoscere quanti distributori verranno serviti da ciascun magazzino. Pertanto, viene creato un **grafico a barre** che misura quanti distributori vengono assegnati a ciascun cluster.

1. Un **grafico a barre** viene trascinato sul foglio. Il grafico è denominato: *# osservazioni per cluster*.
2. Configurazione **Dati** per il **grafico a barre**:
  - a. Viene aggiunta una **Dimensione** denominata *Cluster* (l'etichetta può essere aggiunta dopo l'applicazione dell'espressione). Viene inserita la seguente espressione: `=pick(aggr(KMeans2D(vDistClusters,only(latitude),only(longitude)),id)+1, 'Cluster 1', 'Cluster 2', 'Cluster 3', 'Cluster 4', 'Cluster 5')`
  - b. Viene aggiunta una **Misura** denominata *# di osservazioni*. Viene inserita la seguente espressione: `=count(aggr(KMeans2D(vDistClusters,only(latitude),only(longitude)),id))`
3. Configurazione dell'**Aspetto**:
  - a. Sotto **Colori e legenda**, viene scelto **Personalizzato** per **Colori**.
  - b. **Per dimensione** viene selezionato per colorare il grafico.
  - c. Viene inserita la seguente espressione: `=pick(aggr(KMeans2D(vDistClusters,only(latitude),only(longitude)),id)+1, 'Cluster 1', 'Cluster 2', 'Cluster 3', 'Cluster 4', 'Cluster 5')`
  - d. Viene selezionata la casella di controllo per **Colori persistenti**.
  - e. **Mostra legenda** è disattivato.
  - f. Sotto **Presentazione**, **Etichette valore** è impostato su **Auto**.
  - g. In **Asse X**: sono selezionati **Cluster**, **Solo etichette**.

Grafico a barre: # osservazioni per cluster

### # observations per cluster



### Applicazione della funzione **Centroid2D**

Viene aggiunta una seconda tabella per la funzione **Centroid2D** che identificherà le coordinate per le potenziali posizioni del magazzino. Questa tabella mostra la posizione centrale (valori centroide) per i cinque gruppi distributori identificati.

1. Una **Tabella** viene trascinata sul foglio e denominata *Centroidi cluster* con le seguenti colonne aggiunte:
  - a. Viene aggiunta una **Dimensione** denominata *Cluster*. Viene inserita la seguente espressione: `=pick(aggr(KMeans2D(vDistClusters,only(latitude),only(longitude)),id)+1,'Warehouse 1','Warehouse 2','Warehouse 3','Warehouse 4','Warehouse 5')`
  - b. Viene aggiunta una **Misura** denominata *latitudine (D1)*. Viene inserita la seguente espressione: `=only(aggr(KMeansCentroid2D(vDistClusters,0,only(latitude),only(longitude)),id))`  
 Notare che il parametro **coordinate\_no** corrisponde alla prima dimensione(0). In questo caso la dimensione *latitudine* viene tracciata rispetto all'asse x. Se stessimo lavorando con la funzione **CentroidND** e ci fossero fino a sei dimensioni, tali voci di parametro potrebbero corrispondere a uno qualsiasi di sei valori: 0,1,2,3,4,o 5.
  - c. Viene aggiunta una **Misura** denominata *longitudine (D2)*. Viene inserita la seguente espressione: `=only(aggr(KMeansCentroid2D(vDistClusters,1,only(latitude),only(longitude)),id))`  
 Il parametro **coordinate\_no** in questa espressione corrisponde alla seconda dimensione(1). La dimensione *longitudine* viene tracciata rispetto all'asse y.

Tabella: calcoli centroide cluster

| Cluster centroids |          |                 |                  |
|-------------------|----------|-----------------|------------------|
|                   | Clusters | latitude (D1)   | longitude (D2)   |
| <b>Totals</b>     |          | -               | -                |
| Warehouse 1       |          | 40.945422240426 | -73.719966482979 |
| Warehouse 2       |          | 42.590538729412 | -79.067889217647 |
| Warehouse 3       |          | 42.805089516667 | -75.901621883333 |
| Warehouse 4       |          | 42.8581692625   | -77.6800485875   |
| Warehouse 5       |          | 43.436770771429 | -73.734622635714 |

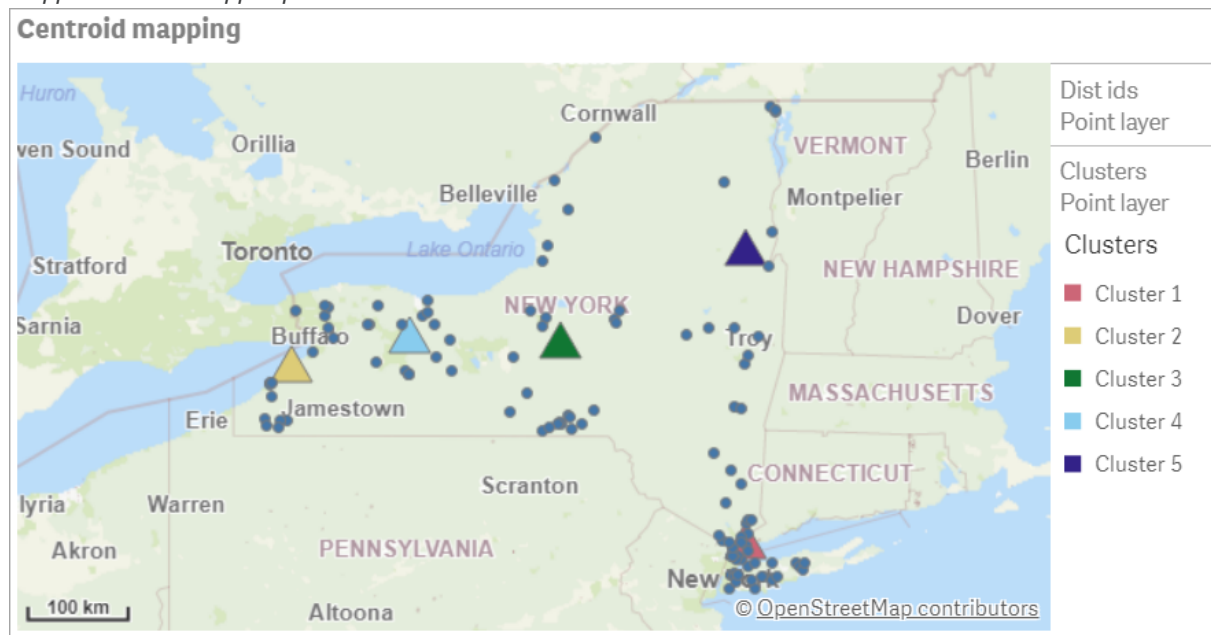
## Mapping centroide

Il passaggio successivo consiste nel mappare i centroidi. Spetta allo sviluppatore di app scegliere se preferisce posizionare la visualizzazione su fogli separati.

1. Una **mappa** denominata *Mapping centroide* viene trascinata sul foglio.
2. Nella sezione **Livelli**. Viene selezionato **Aggiungi livello**, quindi viene selezionato **Livello punti**.
  - a. L'**Id campo** viene selezionato e si aggiunge l'**etichetta Dist ids**.
  - b. Nella sezione **Posizione**, la casella di controllo per **Campi latitudine e longitudine** risulta selezionata.
  - c. Per **Latitudine**, viene selezionato il campo *latitudine*.
  - d. Per **Longitudine**, viene selezionato il campo *longitudine*.
  - e. Nella sezione **Dimensione e forma**, viene selezionato **Bolla per Forma**, quindi la **Dimensione** viene ridotta in base alla preferenza sul cursore.
  - f. Nella sezione **Colori**, viene selezionato **Colore singolo** e viene selezionato blu per il **Colore** e grigio per il colore **Contorno** (tali scelte rappresentano anche una preferenza).
3. Nella sezione **Livelli**, un secondo **Livello punti** viene aggiunto selezionando **Aggiungi livello** e quindi selezionando **Livello punti**.
  - a. Viene inserita la seguente espressione: `=aggr(KMeans2D(vDistClusters,only(latitude),only(longitude)),id)`
  - b. Viene aggiunta l'**Etichetta Cluster**.
  - c. Nella sezione **Posizione**, la casella di controllo per **Campi latitudine e longitudine** risulta selezionata.
  - d. Per la **Latitudine** che in questo caso viene tracciata lungo l'asse x, viene aggiunta la seguente espressione: `=aggr(KMeansCentroid2D(vDistClusters,0,only(latitude),only(longitude)),id)`
  - e. Per la **Longitudine** che in questo caso viene tracciata lungo l'asse y, viene aggiunta la seguente espressione: `=aggr(KMeansCentroid2D(vDistClusters,1,only(latitude),only(longitude)),id)`

- f. Nella sezione **Dimensioni e forma**, viene selezionato **Triangolo** per **Forma** e le **Dimensioni** vengono ridotte sul cursore alla preferenza.
  - g. Sotto **Colori e legenda**, viene selezionato **Personalizzato** per **Colori**.
  - h. **Per dimensione** viene selezionato per colorare il grafico. Viene inserita la seguente espressione: `=pick(aggr(KMeans2D(vDistClusters,only(latitude),only(longitude)),id)+1,'Cluster 1','Cluster 2','Cluster 3','Cluster 4','Cluster 5')`
  - i. La dimensione viene etichettata *Cluster*.
4. In **Impostazioni mappa**, viene selezionato **Adattiva** per **Proiezione**. **Metrica** viene selezionato per **Unità di misura**.

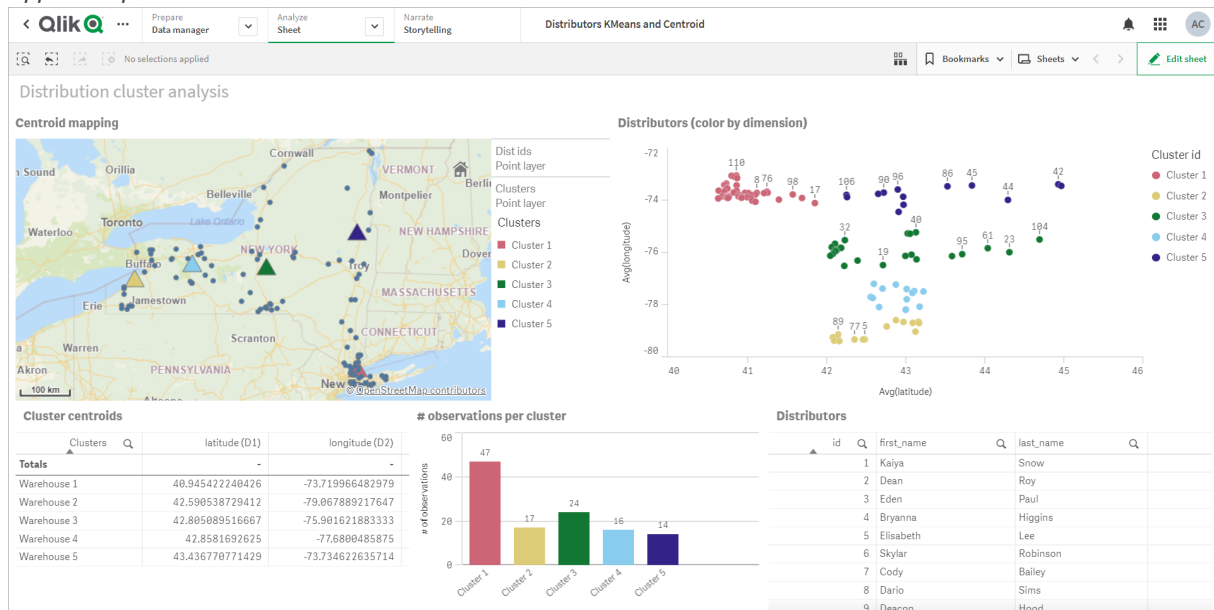
Mappa: centroidi mappati per cluster



## Conclusione

Utilizzando la funzione KMeans per questo scenario del mondo reale, i distributori sono stati segmentati in gruppi simili o cluster basati sulla somiglianza; in questo caso, la prossimità l'uno con l'altro. La funzione Centroide è stata applicata a tali cluster per identificare cinque coordinate di mappatura. Tali coordinate forniscono una posizione centrale iniziale su cui costruire o localizzare i magazzini. La funzione centroide si applica al grafico **mappa**, in modo che gli utenti delle app possano visualizzare dove si trovano i centroidi rispetto ai punti dati cluster circostanti. Le coordinate risultanti rappresentano potenziali posizioni dei magazzini che potrebbero ridurre al minimo i costi di consegna ai distributori nello stato di New York.

## App: esempio di analisi KMeans e centroide



## Set di dati Distributor: caricamento inline per l'editor caricamento dati in Qlik Sense

DistributorData:

Load \* Inline [

id,first\_name,last\_name,telephone,address,city,state,zip,latitude,longitude

1,Kaiya,Snow,(716) 201-1212,6231 Tonawanda Creek Rd #APT 308,Lockport,NY,14094,43.08926,-78.69313

2,Dean,Roy,(716) 201-1588,6884 E High St,Lockport,NY,14094,43.16245,-78.65036

3,Eden,Paul,(716) 202-4596,4647 Southwestern Blvd #APT 350,Hamburg,NY,14075,42.76003,-78.83194

4,Bryanna,Higgins,(716) 203-7041,418 Park Ave,Dunkirk,NY,14048,42.48279,-79.33088

5,Elisabeth,Lee,(716) 203-7043,36 E Courtney St,Dunkirk,NY,14048,42.48299,-79.31928

6,Skylar,Robinson,(716) 203-7166,26 Greco Ln,Dunkirk,NY,14048,42.4612095,-79.3317925

7,Cody,Bailey,(716) 203-7201,114 Lincoln Ave,Dunkirk,NY,14048,42.4801269,-79.322232

8,Dario,Sims,(408) 927-1606,N Castle Dr,Armonk,NY,10504,41.11979,-73.714864

9,Deacon,Hood,(410) 244-6221,4856 44th St,Woodside,NY,11377,40.748372,-73.905445

10,Zackery,Levy,(410) 363-8874,61 Executive Blvd,Farmingdale,NY,11735,40.7197457,-73.430239

11,Rey,Hawkins,(412) 344-8687,4585 Shimerville Rd,Clarence,NY,14031,42.972075,-78.6592452

12,Phillip,Howard,(413) 269-4049,464 Main St #101,Port Washington,NY,11050,40.8273756,-73.7009971

13,Shirley,Tyler,(434) 985-8943,114 Glann Rd,Apalachin,NY,13732,42.0482515,-76.1229725

14,Aniyah,Jarvis,(440) 244-1808,87 N Middletown Rd,Pearl River,NY,10965,41.0629,-74.0159

15,Alayna,Woodard,(478) 335-3704,70 W Red Oak Ln,West Harrison,NY,10604,41.0162722,-73.7234926

16,Jermaine,Lambert,(508) 561-9836,24 Kellogg Rd,New Hartford,NY,13413,43.0555739,-75.2793197

17,Harper,Gibbs,(239) 466-0238,Po Box 33,Cottekill,NY,12419,41.853392,-74.106082

18,Osvaldo,Graham,(252) 246-0816,6878 Sand Hill Rd,East Syracuse,NY,13057,43.073215,-76.081448

19,Roberto,Wade,(270) 469-1211,3936 Holley Rd,Moravia,NY,13118,42.713044,-76.481227

20,Kate,Mcguire,(270) 788-3080,6451 State 64 Rte #3,Naples,NY,14512,42.707366,-77.380489

21,Dale,Andersen,(281) 480-5690,205 W Service Rd,Champlain,NY,12919,44.9645392,-73.4470831

22,Lorelai,Burch,(302) 644-2133,1 Brewster St,Glen Cove,NY,11542,40.865177,-73.633019

23,Amiyah,Flowers,(303) 223-0055,46600 Us Interstate 81 Rte,Alexandria

Bay,NY,13607,44.309626,-75.988365

24,McKinley,Clements,(303) 918-3230,200 Summit Lake Dr,Valhalla,NY,10595,41.101145,-73.778298  
25,Marc,Gibson,(607) 203-1233,25 Robinson St,Binghamton,NY,13901,42.107416,-75.901614  
26,Kali,Norman,(607) 203-1400,1 Ely Park Blvd #APT 15,Binghamton,NY,13905,42.125866,-75.925026  
27,Laci,Cain,(607) 203-1437,16 Zimmer Road,Kirkwood,NY,13795,42.066516,-75.792627  
28,Mohammad,Perez,(607) 203-1652,71 Endicott Ave #APT 12,Johnson City,NY,13790,42.111894,-75.952187  
29,Izabelle,Pham,(607) 204-0392,434 State 369 Rte,Port Crane,NY,13833,42.185838,-75.823074  
30,Kiley,Mays,(607) 204-0870,244 Ballyhack Rd #14,Port Crane,NY,13833,42.175612,-75.814917  
31,Peter,Trevino,(607) 205-1374,125 Melbourne St.,Vestal,NY,13850,42.080254,-76.051124  
32,Ani,Francis,(607) 208-4067,48 Caswell St,Afton,NY,13730,42.232065,-75.525674  
33,Jared,Sheppard,(716) 386-3002,4709 430th Rte,Bemus Point,NY,14712,42.162175,-79.39176  
34,Dulce,Atkinson,(914) 576-2266,501 Pelham Rd,New Rochelle,NY,10805,40.895449,-73.782602  
35,Jayla,Beasley,(716) 526-1054,5010 474th Rte,Ashville,NY,14710,42.096859,-79.375561  
36,Dane,Donovan,(718) 545-3732,5014 31st Ave,Woodside,NY,11377,40.756967,-73.909506  
37,Brendon,Clay,(585) 322-7780,133 Cummings Ave,Gainesville,NY,14066,42.664309,-78.085651  
38,Asia,Nunez,(718) 426-1472,2407 Gilmore ,East Elmhurst,NY,11369,40.766662,-73.869185  
39,Dawson,Odonnell,(718) 342-2179,5019 H Ave,Brooklyn,NY,11234,40.633245,-73.927591  
40,Kyle,Collins,(315) 733-7078,502 Rockhaven Rd,Utica,NY,13502,43.129184,-75.226726  
41,Eliza,Hardin,(315) 331-8072,502 Sladen Place,West Point,NY,10996,41.3993,-73.973003  
42,Kasen,Klein,(518) 298-4581,2407 Lake Shore Rd,Chazy,NY,12921,44.925561,-73.387373  
43,Reuben,Bradford,(518) 298-4581,33 Lake Flats Dr,Champlain,NY,12919,44.928092,-73.387884  
44,Henry,Grimes,(518) 523-3990,2407 Main St,Lake Placid,NY,12946,44.291487,-73.98474  
45,Kyan,Livingston,(518) 585-7364,241 Alexandria Ave,Ticonderoga,NY,12883,43.836553,-73.43155  
46,Kaitlyn,Short,(516) 678-3189,241 Chance Dr,Oceanside,NY,11572,40.638534,-73.63079  
47,Damaris,Jacobs,(914) 664-5331,241 Claremont Ave,Mount Vernon,NY,10552,40.919852,-73.827848  
48,Alivia,Schroeder,(315) 469-4473,241 Lafayette Rd,Syracuse,NY,13205,42.996446,-76.12957  
49,Bridget,Strong,(315) 298-4355,241 Maltby Rd,Pulaski,NY,13142,43.584966,-76.136317  
50,Francis,Lee,(585) 201-7021,166 Ross St,Batavia,NY,14020,43.0031502,-78.17487  
51,Makaila,Phelps,(585) 201-7422,58 S Main St,Batavia,NY,14020,42.99941,-78.1939285  
52,Jazlynn,Stephens,(585) 203-1087,1 Sinclair Dr,Pittsford,NY,14534,43.084157,-77.545452  
53,Ryann,Randolph,(585) 203-1519,331 Eaglehead Rd,East Rochester,NY,14445,43.10785,-77.475552  
54,Rosa,Baker,(585) 204-4011,42 Ossian St,Dansville,NY,14437,42.560761,-77.70088  
55,Marcel,Barry,(585) 204-4013,42 Jefferson St,Dansville,NY,14437,42.557735,-77.702983  
56,Dennis,Schmitt,(585) 204-4061,750 Dansville Mount Morris Rd,Dansville,NY,14437,42.584458,-77.741648  
57,Cassandra,Kim,(585) 204-4138,3 Perine Ave APT1,Dansville,NY,14437,42.562865,-77.69661  
58,Kolton,Jacobson,(585) 206-5047,4925 Upper Holly Rd,Holley,NY,14470,43.175957,-78.074465  
59,Nathanael,Donovan,(718) 393-3501,9604 57th Ave,Corona,NY,11373,40.736077,-73.864858  
60,Robert,Frazier,(718) 271-3067,300 56th Ave,Corona,NY,11373,40.735304,-73.873997  
61,Jessie,Mora,(315) 405-8991,9607 Forsyth Loop,Watertown,NY,13603,44.036466,-75.833437  
62,Martha,Rollins,(347) 242-2642,22 Main St,Corona,NY,11373,40.757727,-73.829331  
63,Emely,Townsend,(718) 699-0751,60 Sanford Ave,Corona,NY,11373,40.755466,-73.831029  
64,Kylie,Coolley,(347) 561-7149,9608 95th Ave,Ozone Park,NY,11416,40.687564,-73.845715  
65,Wendy,Cameron,(585) 571-4185,9608 Union St,Scottsville,NY,14546,43.013327,-77.7907839  
66,Kayley,Peterson,(718) 654-5027,961 E 230th St,Bronx,NY,10466,40.889275,-73.850555  
67,Camden,Ochoa,(718) 760-8699,59 Vark St,Yonkers,NY,10701,40.929322,-73.89957  
68,Priscilla,Castillo,(910) 326-7233,9359 Elm St,Chadwicks,NY,13319,43.024902,-75.26886  
69,Dana,Schultz,(913) 322-4580,99 Washington Ave,Hastings on Hudson,NY,10706,40.99265,-73.879748  
70,Blaze,Medina,(914) 207-0015,60 Elliott Ave,Yonkers,NY,10705,40.921498,-73.896682  
71,Finnegan,Tucker,(914) 207-0015,90 Hillside Drive,Yonkers,NY,10705,40.922514,-73.892911  
72,Pranav,Palmer,(914) 214-8376,5 Bruce Ave,Harrison,NY,10528,40.970916,-73.711493  
73,Kolten,Wong,(914) 218-8268,70 Barker St,Mount Kisco,NY,10549,41.211993,-73.723202  
74,Jasiah,Vazquez,(914) 231-5199,30 Broadway,Dobbs Ferry,NY,10522,41.004629,-73.879825  
75,Lamar,Pierce,(914) 232-0380,68 Ridge Rd,Katonah,NY,10536,41.256662,-73.707964  
76,Carla,Coffey,(914) 232-0469,197 Beaver Dam Rd,Katonah,NY,10536,41.247934,-73.664363

77, Brooklyn, Harmon, (716) 595-3227, 8084 Glasgow Rd, Cassadega, NY, 14718, 42.353861, -79.329558  
78, Raquel, Hodges, (585) 398-8125, 809 County Road, Victor, NY, 14564, 43.011745, -77.398806  
79, Jerimiah, Gardner, (585) 787-9127, 809 Houston Rd, Webster, NY, 14580, 43.224204, -77.491353  
80, Clarence, Hammond, (720) 746-1619, 809 Pierpont Ave, Piermont, NY, 10968, 41.0491181, -73.918622  
81, Rhys, Gill, (518) 427-7887, 81 Columbia St, Albany, NY, 12210, 42.652824, -73.752096  
82, Edith, Parrish, (845) 452-7621, 81 Glenwood Ave, Poughkeepsie, NY, 12603, 41.691058, -73.910829  
83, Kobe, McIntosh, (845) 371-1101, 81 Heitman Dr, Spring Valley, NY, 10977, 41.103227, -74.054396  
84, Ayden, Waters, (516) 796-2722, 81 Kingfisher Rd, Levittown, NY, 11756, 40.738939, -73.52826  
85, Francis, Rogers, (631) 427-7728, 81 Knollwood Ave, Huntington, NY, 11743, 40.864905, -73.426107  
86, Jaden, Landry, (716) 496-4038, 12839 39th Rte, Chaffee, NY, 14030, 43.527396, -73.462786  
87, Giancarlo, Campos, (518) 885-5717, 1284 Saratoga Rd, Ballston Spa, NY, 12020, 42.968594, -73.862847  
88, Eduardo, Contreras, (716) 285-8987, 1285 Saunders Sett Rd, Niagara Falls, NY, 14305, 43.122963, -79.029274  
89, Gabriela, Davidson, (716) 267-3195, 1286 Mee Rd, Falconer, NY, 14733, 42.147339, -79.137976  
90, Evangeline, Case, (518) 272-9435, 1287 2nd Ave, Watervliet, NY, 12189, 42.723132, -73.703818  
91, Tyrone, Ellison, (518) 843-4691, 1287 Midline Rd, Amsterdam, NY, 12010, 42.9730876, -74.1700608  
92, Bryce, Bass, (518) 943-9549, 1288 Leeds Athens Rd, Athens, NY, 12015, 42.259381, -73.876897  
93, Londyn, Butler, (518) 922-7095, 129 Argersinger Rd, Fultonville, NY, 12072, 42.910969, -74.441917  
94, Graham, Becker, (607) 655-1318, 129 Baker Rd, Windsor, NY, 13865, 42.107271, -75.66408  
95, Rolando, Fitzgerald, (315) 465-4166, 17164 County 90 Rte, Mannsville, NY, 13661, 43.713443, -76.06232  
96, Grant, Hoover, (518) 692-8363, 1718 County 113 Rte, Schaghticote, NY, 12154, 42.900648, -73.585036  
97, Mark, Goodwin, (631) 584-6761, 172 Cambon Ave, Saint James, NY, 11780, 40.871152, -73.146032  
98, Deacon, Cantu, (845) 221-7940, 172 Carpenter Rd, Hopewell Junction, NY, 12533, 41.57388, -73.77609  
99, Tristian, Walsh, (516) 997-4750, 172 E Cabot Ln, Westbury, NY, 11590, 40.7480397, -73.54819  
100, Abram, Alexander, (631) 588-3817, 172 Lorenzo Cir, Ronkonkoma, NY, 11779, 40.837123, -73.09367  
101, Lesly, Bush, (516) 489-3791, 172 Nassau Blvd, Garden City, NY, 11530, 40.71147, -73.660753  
102, Pamela, Espinoza, (716) 201-1520, 172 Niagara St, Lockport, NY, 14094, 43.169871, -78.70093  
103, Bryanna, Newton, (914) 328-4332, 172 Warren Ave, White Plains, NY, 10603, 41.047207, -73.79572  
104, Marcelo, Schmitt, (315) 393-4432, 319 Mansion Ave, Ogdensburg, NY, 13669, 44.690246, -75.49992  
105, Layton, Valenzuela, (631) 676-2113, 319 Singingwood Dr, Holbrook, NY, 11741, 40.801391, -73.058993  
106, Roderick, Rocha, (518) 671-6037, 319 Warren St, Hudson, NY, 12534, 42.252527, -73.790629  
107, Camryn, Terrell, (315) 635-1680, 3192 Olive Dr, Baldwinsville, NY, 13027, 43.136843, -76.260303  
108, Summer, Callahan, (585) 394-4195, 3192 Smith Road, Canandaigua, NY, 14424, 42.875457, -77.228039  
109, Pierre, Novak, (716) 665-2524, 3194 Falconer Kimball Stand Rd, Falconer, NY, 14733, 42.138439, -79.211091  
110, Kennedy, Fry, (315) 543-2301, 32 College Rd, Selden, NY, 11784, 40.861624, -73.04757  
111, Wyatt, Pruitt, (716) 681-4042, 277 Ransom Rd, Lancaster, NY, 14086, 42.87702, -78.591302  
112, Lilly, Jensen, (631) 841-0859, 2772 Schliegel Blvd, Amityville, NY, 11701, 40.708021, -73.413015  
113, Tristin, Hardin, (631) 920-0927, 278 Fulton Street, West Babylon, NY, 11704, 40.733578, -73.357321  
114, Tanya, Stafford, (716) 484-0771, 278 Sampson St, Jamestown, NY, 14701, 42.0797, -79.247805  
115, Paris, Cordova, (607) 589-4857, 278 Washburn Rd, Spencer, NY, 14883, 42.225046, -76.510257  
116, Alfonso, Morse, (718) 359-5582, 200 Colden St, Flushing, NY, 11355, 40.750403, -73.822752  
117, Maurice, Hooper, (315) 595-6694, 4435 Italy Hill Rd, Branchport, NY, 14418, 42.597957, -77.199267  
118, Iris, Wolf, (607) 539-7288, 444 Harford Rd, Brooktondale, NY, 14817, 42.392164, -76.30756  
];

### KMeans2D - funzione per grafici

**KMeans2D()** valuta le righe del grafico applicando il clustering K-means, e per ciascuna riga del grafico visualizza l'id cluster del cluster a cui è stato assegnato questo punto dati. Le colonne utilizzate dall'algoritmo di clustering sono determinate rispettivamente dai parametri `coordinate_1` e `coordinate_2`. Sono entrambe aggregazioni. Il numero di cluster creati è determinato dal parametro `num_clusters`. I dati possono essere normalizzati in via opzionale dal parametro `norm`.

**KMeans2D** restituisce un valore per punto dati. Il valore restituito è duale ed è un valore intero corrispondente al cluster a cui ciascun punto dati è stato assegnato.

**Sintassi:**

```
KMeans2D (num_clusters, coordinate_1, coordinate_2 [, norm])
```

**Tipo di dati restituiti:** duale

**Argomenti:**

| Argomenti    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Argomento    | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| num_clusters | Intero che specifica il numero di cluster.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| coordinate_1 | L'aggregazione che calcola la prima coordinata, in genere l'asse x del grafico a dispersione che può essere effettuato dal grafico. Il parametro aggiuntivo, coordinate_2, calcola la seconda coordinata.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| norm         | <p>Il metodo di normalizzazione opzionale applicato alle serie di dati prima del clustering K-means.</p> <p>Possibili valori:</p> <p>0 o 'nessuno' per l'assenza di normalizzazione</p> <p>1 o 'zscore' per la normalizzazione z-score</p> <p>2 o 'minmax' per la normalizzazione min-max</p> <p>Se non viene fornito alcun parametro o se il parametro fornito risulta errato, non viene applicata alcuna normalizzazione.</p> <p>Z-score normalizza i dati in base alla deviazione standard e media della funzionalità. Z-score non assicura che ciascuna funzionalità abbia la stessa scala, ma rappresenta un approccio migliore a min-max quando si ha a che fare con outlier.</p> <p>La normalizzazione min-max assicura che le funzionalità abbiano la stessa scala prelevando i valori minimo e massimo di ciascuna di esse e ricalcolando ciascun datapoint.</p> |

**Esempio: espressione del grafico**

In questo esempio, creiamo un grafico a dispersione usando la serie di dati *Iris*, quindi utilizziamo KMeans per colorare i dati per espressione.

Creiamo inoltre una variabile per l'argomento *num\_clusters*, quindi utilizziamo una casella di input variabile per modificare il numero di cluster.

La serie di dati *Iris* è disponibile pubblicamente in una serie di formati. I dati sono stati forniti come tabella inline da caricare usando l'editor caricamento dati in Qlik Sense. Notare che è stata aggiunta una colonna *Id* alla tabella dati per questo esempio.

Dopo il caricamento dei dati in Qlik Sense, è possibile compiere le seguenti operazioni:

1. Trascinare un **Grafico a dispersione** in un nuovo foglio. Denominare il grafico *Petal (colore per espressione)*.
2. Creare una variabile per specificare il numero di cluster. Per la variabile **Nome**, inserire *KmeansPetalClusters*. Per la variabile **Definizione**, inserire *=2*.
3. Configurare **Dati** per il grafico:
  - i. Sotto **Dimensioni**, scegliere *Id* per il campo per **Bolla**. Inserire *Id* cluster per l'etichetta.
  - ii. Sotto **Misure**, scegliere *Sum([petal.length])* per l'espressione per **asse X**.
  - iii. Sotto **Misure**, scegliere *Sum([petal.width])* per l'espressione per **asse Y**.

*Impostazioni dati per il grafico Petal (colore per espressione)*

**Data**

**Dimensions**

Bubble

Id

Alternative dimensions

Add alternative

**Measures**

X-axis

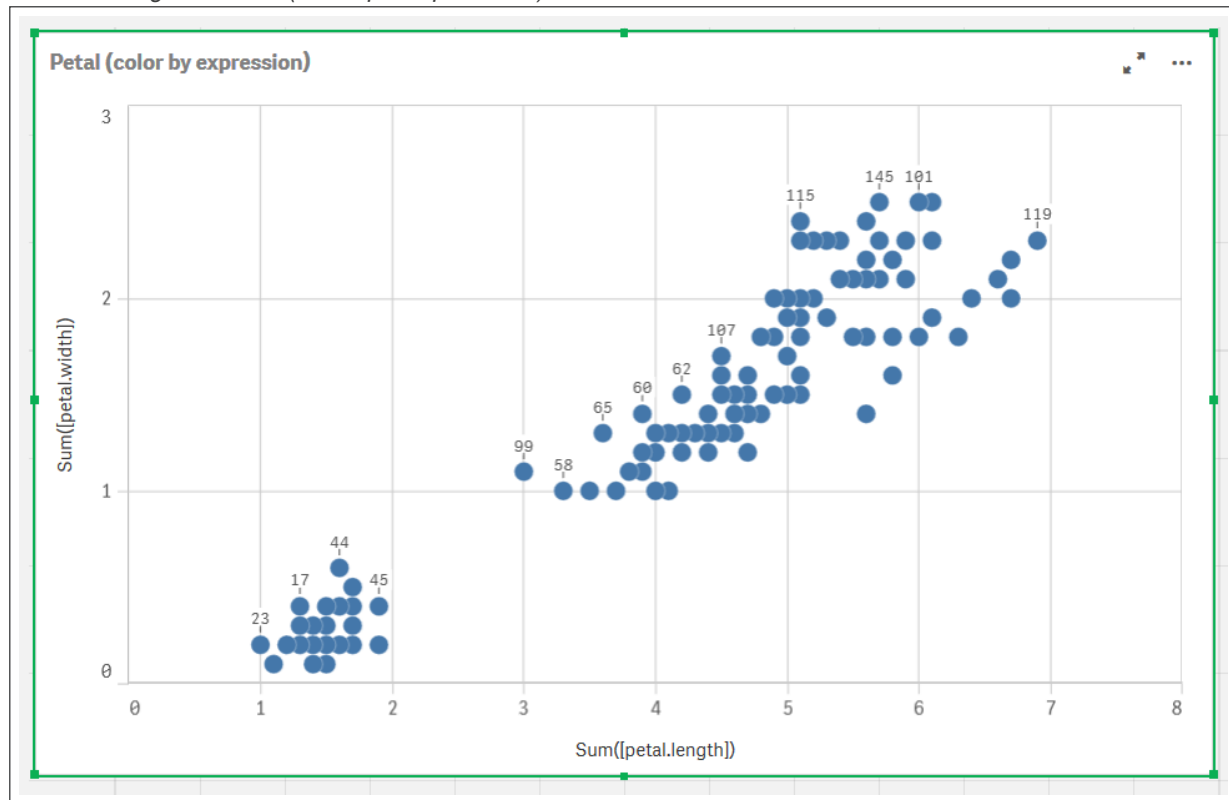
Sum [petal.length]

Y-axis

Sum [petal.width]

I punti dati vengono riportati sul grafico.

Punti dati sul grafico Petal (colore per espressione)



4. Configurare **Aspetto** per il grafico:

- i. Sotto **Colori e legenda**, scegliere **Personalizzato** per **Colori**.
- ii. Scegliere di colorare il grafico **Per espressione**.
- iii. Inserire quanto segue per **Expression**: `kmeans2d($(KmeansPetalClusters), Sum([petal.length]), Sum([petal.width]))`  
 Notare che `KmeansPetalClusters` è la variabile che impostiamo a 2.  
 In alternativa, inserire quanto segue: `kmeans2d(2, Sum([petal.length]), Sum([petal.width]))`
- iv. Deselezionare la casella di controllo per **L'espressione è un codice cromatico**.

v. Inserire quanto segue per **Etichetta**: *ID cluster*

*Impostazioni aspetto per il grafico Petal (colore per espressione)*

Appearance

▼ Colors and legend

Colors ☐

Custom

By expression ▼

Expression

kmeans2d(\$(KmeansPetalC) 

☐ The expression is a color code

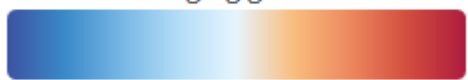
Label

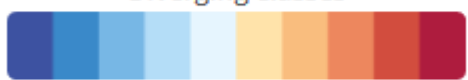
Cluster Id

Color scheme

Sequential gradient 

Sequential classes 

Diverging gradient 

Diverging classes 

☐ Reverse colors

Range ☒

Auto

Show legend ☒

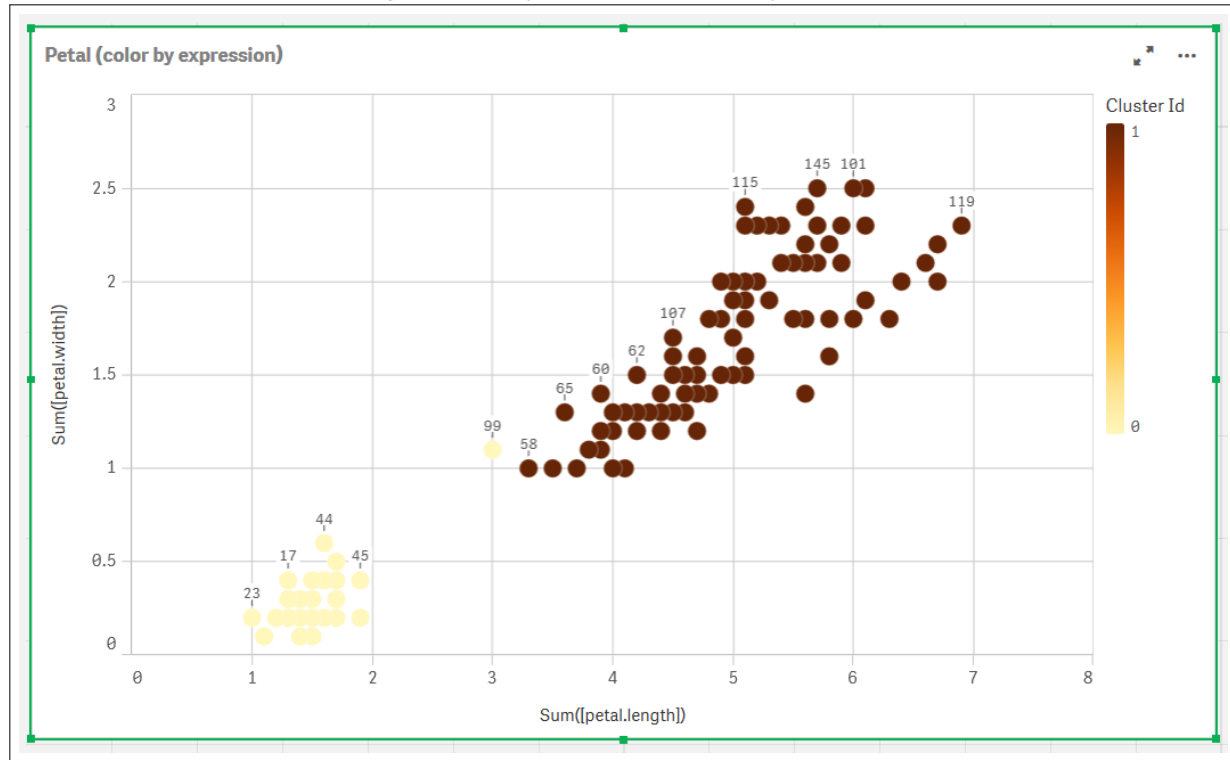
Auto

Legend position

After 

☒ Show legend title

I due cluster sul grafico sono colorati in base all'espressione KMeans.  
*Cluster colorati per espressione sul grafico Petal (colore per espressione)*



5. Aggiungi una casella **Input variabile** per il numero di cluster.
  - i. Sotto **Oggetti personalizzati** nel pannello **Risorse**, scegli **Qlik Dashboard bundle**. Se non avessimo avuto accesso al dashboard bundle, potremmo comunque modificare il numero di cluster utilizzando la variabile che abbiamo creato, o direttamente come numero intero nell'espressione.
  - ii. Trascina una casella **Input variabile** sul foglio.
  - iii. Sotto **Aspetto**, fai clic su **Generale**.
  - iv. Immettere quanto segue per **Titolo**: *Cluster*
  - v. Fare clic su **Variabile**.
  - vi. Scegliere la seguente variabile per **Nome**: *KmeansPetalClusters*.
  - vii. Scegliere **Slider** per **Mostra come**.

viii. Scegliere **Valori** e configurare le impostazioni come richiesto,

*Aspetto per la casella di input della variabile Cluster*

▼ General

Show titles

On

Title

Clusters

*fx*

Subtitle

*fx*

Footnote

*fx*

☐ Disable hover menu

▼ Variable

Name

KmeansPetalClusters

▼

Show as

Slider

▼

☐ Update on drag

▼ Values

Min

2

*fx*

Max

10

*fx*

Step

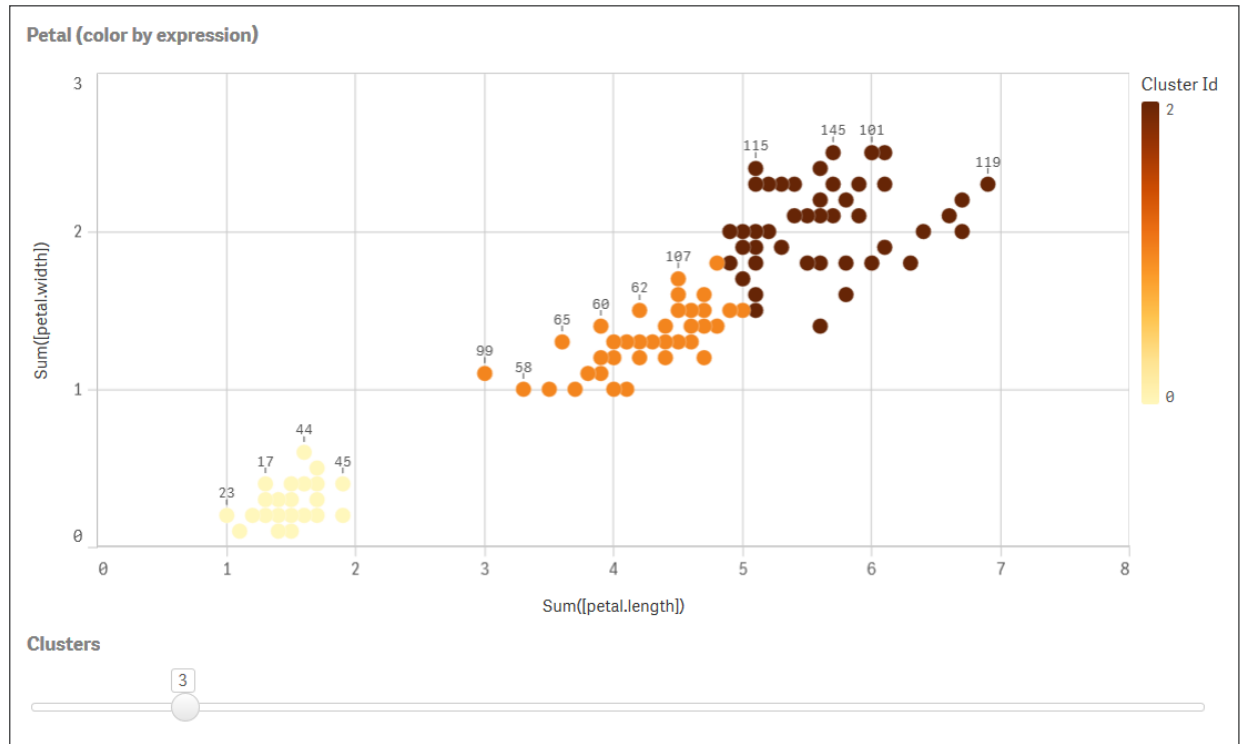
1

*fx*

☒ Slider label

Quando abbiamo finito di modificare, possiamo cambiare il numero di cluster usando il cursore nella casella di input della variabile *Cluster*.

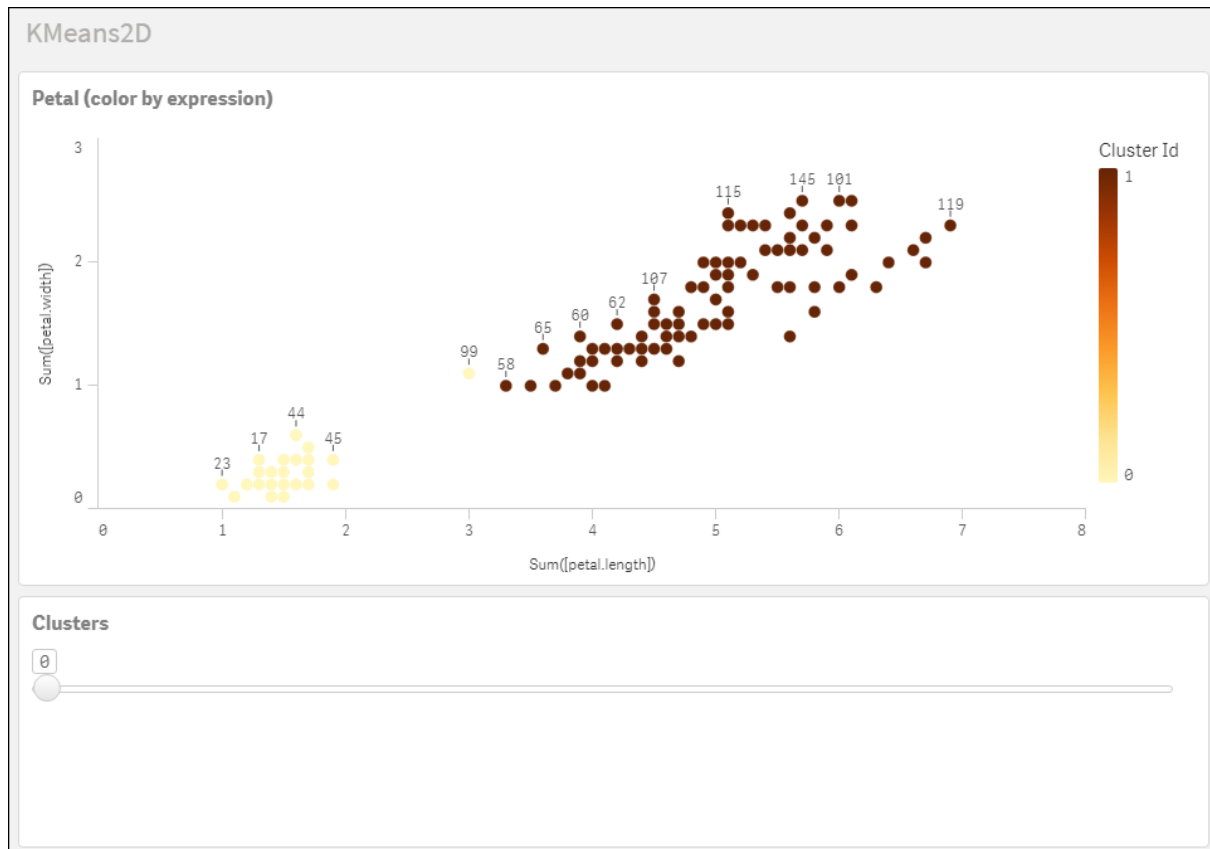
*Cluster colorati per espressione sul grafico Petal (colore per espressione)*



### Clustering automatico

Le funzioni **KMeans** supportano il clustering automatico mediante un metodo chiamato differenza di profondità (DeD, Depth Difference). Quando un utente imposta lo 0 per il numero di cluster, viene determinato un numero ottimale di cluster per tale set di dati. Notare che mentre un valore intero per il numero di cluster ( $k$ ) non viene restituito esplicitamente, viene calcolato all'interno dell'algoritmo KMeans. Ad esempio, se viene specificato 0 nella funzione del valore di *KmeansPetalClusters* o se viene impostato mediante una casella di input variabile, le assegnazioni cluster vengono calcolate automaticamente per il set di dati in base a un numero ottimale di cluster.

Il metodo di differenza di profondità KMeans determina il numero ottimale di cluster quando (k) viene impostato a 0.



### Iris set di dati: Caricamento inline per l'editor caricamento dati in Qlik Sense

IrisData:

Load \* Inline [

sepal.length, sepal.width, petal.length, petal.width, variety, id

5.1, 3.5, 1.4, 0.2, Setosa, 1

4.9, 3, 1.4, 0.2, Setosa, 2

4.7, 3.2, 1.3, 0.2, Setosa, 3

4.6, 3.1, 1.5, 0.2, Setosa, 4

5, 3.6, 1.4, 0.2, Setosa, 5

5.4, 3.9, 1.7, 0.4, Setosa, 6

4.6, 3.4, 1.4, 0.3, Setosa, 7

5, 3.4, 1.5, 0.2, Setosa, 8

4.4, 2.9, 1.4, 0.2, Setosa, 9

4.9, 3.1, 1.5, 0.1, Setosa, 10

5.4, 3.7, 1.5, 0.2, Setosa, 11

4.8, 3.4, 1.6, 0.2, Setosa, 12

4.8, 3, 1.4, 0.1, Setosa, 13

4.3, 3, 1.1, 0.1, Setosa, 14

5.8, 4, 1.2, 0.2, Setosa, 15

5.7, 4.4, 1.5, 0.4, Setosa, 16

5.4, 3.9, 1.3, 0.4, Setosa, 17

5.1, 3.5, 1.4, 0.3, Setosa, 18

5.7, 3.8, 1.7, 0.3, Setosa, 19

5.1, 3.8, 1.5, 0.3, Setosa, 20

5.4, 3.4, 1.7, 0.2, Setosa, 21

5.1, 3.7, 1.5, 0.4, Setosa, 22  
4.6, 3.6, 1, 0.2, Setosa, 23  
5.1, 3.3, 1.7, 0.5, Setosa, 24  
4.8, 3.4, 1.9, 0.2, Setosa, 25  
5, 3, 1.6, 0.2, Setosa, 26  
5, 3.4, 1.6, 0.4, Setosa, 27  
5.2, 3.5, 1.5, 0.2, Setosa, 28  
5.2, 3.4, 1.4, 0.2, Setosa, 29  
4.7, 3.2, 1.6, 0.2, Setosa, 30  
4.8, 3.1, 1.6, 0.2, Setosa, 31  
5.4, 3.4, 1.5, 0.4, Setosa, 32  
5.2, 4.1, 1.5, 0.1, Setosa, 33  
5.5, 4.2, 1.4, 0.2, Setosa, 34  
4.9, 3.1, 1.5, 0.1, Setosa, 35  
5, 3.2, 1.2, 0.2, Setosa, 36  
5.5, 3.5, 1.3, 0.2, Setosa, 37  
4.9, 3.1, 1.5, 0.1, Setosa, 38  
4.4, 3, 1.3, 0.2, Setosa, 39  
5.1, 3.4, 1.5, 0.2, Setosa, 40  
5, 3.5, 1.3, 0.3, Setosa, 41  
4.5, 2.3, 1.3, 0.3, Setosa, 42  
4.4, 3.2, 1.3, 0.2, Setosa, 43  
5, 3.5, 1.6, 0.6, Setosa, 44  
5.1, 3.8, 1.9, 0.4, Setosa, 45  
4.8, 3, 1.4, 0.3, Setosa, 46  
5.1, 3.8, 1.6, 0.2, Setosa, 47  
4.6, 3.2, 1.4, 0.2, Setosa, 48  
5.3, 3.7, 1.5, 0.2, Setosa, 49  
5, 3.3, 1.4, 0.2, Setosa, 50  
7, 3.2, 4.7, 1.4, versicolor, 51  
6.4, 3.2, 4.5, 1.5, versicolor, 52  
6.9, 3.1, 4.9, 1.5, versicolor, 53  
5.5, 2.3, 4, 1.3, versicolor, 54  
6.5, 2.8, 4.6, 1.5, versicolor, 55  
5.7, 2.8, 4.5, 1.3, versicolor, 56  
6.3, 3.3, 4.7, 1.6, versicolor, 57  
4.9, 2.4, 3.3, 1, versicolor, 58  
6.6, 2.9, 4.6, 1.3, versicolor, 59  
5.2, 2.7, 3.9, 1.4, versicolor, 60  
5, 2, 3.5, 1, versicolor, 61  
5.9, 3, 4.2, 1.5, versicolor, 62  
6, 2.2, 4, 1, versicolor, 63  
6.1, 2.9, 4.7, 1.4, versicolor, 64  
5.6, 2.9, 3.6, 1.3, versicolor, 65  
6.7, 3.1, 4.4, 1.4, versicolor, 66  
5.6, 3, 4.5, 1.5, versicolor, 67  
5.8, 2.7, 4.1, 1, versicolor, 68  
6.2, 2.2, 4.5, 1.5, versicolor, 69  
5.6, 2.5, 3.9, 1.1, versicolor, 70  
5.9, 3.2, 4.8, 1.8, versicolor, 71  
6.1, 2.8, 4, 1.3, versicolor, 72  
6.3, 2.5, 4.9, 1.5, versicolor, 73  
6.1, 2.8, 4.7, 1.2, versicolor, 74  
6.4, 2.9, 4.3, 1.3, versicolor, 75  
6.6, 3, 4.4, 1.4, versicolor, 76

6.8, 2.8, 4.8, 1.4, versicolor, 77  
6.7, 3, 5, 1.7, versicolor, 78  
6, 2.9, 4.5, 1.5, versicolor, 79  
5.7, 2.6, 3.5, 1, versicolor, 80  
5.5, 2.4, 3.8, 1.1, versicolor, 81  
5.5, 2.4, 3.7, 1, versicolor, 82  
5.8, 2.7, 3.9, 1.2, versicolor, 83  
6, 2.7, 5.1, 1.6, versicolor, 84  
5.4, 3, 4.5, 1.5, versicolor, 85  
6, 3.4, 4.5, 1.6, versicolor, 86  
6.7, 3.1, 4.7, 1.5, versicolor, 87  
6.3, 2.3, 4.4, 1.3, versicolor, 88  
5.6, 3, 4.1, 1.3, versicolor, 89  
5.5, 2.5, 4, 1.3, versicolor, 90  
5.5, 2.6, 4.4, 1.2, versicolor, 91  
6.1, 3, 4.6, 1.4, versicolor, 92  
5.8, 2.6, 4, 1.2, versicolor, 93  
5, 2.3, 3.3, 1, versicolor, 94  
5.6, 2.7, 4.2, 1.3, versicolor, 95  
5.7, 3, 4.2, 1.2, versicolor, 96  
5.7, 2.9, 4.2, 1.3, versicolor, 97  
6.2, 2.9, 4.3, 1.3, versicolor, 98  
5.1, 2.5, 3, 1.1, versicolor, 99  
5.7, 2.8, 4.1, 1.3, versicolor, 100  
6.3, 3.3, 6, 2.5, virginica, 101  
5.8, 2.7, 5.1, 1.9, virginica, 102  
7.1, 3, 5.9, 2.1, virginica, 103  
6.3, 2.9, 5.6, 1.8, virginica, 104  
6.5, 3, 5.8, 2.2, virginica, 105  
7.6, 3, 6.6, 2.1, virginica, 106  
4.9, 2.5, 4.5, 1.7, virginica, 107  
7.3, 2.9, 6.3, 1.8, virginica, 108  
6.7, 2.5, 5.8, 1.8, virginica, 109  
7.2, 3.6, 6.1, 2.5, virginica, 110  
6.5, 3.2, 5.1, 2, virginica, 111  
6.4, 2.7, 5.3, 1.9, virginica, 112  
6.8, 3, 5.5, 2.1, virginica, 113  
5.7, 2.5, 5, 2, virginica, 114  
5.8, 2.8, 5.1, 2.4, virginica, 115  
6.4, 3.2, 5.3, 2.3, virginica, 116  
6.5, 3, 5.5, 1.8, virginica, 117  
7.7, 3.8, 6.7, 2.2, virginica, 118  
7.7, 2.6, 6.9, 2.3, virginica, 119  
6, 2.2, 5, 1.5, virginica, 120  
6.9, 3.2, 5.7, 2.3, virginica, 121  
5.6, 2.8, 4.9, 2, virginica, 122  
7.7, 2.8, 6.7, 2, virginica, 123  
6.3, 2.7, 4.9, 1.8, virginica, 124  
6.7, 3.3, 5.7, 2.1, virginica, 125  
7.2, 3.2, 6, 1.8, virginica, 126  
6.2, 2.8, 4.8, 1.8, virginica, 127  
6.1, 3, 4.9, 1.8, virginica, 128  
6.4, 2.8, 5.6, 2.1, virginica, 129  
7.2, 3, 5.8, 1.6, virginica, 130  
7.4, 2.8, 6.1, 1.9, virginica, 131

```

7.9, 3.8, 6.4, 2, virginica, 132
6.4, 2.8, 5.6, 2.2, virginica, 133
6.3, 2.8, 5.1, 1.5, virginica, 134
6.1, 2.6, 5.6, 1.4, virginica, 135
7.7, 3, 6.1, 2.3, virginica, 136
6.3, 3.4, 5.6, 2.4, virginica, 137
6.4, 3.1, 5.5, 1.8, virginica, 138
6, 3, 4.8, 1.8, virginica, 139
6.9, 3.1, 5.4, 2.1, virginica, 140
6.7, 3.1, 5.6, 2.4, virginica, 141
6.9, 3.1, 5.1, 2.3, virginica, 142
5.8, 2.7, 5.1, 1.9, virginica, 143
6.8, 3.2, 5.9, 2.3, virginica, 144
6.7, 3.3, 5.7, 2.5, virginica, 145
6.7, 3, 5.2, 2.3, virginica, 146
6.3, 2.5, 5, 1.9, virginica, 147
6.5, 3, 5.2, 2, virginica, 148
6.2, 3.4, 5.4, 2.3, virginica, 149
5.9, 3, 5.1, 1.8, virginica, 150
];

```

## KMeansND - funzione per grafici

**KMeansND()** valuta le righe del grafico applicando il clustering K-means, e per ciascuna riga del grafico visualizza l'id cluster del cluster a cui è stato assegnato questo punto dati. Le colonne utilizzate dall'algorithm di clustering sono determinate dai parametri `coordinate_1`, `coordinate_2`, ecc. fino a `n` colonne. Sono tutte aggregazioni. Il numero di cluster creati è determinato dal parametro `num_clusters`.

**KMeansND** restituisce un valore per punto dati. Il valore restituito è duale ed è un valore intero corrispondente al cluster a cui ciascun punto dati è stato assegnato.

### Sintassi:

```
KMeansND(num_clusters, num_iter, coordinate_1, coordinate_2 [,coordinate_3 [,
...]])
```

**Tipo di dati restituiti:** duale

### Argomenti:

#### Argomenti

| Argomento                 | Descrizione                                                                                                                                                                                                          |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>num_clusters</code> | Intero che specifica il numero di cluster.                                                                                                                                                                           |
| <code>num_iter</code>     | Il numero di ripetizioni di clustering con centri cluster reinizializzati.                                                                                                                                           |
| <code>coordinate_1</code> | L'aggregazione che calcola la prima coordinata, in genere l'asse x (di un grafico a dispersione che può essere effettuato dal grafico). I parametri aggiuntivi calcolano le seconde, terze e quarte coordinate, ecc. |

### Esempio: espressione del grafico

In questo esempio, creiamo un grafico a dispersione usando la serie di dati *Iris*, quindi utilizziamo KMeans per colorare i dati per espressione.

Creiamo inoltre una variabile per l'argomento *num\_clusters*, quindi utilizziamo una casella di input variabile per modificare il numero di cluster.

Inoltre, creiamo una variabile per l'argomento *num\_iter*, quindi utilizziamo una seconda casella di input variabile per modificare il numero di ripetizioni.

La serie di dati *Iris* è disponibile pubblicamente in una serie di formati. I dati sono stati forniti come tabella inline da caricare usando l'editor caricamento dati in Qlik Sense. Notare che è stata aggiunta una colonna *Id* alla tabella dati per questo esempio.

Dopo il caricamento dei dati in Qlik Sense, è possibile compiere le seguenti operazioni:

1. Trascinare un **Grafico a dispersione** in un nuovo foglio. Denominare il grafico *Petal (colore per espressione)*.
2. Creare una variabile per specificare il numero di cluster. Per la variabile **Nome**, inserire *KmeansPetalClusters*. Per la variabile **Definizione**, inserire *=2*.
3. Creare una variabile per specificare il numero di ripetizioni. Per la variabile **Nome**, inserire *KmeansNumberIterations*. Per la variabile **Definizione**, inserire *=1*.
4. Configurare **Dati** per il grafico:
  - i. Sotto **Dimensioni**, scegliere *id* per il campo per **Bolla**. Inserire *Id cluster* per l'etichetta.
  - ii. Sotto **Misure**, scegliere *Sum([petal.length])* per l'espressione per **asse X**.
  - iii. Sotto **Misure**, scegliere *Sum([petal.width])* per l'espressione per **asse Y**.

*Impostazioni dati per il grafico Petal (colore per espressione)*

Data

Dimensions

Bubble

Id

>

Alternative dimensions

Add alternative

Measures

X-axis

Sum

[petal.length]

>

Y-axis

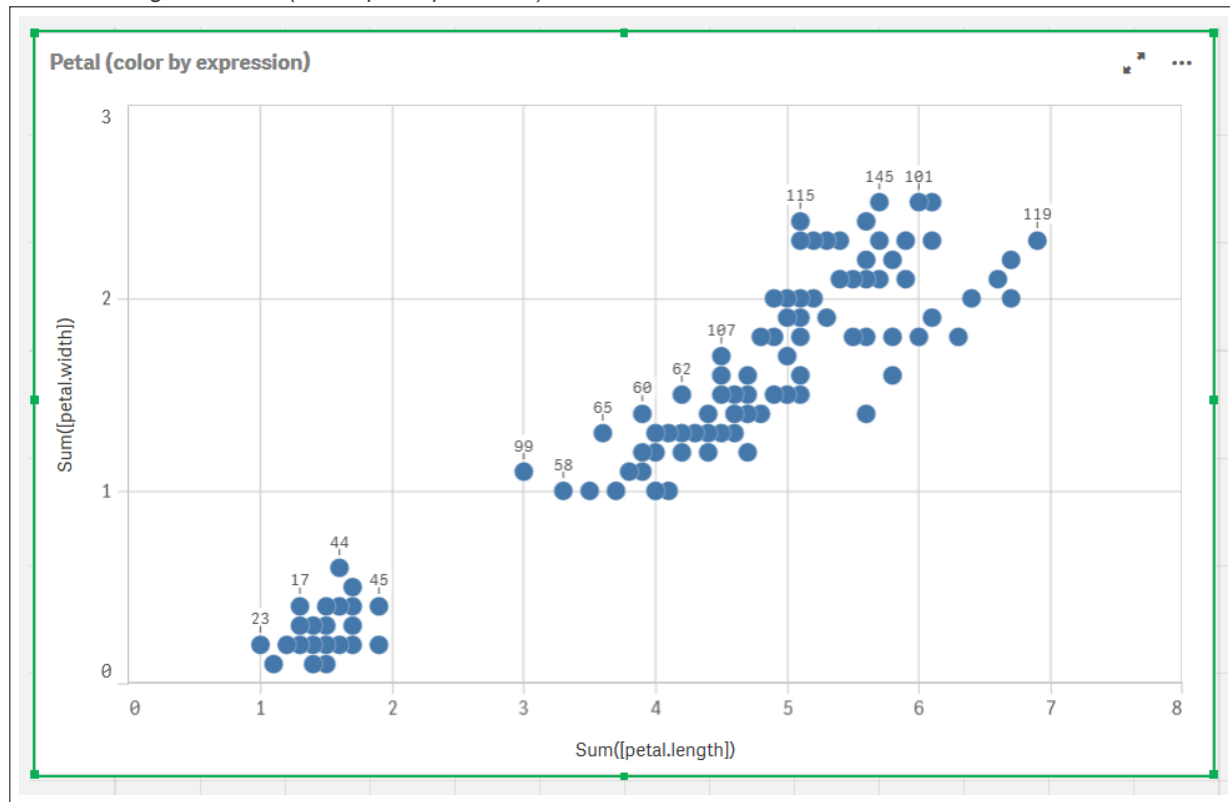
Sum

[petal.width]

>

I punti dati vengono riportati sul grafico.

Punti dati sul grafico Petal (colore per espressione)



5. Configurare **Aspetto** per il grafico:

- i. Sotto **Colori e legenda**, scegliere **Personalizzato** per **Colori**.
- ii. Scegliere di colorare il grafico **Per espressione**.
- iii. Inserire quanto segue per **Expression**: `kmeansnd`  
`$(KmeansPetalClusters),$(KmeansNumberIterations), Sum([petal.length]), Sum`  
`([petal.width]),Sum([sepal.length]), Sum([sepal.width]))`  
 Notare che `KmeansPetalClusters` è la variabile che impostiamo a 2.  
`KmeansNumberIterations` è la variabile che impostiamo a 1.  
 In alternativa, inserire quanto segue: `kmeansnd(2, 2, Sum([petal.length]), Sum`  
`([petal.width]),Sum([sepal.length]), Sum([sepal.width]))`
- iv. Deselezionare la casella di controllo per **L'espressione è un codice cromatico**.

v. Inserire quanto segue per **Etichetta**: *Id cluster*

*Impostazioni aspetto per il grafico Petal (colore per espressione)*

Appearance

▼ Colors and legend

Colors ☐

Custom

By expression ▼

Expression

kmeansnd(\$(KmeansPetal( *fx*

☐ The expression is a color code

Label

Cluster Id

Color scheme

Sequential gradient

Sequential classes

Diverging gradient

Diverging classes

☐ Reverse colors

Range ☒

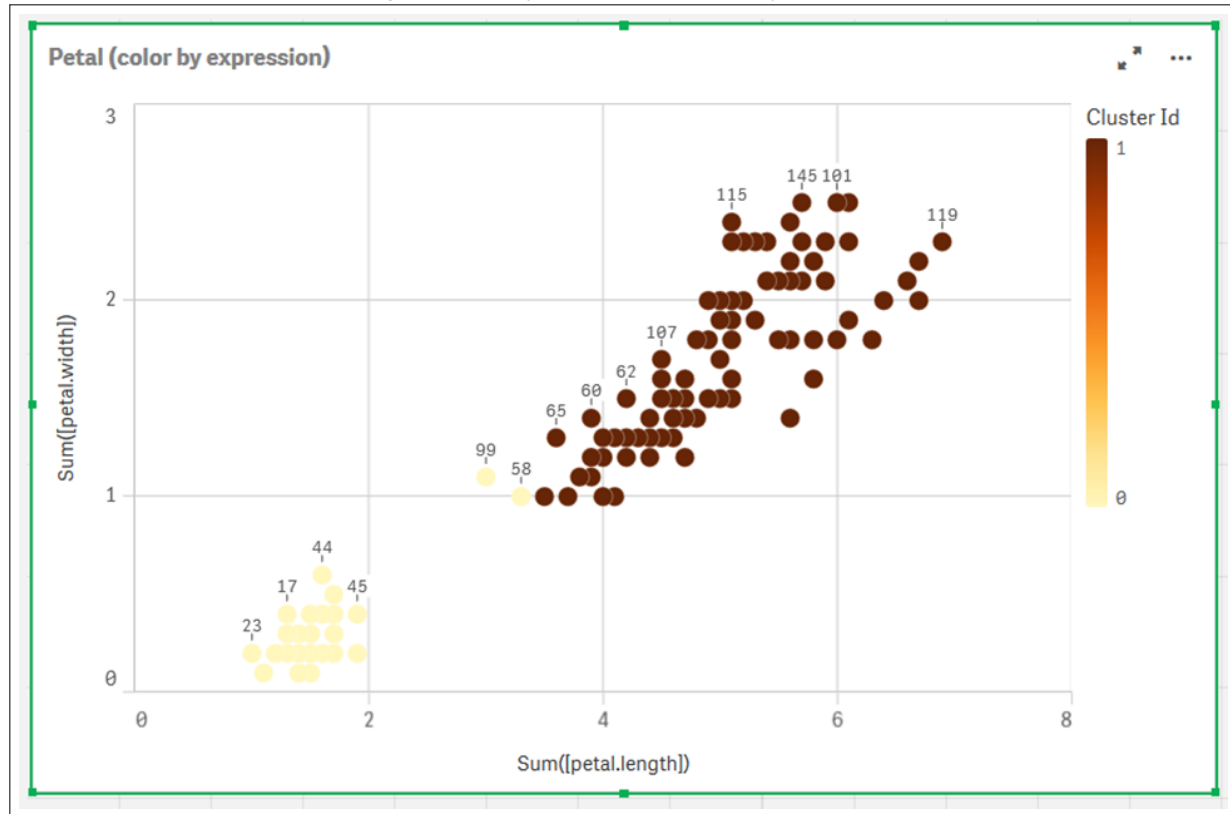
Auto

Show legend ☒

Auto

Legend position

I due cluster sul grafico sono colorati in base all'espressione KMeans.  
*Cluster colorati per espressione sul grafico Petal (colore per espressione)*



6. Aggiungi una casella **Input variabile** per il numero di cluster.
  - i. Sotto **Oggetti personalizzati** nel pannello **Risorse**, scegli **Qlik Dashboard bundle**. Se non avessimo avuto accesso al dashboard bundle, potremmo comunque modificare il numero di cluster utilizzando la variabile che abbiamo creato, o direttamente come numero intero nell'espressione.
  - ii. Trascina una casella **Input variabile** sul foglio.
  - iii. Sotto **Aspetto**, fai clic su **Generale**.
  - iv. Immettere quanto segue per **Titolo**: *Cluster*
  - v. Fare clic su **Variabile**.
  - vi. Scegliere la seguente variabile per **Nome**: *KmeansPetalClusters*.
  - vii. Scegliere **Slider** per **Mostra come**.

viii. Scegliere **Valori** e configurare le impostazioni come richiesto,

*Aspetto per la casella di input della variabile Cluster*

▼ General

Show titles

On

Title

Clusters

*fx*

Subtitle

*fx*

Footnote

*fx*

☐ Disable hover menu

▼ Variable

Name

KmeansPetalClusters

▼

Show as

Slider

▼

☐ Update on drag

▼ Values

Min

2

*fx*

Max

10

*fx*

Step

1

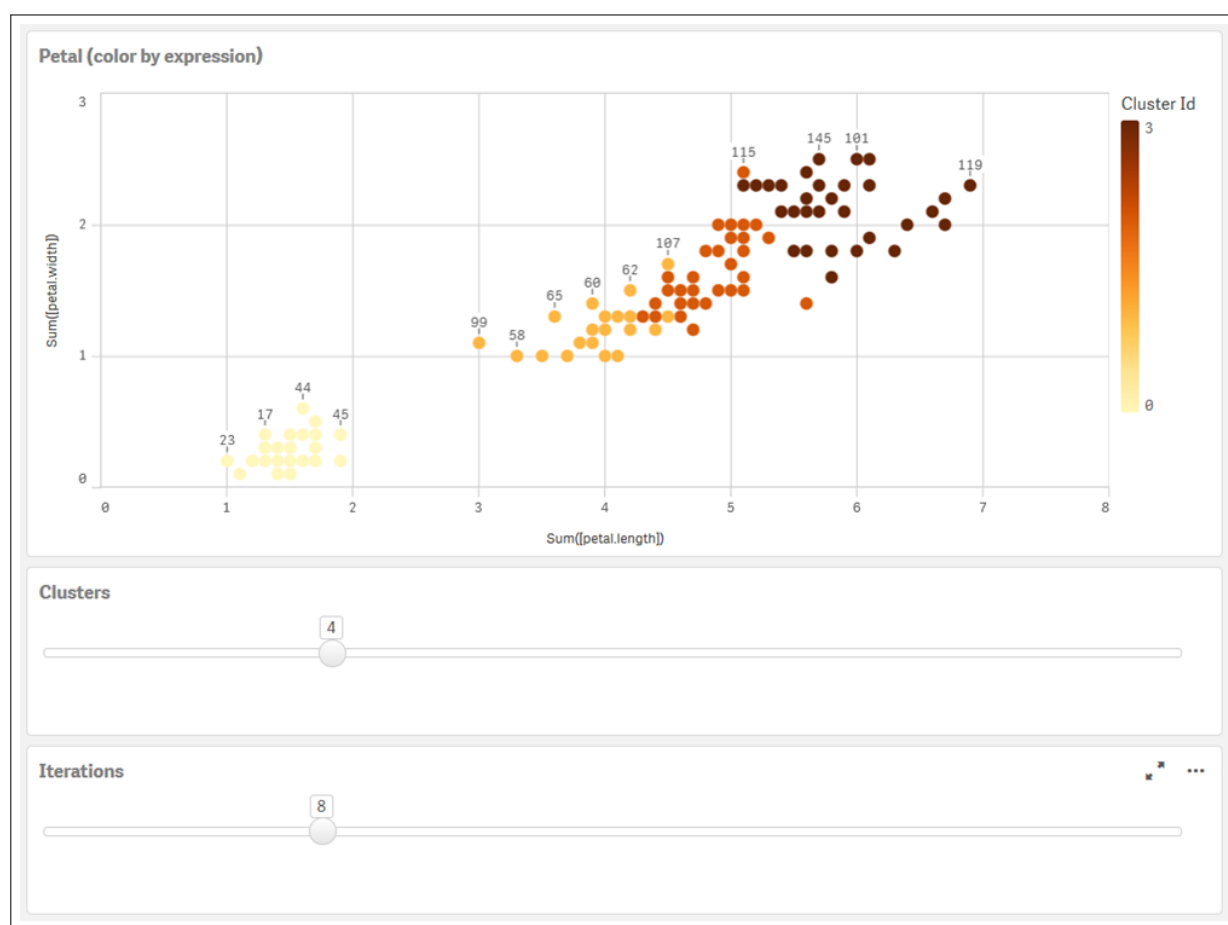
*fx*

☒ Slider label

7. Aggiungere una casella **Input variabile** per il numero di ripetizioni.
  - i. Trascinare una casella **Input variabile** sul foglio.
  - ii. Sotto **Aspetto**, scegliere **Generale**.
  - iii. Inserire quanto segue per **Titolo**: *Ripetizioni*
  - iv. Sotto **Aspetto**, scegliere **Variabile**.
  - v. Scegliere la seguente variabile sotto **Nome**: *KmeansNumberIterations*.
  - vi. Configurare le impostazioni aggiuntive come richiesto,

Ora è possibile modificare il numero di cluster e di ripetizioni usando i cursori nelle caselle di input variabile.

*Cluster colorati per espressione sul grafico Petal (colore per espressione)*

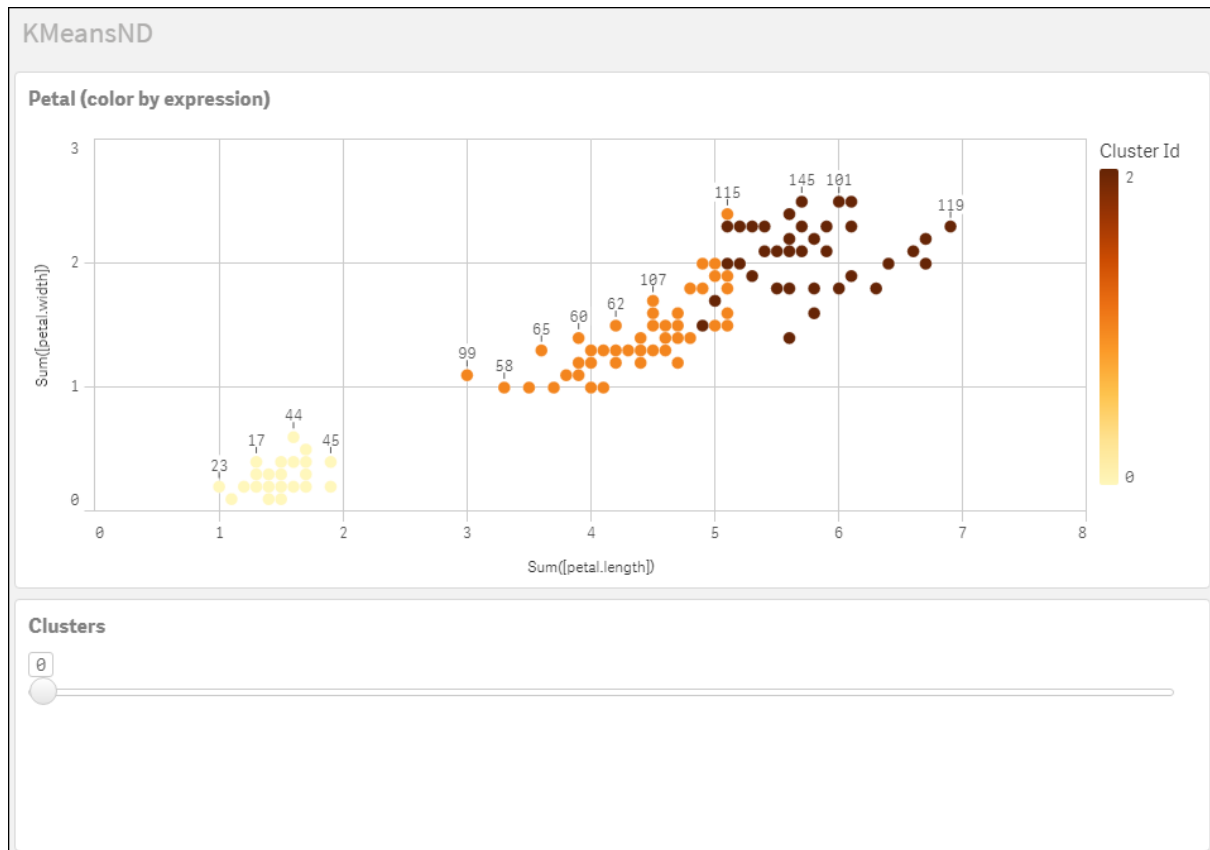


### Clustering automatico

Le funzioni **KMeans** supportano il clustering automatico mediante un metodo chiamato differenza di profondità (DeD, Depth Difference). Quando un utente imposta lo 0 per il numero di cluster, viene determinato un numero ottimale di cluster per tale set di dati. Notare che mentre un valore intero per il numero di cluster ( $k$ ) non viene restituito esplicitamente, viene calcolato all'interno dell'algoritmo KMeans. Ad esempio, se viene specificato 0 nella funzione del valore di *KmeansPetalClusters* o se viene impostato

mediante una casella di input variabile, le assegnazioni cluster vengono calcolate automaticamente per il set di dati in base a un numero ottimale di cluster. In base al set di dati Iris, se viene selezionato 0 per il numero di cluster, l'algoritmo determinerà (clustering automatico) un numero ottimale di cluster (3) per questo set di dati.

*Il metodo di differenza di profondità KMeans determina il numero ottimale di cluster quando (k) viene impostato a 0.*



### Serie di dati Iris: Caricamento inline per l'editor caricamento dati in Qlik Sense

IrisData:

Load \* Inline [

sepal.length, sepal.width, petal.length, petal.width, variety, id

5.1, 3.5, 1.4, 0.2, Setosa, 1

4.9, 3, 1.4, 0.2, Setosa, 2

4.7, 3.2, 1.3, 0.2, Setosa, 3

4.6, 3.1, 1.5, 0.2, Setosa, 4

5, 3.6, 1.4, 0.2, Setosa, 5

5.4, 3.9, 1.7, 0.4, Setosa, 6

4.6, 3.4, 1.4, 0.3, Setosa, 7

5, 3.4, 1.5, 0.2, Setosa, 8

4.4, 2.9, 1.4, 0.2, Setosa, 9

4.9, 3.1, 1.5, 0.1, Setosa, 10

5.4, 3.7, 1.5, 0.2, Setosa, 11

4.8, 3.4, 1.6, 0.2, Setosa, 12

4.8, 3, 1.4, 0.1, Setosa, 13

4.3, 3, 1.1, 0.1, Setosa, 14

5.8, 4, 1.2, 0.2, Setosa, 15

5.7, 4.4, 1.5, 0.4, Setosa, 16  
5.4, 3.9, 1.3, 0.4, Setosa, 17  
5.1, 3.5, 1.4, 0.3, Setosa, 18  
5.7, 3.8, 1.7, 0.3, Setosa, 19  
5.1, 3.8, 1.5, 0.3, Setosa, 20  
5.4, 3.4, 1.7, 0.2, Setosa, 21  
5.1, 3.7, 1.5, 0.4, Setosa, 22  
4.6, 3.6, 1, 0.2, Setosa, 23  
5.1, 3.3, 1.7, 0.5, Setosa, 24  
4.8, 3.4, 1.9, 0.2, Setosa, 25  
5, 3, 1.6, 0.2, Setosa, 26  
5, 3.4, 1.6, 0.4, Setosa, 27  
5.2, 3.5, 1.5, 0.2, Setosa, 28  
5.2, 3.4, 1.4, 0.2, Setosa, 29  
4.7, 3.2, 1.6, 0.2, Setosa, 30  
4.8, 3.1, 1.6, 0.2, Setosa, 31  
5.4, 3.4, 1.5, 0.4, Setosa, 32  
5.2, 4.1, 1.5, 0.1, Setosa, 33  
5.5, 4.2, 1.4, 0.2, Setosa, 34  
4.9, 3.1, 1.5, 0.1, Setosa, 35  
5, 3.2, 1.2, 0.2, Setosa, 36  
5.5, 3.5, 1.3, 0.2, Setosa, 37  
4.9, 3.1, 1.5, 0.1, Setosa, 38  
4.4, 3, 1.3, 0.2, Setosa, 39  
5.1, 3.4, 1.5, 0.2, Setosa, 40  
5, 3.5, 1.3, 0.3, Setosa, 41  
4.5, 2.3, 1.3, 0.3, Setosa, 42  
4.4, 3.2, 1.3, 0.2, Setosa, 43  
5, 3.5, 1.6, 0.6, Setosa, 44  
5.1, 3.8, 1.9, 0.4, Setosa, 45  
4.8, 3, 1.4, 0.3, Setosa, 46  
5.1, 3.8, 1.6, 0.2, Setosa, 47  
4.6, 3.2, 1.4, 0.2, Setosa, 48  
5.3, 3.7, 1.5, 0.2, Setosa, 49  
5, 3.3, 1.4, 0.2, Setosa, 50  
7, 3.2, 4.7, 1.4, versicolor, 51  
6.4, 3.2, 4.5, 1.5, versicolor, 52  
6.9, 3.1, 4.9, 1.5, versicolor, 53  
5.5, 2.3, 4, 1.3, versicolor, 54  
6.5, 2.8, 4.6, 1.5, versicolor, 55  
5.7, 2.8, 4.5, 1.3, versicolor, 56  
6.3, 3.3, 4.7, 1.6, versicolor, 57  
4.9, 2.4, 3.3, 1, versicolor, 58  
6.6, 2.9, 4.6, 1.3, versicolor, 59  
5.2, 2.7, 3.9, 1.4, versicolor, 60  
5, 2, 3.5, 1, versicolor, 61  
5.9, 3, 4.2, 1.5, versicolor, 62  
6, 2.2, 4, 1, versicolor, 63  
6.1, 2.9, 4.7, 1.4, versicolor, 64  
5.6, 2.9, 3.6, 1.3, versicolor, 65  
6.7, 3.1, 4.4, 1.4, versicolor, 66  
5.6, 3, 4.5, 1.5, versicolor, 67  
5.8, 2.7, 4.1, 1, versicolor, 68  
6.2, 2.2, 4.5, 1.5, versicolor, 69  
5.6, 2.5, 3.9, 1.1, versicolor, 70

5.9, 3.2, 4.8, 1.8, versicolor, 71  
6.1, 2.8, 4, 1.3, versicolor, 72  
6.3, 2.5, 4.9, 1.5, versicolor, 73  
6.1, 2.8, 4.7, 1.2, versicolor, 74  
6.4, 2.9, 4.3, 1.3, versicolor, 75  
6.6, 3, 4.4, 1.4, versicolor, 76  
6.8, 2.8, 4.8, 1.4, versicolor, 77  
6.7, 3, 5, 1.7, versicolor, 78  
6, 2.9, 4.5, 1.5, versicolor, 79  
5.7, 2.6, 3.5, 1, versicolor, 80  
5.5, 2.4, 3.8, 1.1, versicolor, 81  
5.5, 2.4, 3.7, 1, versicolor, 82  
5.8, 2.7, 3.9, 1.2, versicolor, 83  
6, 2.7, 5.1, 1.6, versicolor, 84  
5.4, 3, 4.5, 1.5, versicolor, 85  
6, 3.4, 4.5, 1.6, versicolor, 86  
6.7, 3.1, 4.7, 1.5, versicolor, 87  
6.3, 2.3, 4.4, 1.3, versicolor, 88  
5.6, 3, 4.1, 1.3, versicolor, 89  
5.5, 2.5, 4, 1.3, versicolor, 90  
5.5, 2.6, 4.4, 1.2, versicolor, 91  
6.1, 3, 4.6, 1.4, versicolor, 92  
5.8, 2.6, 4, 1.2, versicolor, 93  
5, 2.3, 3.3, 1, versicolor, 94  
5.6, 2.7, 4.2, 1.3, versicolor, 95  
5.7, 3, 4.2, 1.2, versicolor, 96  
5.7, 2.9, 4.2, 1.3, versicolor, 97  
6.2, 2.9, 4.3, 1.3, versicolor, 98  
5.1, 2.5, 3, 1.1, versicolor, 99  
5.7, 2.8, 4.1, 1.3, versicolor, 100  
6.3, 3.3, 6, 2.5, virginica, 101  
5.8, 2.7, 5.1, 1.9, virginica, 102  
7.1, 3, 5.9, 2.1, virginica, 103  
6.3, 2.9, 5.6, 1.8, virginica, 104  
6.5, 3, 5.8, 2.2, virginica, 105  
7.6, 3, 6.6, 2.1, virginica, 106  
4.9, 2.5, 4.5, 1.7, virginica, 107  
7.3, 2.9, 6.3, 1.8, virginica, 108  
6.7, 2.5, 5.8, 1.8, virginica, 109  
7.2, 3.6, 6.1, 2.5, virginica, 110  
6.5, 3.2, 5.1, 2, virginica, 111  
6.4, 2.7, 5.3, 1.9, virginica, 112  
6.8, 3, 5.5, 2.1, virginica, 113  
5.7, 2.5, 5, 2, virginica, 114  
5.8, 2.8, 5.1, 2.4, virginica, 115  
6.4, 3.2, 5.3, 2.3, virginica, 116  
6.5, 3, 5.5, 1.8, virginica, 117  
7.7, 3.8, 6.7, 2.2, virginica, 118  
7.7, 2.6, 6.9, 2.3, virginica, 119  
6, 2.2, 5, 1.5, virginica, 120  
6.9, 3.2, 5.7, 2.3, virginica, 121  
5.6, 2.8, 4.9, 2, virginica, 122  
7.7, 2.8, 6.7, 2, virginica, 123  
6.3, 2.7, 4.9, 1.8, virginica, 124  
6.7, 3.3, 5.7, 2.1, virginica, 125

```
7.2, 3.2, 6, 1.8, virginica, 126
6.2, 2.8, 4.8, 1.8, virginica, 127
6.1, 3, 4.9, 1.8, virginica, 128
6.4, 2.8, 5.6, 2.1, virginica, 129
7.2, 3, 5.8, 1.6, virginica, 130
7.4, 2.8, 6.1, 1.9, virginica, 131
7.9, 3.8, 6.4, 2, virginica, 132
6.4, 2.8, 5.6, 2.2, virginica, 133
6.3, 2.8, 5.1, 1.5, virginica, 134
6.1, 2.6, 5.6, 1.4, virginica, 135
7.7, 3, 6.1, 2.3, virginica, 136
6.3, 3.4, 5.6, 2.4, virginica, 137
6.4, 3.1, 5.5, 1.8, virginica, 138
6, 3, 4.8, 1.8, virginica, 139
6.9, 3.1, 5.4, 2.1, virginica, 140
6.7, 3.1, 5.6, 2.4, virginica, 141
6.9, 3.1, 5.1, 2.3, virginica, 142
5.8, 2.7, 5.1, 1.9, virginica, 143
6.8, 3.2, 5.9, 2.3, virginica, 144
6.7, 3.3, 5.7, 2.5, virginica, 145
6.7, 3, 5.2, 2.3, virginica, 146
6.3, 2.5, 5, 1.9, virginica, 147
6.5, 3, 5.2, 2, virginica, 148
6.2, 3.4, 5.4, 2.3, virginica, 149
5.9, 3, 5.1, 1.8, virginica, 150
];
```

## KMeansCentroid2D - funzione per grafici

**KMeansCentroid2D()** valuta le righe del grafico applicando il clustering K-means, e per ciascuna riga del grafico visualizza la coordinata desiderata del cluster a cui è stato assegnato questo punto dati. Le colonne utilizzate dall'algoritmo di clustering sono determinate rispettivamente dai parametri `coordinate_1` e `coordinate_2`. Sono entrambe aggregazioni. Il numero di cluster creati è determinato dal parametro `num_clusters`. I dati possono essere normalizzati in via opzionale dal parametro `norm`.

**KMeansCentroid2D** restituisce un valore per punto dati. Il valore restituito è duale ed è una delle coordinate della posizione corrispondente al centro del cluster a cui ciascun punto dati è stato assegnato.

### Sintassi:

```
KMeansCentroid2D(num_clusters, coordinate_no, coordinate_1, coordinate_2 [, norm])
```

**Tipo di dati restituiti:** duale

### Argomenti:

#### Argomenti

| Argomento    | Descrizione                                |
|--------------|--------------------------------------------|
| num_clusters | Intero che specifica il numero di cluster. |

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| coordinate_no | Il numero di coordinata desiderato dei centroid (corrispondente, ad esempio, all'asse x, y o z).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| coordinate_1  | L'aggregazione che calcola la prima coordinata, in genere l'asse x del grafico a dispersione che può essere effettuato dal grafico. Il parametro aggiuntivo, coordinate_2, calcola la seconda coordinata.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| norm          | <p>Il metodo di normalizzazione opzionale applicato alle serie di dati prima del clustering K-means.</p> <p>Possibili valori:</p> <p>0 o 'nessuno' per l'assenza di normalizzazione</p> <p>1 o 'zscore' per la normalizzazione z-score</p> <p>2 o 'minmax' per la normalizzazione min-max</p> <p>Se non viene fornito alcun parametro o se il parametro fornito risulta errato, non viene applicata alcuna normalizzazione.</p> <p>Z-score normalizza i dati in base alla deviazione standard e media della funzionalità. Z-score non assicura che ciascuna funzionalità abbia la stessa scala, ma rappresenta un approccio migliore a min-max quando si ha a che fare con outlier.</p> <p>La normalizzazione min-max assicura che le funzionalità abbiano la stessa scala prelevando i valori minimo e massimo di ciascuna di esse e ricalcolando ciascun datapoint.</p> |

## Clustering automatico

Le funzioni **KMeans** supportano il clustering automatico mediante un metodo chiamato differenza di profondità (DeD, Depth Difference). Quando un utente imposta lo 0 per il numero di cluster, viene determinato un numero ottimale di cluster per tale set di dati. Notare che mentre un valore intero per il numero di cluster ( $k$ ) non viene restituito esplicitamente, viene calcolato all'interno dell'algoritmo KMeans. Ad esempio, se viene specificato 0 nella funzione del valore di *KmeansPetalClusters* o se viene impostato mediante una casella di input variabile, le assegnazioni cluster vengono calcolate automaticamente per il set di dati in base a un numero ottimale di cluster.

## KMeansCentroidND - funzione per grafici

**KMeansCentroidND()** valuta le righe del grafico applicando il clustering K-means, e per ciascuna riga del grafico visualizza la coordinata desiderata del cluster a cui è stato assegnato questo punto dati. Le colonne utilizzate dall'algoritmo di clustering sono determinate dai parametri coordinate\_1, coordinate\_2, ecc. fino a n colonne. Sono tutte aggregazioni. Il numero di cluster creati è determinato dal parametro num\_clusters.

**KMeansCentroidND** restituisce un valore per riga. Il valore restituito è duale ed è una delle coordinate della posizione corrispondente al centro del cluster a cui ciascun punto dati è stato assegnato.

**Sintassi:**

```
KMeansCentroidND(num_clusters, num_iter, coordinate_no, coordinate_1,  
coordinate_2 [,coordinate_3 [, ...]])
```

**Tipo di dati restituiti:** duale

**Argomenti:**

Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                          |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| num_clusters  | Intero che specifica il numero di cluster.                                                                                                                                                                           |
| num_iter      | Il numero di ripetizioni di clustering con centri cluster reinizializzati.                                                                                                                                           |
| coordinate_no | Il numero di coordinata desiderato dei centroid (corrispondente, ad esempio, all'asse x, y o z).                                                                                                                     |
| coordinate_1  | L'aggregazione che calcola la prima coordinata, in genere l'asse x (di un grafico a dispersione che può essere effettuato dal grafico). I parametri aggiuntivi calcolano le seconde, terze e quarte coordinate, ecc. |

### Clustering automatico

Le funzioni **KMeans** supportano il clustering automatico mediante un metodo chiamato differenza di profondità (DeD, Depth Difference). Quando un utente imposta lo 0 per il numero di cluster, viene determinato un numero ottimale di cluster per tale set di dati. Notare che mentre un valore intero per il numero di cluster (*k*) non viene restituito esplicitamente, viene calcolato all'interno dell'algoritmo KMeans. Ad esempio, se viene specificato 0 nella funzione del valore di *KmeansPetalClusters* o se viene impostato mediante una casella di input variabile, le assegnazioni cluster vengono calcolate automaticamente per il set di dati in base a un numero ottimale di cluster.

### STL\_Trend - funzione per grafici

**STL\_Trend** è una funzione di scomposizione delle serie temporali. Insieme a **STL\_Seasonal** e a **STL\_Residual**, questa funzione viene utilizzata per scomporre una serie temporale in componenti stagionali, di tendenza e residuali. Nel contesto dell'algoritmo STL, la scomposizione delle serie temporali viene utilizzata per identificare sia un modello stagionale ricorrente sia una tendenza generale, data una metrica di ingresso e altri parametri. La funzione **STL\_Trend** identifica una tendenza generale, indipendente da schemi o cicli stagionali, dai dati delle serie temporali.

Le tre funzioni STL sono correlate alla metrica di input attraverso una semplice somma:

**STL\_Trend + STL\_Seasonal + STL\_Residual = Metrica di input**

STL (seasonal and trend decomposition using Loess) impiega tecniche di smoothing dei dati e, attraverso i suoi parametri di input, consente all'utente di regolare la periodicità dei calcoli che esegue. Questa periodicità determina il modo in cui la dimensione temporale della metrica di input (una misura) viene segmentata nell'analisi.

Come minimo, **STL\_Trend** prende una metrica di input (`target_measure`) e un valore intero per il suo `period_int`, e restituisce un valore in virgola mobile. La metrica di input avrà la forma di un'aggregazione che varia lungo la dimensione temporale. Facoltativamente, è possibile includere i valori per `seasonal_smoother` e `trend_smoother` al fine di regolare l'algoritmo di smoothing.

È possibile utilizzare questa funzione direttamente nell'editor delle espressioni di un grafico.

### Sintassi:

```
STL_Trend(target_measure, period_int [,seasonal_smoother [,trend_smoother]])
```

Tipo di dati restituiti: duale

| Argomenti                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Argomento                | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>target_measure</b>    | <p>La misura da scomporre in componenti stagionali e di tendenza. Questa dovrebbe essere una misura come Sum(Sales) o Sum(Passengers) che varia nella dimensione temporale.</p> <p>Non deve essere un valore costante.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>period_int</b>        | <p>La periodicità del set di dati. Questo parametro è un valore intero che rappresenta il numero di passi discreti che costituiscono un periodo, o ciclo stagionale, del segnale.</p> <p>Ad esempio, se la serie temporale è segmentata in una sezione per ogni trimestre dell'anno, è necessario impostare <b>period_int</b> a un valore di 4 per definire la periodicità come Anno.</p>                                                                                                                                                                                                                                                                                                                                                             |
| <b>seasonal_smoother</b> | <p>Durata del seasonal smoother. Deve essere un numero intero dispari. Il seasonal smoother utilizza i dati relativi a una particolare fase della variazione stagionale, per un certo numero di periodi. Per ogni periodo viene utilizzato un passo discreto della dimensione temporale. Il seasonal smoother indica il numero di periodi utilizzati per lo smoothing.</p> <p>Ad esempio, se la dimensione temporale è segmentata per mese e il periodo è l'anno (12), la componente stagionale sarà elaborata in modo che ogni particolare mese di ogni anno sia calcolato a partire dai dati dello stesso mese, sia in quell'anno che negli anni adiacenti. Il valore <b>seasonal_smoother</b> è il numero di anni utilizzato per lo smoothing.</p> |
| <b>trend_smoother</b>    | <p>Durata del trend smoother. Deve essere un numero intero dispari. Il trend smoother utilizza la stessa scala temporale del parametro <b>period_int</b> e il suo valore è il numero di granuli utilizzati per lo smoothing.</p> <p>Ad esempio, se una serie temporale è segmentata per mese, il trend smoother corrisponderà al numero di mesi utilizzato per lo smoothing.</p>                                                                                                                                                                                                                                                                                                                                                                      |

La funzione grafico **STL\_Trend** viene spesso utilizzata in combinazione con le seguenti funzioni:

### Funzioni correlate

| Funzione                                                               | Interazione                                                                                                                                                                                                                                                      |
|------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#"><u>STL_Seasonal - funzione per grafici (page 1677)</u></a> | È la funzione utilizzata per calcolare la componente stagionale di una serie temporale.                                                                                                                                                                          |
| <a href="#"><u>STL_Residual - funzione per grafici (page 1679)</u></a> | Quando si scompone una metrica di input in componente stagionale e componente di tendenza, parte della variazione delle misure non rientra in nessuna delle due componenti principali. La funzione <b>STL_Residual</b> calcola questa parte della scomposizione. |

Per un tutorial con un esempio completo che mostra come utilizzare questa funzione, vedere [Tutorial - Scomposizione delle serie temporali in Qlik Sense \(page 1681\)](#).

### STL\_Seasonal - funzione per grafici

**STL\_Seasonal** è una funzione di scomposizione delle serie temporali. Insieme a **STL\_Trend** e a **STL\_Residual**, questa funzione viene utilizzata per scomporre una serie temporale in componenti stagionali, di tendenza e residui. Nel contesto dell'algoritmo STL, la scomposizione delle serie temporali viene utilizzata per identificare sia un modello stagionale ricorrente sia una tendenza generale, data una metrica di ingresso e altri parametri. La funzione **STL\_Seasonal** è in grado di identificare un modello stagionale all'interno di una serie temporale, separandolo dalla tendenza generale mostrata dai dati.

Le tre funzioni STL sono correlate alla metrica di input attraverso una semplice somma:

**STL\_Trend + STL\_Seasonal + STL\_Residual = Metrica di input**

STL (seasonal and trend decomposition using Loess) impiega tecniche di smoothing dei dati e, attraverso i suoi parametri di input, consente all'utente di regolare la periodicità dei calcoli che esegue. Questa periodicità determina il modo in cui la dimensione temporale della metrica di input (una misura) viene segmentata nell'analisi.

Come minimo, **Seasonal** prende una metrica di input (`target_measure`) e un valore intero per il suo `period_int`, e restituisce un valore in virgola mobile. La metrica di input avrà la forma di un'aggregazione che varia lungo la dimensione temporale. Facoltativamente, è possibile includere i valori per `seasonal_smoother` e `trend_smoother` al fine di regolare l'algoritmo di smoothing.

È possibile utilizzare questa funzione direttamente nell'editor delle espressioni di un grafico.

### Sintassi:

```
STL_Seasonal(target_measure, period_int [,seasonal_smoother [,trend_smoother]])
```

**Tipo di dati restituiti:** duale

| Argomenti                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Argomento                | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>target_measure</b>    | <p>La misura da scomporre in componenti stagionali e di tendenza. Questa dovrebbe essere una misura come Sum(Sales) o Sum(Passengers) che varia nella dimensione temporale.</p> <p>Non deve essere un valore costante.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>period_int</b>        | <p>La periodicità del set di dati. Questo parametro è un valore intero che rappresenta il numero di passi discreti che costituiscono un periodo, o ciclo stagionale, del segnale.</p> <p>Ad esempio, se la serie temporale è segmentata in una sezione per ogni trimestre dell'anno, è necessario impostare <b>period_int</b> a un valore di 4 per definire la periodicità come Anno.</p>                                                                                                                                                                                                                                                                                                                                                             |
| <b>seasonal_smoother</b> | <p>Durata del seasonal smoother. Deve essere un numero intero dispari. Il seasonal smoother utilizza i dati relativi a una particolare fase della variazione stagionale, per un certo numero di periodi. Per ogni periodo viene utilizzato un passo discreto della dimensione temporale. Il seasonal smoother indica il numero di periodi utilizzati per lo smoothing.</p> <p>Ad esempio, se la dimensione temporale è segmentata per mese e il periodo è l'anno (12), la componente stagionale sarà elaborata in modo che ogni particolare mese di ogni anno sia calcolato a partire dai dati dello stesso mese, sia in quell'anno che negli anni adiacenti. Il valore <b>seasonal_smoother</b> è il numero di anni utilizzato per lo smoothing.</p> |

| Argomento             | Descrizione                                                                                                                                                                                                                                                                                                                                                                      |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>trend_smoother</b> | <p>Durata del trend smoother. Deve essere un numero intero dispari. Il trend smoother utilizza la stessa scala temporale del parametro <b>period_int</b> e il suo valore è il numero di granuli utilizzati per lo smoothing.</p> <p>Ad esempio, se una serie temporale è segmentata per mese, il trend smoother corrisponderà al numero di mesi utilizzato per lo smoothing.</p> |

La funzione grafico **STL\_Seasonal** viene spesso utilizzata in combinazione con le seguenti funzioni:

#### Funzioni correlate

| Funzione                                                               | Interazione                                                                                                                                                                                                                                                      |
|------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#"><i>STL_Trend - funzione per grafici (page 1675)</i></a>    | È la funzione utilizzata per calcolare la componente di tendenza di una serie temporale.                                                                                                                                                                         |
| <a href="#"><i>STL_Residual - funzione per grafici (page 1679)</i></a> | Quando si scompone una metrica di input in componente stagionale e componente di tendenza, parte della variazione delle misure non rientra in nessuna delle due componenti principali. La funzione <b>STL_Residual</b> calcola questa parte della scomposizione. |

Per un tutorial con un esempio completo che mostra come utilizzare questa funzione, vedere [\*Tutorial - Scomposizione delle serie temporali in Qlik Sense \(page 1681\)\*](#).

## STL\_Residual - funzione per grafici

**STL\_Residual** è una funzione di scomposizione delle serie temporali. Insieme a **STL\_Seasonal** e a **STL\_Trend**, questa funzione viene utilizzata per scomporre una serie temporale in componenti stagionali, di tendenza e residuali. Nel contesto dell'algoritmo STL,

la scomposizione delle serie temporali viene utilizzata per identificare sia un modello stagionale ricorrente sia una tendenza generale, data una metrica di ingresso e altri parametri. Nell'eseguire questa operazione, una parte della variazione della metrica di input non rientrerà né nella componente stagionale né in quella di tendenza e sarà definita come componente residua. La funzione grafico **STL\_Residual** acquisisce questa parte del calcolo.

Le tre funzioni STL sono correlate alla metrica di input attraverso una semplice somma:

**STL\_Trend + STL\_Seasonal + STL\_Residual = Metrica di input**

STL (seasonal and trend decomposition using Loess) impiega tecniche di smoothing dei dati e, attraverso i suoi parametri di input, consente all'utente di regolare la periodicità dei calcoli che esegue. Questa periodicità determina il modo in cui la dimensione temporale della metrica di input (una misura) viene segmentata nell'analisi.

Poiché la scomposizione delle serie temporali cerca principalmente la stagionalità e le variazioni generali dei dati, l'informazione nel residuo è considerata la meno significativa delle tre componenti. Tuttavia, una componente residua obliqua o periodica può aiutare a identificare problemi nel calcolo, come ad esempio un'impostazione errata della periodicità.

Come minimo, **STL\_Residual** prende una metrica di input (`target_measure`) e un valore intero per il suo `period_int`, e restituisce un valore in virgola mobile. La metrica di input avrà la forma di un'aggregazione che varia lungo la dimensione temporale. Facoltativamente, è possibile includere i valori per `seasonal_smoother` e `trend_smoother` al fine di regolare l'algoritmo di smoothing.

È possibile utilizzare questa funzione direttamente nell'editor delle espressioni di un grafico.

### Sintassi:

```
STL_Residual(target_measure, period_int [,seasonal_smoother [,trend_smoother]])
```

**Tipo di dati restituiti:** duale

### Argomenti

| Argomento             | Descrizione                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>target_measure</b> | La misura da scomporre in componenti stagionali e di tendenza. Questa dovrebbe essere una misura come Sum(Sales) o Sum(Passengers) che varia nella dimensione temporale.<br><br>Non deve essere un valore costante.                                                                                                                                                                |
| <b>period_int</b>     | La periodicità del set di dati. Questo parametro è un valore intero che rappresenta il numero di passi discreti che costituiscono un periodo, o ciclo stagionale, del segnale.<br><br>Ad esempio, se la serie temporale è segmentata in una sezione per ogni trimestre dell'anno, è necessario impostare <b>period_int</b> a un valore di 4 per definire la periodicità come Anno. |

| Argomento                | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>seasonal_smoother</b> | <p>Durata del seasonal smoother. Deve essere un numero intero dispari. Il seasonal smoother utilizza i dati relativi a una particolare fase della variazione stagionale, per un certo numero di periodi. Per ogni periodo viene utilizzato un passo discreto della dimensione temporale. Il seasonal smoother indica il numero di periodi utilizzati per lo smoothing.</p> <p>Ad esempio, se la dimensione temporale è segmentata per mese e il periodo è l'anno (12), la componente stagionale sarà elaborata in modo che ogni particolare mese di ogni anno sia calcolato a partire dai dati dello stesso mese, sia in quell'anno che negli anni adiacenti. Il valore <b>seasonal_smoother</b> è il numero di anni utilizzato per lo smoothing.</p> |
| <b>trend_smoother</b>    | <p>Durata del trend smoother. Deve essere un numero intero dispari. Il trend smoother utilizza la stessa scala temporale del parametro <b>period_int</b> e il suo valore è il numero di granuli utilizzati per lo smoothing.</p> <p>Ad esempio, se una serie temporale è segmentata per mese, il trend smoother corrisponderà al numero di mesi utilizzato per lo smoothing.</p>                                                                                                                                                                                                                                                                                                                                                                      |

La funzione grafico **STL\_Residual** viene spesso utilizzata in combinazione con le seguenti funzioni:

#### Funzioni correlate

| Funzione                                                               | Interazione                                                                              |
|------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
| <a href="#"><i>STL_Seasonal - funzione per grafici (page 1677)</i></a> | È la funzione utilizzata per calcolare la componente stagionale di una serie temporale.  |
| <a href="#"><i>STL_Trend - funzione per grafici (page 1675)</i></a>    | È la funzione utilizzata per calcolare la componente di tendenza di una serie temporale. |

Per un tutorial con un esempio completo che mostra come utilizzare questa funzione, vedere [\*Tutorial - Scomposizione delle serie temporali in Qlik Sense \(page 1681\)\*](#).

## Tutorial - Scomposizione delle serie temporali in Qlik Sense

Questo tutorial dimostra l'utilizzo di tre funzioni grafiche per scomporre una serie temporale utilizzando l'algoritmo STL.

Questo tutorial utilizza i dati delle serie temporali del numero di passeggeri di una compagnia aerea al mese per dimostrare la funzionalità dell'algoritmo STL. Le funzioni del grafico **STL\_Trend**, **STL\_Seasonal** e **STL\_Residual** verranno utilizzate per creare le visualizzazioni. Per ulteriori informazioni sulla scomposizione delle serie temporali in Qlik Sense, vedere [Funzioni di scomposizione serie temporale \(page 1626\)](#).

### Creare un'app

Per iniziare, creare una nuova app, quindi importare il set di dati.

Scaricare questo set di dati:

[Tutorial - Scomposizione delle serie temporali](#)

Questo file contiene i dati relativi al numero di passeggeri mensili di una compagnia aerea.

**Procedere come indicato di seguito:**

1. Nell'hub, fare clic su **Crea nuova app**.
2. Aprire l'app e rilasciare il file *Tutorial - Time series decomposition.csv* su di essa.

### Preparare e caricare i dati

Per permettere a Qlik Sense di interpretare correttamente il campo YearMonth, potrebbe essere necessario utilizzare Gestione dati per riconoscere il campo come un campo data e non come un campo con valori stringa. In genere questo passaggio è gestito automaticamente, ma in questo caso le date sono presentate nel formato YYYY-MM, un po' insolito.

1. In Gestione dati selezionare una tabella e fare clic su .
2. Con il campo *YearMonth* selezionato, fare clic su  e impostare il **Tipo campo** su **Data**.
3. Sotto **Formato input**, inserire YYYY-MM.
4. Sotto **Formato visualizzazione**, inserire YYYY-MM e fare clic su **OK**.  
Il campo dovrebbe ora mostrare l'icona del calendario.
5. Fare clic su **Carica dati**.

Ora si è pronti per iniziare a usare le funzioni STL per rappresentare visivamente i propri dati.

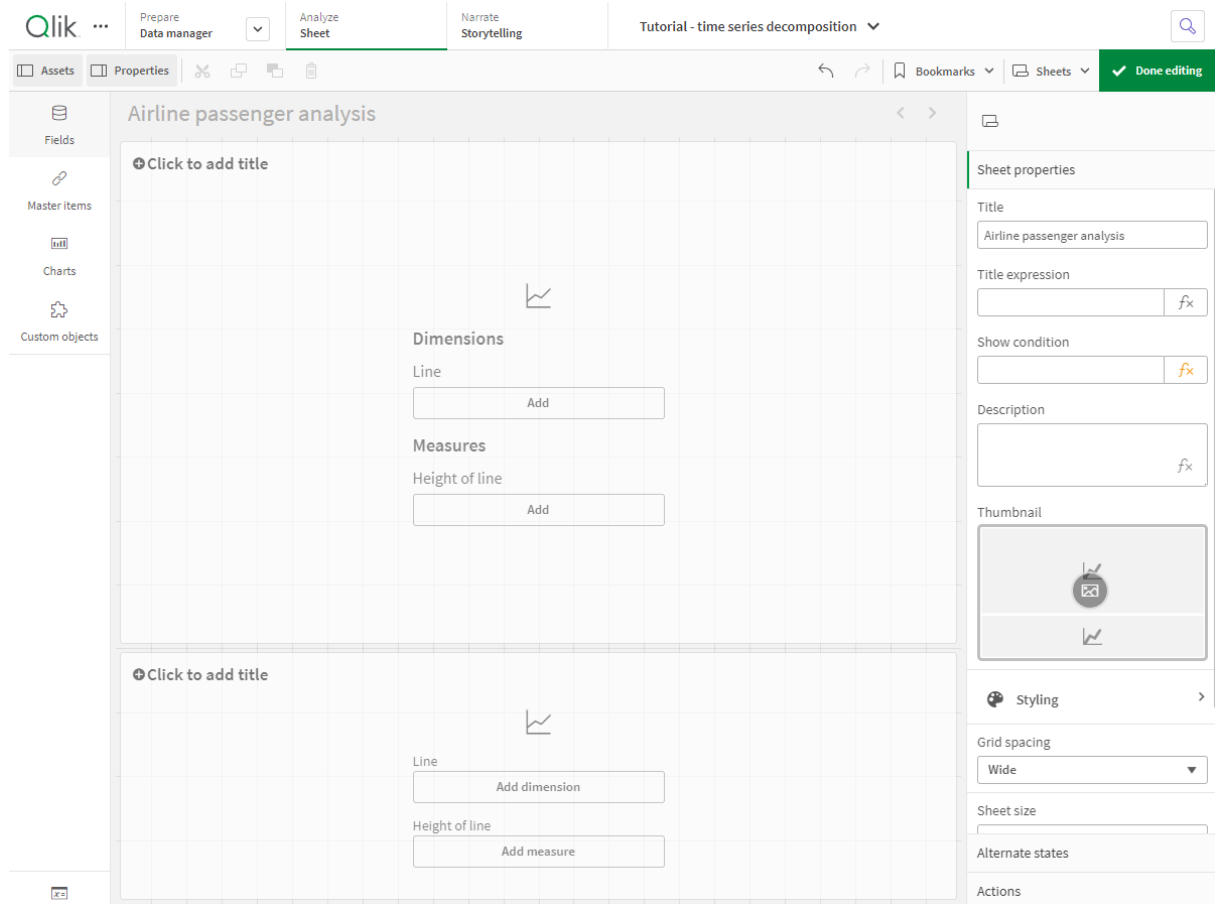
### Creare le visualizzazioni

Ora, creare due grafici lineari per dimostrare la funzionalità delle funzioni grafico **STL\_Trend**, **STL\_Seasonal** e **STL\_Residual**.

Aprire un nuovo foglio, quindi assegnargli un titolo.

Aggiungere due grafici lineari al foglio. Ridimensionare e riposizionare i grafici in modo che corrispondano all'immagine seguente.

Contorno della griglia di Qlik Sense per il foglio vuoto dell'app



### Primo grafico lineare: tendenza e componenti stagionali

Procedere come indicato di seguito:

1. Aggiungere il titolo *Stagionale e tendenza* al primo grafico lineare.
2. Aggiungere *YearMonth* come dimensione, quindi assegnare l'etichetta *Data*.
3. Aggiungere la seguente misura ed etichettarla *Passeggeri al mese*:  
 $=\text{Sum}(\text{Passengers})$
4. Sotto **Dati**, espandere la misura *Passeggeri al mese* e fare clic su **Aggiungi linea di tendenza**.
5. Impostare il **Tipo** su **Lineare**.  
Questa linea di tendenza verrà confrontata con l'output uniformato del componente di tendenza.
6. Aggiungere la seguente misura per tracciare la componente di tendenza ed etichettarla *Tendenza*:  
 $=\text{STL\_Trend}(\text{SUM}(\text{Passengers}), 12)$
7. Quindi, aggiungere la seguente misura per tracciare la componente stagionale ed etichettarla *Stagionale*:  
 $=\text{STL\_Seasonal}(\text{SUM}(\text{Passengers}), 12)$
8. Sotto **Aspetto > Presentazione**, impostare **Barra di scorrimento** su **Nessuna**.
9. Mantenere i colori predefiniti o modificarli in base alle proprie preferenze.

### Secondo grafico lineare: componente residuo

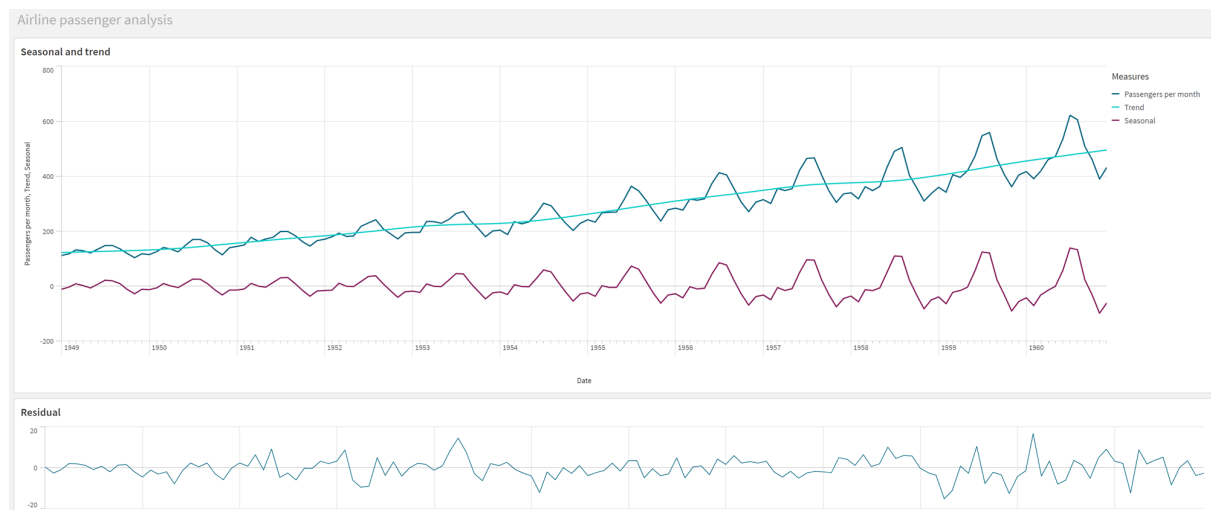
Quindi, configurare il secondo grafico lineare. Questa visualizzazione mostra la componente residua della serie temporale.

Procedere come indicato di seguito:

1. Trascinare un grafico lineare sul foglio. Aggiungere il titolo *Residuo*.
2. Aggiungere *Data* come dimensione.
3. Aggiungere la seguente misura ed etichettarla come *Residua*:  
 $=STL\_Residual(SUM(Passengers), 12)$
4. Sotto **Aspetto > Presentazione**, impostare **Barra di scorrimento** su **Nessuna**.

Il proprio foglio dovrebbe ora assomigliare a quello qui sotto.

*Foglio Qlik Sense per l'analisi dei passeggeri delle compagnie aeree*



### Interpretazione e spiegazione dei dati

Con le funzioni STL, possiamo ottenere una serie di informazioni strategiche dai dati delle serie temporali.

#### Componente di tendenza

Le informazioni statistiche nella componente di tendenza sono destagionalizzate. In questo modo è più facile vedere le fluttuazioni generali e non ripetute nel tempo. Rispetto alla linea di tendenza rettilinea e lineare per i *Passeggeri al mese*, la componente di tendenza STL cattura le tendenze in evoluzione. Mostra alcune chiare deviazioni, pur presentando le informazioni in modo leggibile. I comportamenti di uniformità nell'algoritmo STL hanno contribuito a catturare questo aspetto.

Il calo del numero di passeggeri delle compagnie aeree, visibile nel grafico del trend di STL, può essere spiegato come parte dell'impatto economico delle recessioni che si sono verificate negli anni Cinquanta.

### Componente stagionale

La componente stagionale priva di trend ha isolato le fluttuazioni ricorrenti nella serie temporale, eliminando le informazioni sulle tendenze generali da questa parte dell'analisi. Abbiamo iniziato con un set di dati composto da aggregazioni anno-mese. Con questi dati, è implicito che stiamo segmentando i dati in granuli di un mese. Definendo un valore di periodo pari a 12, si imposta il grafico per modellare i modelli stagionali nel corso di cicli di un anno (dodici mesi).

I dati mostrano un modello stagionale ripetuto di aumento dei passeggeri delle compagnie aeree nei mesi estivi, seguito da un calo nei mesi invernali. Questo è in linea con l'idea che l'estate sia un periodo molto gettonato per le vacanze e i viaggi. Si nota anche che, nel corso della serie temporale, questi cicli stagionali aumentano drasticamente in ampiezza.

### Componente residuo

Il grafico della componente residua mostra tutte le informazioni che non sono state acquisite dalla scomposizione per trend e stagionalità. La componente residua comprende il rumore statistico, ma può anche indicare un'impostazione errata degli argomenti delle funzioni STL di tendenza e stagionali. In generale, se si verificano oscillazioni periodiche nella componente residua del segnale, o se le informazioni visualizzate non sono chiaramente casuali, di solito si tratta di un segnale che indica la presenza di informazioni nella serie temporale che non sono attualmente acquisite nelle componenti stagionali o di tendenza. In questo caso, è necessario rivedere le definizioni di ciascun argomento della funzione ed eventualmente modificare la periodicità.

### Valori smoother

Poiché non sono stati specificati valori per gli smoother di tendenza e stagionali, la funzione utilizzerà i valori predefiniti per questi parametri. In Qlik Sense, i valori predefiniti dello smoother nell'algoritmo STL producono risultati efficaci. Di conseguenza, nella maggior parte dei casi, questi argomenti possono essere lasciati fuori dalle espressioni.



*Impostando gli argomenti dello smoother stagionale o di tendenza come 0 in una delle tre funzioni STL, l'algoritmo utilizzerà valori predefiniti, anziché valori pari a 0.*

Il valore dello smoother di tendenza utilizza la dimensione specificata nel grafico. Poiché il campo *YearMonth* presenta i dati per mesi, il valore di trend smoother sarà il numero di mesi. Lo smoother stagionale rifletterà la periodicità definita. In questo caso, poiché abbiamo definito che un periodo dura dodici mesi (un anno), il valore dello smoother stagionale è il numero di anni. Ciò può sembrare confuso, ma in realtà significa che per trovare la stagionalità, dobbiamo esaminare un certo numero di stagioni. Questo numero è lo smoother stagionale.

### Altre informazioni utili

Dato che i cicli stagionali aumentano di ampiezza nel tempo, un approccio analitico più avanzato potrebbe fare uso di funzioni logaritmiche per creare una scomposizione moltiplicativa. In pratica, in Qlik Sense è possibile creare una semplice misura dell'ampiezza relativa dividendo la componente stagionale per quella di tendenza. In questo modo, si nota che nel corso del tempo i picchi estivi di ciascun ciclo aumentano in ampiezza relativa. L'ampiezza dei punti bassi invernali, tuttavia, non aumenta nel tempo.

## 8.23 Funzioni di distribuzione statistica

Le funzioni di distribuzione statistica restituiscono le probabilità di accadimento dei diversi risultati possibili per una determinata variabile di input. È possibile utilizzare queste funzioni per calcolare i valori potenziali dei punti dati.

I tre gruppi delle funzioni di distribuzione statistica descritti di seguito sono tutte implementati in Qlik Sense mediante la libreria di funzioni Cephес. Per ottenere riferimenti e informazioni dettagliate sugli algoritmi utilizzati, l'accuratezza e così via, visitare il sito [≤ Cephес library](#). La libreria di funzioni Cephес è utilizzata su concessione.

- Le funzioni di probabilità calcolano la probabilità nel punto della distribuzione dato dal valore fornito.
  - Le funzioni di frequenza sono utilizzate per le distribuzioni discrete.
  - Le funzioni di densità vengono utilizzate per le funzioni continue.
- Le funzioni Dist calcolano la probabilità cumulativa della distribuzione nel punto della distribuzione dato dal valore fornito.
- Le funzioni Inv calcolano il valore inverso, data la probabilità cumulativa della distribuzione.

Tutte le funzioni possono essere utilizzate sia nello script di caricamento dei dati che nelle espressioni grafiche.

### Panoramica sulle funzioni di distribuzione statistica

Ciascuna funzione viene descritta ulteriormente dopo la panoramica. È inoltre possibile fare clic sul nome della funzione nella sintassi per accedere immediatamente ai dettagli per tale funzione specifica.

#### BetaDensity

BetaDensity() restituisce la probabilità della distribuzione Beta.

```
BetaDensity (value, alpha, beta)
```

#### BetaDist

BetaDist() restituisce la probabilità cumulativa della distribuzione Beta.

```
BetaDist (value, alpha, beta)
```

#### BetaInv

BetaINV() restituisce l'inverso della probabilità cumulativa della distribuzione Beta.

```
BetaInv (prob, alpha, beta)
```

#### BinomDist

BinomDist() restituisce la probabilità cumulativa della distribuzione Binomiale.

```
BinomDist (value, trials, trial_probability)
```

#### BinomFrequency

BinomFrequency() restituisce la distribuzione di probabilità Binomiale.

**BinomFrequency** (value, trials, trial\_probability)

### BinomInv

BinomInv() restituisce l'inverso della probabilità cumulativa della distribuzione Binomiale.

**BinomInv** (prob, trials, trial\_probability)

### ChiDensity

ChiDensity() restituisce la probabilità a una coda della distribuzione  $\chi^2$ . La funzione di densità  $\chi^2$  è associata a un test  $\chi^2$ .

**ChiDensity** (value, degrees\_freedom)

### ChiDist

ChiDist() restituisce la probabilità a una coda della distribuzione  $\chi^2$ . La distribuzione  $\chi^2$  è associata a un test  $\chi^2$ .

**ChiDist** (value, degrees\_freedom)

### ChiInv

ChiInv() restituisce l'inverso della probabilità a una coda della distribuzione  $\chi^2$ .

**ChiInv** (prob, degrees\_freedom)

### FDensity

FDensity() restituisce la probabilità della distribuzione F.

**FDensity** (value, degrees\_freedom1, degrees\_freedom2)

### FDist

FDist() restituisce la probabilità cumulativa della distribuzione F.

**FDist** (value, degrees\_freedom1, degrees\_freedom2)

### FInv

FInv() restituisce l'inverso della probabilità cumulativa della distribuzione F.

**FInv** (prob, degrees\_freedom1, degrees\_freedom2)

### GammaDensity

GammaDensity() restituisce la probabilità della distribuzione Gamma.

**GammaDensity** (value, k,  $\theta$ )

### GammaDist

GammaDist() restituisce la probabilità accumulata della distribuzione Gamma.

**GammaDist** (value, k,  $\theta$ )

### GammaInv

GammaInv() restituisce l'inverso della probabilità cumulativa della distribuzione Gamma.

**GammaInv** (prob, k,  $\theta$ )

### NormDist

NormDist() restituisce la distribuzione cumulativa normale per la media e la deviazione standard specificate. Se mean = 0 e standard\_dev = 1, la funzione restituisce la distribuzione normale standard.

```
NormDist (value, mean, standard_dev)
```

### NormInv

NormInv() restituisce il valore contrario della distribuzione cumulativa normale per la media e la deviazione standard specificate.

```
NormInv (prob, mean, standard_dev)
```

### PoissonDist

PoissonDist() restituisce la probabilità cumulativa della distribuzione Poisson.

```
PoissonDist (value, mean)
```

### PoissonFrequency

PoissonFrequency() restituisce la distribuzione di probabilità di Poisson.

```
PoissonFrequency (value, mean)
```

### PoissonInv

PoissonInv() restituisce l'inverso della probabilità cumulativa della distribuzione Poisson.

```
PoissonInv (prob, mean)
```

### TDensity

TDensity() restituisce il valore della funzione di densità t per lo studente, dove un valore numerico è un valore calcolato di t per il quale la probabilità deve essere elaborata.

```
TDensity (value, degrees_freedom, tails)
```

### TDist

TDist() restituisce la probabilità per la distribuzione t dello studente in cui un valore numerico è un valore calcolato di t per il quale deve essere calcolata la probabilità.

```
TDist (value, degrees_freedom, tails)
```

### TInv

TInv() restituisce il valore t della distribuzione t dello studente in funzione della probabilità e dei gradi di libertà.

```
TInv (prob, degrees_freedom)
```

---

### Vedere anche:

p [Funzioni di aggregazione statistica \(page 450\)](#)

## BetaDensity

`BetaDensity()` restituisce la probabilità della distribuzione Beta.

### Sintassi:

```
BetaDensity(value, alpha, beta)
```

**Tipo di dati restituiti:** numero

| Argomenti |                                                                                                                        |
|-----------|------------------------------------------------------------------------------------------------------------------------|
| Argomento | Descrizione                                                                                                            |
| value     | Il valore in corrispondenza del quale si desidera valutare la distribuzione. Il valore deve essere compreso tra 0 e 1. |
| alpha     | Un numero positivo che definisce il primo parametro di forma. È l'esponente della variabile casuale                    |
| beta      | Un numero positivo che definisce il secondo parametro di forma. Indica il numero di gradi di libertà del denominatore. |

## BetaDist

`BetaDist()` restituisce la probabilità cumulativa della distribuzione Beta.

### Sintassi:

```
BetaDist(value, alpha, beta)
```

**Tipo di dati restituiti:** numero

| Argomenti |                                                                                                                           |
|-----------|---------------------------------------------------------------------------------------------------------------------------|
| Argomento | Descrizione                                                                                                               |
| value     | Il valore in corrispondenza del quale si desidera valutare la distribuzione. Il valore deve essere compreso tra 0 e 1.    |
| alpha     | Un numero positivo che definisce il primo parametro di forma. È l'esponente della variabile casuale                       |
| beta      | Un numero positivo che definisce il secondo parametro di forma. È l'esponente che controlla la forma della distribuzione. |

Questa funzione è correlata alla funzione `BetaInv` nel modo seguente:

If `prob = BetaDist(value, alpha, beta)`, then `BetaInv(prob, alpha, beta) = value`

## BetaInv

`BetaInv()` restituisce l'inverso della probabilità cumulativa della distribuzione Beta.

### Sintassi:

```
BetaInv(prob, alpha, beta)
```

**Tipo di dati restituiti:** numero

#### Argomenti

| Argomento | Descrizione                                                                                                               |
|-----------|---------------------------------------------------------------------------------------------------------------------------|
| prob      | Una probabilità associata alla distribuzione di probabilità Beta. Deve essere un numero compreso tra 0 e 1.               |
| alpha     | Un numero positivo che definisce il primo parametro di forma. È l'esponente della variabile casuale                       |
| beta      | Un numero positivo che definisce il secondo parametro di forma. È l'esponente che controlla la forma della distribuzione. |

Questa funzione è correlata alla funzione `BetaDist` nel modo seguente:

If `prob = BetaDist(value, alpha, beta)`, then `BetaInv(prob, alpha, beta) = value`

## BinomDist

`BinomDist()` restituisce la probabilità cumulativa della distribuzione Binomiale.

### Sintassi:

```
BinomDist(value, trials, trial_probability)
```

**Tipo di dati restituiti:** numero

#### Argomenti

| Argomento         | Descrizione                                                                                                                                                                  |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value             | Il valore in corrispondenza del quale si desidera valutare la distribuzione. Il valore deve essere un numero intero non inferiore a zero e non superiore al numero di prove. |
| trials            | Un numero intero positivo che indica il numero di prove.                                                                                                                     |
| trial_probability | La probabilità di successo per ogni prova. È sempre un numero compreso tra 0 e 1.                                                                                            |

Questa funzione è correlata alla funzione `BinomInv` nel modo seguente:

If `prob = BinomDIST(value, trials, trial_probability)`, then `BinomInv(prob, trials, trial_probability) = value`

## BinomFrequency

`BinomFrequency()` restituisce la distribuzione di probabilità Binomiale.

### Sintassi:

```
BinomFrequency(value, trials, trial_probability)
```

Tipo di dati restituiti: numero

#### Argomenti

| Argomento         | Descrizione                                                                                                                                                                  |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value             | Il valore in corrispondenza del quale si desidera valutare la distribuzione. Il valore deve essere un numero intero non inferiore a zero e non superiore al numero di prove. |
| trials            | Un numero intero positivo che indica il numero di prove                                                                                                                      |
| trial_probability | La probabilità di successo per ogni prova. È sempre un numero compreso tra 0 e 1.                                                                                            |

## BinomInv

`BinomInv()` restituisce l'inverso della probabilità cumulativa della distribuzione Binomiale.

#### Sintassi:

```
BinomInv(prob, trials, trial_probability)
```

Tipo di dati restituiti: numero

#### Argomenti

| Argomento         | Descrizione                                                                                                      |
|-------------------|------------------------------------------------------------------------------------------------------------------|
| prob              | Una probabilità associata alla distribuzione di probabilità Binomiale. Deve essere un numero compreso tra 0 e 1. |
| trials            | Un numero intero positivo che indica il numero di prove.                                                         |
| trial_probability | La probabilità di successo per ogni prova. È sempre un numero compreso tra 0 e 1.                                |

Questa funzione è correlata alla funzione `BinomDist` nel modo seguente:

If `prob = BinomDist(value, trials, trial_probability)`, then `BinomInv(prob, trials, trial_probability) = value`

## ChiDensity

`ChiDensity()` restituisce la probabilità a una coda della distribuzione  $\chi^2$ . La funzione di densità  $\chi^2$  è associata a un test  $\chi^2$ .

#### Sintassi:

```
ChiDensity(value, degrees_freedom)
```

Tipo di dati restituiti: numero

#### Argomenti

| Argomento       | Descrizione                                                                                                      |
|-----------------|------------------------------------------------------------------------------------------------------------------|
| value           | Il valore in corrispondenza del quale si desidera valutare la distribuzione. Il valore non deve essere negativo. |
| degrees_freedom | Numero intero positivo corrispondente al numero di gradi di libertà del numeratore.                              |

## ChiDist

`ChiDist()` restituisce la probabilità a una coda della distribuzione  $\chi^2$ . La distribuzione  $\chi^2$  è associata a un test  $\chi^2$ .

#### Sintassi:

```
CHIDIST(value, degrees_freedom)
```

Tipo di dati restituiti: numero

#### Argomenti:

#### Argomenti

| Argomento       | Descrizione                                                                                                      |
|-----------------|------------------------------------------------------------------------------------------------------------------|
| value           | Il valore in corrispondenza del quale si desidera valutare la distribuzione. Il valore non deve essere negativo. |
| degrees_freedom | Un numero intero positivo corrispondente al numero di gradi di libertà.                                          |

Questa funzione è correlata alla funzione **ChiInv** nel modo seguente:

If `prob = CHIDIST(value,df)`, then `CHIINV(prob, df) = value`

#### Limiti:

Tutti gli argomenti devono essere numerici, altrimenti viene restituito un valore NULL.

#### Esempi e risultati:

| Esempio                      | Risultato          |
|------------------------------|--------------------|
| <code>CHIDIST( 8, 15)</code> | Restituisce 0,9238 |

## ChiInv

`ChiInv()` restituisce l'inverso della probabilità a una coda della distribuzione  $\chi^2$ .

#### Sintassi:

```
CHIINV(prob, degrees_freedom)
```

**Tipo di dati restituiti:** numero

**Argomenti:**

Argomenti

| Argomento       | Descrizione                                                                                          |
|-----------------|------------------------------------------------------------------------------------------------------|
| prob            | Una probabilità associata alla distribuzione di $\chi^2$ . Deve essere un numero compreso tra 0 e 1. |
| degrees_freedom | Un numero intero corrispondente al numero di gradi di libertà.                                       |

Questa funzione è correlata alla funzione **ChiDist** nel modo seguente:

If prob = CHIDIST(value,df), then CHIINV(prob, df) = value

**Limiti:**

Tutti gli argomenti devono essere numerici, altrimenti viene restituito un valore NULL.

**Esempi e risultati:**

| Esempio               | Risultato          |
|-----------------------|--------------------|
| CHIINV(0.9237827, 15) | Restituisce 8,0000 |

## FDensity

**FDensity()** restituisce la probabilità della distribuzione F.

**Sintassi:**

```
FDensity(value, degrees_freedom1, degrees_freedom2)
```

**Tipo di dati restituiti:** numero

Argomenti

| Argomento        | Descrizione                                                                                                      |
|------------------|------------------------------------------------------------------------------------------------------------------|
| value            | Il valore in corrispondenza del quale si desidera valutare la distribuzione. Il valore non deve essere negativo. |
| degrees_freedom1 | Numero intero positivo corrispondente al numero di gradi di libertà del numeratore.                              |
| degrees_freedom2 | Numero intero positivo corrispondente al numero di gradi di libertà del denominatore.                            |

## FDist

**FDist()** restituisce la probabilità cumulativa della distribuzione F.

**Sintassi:**

```
FDist(value, degrees_freedom1, degrees_freedom2)
```

**Tipo di dati restituiti:** numero

**Argomenti:**

## Argomenti

| Argomento        | Descrizione                                                                                                      |
|------------------|------------------------------------------------------------------------------------------------------------------|
| value            | Il valore in corrispondenza del quale si desidera valutare la distribuzione. Il valore non deve essere negativo. |
| degrees_freedom1 | Numero intero positivo corrispondente al numero di gradi di libertà del numeratore.                              |
| degrees_freedom2 | Numero intero positivo corrispondente al numero di gradi di libertà del denominatore.                            |

Questa funzione è correlata alla funzione **FInv** nel modo seguente:

If prob = FDIST(value, df1, df2), then FINV(prob, df1, df2) = value

**Limiti:**

Tutti gli argomenti devono essere numerici, altrimenti viene restituito un valore NULL.

**Esempi e risultati:**

| Esempio         | Risultato          |
|-----------------|--------------------|
| FDIST(15, 8, 6) | Restituisce 0,0019 |

**FInv**

**FInv()** restituisce l'inverso della probabilità cumulativa della distribuzione F.

**Sintassi:**

```
FInv(prob, degrees_freedom1, degrees_freedom2)
```

**Tipo di dati restituiti:** numero

**Argomenti:**

## Argomenti

| Argomento       | Descrizione                                                                                          |
|-----------------|------------------------------------------------------------------------------------------------------|
| prob            | Una probabilità associata alla distribuzione della probabilità F che deve essere compresa tra 0 e 1. |
| degrees_freedom | Un numero intero corrispondente al numero di gradi di libertà.                                       |

Questa funzione è correlata alla funzione **FDist** nel modo seguente:

If  $\text{prob} = \text{FDIST}(\text{value}, \text{df1}, \text{df2})$ , then  $\text{FINV}(\text{prob}, \text{df1}, \text{df2}) = \text{value}$

### Limiti:

Tutti gli argomenti devono essere numerici, altrimenti viene restituito un valore NULL.

### Esempi e risultati:

| Esempio                             | Risultato           |
|-------------------------------------|---------------------|
| <code>FINV( 0.0019369, 8, 6)</code> | Restituisce 15,0000 |

## GammaDensity

`GammaDensity()` restituisce la probabilità della distribuzione Gamma.

### Sintassi:

```
GammaDensity(value, k,  $\theta$ )
```

**Tipo di dati restituiti:** numero

#### Argomenti

| Argomento                        | Descrizione                                                                                                      |
|----------------------------------|------------------------------------------------------------------------------------------------------------------|
| <code>value</code>               | Il valore in corrispondenza del quale si desidera valutare la distribuzione. Il valore non deve essere negativo. |
| <code>k</code>                   | Un numero positivo che definisce il parametro di forma.                                                          |
| <code><math>\theta</math></code> | Un numero positivo che definisce il parametro di scaling.                                                        |

## GammaDist

`GammaDist()` restituisce la probabilità accumulata della distribuzione Gamma.

### Sintassi:

```
GammaDist(value, k,  $\theta$ )
```

**Tipo di dati restituiti:** numero

#### Argomenti

| Argomento                        | Descrizione                                                                                                      |
|----------------------------------|------------------------------------------------------------------------------------------------------------------|
| <code>value</code>               | Il valore in corrispondenza del quale si desidera valutare la distribuzione. Il valore non deve essere negativo. |
| <code>k</code>                   | Un numero positivo che definisce il parametro di forma.                                                          |
| <code><math>\theta</math></code> | Un numero positivo che definisce il parametro di scaling.                                                        |

Questa funzione è correlata alla funzione `GammaInv` nel modo seguente:

If `prob = GammaDist(value, k,  $\theta$ )`, then `GammaInv(prob, k,  $\theta$ ) = value`

### GammaInv

`GammaInv()` restituisce l'inverso della probabilità cumulativa della distribuzione Gamma.

#### Sintassi:

```
GammaInv(prob, k,  $\theta$ )
```

**Tipo di dati restituiti:** numero

#### Argomenti

| Argomento | Descrizione                                                                                                  |
|-----------|--------------------------------------------------------------------------------------------------------------|
| prob      | Una probabilità associata alla distribuzione di probabilità Gamma. Deve essere un numero compreso tra 0 e 1. |
| k         | Un numero positivo che definisce il parametro di forma.                                                      |
| $\theta$  | Un numero positivo che definisce il parametro di scaling.                                                    |

Questa funzione è correlata alla funzione `GammaDist` nel modo seguente:

If `prob = GammaDist(value, k,  $\theta$ )`, then `GammaInv(prob, k,  $\theta$ ) = value`

### NormDist

`NormDist()` restituisce la distribuzione cumulativa normale per la media e la deviazione standard specificate. Se `mean = 0` e `standard_dev = 1`, la funzione restituisce la distribuzione normale standard.

#### Sintassi:

```
NORMDIST(value, [mean], [standard_dev], [cumulative])
```

**Tipo di dati restituiti:** numero

#### Argomenti:

#### Argomenti

| Argomento    | Descrizione                                                                                                                                          |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| value        | Il valore in corrispondenza del quale si desidera valutare la distribuzione.                                                                         |
| mean         | Valore opzionale che indica la media aritmetica della distribuzione.<br>Se non si specifica questo argomento, il valore predefinito è 0.             |
| standard_dev | Valore positivo opzionale che indica la deviazione standard della distribuzione.<br>Se non si specifica questo argomento, il valore predefinito è 1. |

| Argomento  | Descrizione                                                                                                                                                                                                      |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| cumulative | <p>È possibile scegliere di utilizzare una distribuzione normale standard o una distribuzione cumulativa.</p> <p>0 = distribuzione normale standard</p> <p>1 = distribuzione cumulativa (valore predefinito)</p> |

Questa funzione è correlata alla funzione **NormInv** nel modo seguente:

If prob = NORMDIST(value, m, sd), then NORMINV(prob, m, sd) = value

#### Limiti:

Tutti gli argomenti devono essere numerici, altrimenti viene restituito un valore NULL.

#### Esempi e risultati:

| Esempio              | Risultato          |
|----------------------|--------------------|
| NORMDIST( 0.5, 0, 1) | Restituisce 0,6915 |

## NormInv

**NormInv()** restituisce il valore contrario della distribuzione cumulativa normale per la media e la deviazione standard specificate.

#### Sintassi:

```
NORMINV(prob, mean, standard_dev)
```

**Tipo di dati restituiti:** numero

#### Argomenti:

##### Argomenti

| Argomento    | Descrizione                                                                                     |
|--------------|-------------------------------------------------------------------------------------------------|
| prob         | Una probabilità associata alla distribuzione normale. Deve essere un numero compreso tra 0 e 1. |
| mean         | Un valore che indica la media aritmetica della distribuzione.                                   |
| standard_dev | Un valore positivo che indica la deviazione standard della distribuzione.                       |

Questa funzione è correlata alla funzione **NormDist** nel modo seguente:

If prob = NORMDIST(value, m, sd), then NORMINV(prob, m, sd) = value

#### Limiti:

Tutti gli argomenti devono essere numerici, altrimenti viene restituito un valore NULL.

Esempi e risultati:

| Esempio                    | Risultato          |
|----------------------------|--------------------|
| NORMINV( 0.6914625, 0, 1 ) | Restituisce 0,5000 |

## PoissonDist

PoissonDist() restituisce la probabilità cumulativa della distribuzione Poisson.

**Sintassi:**

```
PoissonDist (value, mean)
```

**Tipo di dati restituiti:** numero

Argomenti

| Argomento | Descrizione                                                                                                      |
|-----------|------------------------------------------------------------------------------------------------------------------|
| value     | Il valore in corrispondenza del quale si desidera valutare la distribuzione. Il valore non deve essere negativo. |
| mean      | Un numero positivo che definisce il risultato medio.                                                             |

Questa funzione è correlata alla funzione PoissonInv nel modo seguente:

If prob = PoissonDist(value, mean), then PoissonInv(prob, mean) = value

## PoissonFrequency

PoissonFrequency() restituisce la distribuzione di probabilità di Poisson.

**Sintassi:**

```
PoissonFrequency (value, mean)
```

**Tipo di dati restituiti:** numero

Argomenti

| Argomento | Descrizione                                                                                                      |
|-----------|------------------------------------------------------------------------------------------------------------------|
| value     | Il valore in corrispondenza del quale si desidera valutare la distribuzione. Il valore non deve essere negativo. |
| mean      | Un numero positivo che definisce il risultato medio.                                                             |

## PoissonInv

PoissonInv() restituisce l'inverso della probabilità cumulativa della distribuzione Poisson.

**Sintassi:**

```
PoissonInv (prob, mean)
```

Tipo di dati restituiti: numero

#### Argomenti

| Argomento | Descrizione                                                                                                    |
|-----------|----------------------------------------------------------------------------------------------------------------|
| prob      | Una probabilità associata alla distribuzione di probabilità Poisson. Deve essere un numero compreso tra 0 e 1. |
| mean      | Un numero positivo che definisce il risultato medio.                                                           |

Questa funzione è correlata alla funzione `PoissonDIST` nel modo seguente:

If `prob = PoissonDist(value, mean)`, then `PoissonInv(prob, mean) = value`

## TDensity

`TDensity()` restituisce il valore della funzione di densità  $t$  per lo studente, dove un valore numerico è un valore calcolato di  $t$  per il quale la probabilità deve essere elaborata.

Sintassi:

```
TDensity(value, degrees_freedom)
```

Tipo di dati restituiti: numero

#### Argomenti

| Argomento       | Descrizione                                                                                                      |
|-----------------|------------------------------------------------------------------------------------------------------------------|
| value           | Il valore in corrispondenza del quale si desidera valutare la distribuzione. Il valore non deve essere negativo. |
| degrees_freedom | Un numero intero positivo corrispondente al numero di gradi di libertà.                                          |

## TDist

`TDist()` restituisce la probabilità per la distribuzione  $t$  dello studente in cui un valore numerico è un valore calcolato di  $t$  per il quale deve essere calcolata la probabilità.

Sintassi:

```
TDist(value, degrees_freedom, tails)
```

**Tipo di dati restituiti:** numero

**Argomenti:**

Argomenti

| Argomento       | Descrizione                                                                                                      |
|-----------------|------------------------------------------------------------------------------------------------------------------|
| value           | Il valore in corrispondenza del quale si desidera valutare la distribuzione. Il valore non deve essere negativo. |
| degrees_freedom | Un numero intero positivo corrispondente al numero di gradi di libertà.                                          |
| tails           | Deve essere 1 (distribuzione a una coda) o 2 (distribuzione a due code).                                         |

Questa funzione è correlata alla funzione **TInv** nel modo seguente:

If prob = TDIST(value, df ,2), then TINV(prob, df) = value

**Limiti:**

Tutti gli argomenti devono essere numerici, altrimenti viene restituito un valore NULL.

**Esempi e risultati:**

| Esempio         | Risultato          |
|-----------------|--------------------|
| TDIST(1, 30, 2) | Restituisce 0,3253 |

## TInv

**TINV()** restituisce il valore  $t$  della distribuzione  $t$  dello studente in funzione della probabilità e dei gradi di libertà.

**Sintassi:**

```
TINV(prob, degrees_freedom)
```

**Tipo di dati restituiti:** numero

**Argomenti:**

Argomenti

| Argomento       | Descrizione                                                                                             |
|-----------------|---------------------------------------------------------------------------------------------------------|
| prob            | Una probabilità a due code associata alla distribuzione $t$ . Deve essere un numero compreso tra 0 e 1. |
| degrees_freedom | Un numero intero corrispondente al numero di gradi di libertà.                                          |

**Limiti:**

Tutti gli argomenti devono essere numerici, altrimenti viene restituito un valore NULL.

Questa funzione è correlata alla funzione **TDist** nel modo seguente:

If prob = TDIST(value, df ,2), then TINV(prob, df) = value.

**Esempi e risultati:**

| Esempio              | Risultato          |
|----------------------|--------------------|
| TINV(0.3253086, 30 ) | Restituisce 1,0000 |

## 8.24 Funzioni di stringa

In questa sezione vengono descritte le funzioni per la gestione e l'elaborazione delle stringhe.

Tutte le funzioni possono essere utilizzate sia nello script di caricamento dei dati che nelle espressioni grafiche, ad eccezione della funzione **Evaluate** che può essere utilizzata solo nello script di caricamento dei dati.

### Panoramica sulle funzioni di stringa

Ciascuna funzione viene descritta ulteriormente dopo la panoramica. È inoltre possibile fare clic sul nome della funzione nella sintassi per accedere immediatamente ai dettagli per tale funzione specifica.

**Capitalize**

La funzione **Capitalize()** converte il primo carattere di ogni parola in una stringa di testo in maiuscolo e converte tutti gli altri caratteri in minuscolo.

**Capitalize** (text)

**Chr**

**Chr()** restituisce il carattere Unicode che corrisponde al numero intero di input, noto come punto del codice.

**Chr** (int)

**CountRegEx**

**CountRegEx()** restituisce il numero di occorrenze del modello dell'espressione regolare specificato nel testo della stringa di input. In mancanza di corrispondenze, viene restituito 0.

**CountRegEx** (text, regex)

**Evaluate**

**Evaluate()** rileva se la stringa di testo di input può essere valutata come espressione di Qlik Sense valida e, in caso affermativo, restituisce il valore dell'espressione come stringa. Se la stringa di input non è un'espressione valida, viene restituito NULL.

**Evaluate** (expression\_text)

### ExtractRegEx

**ExtractRegEx()** estrae il testo da un'espressione di stringa di input utilizzando il modello di espressione regolare specificato. La funzione restituisce un valore null se non vengono trovate corrispondenze.

```
ExtractRegEx (text, regex [, field_no])
```

### ExtractRegExGroup

**ExtractRegExGroup()** estrae il testo dall'espressione di una stringa di input utilizzando il modello di espressione regolare composta specificato. Quando si utilizza la funzione, specificare il gruppo da utilizzare all'interno dell'espressione regolare composta. La funzione restituisce un valore null se non vengono trovate corrispondenze.

```
ExtractRegExGroup (text, regex, group [, field_no])
```

### FindOneOf

**FindOneOf()** ricerca una stringa per individuare la posizione dell'occorrenza di un qualsiasi carattere che fa parte di un set di caratteri forniti. Viene restituita la posizione della n-esima occorrenza di qualsiasi carattere del set di ricerca, dove N è il terzo parametro opzionale della funzione. Se non viene fornito un terzo parametro, viene restituita la prima occorrenza. Se non viene trovata una corrispondenza, viene restituito il valore 0.

```
FindOneOf (text, char_set[, count])
```

### Hash128

**Hash128()** restituisce un hash a 128 bit dei valori dell'espressione di input combinati. Il risultato è una stringa da 22 caratteri.

```
Hash128 (expr{, expression})
```

### Hash160

**Hash160()** restituisce un valore hash a 160 bit dei valori di espressione di input combinati. Il risultato è una stringa da 27 caratteri.

```
Hash160 (expr{, expression})
```

### Hash256

**Hash256()** restituisce un valore hash a 256 bit dei valori di espressione di input combinati. Il risultato è una stringa da 43 caratteri.

```
Hash256 (expr{, expression})
```

### Index

**Index()** ricerca una stringa per individuare la posizione iniziale della n-esima occorrenza di una sottostringa fornita. Un terzo argomento opzionale fornisce il valore di n, che corrisponde a 1 se omissivo. Un valore negativo esegue la ricerca dalla fine della stringa. Le posizioni nella stringa sono numerate da sinistra a destra a partire dal valore 1.

```
Index (text, substring[, count])
```

### IndexRegEx

**IndexRegEx()** effettua una ricerca nella stringa di input e restituisce la posizione iniziale dell'occorrenza n-esima del modello di espressione regolare specificato. Un terzo argomento opzionale **count** fornisce il valore di n, che corrisponde a 1 se omesso. Le posizioni nella stringa sono numerate da sinistra a destra a partire dal valore 1 in poi. Se non viene trovata alcuna corrispondenza, la funzione restituisce 0.

```
IndexRegEx (text, regex [, count])
```

### IndexRegExGroup

**ExtractRegExGroup()** estrae il testo dall'espressione di una stringa di input utilizzando il modello di espressione regolare composta specificato. Quando si utilizza la funzione, specificare il gruppo da utilizzare all'interno dell'espressione regolare composta. La funzione restituisce un valore null se non vengono trovate corrispondenze.

```
IndexRegExGroup (text, regex, group [, count])
```

### IsJson

**IsJson()** verifica se una stringa specificata contiene dati JSON (JavaScript Object Notation) validi. È anche possibile convalidare un tipo specifico di dati JSON.

```
IsJson (json [, type])
```

### IsRegEx

**IsRegEx()** restituisce se il testo specificato è un'espressione regolare valida.

```
IsRegEx (expr [, debug])
```

### JsonGet

**JsonGet()** restituisce il percorso di una stringa dati JSON (JavaScript Object Notation). I dati devono essere dati JSON validi ma possono contenere spazi extra o righe nuove.

```
JsonGet (json, path)
```

### JsonSet

**JsonSet()** modifica una stringa contenente dati JSON (JavaScript Object Notation). Può impostare o inserire un valore JSON con la nuova posizione specificata dal percorso. I dati devono essere dati JSON validi ma possono contenere spazi extra o righe nuove.

```
JsonSet (json, path, value)
```

### KeepChar

**KeepChar()** restituisce una stringa costituita da qualsiasi carattere nella prima stringa che corrisponde ai caratteri nella seconda stringa. Questa funzione è sensibile alle maiuscole e alle minuscole.

```
KeepChar (text, keep_chars)
```

### Left

**Left()** restituisce una stringa costituita dai primi caratteri (posizionati più a sinistra) della stringa di input, in cui il numero di caratteri viene stabilito dal secondo argomento.

```
Left (text, count)
```

### Len

**Len()** restituisce la lunghezza della stringa di input.

```
Len (text)
```

### LevenshteinDist

**LevenshteinDist()** restituisce la distanza Levenshtein tra due stringhe. Viene definita come il numero minimo di modifiche a un singolo carattere (inserimenti, eliminazioni o sostituzioni) richiesto per cambiare una stringa con un'altra. La funzione è utile per i confronti tra stringhe fuzzy.

```
LevenshteinDist (text1, text2)
```

### Lower

**Lower()** applica il formato minuscolo a tutti i caratteri della stringa di input.

```
Lower (text)
```

### LTrim

**LTrim()** restituisce la stringa di input senza spazi iniziali.

```
LTrim (text)
```

### MatchRegex

**MatchRegex()** confronta la stringa di input con uno o più modelli di espressione regolare specificati e restituisce la posizione numerica dei modelli di espressione regolare corrispondenti. Se non vengono trovate corrispondenze, la funzione restituisce 0. Questa funzione cerca solo le corrispondenze esatte.

```
MatchRegex (text, regex1 [ , regex2, ...regexN])
```

### Mid

**Mid()** restituisce la parte della stringa di input che inizia nella posizione del carattere definito dal secondo argomento, 'start', e che restituisce il numero di caratteri definito dal terzo argomento, 'count'. Se viene omesso 'count', viene restituita la parte rimanente della stringa di input. Il primo carattere nella stringa di input viene contrassegnato con il numero 1.

```
Mid (text, start[, count])
```

### Ord

**Ord()** restituisce il valore numerico (ASCII o Unicode) del primo carattere di una stringa. Questa funzione è utile per valutare o confrontare le stringhe in base ai codici dei caratteri sottostanti, ad esempio quando si ordinano o si filtrano stringhe con caratteri non standard.

```
Ord (text)
```

### PurgeChar

**PurgeChar()** restituisce una stringa costituita dai caratteri contenuti nella stringa di input ('text'), ad eccezione di qualsiasi carattere presente nel secondo argomento ('remove\_chars').

```
PurgeChar (text, remove_chars)
```

### Repeat

**Repeat()** crea una stringa costituita dalla stringa di input ripetuta il numero di volte stabilito dal secondo argomento.

```
Repeat (text[, repeat_count])
```

### Replace

**Replace()** restituisce una stringa dopo la sostituzione di tutte le occorrenze di una sottostringa fornita all'interno della stringa di input con un'altra sottostringa. La funzione non è ricorrente e viene applicata da sinistra verso destra.

```
Replace (text, from_str, to_str)
```

### ReplaceRegEx

**ReplaceRegEx()** restituisce una stringa dopo aver sostituito una o più corrispondenze tra una stringa di input e un modello di espressione regolare specificato. Il testo che sostituisce il testo corrispondente è specificato nell'argomento **to\_str**. La funzione viene applicata da sinistra verso destra, ma se si specifica un valore **occurrence** negativo, le funzioni leggono i valori da destra a sinistra.

```
ReplaceRegEx (text, regex, to_str [, occurrence])
```

### ReplaceRegExGroup

**ReplaceRegExGroup()** restituisce una stringa dopo aver sostituito una o più corrispondenze tra una stringa di input e il modello di espressione regolare composta specificato. Il testo che sostituisce il testo corrispondente è specificato nell'argomento **to\_str**. La funzione viene applicata da sinistra verso destra, ma se si specifica un valore **occurrence** negativo, le funzioni leggono i valori da destra a sinistra.

```
ReplaceRegExGroup (text, regex, to_str, group [, occurrence])
```

### Right

**Right()** restituisce una stringa costituita dagli ultimi caratteri (posizionati più a destra) della stringa di input, in cui il numero di caratteri viene stabilito dal secondo argomento.

```
Right (text, count)
```

### RTrim

**RTrim()** restituisce la stringa di input senza spazi finali.

```
RTrim (text)
```

### SubField

**SubField()** consente di estrarre i componenti della sottostringa da un campo della stringa padre, in cui i campi del record originali sono costituiti da due o più parti separate da un delimitatore.

```
SubField (text, delimiter[, field_no ])
```

### SubFieldRegEx

**SubFieldRegEx()** estrae il testo da un'espressione di stringa di input utilizzando il modello di espressione regolare specificato come delimitatore. La funzione restituisce un valore null se non vengono trovate corrispondenze.

```
SubFieldRegEx (text, regex_delimiter [, field_no])
```

### SubStringCount

**SubStringCount()** restituisce il numero di occorrenze della sottostringa specificata nel testo della stringa di input. In mancanza di corrispondenze, viene restituito 0.

```
SubStringCount (text, substring)
```

### TextBetween

**TextBetween()** restituisce il testo nella stringa di input che si trova tra i caratteri specificati come delimitatori.

```
TextBetween (text, delimiter1, delimiter2[, n])
```

### Trim

**Trim()** restituisce la stringa di input senza spazi iniziali e finali.

```
Trim (text)
```

### Upper

**Upper()** applica il carattere maiuscolo a tutti i caratteri della stringa di input per tutti i caratteri di testo nell'espressione. I numeri e i simboli vengono ignorati.

```
Upper (text)
```

## Utilizzo delle espressioni regolari negli script di caricamento e nelle espressioni dei grafici

È possibile migliorare le analisi e la preparazione dei dati utilizzando le espressioni regolari (regex) negli script di caricamento dei dati e nelle espressioni dei grafici. Le espressioni regolari offrono opzioni avanzate, flessibili e potenti per trovare e sostituire i dati basati su testo. Per utilizzare le espressioni regolari in Qlik Sense, utilizzare le [funzioni per script e grafici disponibili](#).

### Come utilizzare le espressioni regolari

Le espressioni regolari (regex) offrono ampie possibilità estese per la corrispondenza avanzata dei modelli. Scrivere un'espressione regolare per specificare un particolare modello di testo che si desidera identificare. Ad esempio, potrebbe essere necessario trovare le occorrenze di indirizzi e-mail o di URL web all'interno di grandi volumi nei corpora in linguaggio naturale di grandi dimensioni.

Esistono molti formati diversi in cui si possono scrivere le espressioni regolari, ognuno con le proprie regole di sintassi. Le funzioni di espressioni regolari disponibili utilizzano la sintassi delle espressioni regolari Perl.

Utilizzare le espressioni regolari in Qlik Sense, utilizzare le funzioni regex disponibili per script e grafici. L'uso di queste funzioni presuppone una conoscenza di base del funzionamento delle espressioni regolari. Per maggiori informazioni, inclusi degli esempi, vedere [Funzioni ed esempi disponibili \(page 1707\)](#).

### Casi d'uso

È possibile utilizzare le espressioni regolari per eseguire le seguenti operazioni:

- Estrarre e separare le informazioni in un testo che contiene più componenti distinti. Ad esempio, indirizzi e-mail, numeri di telefono, URL web e altro ancora.
- Standardizzare la formattazione dei dati.
- Cercare e/o sostituire del testo.

Con le funzioni regex disponibili, è possibile eseguire le seguenti operazioni negli script di caricamento:

- Trasformare i dati e caricarli per utilizzarli nelle analisi.
- Convalidare i dati per assicurarsi che seguano la formattazione e altri standard di conformità.
- Identificare, mascherare o alterare le informazioni di identificazione personale (PII) in una forma appropriata per i consumatori dei contenuti delle proprie analisi.
- Caricare in modo selettivo i dati nelle app in base alla corrispondenza o meno con specifici modelli di testo.

I casi d'uso delle funzioni regex nelle espressioni dei grafici sono simili a quelli degli script di caricamento. L'utilizzo delle funzioni nelle espressioni dei grafici consente anche di eseguire le seguenti azioni:

- Trasformare i dati istantaneamente, in modo condizionale o statico, a seconda delle esigenze e delle interazioni dell'utente.
- Utilizzare le funzioni per definire le etichette, le impostazioni del colore e altre proprietà specifiche del grafico.

### Considerazioni sulle prestazioni

Le funzioni delle espressioni regolari disponibili negli script di caricamento e nelle espressioni dei grafici sono utilizzate al meglio per le trasformazioni avanzate delle stringhe, dove la flessibilità è un requisito fondamentale. L'uso delle funzioni regex richiede più risorse rispetto alle alternative senza regex. Per trasformazioni di dati più semplici, considerare l'utilizzo di altre funzioni script e grafico disponibili, incluse le funzioni stringa. Per un elenco delle funzioni stringa disponibili, consultare [Funzioni di stringa \(page 1701\)](#).

Quando si utilizzano grandi volumi di dati, le operazioni regex per cercare corrispondenze di valori **n** negativi o di grandi dimensioni possono incidere negativamente sulle prestazioni. Nel contesto delle funzioni regex disponibili, il valore **n** corrisponde ad argomenti della funzione come **occurrence**, **count** e **field\_no**.

### Funzioni ed esempi disponibili

- [CountRegEx \(page 1713\)](#)
- [ExtractRegEx \(page 1722\)](#)
- [ExtractRegExGroup \(page 1728\)](#)
- [IndexRegEx \(page 1760\)](#)
- [IndexRegExGroup \(page 1763\)](#)
- [IsRegEx \(page 1772\)](#)

- [MatchRegEx \(page 1806\)](#)
- [ReplaceRegEx \(page 1827\)](#)
- [ReplaceRegExGroup \(page 1833\)](#)
- [SubFieldRegEx \(page 1851\)](#)

## Capitalize

La funzione **Capitalize()** converte il primo carattere di ogni parola in una stringa di testo in maiuscolo e converte tutti gli altri caratteri in minuscolo.

Sintassi:

**Capitalize**(text)

Tipo di dati restituiti: stringa

### Argomenti

| Argomento | Descrizione             |
|-----------|-------------------------|
| text      | La stringa da valutare. |

### Esempio: espressioni del grafico

| Esempio                     | Risultato               |
|-----------------------------|-------------------------|
| Capitalize( 'star trek' )   | Restituisce star Trek   |
| Capitalize( 'AA bb cC dD' ) | Restituisce Aa Bb Cc Dd |

## Esempio: principi fondamentali della funzione Capitalize

Espressione del grafico

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata Example.
- Un campo della tabella dati denominata TextToCapitalize.

### Script di caricamento

```
Example:
Load * inline [
TextToCapitalize
50 Cent
a-ha
BIGBANG
girl in red
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- TextToCapitalize

Creare la seguente dimensione calcolata:

- =Capitalize(TextToCapitalize), per calcolare i caratteri che si desidera rendere in maiuscolo.

Tabella dei risultati

| TextToCapitalize | Capitalize(TextToCapitalize) |
|------------------|------------------------------|
| 50 Cent          | 50 Cent                      |
| a-ha             | A-Ha                         |
| BIGBANG          | Bigbang                      |
| girl in red      | Girl In Red                  |

Non c'è alcuna modifica alla prima riga di dati, poiché il primo set di caratteri inizia con un numero e il secondo è già in maiuscolo.

Nella seconda riga, il primo e il terzo carattere sono in maiuscolo.

Nella terza riga, solo la prima lettera è in maiuscolo e tutte le lettere successive sono impostate come minuscole.

Nella quarta riga, la prima lettera di ogni parola è in maiuscolo.

### Script di caricamento

Il codice seguente mostra come utilizzare la funzione in uno script di caricamento.

```
Load
String,
Capitalize(String)
Inline
[String
rHode iSland
washingTon d.C.
new york];
```

Tabella dei risultati

| Stringa         | Capitalize(String) |
|-----------------|--------------------|
| rHode iSland    | Rhode Island       |
| washingTon d.C. | Washington D.C.    |
| new york        | New York           |

### Esempio: scenario per l'applicazione della funzione Capitalize

Espressione del grafico

#### Panoramica

Un set di dati di nomi di clienti viene inserito utilizzando vari formati, come tutte minuscole, tutte maiuscole o lettere maiuscole e minuscole. La funzione `capitalize` viene utilizzata per pulire i dati in modo che il nome e il cognome del cliente siano correttamente in maiuscolo quando vengono visualizzati nel dashboard.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Un campo della tabella dati denominata `CustomerName`.

#### Script di caricamento

```
Example:
Load * inline [
CustomerName
john smith
JANE DOE
miCHAEl Brown
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `CustomerName`

Creare la seguente dimensione calcolata:

- `=Capitalize(CustomerName)`, per scrivere in maiuscolo i nomi dei clienti.

Tabella dei risultati

| CustomerName  | Capitalize(CustomerName) |
|---------------|--------------------------|
| JANE DOE      | Jane Doe                 |
| john smith    | John Smith               |
| miCHAEl Brown | Michael Brown            |

Confrontare l'output della funzione `capitalize` con i valori delle stringhe originali di `CustomerName` caricate nello script. È possibile vedere come la funzione abbia convertito con successo i nomi per utilizzare le maiuscole e le minuscole in modo corretto. La visualizzazione dei nomi dei clienti con l'uso di maiuscole e minuscole standardizzato migliora l'aspetto, la coerenza e la leggibilità del dashboard.

## Chr

**Chr()** restituisce il carattere Unicode che corrisponde al numero intero di input, noto come punto del codice.

### Sintassi:

```
Chr (int)
```

**Tipo di dati restituiti:** stringa

#### Argomenti

| Argomento | Descrizione                   |
|-----------|-------------------------------|
| int       | Il numero intero da valutare. |

#### Esempio: espressioni del grafico

| Esempio  | Risultato                |
|----------|--------------------------|
| Chr(65)  | Restituisce la stringa A |
| Chr(163) | Restituisce la stringa £ |
| Chr(35)  | Restituisce la stringa # |

## Esempio: principi fondamentali della funzione Chr

Espressione del grafico

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Un campo della tabella dati denominata `InputInt`. Il campo contiene il valore intero da valutare.

### Script di caricamento

Example:

```
Load * inline [  
InputInt  
65  
163  
35  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- InputInt

Creare la seguente misura:

- =Chr(InputInt), per calcolare i valori in InputInt e restituire un carattere Unicode.

Tabella dei risultati

| InputInt | Chr(InputInt) |
|----------|---------------|
| 35       | #             |
| 65       | A             |
| 163      | £             |

La prima riga restituisce la stringa #, in quanto si tratta del carattere Unicode che corrisponde al valore 35.

La seconda riga restituisce la stringa A, in quanto si tratta del carattere Unicode che corrisponde al valore 65.

La terza riga restituisce la stringa £, in quanto si tratta del carattere Unicode che corrisponde al valore 163.

### Esempio: scenario per l'applicazione della funzione Chr

Espressione del grafico

#### Panoramica

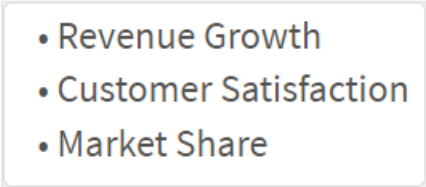
Questo esempio mostra come utilizzare la funzione chr per creare un elenco puntato delle stringhe di testo in un oggetto di testo.

#### Risultati

Aprire un foglio. Creare un nuovo oggetto di testo, quindi aggiungere il seguente codice a una misura:

- = Chr(149) & ' Revenue Growth' & Chr(10) & Chr(149) & ' Customer Satisfaction' & Chr(10) & Chr(149) & ' Market Share'

*L'output dell'espressione della misura nell'oggetto di testo*

- 
- Revenue Growth
  - Customer Satisfaction
  - Market Share

L'output viene ricavato nel modo seguente:

- Chr(149) restituisce un punto elenco (•).
- Chr(10) inserisce un'interruzione di riga (new line) tra gli elementi.
- La e commerciale (&) è un operatore di concatenazione delle stringhe che concatena i singoli frammenti di testo in un'unica stringa di testo.

## CountRegEx

**CountRegEx()** restituisce il numero di occorrenze del modello dell'espressione regolare specificato nel testo della stringa di input. In mancanza di corrispondenze, viene restituito 0.

Questa funzione esegue operazioni regex che distinguono tra maiuscole e minuscole. In alternativa, è possibile utilizzare la variante **CountRegExI()** per eseguire operazioni regex senza distinzione tra maiuscole e minuscole.

### Sintassi:

```
CountRegEx (text, regex)
```

**Tipo di dati restituiti:** numerico

### Argomenti

| Argomento    | Descrizione                                                                         |
|--------------|-------------------------------------------------------------------------------------|
| <b>text</b>  | Il testo della stringa di input in cui si desidera cercare un'espressione regolare. |
| <b>regex</b> | Il modello di espressione regolare su cui si desidera eseguire una ricerca.         |

### Esempi di funzioni

| Esempio                             | Risultato                                                                                                                                       |
|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| CountRegEx ('abc123', '[a-z]')      | Restituisce 3.                                                                                                                                  |
| CountRegEx ('abc123', '[a-z][0-9]') | Restituisce 1.                                                                                                                                  |
| CountRegEx ('abc123', '[0-9]')      | Restituisce 3.                                                                                                                                  |
| CountRegEx('ABC', '[a-z]{3}')       | Restituisce 0.                                                                                                                                  |
| CountRegExI('ABC', '[a-z]{3}')      | Restituisce 1. Poiché viene utilizzata la variante <b>CountRegExI()</b> , le ricerche di testo non fanno distinzione tra maiuscole e minuscole. |

## Casi di utilizzo

Gli esempi di casi d'uso per **CountRegEx()** includono:

- Il conteggio del numero di volte in cui un determinato modello di testo ricorre all'interno di una stringa. Ad esempio, è possibile cercare le istanze di indirizzi e-mail, numeri di telefono e altre informazioni nei documenti e nei messaggi e-mail.
- La convalida se i dati corrispondono a un requisito di sintassi specifico. Ad esempio, è possibile identificare i casi di errore nell'inserimento dei dati.

### Esempio 1 - script di caricamento per contare gli intervalli di anni validi

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Una scheda denominata `vehicles` contenente un elenco di veicoli, con le condizioni e l'anno del modello di ciascun veicolo.
- Per ogni record in `vehicles`, viene effettuato un calcolo per determinare se l'intervallo di anni per il veicolo soddisfa la sintassi prevista. Gli intervalli di anni validi includono gli anni '90, 2000, 2010 e 2020. Questo calcolo viene eseguito con la funzione **CountRegex()**.

#### Script di caricamento

```
Vehicles:
Load CountRegex(Model_YearRange, '^19[0-9][0-9]s$|^20[0-9][0-9]s$') as Valid_Year_Data,
* Inline [
Vehicle_ID, Condition, Model_YearRange
1, Good,20100s
2,Fair,2020s
3, Poor,2020s
4, Good,210s
5, Excellent,2000s
6, Excellent,2020s
7, Excellent,202s
8, Good,2010s
9, Poor,2010ss
10, Poor,1990s
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `Vehicle_ID`
- `Valid_Year_Data`

Tabella dei risultati

| Vehicle_ID | Valid_Year_Data |
|------------|-----------------|
| 1          | 0               |
| 2          | 1               |
| 3          | 1               |

| Vehicle_ID | Valid_Year_Data |
|------------|-----------------|
| 4          | 0               |
| 5          | 1               |
| 6          | 1               |
| 7          | 0               |
| 8          | 1               |
| 9          | 0               |
| 10         | 1               |

È possibile anche aggiungere un KPI al foglio e aggiungere questa misura:

`Sum(Valid_Year_Data)`

Questo grafico mostra il numero totale di record con valori *Model\_YearRange* validi.



La funzione **MatchRegEx()** è utile anche per questi tipi di casi d'uso di convalida.

### Esempio 2 - script di caricamento per contare il numero di indirizzi e-mail nelle stringhe di testo

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Una tabella denominata `Employees` contenente i dati dei dipendenti di un'azienda. Tutti gli indirizzi e-mail di ciascun dipendente sono contenuti in un'unica stringa.
- La creazione di un campo, `NoOfEmails`, per calcolare il numero di indirizzi e-mail contenuti nella stringa di input per ogni dipendente.

#### Script di caricamento

`Employees:`

```
Load CountRegEx(Emails, '@[a-zA-Z0-9.-]*\.[a-zA-Z0-9]*\.com') as NoOfEmails,
* Inline `
ID      Emails
1      john_doe@test.com john_michael_doe@example.com j.doe123@example123.com
2      jane_doe@test.com
3      Michael_Andrews@test.com Michael_John_Andrews@example.com
```

```
4      charlie_kent@support.test.com charlie_peter_kent@support.example.com
cp.kent.1985@test123.com c_p_kent@test123.com
5      frances_johnson@test.com frances_johnson@example.com
6      rebecca_smith@hr.test.com
` (delimiter is '\t');
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- ID
- Emails
- NoOfEmails

Tabella dei risultati

| ID | Emails                                                                                                                | NoOfEmails |
|----|-----------------------------------------------------------------------------------------------------------------------|------------|
| 1  | john_doe@test.com john_michael_doe@example.com<br>j.doe123@example123.com                                             | 3          |
| 2  | jane_doe@test.com                                                                                                     | 1          |
| 3  | Michael_Andrews@test.com Michael_John_Andrews@example.com                                                             | 2          |
| 4  | charlie_kent@support.test.com charlie_peter_kent@support.example.com<br>cp.kent.1985@test123.com c_p_kent@test123.com | 4          |
| 5  | frances_johnson@test.com frances_johnson@example.com                                                                  | 2          |
| 6  | rebecca_smith@hr.test.com                                                                                             | 1          |

### Esempio 3 - espressione del grafico per contare le occorrenze della stringa

Espressione del grafico e risultati

#### Panoramica

Aprire Editor caricamento dati, quindi aggiungere il seguente script di caricamento in una nuova sezione.

Lo script di caricamento contiene il testo di una serie di corrispondenze aziendali inviate ai dipendenti di un'organizzazione. I dati sono già stati puliti e non contengono informazioni di identificazione personale (PII), a parte il nome dell'azienda e l'indirizzo e-mail che ha inviato ogni corrispondenza. Il corpo di ogni messaggio è stato compresso in un'unica stringa continua. In altre parole, sono stati rimossi tutti i caratteri di formattazione e di interruzione di riga.

Requisiti:

- Identificare quali messaggi contengono una stringa che include l'indirizzo e-mail *TestCompanyNameCorrespondence@test.com*, oltre a un'introduzione specifica a questo indirizzo che indica che si tratta del mittente.

- Per quanto riguarda le introduzioni da identificare, è necessaria una certa flessibilità, ma anche disporre di restrizioni precise. I valori delle introduzioni `sender` e `sent by` sono entrambi considerati validi. Per raggiungere questo obiettivo, si possono utilizzare i modelli regex.
- La distinzione tra maiuscole e minuscole per gli indirizzi e-mail non è un fattore. Pertanto, la ricerca non deve distinguere tra maiuscole e minuscole.

### Script di caricamento

Correspondences:

```
Load * Inline [
```

```
ID|MessageBody
```

```
1|Sent by-- TestCompanyNameCorrespondence@test.com Good morning TestCompanyName team! I hope you are all well. I am just reaching out about the fabulous benefits package we launched just year. Our goal is to take a poll of what you all think. Can you please rate the new options from 1-10 in a response to this email, by this Tuesday? Thank you!
```

```
2|Sender: TestCompanyNameIT@test.com Did you know you can access our IT portal anytime, from any of your company devices? It's true! Reach out to your team lead to learn more.
```

```
3|Sender: TESTCOMPANYNAMECORRESPONDENCE@test.COM URGENT: This is a message to inform employees of an ongoing issue with our local branch. The branch will be closed until further notice. Thank you
```

```
4|Sender: TestCompanyNameVolunteering@test.com Dear team! We are looking for volunteers to help with this year's charity event! We need 40 volunteers to help with the event. If you are interested, please send an email to your direct supervisor. Thanks, TestCompanyName team members, for helping make the world a better place!
```

```
5|Sender is TestCompanyNameCorrespondence@test.com Hi folks, it's time to announce the annual TestCompanyName staff party! The party will be taking place at the office this year. Please make sure to RSVP, and we hope to see you there! Sincerely, the TestCompanyName management team
```

```
] (delimiter is |);
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- ID
- MessageBody

Aggiungere la misura seguente:

```
=CountRegExI(MessageBody, '(Sender.*TestCompanyNameCorrespondence@test.com)|(Sent by.*TestCompanyNameCorrespondence@test.com)')
```

Nelle proprietà della misura, impostare la **funzione Totali** su **Sum**. Questa configura la riga dei totali nella parte superiore della tabella per contare il numero totale di record che contribuiscono al conteggio.

La misura conta il numero di corrispondenze per il modello regex specificato all'interno del contenuto del messaggio. Nel modello regex, `.` indica un numero qualsiasi di qualunque carattere, ad eccezione dei caratteri di interruzione di riga. Il modello tiene conto anche delle variazioni nel modo in cui viene introdotto l'indirizzo e-mail: sia `sender` che `sent by` sono corrispondenze valide. La variante **CountRegExI()** della funzione assicura la ricerca senza distinzione tra maiuscole e minuscole.

Tabella dei risultati

| ID     | MessageBody                                                                                                                                                                                                                                                                                                                                                   | =CountRegExI(MessageBody,'(Sender.*TestCompanyNameCorrespondence@test.com) '(Sent by.*TestCompanyNameCorrespondence@test.com)') |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| Totals | -                                                                                                                                                                                                                                                                                                                                                             | 3                                                                                                                               |
| 1      | Sent by--<br>TestCompanyNameCorrespondence@test.com<br>Good morning TestCompanyName team! I hope you are all well. I am just reaching out about the fabulous benefits package we launched just year. Our goal is to take a poll of what you all think. Can you please rate the new options from 1-10 in a response to this email, by this Tuesday? Thank you! | 1                                                                                                                               |
| 2      | Sender: TestCompanyNameIT@test.com Did you know you can access our IT portal anytime, from any of your company devices? It's true! Reach out to your team lead to learn more.                                                                                                                                                                                 | 0                                                                                                                               |
| 3      | Sender: TESTCOMPANYNAMECORRESPONDENCE@TEST.COM URGENT: This is a message to inform employees of an ongoing issue with our local branch. The branch will be closed until further notice. Thank you                                                                                                                                                             | 1                                                                                                                               |
| 4      | Sender: TestCompanyNameVolunteering@test.com Dear team! We are looking for volunteers to help with this year's charity event! We need 40 volunteers to help with the event. If you are interested, please send an email to your direct supervisor. Thanks, TestCompanyName team members, for helping make the world a better place!                           | 0                                                                                                                               |
| 5      | Sender is<br>TestCompanyNameCorrespondence@test.com<br>Hi folks, it's time to announce the annual TestCompanyName staff party! The party will be taking place at the office this year. Please make sure to RSVP, and we hope to see you there!<br>Sincerely, the TestCompanyName management team                                                              | 1                                                                                                                               |

Tra i cinque messaggi, ce ne sono tre che corrispondono al modello regex specificato. Ogni messaggio contiene una corrispondenza.



La funzione **MatchRegEx()** è utile anche per questi tipi di casi d'uso di convalida.

### Evaluate

**Evaluate()** rileva se la stringa di testo di input può essere valutata come espressione di Qlik Sense valida e, in caso affermativo, restituisce il valore dell'espressione come stringa. Se la stringa di input non è un'espressione valida, viene restituito NULL.

Sintassi:

```
Evaluate(expression_text)
```

Tipo di dati restituiti: duale

#### Argomenti

| Argomento | Descrizione             |
|-----------|-------------------------|
| text      | La stringa da valutare. |



La funzione stringa non può essere utilizzata nelle espressioni grafiche.

#### Esempio di funzione e risultati

| Esempio di funzione | Risultato      |
|---------------------|----------------|
| Evaluate( 5 * 8 )   | Restituisce 40 |

### Esempio: principi fondamentali della funzione Evaluate

Script di caricamento

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata Example.
- Un campo della tabella dati denominata ExpressionText.
- Un caricamento precedente che utilizza la funzione Evaluate per valutare il campo ExpressionText e restituire due nuovi campi: Evaluated e Evaluatedx4.

### Script di caricamento

Example:

```
Load
ExpressionText
, Evaluate(ExpressionText) as Evaluated
, Evaluate(ExpressionText)*4 as Evaluatedx4
;
Load * Inline
[ExpressionText
4
5+3
100/5
123*2
329-9
0123456789012345678
1234567890123456789
0123456.5512
0123456.5512479994578952364859346469
Today()
Bob
];
```

### Risultati

I seguenti campi vengono caricati nella tabella dati:

- ExpressionText
- Evaluated
- Evaluatedx4

Results table

| ExpressionText      | Evaluated           | Evaluatedx4         |
|---------------------|---------------------|---------------------|
| 4                   | 4                   | 16                  |
| 5+3                 | 8                   | 32                  |
| 100/5               | 20                  | 80                  |
| 123*2               | 246                 | 984                 |
| 329-9               | 320                 | 1280                |
| 0123456789012345678 | 0123456789012345678 | 4.9382715604938e+17 |
| 1234567890123456789 | 1234567890123456789 | 4.9382715604938e+18 |
| 0123456.5512        | 0123456.5512        | 493826.2048         |

| ExpressionText                       | Evaluated                            | Evaluatedx4   |
|--------------------------------------|--------------------------------------|---------------|
| 0123456.5512479994578952364859346469 | 0123456.5512479994578952364859346469 | 493826.204992 |
| Today()                              | 11/5/2024                            | 182404        |
| Bob                                  | -                                    | -             |

L'output della funzione `Evaluate` restituisce i valori di tutte le stringhe `ExpressionText` tranne l'ultima riga. La stringa `bob` non può essere valutata, quindi la funzione restituisce un valore nullo, `NULL`. Tutte le righe sono state valutate, ad esempio la seconda riga aggiunge 5 e 3 per restituire 8. Alcune stringhe `ExpressionText` restituiscono la stringa così com'è, perché la stringa è già valutata come un numero. Ad esempio, la stringa `0123456789012345678` valuta lo stesso numero.

Sebbene Qlik Sense abbia un limite di precisione di 14 cifre per i valori numerici, la funzione `Evaluate` ha valutato le stringhe di 18 cifre come un numero e ha applicato la moltiplicazione all'interno dello script. Questa opzione è utile per gestire numeri molto grandi negli script.

## Esempio: scenario per l'applicazione della funzione Evaluate

### Script di caricamento

#### Panoramica

Un set di dati sulle vendite contiene prodotti, prezzi e sconti. In questo esempio si mostra come ricavare il prezzo scontato dei prodotti.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Product`
  - `Price`
  - `Discount %`
- Un carico precedente che prende i valori `Price` e `Discount %` e calcola un campo aggiuntivo denominato `DiscountedPrice`. Questo nuovo campo utilizza la funzione `Evaluate` per calcolare il prezzo meno lo sconto.

### Script di caricamento

Example:

```
LOAD
Product,
Price,
[Discount (%)],
Price * (1 - Evaluate([Discount (%)] / 100)) AS DiscountedPrice
INLINE [
```

```
Product, Price, Discount (%)
Laptop, 1000, 15
Smartphone, 800, 10
Tablet, 600, 20
Headphones, 200, 5
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- Product
- Price
- Discount %
- DiscountedPrice

| Results table |       |            |                 |
|---------------|-------|------------|-----------------|
| Product       | Price | Discount % | DiscountedPrice |
| Headphones    | 200   | 5          | 190             |
| Tablet        | 600   | 20         | 480             |
| Smartphone    | 800   | 10         | 720             |
| Laptop        | 1000  | 15         | 850             |

L'output della funzione `Evaluate` dimostra come è possibile utilizzare la funzione per calcolare i valori.

## ExtractRegEx

**ExtractRegEx()** estrae il testo da un'espressione di stringa di input utilizzando il modello di espressione regolare specificato. La funzione restituisce un valore null se non vengono trovate corrispondenze.

Questa funzione esegue operazioni regex che distinguono tra maiuscole e minuscole. In alternativa, è possibile utilizzare la variante **ExtractRegExI()** per eseguire operazioni regex senza distinzione tra maiuscole e minuscole.

### Sintassi:

```
ExtractRegEx (text, regex [, field_no])
```

**Tipo di dati restituiti:** stringa

| Argomenti   |                                                                              |
|-------------|------------------------------------------------------------------------------|
| Argomento   | Descrizione                                                                  |
| <b>text</b> | L'espressione stringa contenente il testo da estrarre nel valore restituito. |

| Argomento       | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>regex</b>    | L'espressione regolare da utilizzare per estrarre il testo.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>field_no</b> | <p>Il numero della corrispondenza da estrarre. Questo è utile quando nel testo si possono trovare più corrispondenze per l'espressione regolare. Per esempio, specificare il valore 4 per estrarre la quarta corrispondenza. Questo argomento è facoltativo. Specificare un valore negativo per invertire l'ordine delle corrispondenze.</p> <p>Le seguenti indicazioni si applicano indipendentemente dal fatto che la funzione sia utilizzata in uno script di caricamento o in un'espressione del grafico:</p> <ul style="list-style-type: none"> <li>Se <b>field_no</b> è un valore positivo, la funzione restituirà un valore, identificando le corrispondenze da sinistra a destra.</li> <li>Se <b>field_no</b> è un valore negativo, la funzione restituirà un valore, identificando le corrispondenze da destra a sinistra.</li> </ul> <p>Quando si utilizza la funzione in uno script di caricamento:</p> <ul style="list-style-type: none"> <li>Se si utilizza la funzione <b>ExtractRegEx()</b> in un'istruzione <b>LOAD</b> e il valore <b>field_no</b> viene omissso, la funzione genera tanti record quante sono le corrispondenze.</li> <li>Se diversi campi vengono caricati utilizzando <b>ExtractRegEx()</b> e nessuno di essi specifica un argomento <b>field_no</b>, vengono creati i prodotti cartesiani per tutte le combinazioni.</li> </ul> <p>Quando utilizzare la funzione in un'espressione del grafico:</p> <ul style="list-style-type: none"> <li>Se <b>field_no</b> viene omissso, il valore predefinito è 1.</li> </ul> |

## Esempi di funzioni

| Esempio                                                   | Risultato                                                                                                                                                     |
|-----------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>ExtractRegEx('a,b c;1 2,3','[a-z] [0-9]', 1)</code> | Restituisce a, che è la prima corrispondenza trovata.                                                                                                         |
| <code>ExtractRegEx('a,b c;1 2,3','[a-z] [0-9]', 2)</code> | Restituisce b, che è la seconda corrispondenza trovata.                                                                                                       |
| <code>ExtractRegEx('a,b c;1 2,3','[a-z] [0-9]', 4)</code> | Restituisce 1, che è la quarta corrispondenza trovata.                                                                                                        |
| <code>ExtractRegEx('abc','a b',-1)</code>                 | Restituisce b. Quando viene specificato un valore negativo, le corrispondenze vengono conteggiate alla rovescia.                                              |
| <code>ExtractRegEx('A,B C;1 2,3','[a-c] [4-9]')</code>    | Restituisce un valore null, perché <b>ExtractRegEx()</b> fa distinzione tra maiuscole e minuscole, e non è stata utilizzata la variante che non le distingue. |
| <code>ExtractRegExI('A,B C;1 2,3','[a-c] [4-9]')</code>   | Restituisce A, perché è stata utilizzata la variante <b>ExtractRegExI()</b> che non fa distinzione tra maiuscole e minuscole.                                 |

### Casi di utilizzo

È possibile utilizzare **Extract RegEx()** per estrarre le informazioni che si desidera isolare dai dati che possono contenere anche altre informazioni (ad esempio, testo libero o stringhe JSON). Ad esempio:

- Estrarre indirizzi e-mail, numeri di telefono, numeri di conto e altre informazioni da un testo.
- Estrarre valori numerici da un testo (ad esempio, la valuta).
- Standardizzare la formattazione dei dati numerici o di testo.

### Esempio 1 - script di caricamento per estrarre gli indirizzi e-mail da JSON

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Una tabella `orderDetails` contenente un campo `orderJson`.
- Il campo `orderJson` contiene una stringa JSON con i dati dell'ordine. La formattazione, come l'indentazione e le interruzioni di riga, è stata rimossa dal JSON. I dati dell'ordine contengono indirizzi e-mail che bisogna estrarre in valori separati.
- Una nuova tabella `EmailAddresses`, che viene caricata dopo `orderDetails`. Nell'istruzione `EmailAddresses LOAD`, è necessario caricare il campo `orderJson` dalla tabella `orderDetails` e creare un nuovo campo calcolato, `Email`, che contiene gli indirizzi e-mail estratti. Il valore `orderDetails` viene quindi eliminato.

Requisiti:

- Indirizzi e-mail in un dominio specifico: `example.com`.
- Indirizzi e-mail che contengono un numero qualsiasi di caratteri alfanumerici, oltre a una serie definita di caratteri speciali.

#### Script di caricamento

`orderDetails:`

```
Load * Inline `
orderJson
{"Orders":[{"order_id":"12345","customer":{"name":"John
Doe","email":"john.doe@example.com"},"items":
{"product":"Laptop","quantity":2,"price":1200},"total_price":2400},{ "order_
id":"12346","customer":{"name":"Jane Doe","email":"jane.doe@example.com"},"items":
{"product":"Tablet","quantity":4,"price":400},"total_price":1600},{ "order_
id":"12347","customer":{"name":"Amalia Craig","email":"amalia.craig@example.com"},"items":
{"product":"Desktop Computer","quantity":1,"price":1900},"total_price":1900},{ "order_
id":"12348","customer":{"name":"Ken Roberts","email":"ken.roberts@example.com"},"items":
{"product":"Phone Charger Cable","quantity":110,"price":1900},"total_price":1210}}]
` (delimiter is '\t');
```

EmailAddresses:

```
Load OrderJson, ExtractRegEx(OrderJson, '[a-zA-Z0-9!#$%^&*~_+=~{|}\./\']+'@example.com') as  
Email Resident OrderDetails;
```

```
Drop table OrderDetails;
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- Email

Tabella dei risultati

| Email                    |
|--------------------------|
| amalia.craig@example.com |
| jane.doe@example.com     |
| john.doe@example.com     |
| ken.roberts@example.com  |

Se non si specifica un valore per l'argomento di **field\_no**, tutte le corrispondenze vengono estratte automaticamente in valori separati.

### Esempio 2 - script di caricamento per estrarre i valori numerici dagli importi in valuta

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Una tabella `sales` contenente i dati delle vendite.
- All'interno della tabella `sales`, un campo denominato `Amount`, che contiene gli importi delle transazioni in USD. Questi importi sono espressi come stringhe, che includono simboli e codici di valuta.
- La definizione di un campo personalizzato all'interno di `sales`, `OrderAmountNumeric_USD`, che estrae ogni importo in un valore puramente numerico. Questi dati devono essere in forma numerica, in modo da poterli utilizzare nelle aggregazioni dei grafici.

#### Script di caricamento

```
SET DateFormat='MM-DD-YYYY';
```

Sales:

```
Load ExtractRegEx(Amount, '[0-9]+.[0-9]{2}') as OrderAmountNumeric_USD, * Inline `
```

```
OrderID, OrderDate, Product, Amount
1, '12-27-2024', Software, $100.31 USD
2, '12-28-2024', Electronics, $513.93 USD
3, '12-29-2024', Grocery, $34.66 USD
4, '12-30-2024', Furniture, $1124.58 USD
5, '12-31-2024', Software, $95.95 USD
6, '01-01-2025', Software, $601.00 USD
7, '01-02-2025', Grocery, $1534.20 USD
8, '01-03-2025', Produce, $51.49 USD
9, '01-04-2025', Home Improvement, $176.12 USD
10, '01-05-2025', Produce, $432.77 USD
11, '01-06-2025', Software, $270.01 USD
12, '01-07-2025', Electronics, $83.85 USD
13, '01-08-2025', Furniture, $65.65 USD
14, '01-09-2025', Electronics, $1526.39 USD
15, '01-10-2025', Software, $432.44 USD
16, '01-11-2025', Furniture, $888.15 USD
17, '01-12-2025', Furniture, $300.00 USD
18, '01-13-2025', Grocery, $109.53 USD
19, '01-14-2025', Furniture, $487.97 USD
20, '01-15-2025', Software, $65.09 USD
`;
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- OrderID
- Amount
- OrderAmountNumeric\_USD

Tabella dei risultati

| OrderID | Amount        | OrderAmountNumeric_USD |
|---------|---------------|------------------------|
| 1       | \$100.31 USD  | 100.31                 |
| 2       | \$513.93 USD  | 513.93                 |
| 3       | \$34.66 USD   | 34.66                  |
| 4       | \$1124.58 USD | 1124.58                |
| 5       | \$95.95 USD   | 95.95                  |
| 6       | \$601.00 USD  | 601.00                 |
| 7       | \$1534.20 USD | 1534.20                |
| 8       | \$51.49 USD   | 51.49                  |

| OrderID | Amount        | OrderAmountNumeric_USD |
|---------|---------------|------------------------|
| 9       | \$176.12 USD  | 176.12                 |
| 10      | \$432.77 USD  | 432.77                 |
| 11      | \$270.01 USD  | 270.01                 |
| 12      | \$83.85 USD   | 83.85                  |
| 13      | \$65.65 USD   | 65.65                  |
| 14      | \$1526.39 USD | 1526.39                |
| 15      | \$432.44 USD  | 432.44                 |
| 16      | \$888.15 USD  | 888.15                 |
| 17      | \$300.00 USD  | 300.00                 |
| 18      | \$109.53 USD  | 109.53                 |
| 19      | \$487.97 USD  | 487.97                 |
| 20      | \$65.09 USD   | 65.09                  |

### Esempio 3 - espressioni dei grafici per estrarre gli indirizzi e-mail da JSON

Espressione del grafico e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Una tabella `orderDetails` contenente un campo `orderJson`.
- Il campo `orderJson` contiene una stringa JSON con i dati dell'ordine. La formattazione, come l'indentazione e le interruzioni di riga, è stata rimossa dal JSON. I dati dell'ordine contengono indirizzi e-mail che bisogna estrarre in valori separati.

Requisiti:

- Indirizzi e-mail in un dominio specifico: `example.com`.
- Indirizzi e-mail che contengono un numero qualsiasi di caratteri alfanumerici, oltre a una serie definita di caratteri speciali.

#### Script di caricamento

`OrderDetails:`

```
Load * Inline `
OrderJson
{"orders":[{"order_id":"12345","customer":{"name":"John
Doe","email":"john.doe@example.com"},"items":
```

```
{
  "product": "Laptop", "quantity": 2, "price": 1200, "total_price": 2400, {"order_id": "12346", "customer": {"name": "Jane Doe", "email": "jane.doe@example.com"}, "items": [{"product": "Tablet", "quantity": 4, "price": 400, "total_price": 1600}, {"order_id": "12347", "customer": {"name": "Amalia Craig", "email": "amalia.craig@example.com"}, "items": [{"product": "Desktop Computer", "quantity": 1, "price": 1900, "total_price": 1900}, {"order_id": "12348", "customer": {"name": "Ken Roberts", "email": "ken.roberts@example.com"}, "items": [{"product": "Phone Charger Cable", "quantity": 110, "price": 1900, "total_price": 1210}]}]}
` (delimiter is '\t');
```

## Risultati

Caricare i dati e aprire un foglio. Creare una tabella e aggiungere ognuna delle seguenti espressioni come dimensioni calcolate:

- =ExtractRegEx(OrderJson, '[a-zA-Z0-9!#\$%^&\*~\_+=~{|}\./.'']+@example.com', 1)
- =ExtractRegEx(OrderJson, '[a-zA-Z0-9!#\$%^&\*~\_+=~{|}\./.'']+@example.com', 2)
- =ExtractRegEx(OrderJson, '[a-zA-Z0-9!#\$%^&\*~\_+=~{|}\./.'']+@example.com', 3)
- =ExtractRegEx(OrderJson, '[a-zA-Z0-9!#\$%^&\*~\_+=~{|}\./.'']+@example.com', 4)

Tabella dei risultati

| =ExtractRegEx<br>(OrderJson, '[a-zA-Z0-9!#\$%^&*~_+=~{ }\./.'']+@example.com', 1) | =ExtractRegEx<br>(OrderJson, '[a-zA-Z0-9!#\$%^&*~_+=~{ }\./.'']+@example.com', 2) | =ExtractRegEx<br>(OrderJson, '[a-zA-Z0-9!#\$%^&*~_+=~{ }\./.'']+@example.com', 3) | =ExtractRegEx<br>(OrderJson, '[a-zA-Z0-9!#\$%^&*~_+=~{ }\./.'']+@example.com', 4) |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|
| john.doe@example.com                                                              | jane.doe@example.com                                                              | amalia.craig@example.com                                                          | ken.roberts@example.com                                                           |

## ExtractRegExGroup

**ExtractRegExGroup()** estrae il testo dall'espressione di una stringa di input utilizzando il modello di espressione regolare composita specificato. Quando si utilizza la funzione, specificare il gruppo da utilizzare all'interno dell'espressione regolare composita. La funzione restituisce un valore null se non vengono trovate corrispondenze.

Se si utilizza la funzione **ExtractRegExGroup()** in un'istruzione **LOAD** e il valore **field\_no** viene omesso, la funzione restituirà più record. Se diversi campi vengono caricati utilizzando **ExtractRegExGroup()**, vengono creati i prodotti cartesiani di tutte le combinazioni.

Questa funzione esegue operazioni regex che distinguono tra maiuscole e minuscole. In alternativa, è possibile utilizzare la variante **ExtractRegExGroupI()** per eseguire operazioni regex senza distinzione tra maiuscole e minuscole.

### Sintassi:

```
ExtractRegExGroup (text, regex, group [, field_no])
```

Tipo di dati restituiti: stringa

#### Argomenti

| Argomento       | Descrizione                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>text</b>     | L'espressione stringa contenente il testo da estrarre nel valore restituito.                                                                                                                                                                                                                                                                                                                        |
| <b>regex</b>    | L'espressione regolare da utilizzare per estrarre il testo.                                                                                                                                                                                                                                                                                                                                         |
| <b>group</b>    | <p>Il numero del gruppo, nel caso di un'espressione regolare composta.</p> <p>Se l'espressione regolare contiene solo un gruppo, utilizzare invece la funzione <b>ExtractRegEx()</b>. In alternativa, utilizzare <b>ExtractRegExGroup()</b> con un valore <b>group</b> di 0.</p> <p>È possibile specificare un valore <b>group</b> negativo per cercare le corrispondenze da destra a sinistra.</p> |
| <b>field_no</b> | <p>Il numero della corrispondenza da estrarre. Questo è utile quando nel testo si possono trovare più corrispondenze per l'espressione regolare. Per esempio, specificare il valore 4 per estrarre la quarta corrispondenza.</p> <p>Questo argomento è facoltativo. Se non viene specificata, l'impostazione predefinita è 1.</p>                                                                   |

#### Esempi di funzioni

| Esempio                                                                   | Risultato                                                      |
|---------------------------------------------------------------------------|----------------------------------------------------------------|
| <code>ExtractRegExGroup('abc123 def456', '([a-z]+)([0-9]+)', 1)</code>    | Restituisce abc (primo gruppo della prima corrispondenza).     |
| <code>ExtractRegExGroup('abc123 def456', '([a-z]+)([0-9]+)', 1, 2)</code> | Restituisce def (primo gruppo della seconda corrispondenza).   |
| <code>ExtractRegExGroup('abc123 def456', '([a-z]+)([0-9]+)', 2)</code>    | Restituisce 123 (secondo gruppo della prima corrispondenza).   |
| <code>ExtractRegExGroup('abc123 def456', '([a-z]+)([0-9]+)', 2, 2)</code> | Restituisce 456 (secondo gruppo della seconda corrispondenza). |

### Casi di utilizzo

È possibile utilizzare **Extract RegEx()** per estrarre le informazioni che si desidera isolare dai dati che possono contenere anche altre informazioni (ad esempio, testo libero o stringhe JSON). Ad esempio:

- Estrarre indirizzi e-mail, numeri di telefono, numeri di conto e altre informazioni da un testo.
- Estrarre valori numerici da un testo (ad esempio, la valuta).
- Standardizzare la formattazione dei dati numerici o di testo.

### Esempio 1 - script di caricamento per analizzare i codici delle transazioni

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Una tabella denominata `Transactions`, che contiene informazioni sulle transazioni. Alcuni dettagli vengono acquisiti con i codici delle transazioni utilizzando la seguente sintassi:  
*Anno della transazione-Origine della transazione* (cioè acquisto online o in negozio)-*Centro di distribuzione associato*
- La creazione di diversi nuovi campi per estrarre ogni dettaglio, ognuno dei quali riutilizza la stessa espressione regolare composita.

Requisiti:

- L'anno può essere una qualsiasi combinazione di quattro numeri.
- I valori *ONLINE* e *INSTORE* sono gli unici accettabili per l'origine della transazione.
- Il centro di distribuzione deve avere esattamente cinque numeri.

#### Script di caricamento

Transactions:

Load

```
    recno() as RecordID,
    ExtractRegExGroup(TransactionCode, '([0-9]{4})-(ONLINE|INSTORE)-([0-9]{5})', 0) as
TransactionCode_Unparsed,
    ExtractRegExGroup(TransactionCode, '([0-9]{4})-(ONLINE|INSTORE)-([0-9]{5})', 1) as
TransactionYear,
    ExtractRegExGroup(TransactionCode, '([0-9]{4})-(ONLINE|INSTORE)-([0-9]{5})', 2) as
TransactionSource,
    ExtractRegExGroup(TransactionCode, '([0-9]{4})-(ONLINE|INSTORE)-([0-9]{5})', 3) as
TransactionDC,
* Inline `
TransactionCode, Category
2025-ONLINE-60019, Product A
2024-INSTORE-60020, Product B
2025-ONLINE-60018, Product C
2024-ONLINE-60020, Product A
2025-INSTORE-60019, Product B
2025-ONLINE-60017, Product D
`;
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- RecordID
- TransactionCode
- TransactionCode\_Unparsed
- TransactionYear
- TransactionSource
- TransactionDC

Tabella dei risultati

| RecordID | TransactionCode    | TransactionCode_Unparsed | TransactionYear | TransactionSource | TransactionDC |
|----------|--------------------|--------------------------|-----------------|-------------------|---------------|
| 1        | 2025-ONLINE-60019  | 2025-ONLINE-60019        | 2025            | ONLINE            | 60019         |
| 2        | 2024-INSTORE-60020 | 2024-INSTORE-60020       | 2024            | INSTORE           | 60020         |
| 3        | 2025-ONLINE-60018  | 2025-ONLINE-60018        | 2025            | ONLINE            | 60018         |
| 4        | 2024-ONLINE-60020  | 2024-ONLINE-60020        | 2024            | ONLINE            | 60020         |
| 5        | 2025-INSTORE-60019 | 2025-INSTORE-60019       | 2025            | INSTORE           | 60019         |
| 6        | 2025-ONLINE-60017  | 2025-ONLINE-60017        | 2025            | ONLINE            | 60017         |

Questi risultati evidenziano come l'argomento **group** permette di riutilizzare una singola espressione regex per più operazioni. Il campo `TransactionCode_Unparsed`, che utilizza un valore **group** di 0, non fornisce alcun valore aggiuntivo in questo caso, ma viene mostrato qui per dimostrare la funzione.

### Esempio 2 - script di caricamento per estrarre le informazioni di contatto dell'azienda

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Una tabella denominata `businessContactInfo`, che contiene informazioni sulle aziende trovate sul sito web di ciascuna azienda.

- `ContactInfo` è un campo che contiene testo libero, compresi gli indirizzi e-mail e i numeri di telefono dell'azienda.
- La creazione di diversi nuovi campi per estrarre ogni dettaglio, ognuno dei quali riutilizza la stessa espressione regolare composita.

Requisiti:

- Gli indirizzi e-mail devono soddisfare una serie di requisiti e di sintassi specifici.
- I numeri di telefono devono seguire il sistema NANP (piano di numerazione nordamericano), con 10 cifre. Bisogna consentire che il prefisso sia racchiuso tra parentesi, oltre alla presenza di spazi vuoti e trattini in alcuni punti.

### Script di caricamento

`BusinessContactInfo:`

Load

```
ExtractRegExGroupI(ContactInfo, '([a-zA-Z0-9!#$%^&*~_+=~{|}\./.'')+@[a-zA-Z0-9!#$%^&*~_+=~{|}\./.'']{1,50}\.[a-zA-Z0-9!#$%^&*~_+=~{|}\./.'']{1,50})|(\{0,1}[0-9]{3}\){0,1}[-]*[0-9]{3}[-]*[0-9]{4})',1,1) as CompanyEmail,
ExtractRegExGroupI(ContactInfo, '([a-zA-Z0-9!#$%^&*~_+=~{|}\./.'')+@[a-zA-Z0-9!#$%^&*~_+=~{|}\./.'']{1,50}\.[a-zA-Z0-9!#$%^&*~_+=~{|}\./.'']{1,50})|(\{0,1}[0-9]{3}\){0,1}[-]*[0-9]{3}[-]*[0-9]{4})',2,2) as CompanyPhoneNum,
* Inline `
ID      CompanyName      ContactInfo
1      Company A      Email is: Company1@example.com, Phone number is: (123) 456-7890
2      Company B      Email is: company2@test.com, Phone # is: 0123456790
3      Company C      Email is: company3@placeholder.com, Phone no. is: 234-567-8901

` (delimiter is '\t');
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `CompanyName`
- `CompanyEmail`
- `CompanyPhoneNum`

Tabella dei risultati

| CompanyName | CompanyEmail             | CompanyPhoneNum |
|-------------|--------------------------|-----------------|
| Company A   | Company1@example.com     | (123) 456-7890  |
| Company B   | company2@test.com        | 0123456790      |
| Company C   | company3@placeholder.com | 234-567-8901    |

La stessa espressione regolare composita viene riutilizzata per recuperare informazioni diverse. L'argomento **group** specifica quale dei due gruppi dell'espressione regolare ricercare, e l'argomento **field\_no** specifica quale la corrispondenza complessiva (nell'intera stringa) che si desidera trovare.

La variante **ExtractRegEGroupxl()** della funzione assicura la ricerca senza distinzione tra maiuscole e minuscole.

### Esempio 3 - script di caricamento per estrarre i componenti dai codici ISBN

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- La creazione di una variabile, `ISBN_RegEx`, per memorizzare l'espressione regolare che si desidera utilizzare per tutte le estrazioni.
- Un set di codici ISBN per i libri che una libreria vuole ordinare da diversi fornitori.

Bisogna estrarre i seguenti valori da ciascun codice ISBN:

- EAN
- Gruppo
- Soggetto registrato
- Pubblicazione
- Checksum

#### Script di caricamento

```
SET ISBN_RegEx = 'ISBN[ ]*([0-9]{3})-([0-9]{1})-([0-9]{4})-([0-9]{4})-([0-9]{1})';

ISBN:
LOAD Supplier,
ExtractRegExGroup(Books,'$(ISBN_RegEx)',1) AS EAN,
ExtractRegExGroup(Books,'$(ISBN_RegEx)',2) AS Group,
ExtractRegExGroup(Books,'$(ISBN_RegEx)',3) AS Registrant,
ExtractRegExGroup(Books,'$(ISBN_RegEx)',4) AS Publication,
ExtractRegExGroup(Books,'$(ISBN_RegEx)',5) AS Checksum;
// Split the ISBN with the Group function in a preceding load to avoid generating a cartesian
product
LOAD *,
ExtractRegEx(SupplierBooks, '$(ISBN_RegEx)') AS Books
INLINE [
Supplier, SupplierBooks
Supplier 1, ISBN 123-3-1234-1234-0 ISBN 012-2-0123-0123-4 ISBN 000-1-0123-0123-2 ISBN 234-5-
2345-2345-1 ISBN 555-2-5555-5555-3 ISBN 222-4-2222-2222-2
Supplier 2, ISBN 000-0-3333-3333-3 ISBN 333-3-3333-3333-3 ISBN 555-1-5151-5151-3 ISBN 232-1-
2323-2323-1 ISBN 008-0-7777-7777-3 ISBN 888-0-9999-0000-0
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- Supplier
- EAN
- Group
- Registrant
- Publication
- Checksum

Tabella dei risultati

| Supplier   | EAN | Group | Registrant | Publication | Checksum |
|------------|-----|-------|------------|-------------|----------|
| Supplier 1 | 000 | 1     | 0123       | 0123        | 2        |
| Supplier 1 | 012 | 2     | 0123       | 0123        | 4        |
| Supplier 1 | 123 | 3     | 1234       | 1234        | 0        |
| Supplier 1 | 222 | 4     | 2222       | 2222        | 2        |
| Supplier 1 | 234 | 5     | 2345       | 2345        | 1        |
| Supplier 1 | 555 | 2     | 5555       | 5555        | 3        |
| Supplier 2 | 000 | 0     | 3333       | 3333        | 3        |
| Supplier 2 | 008 | 0     | 7777       | 7777        | 3        |
| Supplier 2 | 232 | 1     | 2323       | 2323        | 1        |
| Supplier 2 | 333 | 3     | 3333       | 3333        | 3        |
| Supplier 2 | 555 | 1     | 5151       | 5151        | 3        |
| Supplier 2 | 888 | 0     | 9999       | 0000        | 0        |

### Esempio 4 - espressioni dei grafici per estrarre informazioni sui contatti aziendali (con confronto con **ExtractRegEx()** ).

Espressione del grafico e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Una tabella denominata `BusinessContactInfo`, che contiene informazioni sulle aziende trovate sul sito web di ciascuna azienda.
- `ContactInfo` è un campo che contiene testo libero, compresi gli indirizzi e-mail e i numeri di telefono dell'azienda. Bisogna estrarre ogni indirizzo e-mail e numero di telefono con le espressioni dei grafici.

Requisiti:

- ## Script di caricamento

```
Load * Inline `
ID      CompanyName      ContactInfo
1      Company A      Email is: Company1@example.com, Phone number is: (123) 456-7890
2      Company B      Email is: company2@test.com, Phone # is: 0123456790
3      Company C      Email is: company3@placeholder.com, Phone no. is: 234-567-8901

` (delimiter is '\t');
```

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- Aggiungere la seguente dimensione calcolata alla tabella:

- ## Tabella dei risultati

La stessa espressione regolare composita viene riutilizzata per recuperare informazioni diverse. L'argomento **group** specifica quale dei due gruppi dell'espressione regolare ricercare, e l'argomento **field\_** **no** specifica quale la corrispondenza complessiva (nell'intera stringa) che si desidera trovare.

Sintassi dello script e funzioni grafiche - Qlik Sense, November 2025

### Esempio 5 - Analisi dell'URL (con confronto con **ExtractRegEx()** )

Espressione del grafico e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Una tabella denominata `correspondence`, che contiene il testo del messaggio e-mail in un campo denominato `EmailBody`.
- Contenuto del messaggio e-mail che contiene URL web.

Bisogna estrarre il secondo URL, se presente, nei seguenti componenti:

- URL completo
- Protocollo
- Dominio
- Percorso

Bisogna utilizzare un'unica espressione regolare per controllare tutte le estrazioni.

#### Script di caricamento

`Correspondence:`

```
Load * Inline `
ID      EmailBody
1      Thanks again for this morning's meeting! You can find the meeting minutes posted here:
https://example.com/resourceexample. If you still have any questions, always feel free to ask
me or one of the other team members. Here are a few learning resources that might help you:
http://www.example.ca/training1.pptx http://www.example.ca/training2.pptx
http://www.example.ca/training3.pptx Thanks again!
2      Hi, you'll want to visit our company website for that, it's available at
https://www.example.se.
3      Hello all, I just wanted to let you know that our online stores are now up and running! I
couldn't be more excited. We are already seeing quite a bit of traffic and volume sold, which
is very promising! For Product A, go to https://www.examplestore1.com/products. For Product B,
you'll want go to https://www.examplestore2.com/products. Product C, go check out
https://www.examplestore3.com/products. Cheers!

` (delimiter is '\t');
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- ID

Aggiungere le seguenti dimensioni calcolate:

- **URL 2:**  
=ExtractRegEx(EmailBody, '([a-zA-Z0-9]+):\\/(\\/([a-zA-Z0-9]+\\.)\*([a-zA-Z0-9]+)\\.([a-zA-Z0-9]+))((\\/([a-zA-Z0-9]+)\*\\/([a-zA-Z0-9]+)\\.([a-zA-Z0-9]+)?))?' , 2)
- **Protocollo URL 2:**  
=ExtractRegExGroup(EmailBody, '([a-zA-Z0-9]+):\\/(\\/([a-zA-Z0-9]+\\.)\*([a-zA-Z0-9]+)\\.([a-zA-Z0-9]+))((\\/([a-zA-Z0-9]+)\*\\/([a-zA-Z0-9]+)\\.([a-zA-Z0-9]+)?))?' , 1, 2)
- **Dominio URL 2:**  
=ExtractRegExGroup(EmailBody, '([a-zA-Z0-9]+):\\/(\\/([a-zA-Z0-9]+\\.)\*([a-zA-Z0-9]+)\\.([a-zA-Z0-9]+))((\\/([a-zA-Z0-9]+)\*\\/([a-zA-Z0-9]+)\\.([a-zA-Z0-9]+)?))?' , 2, 2)
- **Percorso URL 2:**  
=ExtractRegExGroup(EmailBody, '([a-zA-Z0-9]+):\\/(\\/([a-zA-Z0-9]+\\.)\*([a-zA-Z0-9]+)\\.([a-zA-Z0-9]+))((\\/([a-zA-Z0-9]+)\*\\/([a-zA-Z0-9]+)\\.([a-zA-Z0-9]+)?))?' , 6, 2)

Tabella dei risultati

| ID | URL 2                                  | Protocollo URL 2 | Dominio URL 2         | Percorso URL 2  |
|----|----------------------------------------|------------------|-----------------------|-----------------|
| 1  | http://www.example.ca/training1.pptx   | http             | www.example.ca        | /training1.pptx |
| 2  | -                                      | -                | -                     | -               |
| 3  | https://www.examplestore2.com/products | https            | www.examplestore2.com | /products       |

**ExtractRegEx()** restituisce l'intero URL, mentre **ExtractRegExGroup()** restituisce singole parti dell'URL, corrispondenti al valore **group** che è stato utilizzato. Il record con un valore per ID uguale a 2 contiene solo un URL, quindi non viene restituito alcun dato.

La stessa espressione regolare viene utilizzata in tutte le espressioni dei grafici per restituire diverse informazioni. La scomposizione dei gruppi definiti nell'espressione regolare è la seguente.

Gruppi nell'espressione regolare

| Numero del gruppo | Regex                                                    | Parte URL                                  |
|-------------------|----------------------------------------------------------|--------------------------------------------|
| 1                 | ([a-zA-Z0-9]+)                                           | Protocollo                                 |
| 2                 | ((([a-zA-Z0-9]+\\.)*([a-zA-Z0-9]+)\\.([a-zA-Z0-9]+))     | Dominio (gruppi 3, 4 e 5)                  |
| 3                 | ([a-zA-Z0-9]+\\.)*                                       | Dominio radice                             |
| 4                 | ([a-zA-Z0-9]+)                                           | Dominio di primo livello                   |
| 5                 | ([a-zA-Z0-9]+)                                           | Percorso (gruppi 7, 8, 9 e 10) (opzionale) |
| 6                 | ((\\([a-zA-Z0-9]+)*\\([a-zA-Z0-9]+)\\.([a-zA-Z0-9]+)?))? | Sottodirectory                             |
| 7                 | (\\([a-zA-Z0-9]+)*                                       | File (gruppi 9 e 10)                       |

| Numero del gruppo | Regex                               | Parte URL                       |
|-------------------|-------------------------------------|---------------------------------|
| 8                 | (\V([a-zA-Z0-9]+)(\.[a-zA-Z0-9]+)?) | File                            |
| 9                 | ([a-zA-Z0-9]+)                      | Nome file                       |
| 10                | (\.[a-zA-Z0-9]+)?                   | Estensione del file (opzionale) |

## FindOneOf

**FindOneOf()** ricerca una stringa per individuare la posizione dell'occorrenza di un qualsiasi carattere che fa parte di un set di caratteri forniti. Viene restituita la posizione della n-esima occorrenza di qualsiasi carattere del set di ricerca, dove N è il terzo parametro opzionale della funzione. Se non viene fornito un terzo parametro, viene restituita la prima occorrenza. Se non viene trovata una corrispondenza, viene restituito il valore 0.

### Sintassi:

```
FindOneOf(text, char_set[, count])
```

**Tipo di dati restituiti:** numero intero

### Argomenti

| Argomento       | Descrizione                                                                                                                |
|-----------------|----------------------------------------------------------------------------------------------------------------------------|
| <b>text</b>     | La stringa originale.                                                                                                      |
| <b>char_set</b> | Un set di caratteri da ricercare in <b>text</b> .                                                                          |
| <b>count</b>    | Definisce l'occorrenza di qualsiasi carattere da ricercare. Ad esempio, un valore di 2 ricerche per la seconda occorrenza. |

### Esempio: espressioni del grafico

| Esempio                                                       | Risultato                                                                                                                                                                       |
|---------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>FindOneOf( 'my example text string', 'et%s' )</code>    | Restituisce 4 perché e corrisponde al quarto carattere nella stringa di esempio.                                                                                                |
| <code>FindOneOf( 'my example text string', 'et%s', 3 )</code> | Restituisce 12 perché la ricerca viene eseguita per uno qualsiasi dei caratteri e, t, % o s, e t corrisponde alla terza occorrenza nella posizione 12 della stringa di esempio. |
| <code>FindOneOf( 'my example text string', 'x%&amp;' )</code> | Restituisce 0 perché nessuno dei caratteri x, % o & esiste nella stringa di esempio.                                                                                            |

### Esempio: principi fondamentali della funzione FindOneOf

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Un campo nella tabella dati chiamato `Phrase`, che contiene le stringhe di testo originali da elaborare.

#### Script di caricamento

```
Example:
Load *
Inline
[Phrase
Many tiny beads
For a very long time
Has the potential for growth
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `Phrase`

Creare le seguenti dimensioni calcolate:

- `=FindOneOf(Phrase, 'e', 1)`
- `=FindOneOf(Phrase, 'y', 1)`
- `=FindOneOf(Phrase, 'ey', 1)`
- `=FindOneOf(Phrase, 'e', 2)`
- `=FindOneOf(Phrase, 'y', 2)`
- `=FindOneOf(Phrase, 'ey', 2)`

Tabella dei risultati

| Phrase                  | FindOneOf<br>(Phrase, 'e',<br>1) | FindOneOf<br>(Phrase, 'y',<br>1) | FindOneOf<br>(Phrase,<br>'ey', 1) | FindOneOf<br>(Phrase, 'e',<br>2) | FindOneOf<br>(Phrase, 'y',<br>2) | FindOneOf<br>(Phrase,<br>'ey', 2) |
|-------------------------|----------------------------------|----------------------------------|-----------------------------------|----------------------------------|----------------------------------|-----------------------------------|
| For a very<br>long time | 8                                | 10                               | 8                                 | 20                               | 0                                | 10                                |

| Phrase                             | FindOneOf<br>(Phrase, 'e',<br>1) | FindOneOf<br>(Phrase, 'y',<br>1) | FindOneOf<br>(Phrase,<br>'ey', 1) | FindOneOf<br>(Phrase, 'e',<br>2) | FindOneOf<br>(Phrase, 'y',<br>2) | FindOneOf<br>(Phrase,<br>'ey', 2) |
|------------------------------------|----------------------------------|----------------------------------|-----------------------------------|----------------------------------|----------------------------------|-----------------------------------|
| Has the<br>potential<br>for growth | 7                                | 0                                | 7                                 | 12                               | 0                                | 12                                |
| Many tiny<br>beads                 | 12                               | 4                                | 4                                 | 0                                | 9                                | 9                                 |

Per ogni dimensione calcolata, l'output restituisce la posizione dell'occorrenza dei caratteri del set di ricerca per ogni frase. Ad esempio, la prima occorrenza della lettera e si trova nelle posizioni 8, 7 e 12 per ogni frase, rispettivamente. La seconda occorrenza della lettera e in ogni frase si trova nelle posizioni 20, 12 e 0 (non trovata). In modo simile, la prima occorrenza delle lettere "e o y" si trova nelle posizioni 8, 7 e 4, per ogni frase, rispettivamente. Notare che nella frase `For a very long time`, la prima occorrenza di "e o y" si trova nella posizione 8, una e, mentre la seconda occorrenza è nella posizione 10, una y.

### Script di caricamento

Il codice seguente mostra come utilizzare la funzione in uno script di caricamento.

Example:

```
Load *, FindOneOf(InputText, SearchFor, Occurrence) AS FindOneOf_Matches
Inline
InputText, SearchFor, Occurrence
my example text string, et%s,1
my example text string, et%s,3
my example text string, æ%&,1
];
```

Tabella dei risultati

| Testo di input              | Cerca | Occorrenza | Corrispondenze di FindOneOf |
|-----------------------------|-------|------------|-----------------------------|
| esempio di stringa di testo | et%s  | 1          | 4                           |
| esempio di stringa di testo | et%s  | 3          | 12                          |
| esempio di stringa di testo | æ%&   | 1          | 0                           |

## Esempio: scenario per l'applicazione della funzione FindOneOf

Espressione del grafico

### Panoramica

In questo esempio si mostra come utilizzare la funzione `FindOneOf` per analizzare i commenti dei clienti in un set di dati di ordini dei clienti e identificare gli ordini che possono richiedere un'azione. Ogni ordine dispone di un campo `customerComment`, dove i clienti possono lasciare note o commenti sui propri ordini. Analizzando i commenti, è possibile identificare parole chiave o caratteri specifici, come "!", "@", "#", che potrebbero indicare urgenza o richieste speciali.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `TicketID` il numero di identità del ticket
  - La stringa di testo originale di `CustomerComment`

### Script di caricamento

Example:

```
Load * inline [  
TicketID, CustomerComment  
1, I need this order ASAP!  
2, Please confirm my order @12345.  
3, Can you update my order?  
4, I have a question about #discount.  
5, Thank you!  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `TicketID`
- `CustomerComment`

Creare le seguenti dimensioni calcolate:

- `=FindOneOf(CustomerComment, '!,@,#')`
- `=If(FindOneOf(CustomerComment, '!,@,#')>0,'High Priority','Normal Priority')`

Tabella dei risultati

| TicketID | CommentoCliente                                 | FindOneOf<br>(CustomerComment,<br>'!,@,#') | If(FindOneOf(CustomerComment,<br>'!,@,#')>0,'Alta priorità','Priorità<br>normale') |
|----------|-------------------------------------------------|--------------------------------------------|------------------------------------------------------------------------------------|
| 1        | Ho bisogno di questo ordine il prima possibile! | 23                                         | Priorità alta                                                                      |
| 2        | La prego di confermare il mio ordine @12345.    | 25                                         | Priorità alta                                                                      |
| 3        | Può aggiornare il mio ordine?                   | 0                                          | Priorità normale                                                                   |
| 4        | Ho una domanda sullo #sconto.                   | 25                                         | Priorità alta                                                                      |
| 5        | Grazie!                                         | 10                                         | Priorità alta                                                                      |

Confrontare l'output della funzione `Findoneof` con i valori delle stringhe originali di `CustomerComment` caricate nello script. Utilizzando i valori `!`, `@`, `#`, la funzione ha restituito con successo la posizione in cui si trovano questi caratteri nella stringa di commento. La misura finale utilizza un'istruzione condizionale `If` per identificare i record che includono uno qualsiasi dei caratteri di ricerca e assegnarli come `High Priority`. Se non viene trovato nessuno dei caratteri di ricerca `!`, `@`, `#`, ad esempio nella riga 3, il ticket viene classificato come `Normal Priority`.

### Hash128

**Hash128()** restituisce un hash a 128 bit dei valori dell'espressione di input combinati. Il risultato è una stringa da 22 caratteri. I valori hash sono utili per mascherare le informazioni personali identificabili (PII), come i nomi dei clienti, i numeri di previdenza sociale o i numeri di conto.

#### Sintassi:

```
Hash128(expr{, expression})
```

Tipo di dati restituiti: stringa

| Argomenti         |                         |
|-------------------|-------------------------|
| Argomento         | Descrizione             |
| <code>expr</code> | La stringa da valutare. |

Esempio: espressioni del grafico

| Esempio                                         | Risultato                                                |
|-------------------------------------------------|----------------------------------------------------------|
| <code>Hash128( 'abc', 'xyz', '123' )</code>     | Restituisce <code>MA&amp;5]6+3=:&gt;:G%S&lt;U*S2+</code> |
| <code>Hash128 ( Region, Year, Month )</code>    | Restituisce <code>G7*=6GKPJ(Z+)^KM?&lt;\$'A+</code>      |
| Note: Region, Year, and Month are table fields. |                                                          |

### Esempio: principi fondamentali della funzione Hash128

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi saranno utilizzati per creare un valore hash a 128 bit:
  - `Region`
  - `Year`
  - `Month`

### Script di caricamento

```
Example:
Load *
inline [
Region, Year, Month
abc, xyz, 123
EU, 2022, 01
UK, 2022, 02
US, 2022, 02
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- Region
- Year
- Month

Creare la seguente dimensione calcolata:

- =Hash128(Region, Year, Month), per calcolare il valore hash a 128 bit.

Tabella dei risultati

| Area geografica | Anno | Mese | Hash128(Region, Year, Month) |
|-----------------|------|------|------------------------------|
| abc             | xyz  | 123  | MA&5]6+3=:>;>G%S<U*S2+       |
| EU              | 2022 | 01   | B40^K&[T@!;VB'XR]<5=/\$      |
| UK              | 2022 | 02   | O5T;+1?[B&"F&1//MA[MN!       |
| US              | 2022 | 02   | C6@#]4#_G-(J7EQY#KRW0        |

Per ogni riga, le stringhe hash per region, year e month vengono unite e viene restituita una stringa di 22 caratteri.

### Script di caricamento

Il codice seguente mostra come utilizzare la funzione in uno script di caricamento.

```
Hash_128:
Load *,
Hash128(Region, Year, Month) as Hash128_Region_Year_Month;
Load * inline [
Region, Year, Month
abc, xyz, 123
EU, 2022, 01
UK, 2022, 02
US, 2022, 02 ];
```

Tabella dei risultati

| Area geografica | Anno | Mese | Hash128_Region_Year_Month |
|-----------------|------|------|---------------------------|
| abc             | xyz  | 123  | MA&5]6+3=:>;>G%S<U*S2+    |
| EU              | 2022 | 01   | B40^K&[T@!;VB'XR]<5=/\$   |
| UK              | 2022 | 02   | O5T;+1?[B&"F&1//MA[MN!    |
| US              | 2022 | 02   | C6@#]4#_G-(J7EQY#KRW0     |

### Esempio: scenario per l'applicazione della funzione Hash128 con voci duplicate

Espressione del grafico

#### Panoramica

In questo esempio si mostra come creare un valore hash univoco per ogni voce del prodotto. Il valore hash funge da identificatore univoco, consentendo all'azienda di individuare e gestire efficacemente i duplicati.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `TransactionID`
  - `ProductID`
  - `ProductName`
  - `SupplierID`

#### Script di caricamento

```
Example:
Load *
inline [
TransactionID, ProductID, ProductName, SupplierID
1, 101, widget A, S001, 100
2, 102, widget B, S002, 200
3, 101, widget A, S001, 50
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `Transaction ID`
- `ProductID`

- `ProductName`
- `SupplierID`

Creare la seguente dimensione calcolata:

- `=Hash128(ProductID, ProductName, SupplierID)`, per calcolare una chiave di 22 caratteri.

Tabella dei risultati

| TransactionID | ProductID | ProductName | SupplierID | Hash128(ProductID, ProductName, SupplierID) |
|---------------|-----------|-------------|------------|---------------------------------------------|
| 1             | 101       | Widget A    | S001       | CY`&^(N,E/#`TJT4&J/7N%                      |
| 2             | 102       | Widget B    | S002       | O7(1;V1%MG;Z+A[/H)G5&0                      |
| 3             | 101       | Widget A    | S001       | CY`&^(N,E/#`TJT4&J/7N%                      |

Confrontare l'output della funzione `hash128`. Notare che il primo e il terzo record di `TransactionID` hanno valori hash duplicati perché la combinazione delle stringhe `ProductID`, `ProductName` e `SupplierID` è la stessa per queste righe.

### Esempio: scenario per l'applicazione della funzione Hash128 per l'anonimizzazione dei dati

Espressione del grafico

#### Panoramica

Una società di servizi finanziari gestisce le informazioni sensibili sui clienti, come carte di credito, numeri di previdenza sociale e i nomi dei clienti. Per proteggere la privacy e garantire la conformità a normative come il General Data Protection Regulation (GDPR) o l'Health Insurance Portability and Accountability Act (HIPAA), hanno bisogno di rendere anonimi i dati sensibili quando generano report per l'analisi. Tuttavia, devono anche mantenere la coerenza, in modo da poter rintracciare lo stesso cliente in diversi set di dati senza rivelarne l'identità.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `CustomerID`
  - `CustomerName`
  - `SSN`

### Script di caricamento

```
Example:
Load *
inline [
CustomerID, CustomerName, SSN
101, John Doe, 123-45-6789|
102, Jane Smith, 987-65-4321
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- CustomerID
- CustomerName
- SSN

Creare le seguenti dimensioni calcolate:

- =Hash128(CustomerName)
- =Hash128(SSN)

Tabella dei risultati

| CustomerID | CustomerName | SSN         | Hash128<br>(CustomerName)   | Hash128(SSN)                |
|------------|--------------|-------------|-----------------------------|-----------------------------|
| 101        | John Doe     | 123-45-6789 | GMD<br>(*B9*!PQ&MTY2@\$4A\+ | N&<br>[PSH:!\$M?OF<5/9LV&N# |
| 102        | Jane Smith   | 987-65-4321 | EP@9]GSI)5G\*,B?H,C?)       | HCFXMR'2=\3WF6?_<br>\6LN;\$ |

L'output genera chiavi univoche che presentano i seguenti vantaggi:

- Privacy dei dati: i dati sensibili originali non sono direttamente visibili, ma il valore hash rimane coerente tra i diversi record. Ciò consente di effettuare analisi, come l'aggregazione delle attività dei clienti, senza esporre l'identità effettiva del cliente.
- Conformità: grazie all'hashing dei campi sensibili, l'azienda garantisce la conformità alle normative sulla privacy dei dati.
- Sicurezza: la funzione hash produce un output di dimensioni fisse che non è reversibile, aggiungendo un ulteriore livello di protezione per i dati sensibili.

## Hash160

**Hash160()** restituisce un valore hash a 160 bit dei valori di espressione di input combinati. Il risultato è una stringa da 27 caratteri. I valori hash sono utili per mascherare le informazioni personali identificabili (PII), come i nomi dei clienti, i numeri di previdenza sociale o i numeri

di conto.

#### Sintassi:

```
Hash160(expr{, expression})
```

Tipo di dati restituiti: stringa

| Argomenti |                         |
|-----------|-------------------------|
| Argomento | Descrizione             |
| expr      | La stringa da valutare. |

#### Esempio: espressioni del grafico

| Esempio                                                                               | Risultato                                     |
|---------------------------------------------------------------------------------------|-----------------------------------------------|
| Hash160( 'abc', 'xyz', '123' )                                                        | Restituisce<br>MA&5]6+3=:;>G%S<U*S2I: `=X*    |
| Hash160( Region, Year, Month )<br><br>Note: Region, Year, and Month are table fields. | Restituisce G7*=6GKPJ<br>(Z+)^KM?<\$'AI.)?U\$ |

### Esempio: principi fondamentali della funzione Hash160

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata Example.
- I seguenti campi saranno utilizzati per creare un valore hash a 160 bit:
  - Region
  - Year
  - Month

#### Script di caricamento

Example:

```
Load *
inline [
Region, Year, Month
abc, xyz, 123
EU, 2022, 01
UK, 2022, 02
US, 2022, 02
];
```

## Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- Region
- Year
- Month

Creare la seguente dimensione calcolata:

- =Hash160(Region, Year, Month), per calcolare il valore hash a 160 bit.

Tabella dei risultati

| Area geografica | Anno | Mese | Hash160(Region, Year, Month) |
|-----------------|------|------|------------------------------|
| abc             | xyz  | 123  | MA&5]6+3=:>;>G%S<U*S2I:.`=X* |
| EU              | 2022 | 01   | B40^K&[T@!;VB'XR]<5=//_F853  |
| UK              | 2022 | 02   | O5T;+1?[B&"F&1//MA[MN!T"FWZ  |
| US              | 2022 | 02   | C6@#]4#_G-(J7EQY#KRW`@KF+W   |

Per ogni riga, le stringhe hash per Region, Year e Month vengono unite e viene restituita una stringa di 27 caratteri.

## Script di caricamento

Il codice seguente mostra come utilizzare la funzione in uno script di caricamento.

```
Hash_160:
Load *,
Hash160(Region, Year, Month) as Hash160_Region_Year_Month;
Load * inline [
Region, Year, Month
abc, xyz, 123
EU, 2022, 01
UK, 2022, 02
US, 2022, 02 ];
```

Tabella dei risultati

| Area geografica | Anno | Mese | Hash160_Region_Year_Month    |
|-----------------|------|------|------------------------------|
| abc             | xyz  | 123  | MA&5]6+3=:>;>G%S<U*S2I:.`=X* |
| EU              | 2022 | 01   | B40^K&[T@!;VB'XR]<5=//_F853  |
| UK              | 2022 | 02   | O5T;+1?[B&"F&1//MA[MN!T"FWZ  |
| US              | 2022 | 02   | C6@#]4#_G-(J7EQY#KRW`@KF+W   |

### Esempio: scenario per l'applicazione della funzione Hash160 con voci duplicate

Espressione del grafico

#### Panoramica

In questo esempio si mostra come creare un valore hash univoco per ogni voce del prodotto. Il valore hash funge da identificatore univoco, consentendo all'azienda di individuare e gestire efficacemente i duplicati.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `TransactionID`
  - `ProductID`
  - `ProductName`
  - `SupplierID`

#### Script di caricamento

```
Example:
Load *
inline [
TransactionID, ProductID, ProductName, SupplierID
1, 101, widget A, S001
2, 102, widget B, S002
3, 101, widget A, S001
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `TransactionID`
- `ProductID`
- `ProductName`
- `SupplierID`

Creare la seguente dimensione calcolata:

- `=Hash160(ProductID, ProductName, SupplierID)`, per calcolare una chiave di 27 caratteri.

Tabella dei risultati

| TransactionID | ProductID | ProductName | SupplierID | Hash160(ProductID, ProductName, SupplierID) |
|---------------|-----------|-------------|------------|---------------------------------------------|
| 1             | 101       | Widget A    | S001       | CY`&^(N,E/#`TJT4&J/7N1Q+*T3                 |
| 2             | 102       | Widget B    | S002       | O7(1;V1%MG;Z+A[/H)G5&`6RA0(                 |
| 3             | 101       | Widget A    | S001       | CY`&^(N,E/#`TJT4&J/7N1Q+*T3                 |

Confrontare l'output della funzione `hash160`. Notare che il primo e il terzo record di `TransactionID` hanno valori hash duplicati perché la combinazione delle stringhe `ProductID`, `ProductName` e `SupplierID` è la stessa per queste righe.

### Esempio: scenario per l'applicazione della funzione Hash160 per l'anonimizzazione dei dati

Espressione del grafico

#### Panoramica

Una società di servizi finanziari gestisce le informazioni sensibili sui clienti, come carte di credito, numeri di previdenza sociale e i nomi dei clienti. Per proteggere la privacy e garantire la conformità a normative come il General Data Protection Regulation (GDPR) o l'Health Insurance Portability and Accountability Act (HIPAA), hanno bisogno di rendere anonimi i dati sensibili quando generano report per l'analisi. Tuttavia, devono anche mantenere la coerenza, in modo da poter rintracciare lo stesso cliente in diversi set di dati senza rivelarne l'identità.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `CustomerID`
  - `CustomerName`
  - `SSN`

#### Script di caricamento

```
Example:
Load *
inline [
CustomerID, CustomerName, SSN
101, John Doe, 123-45-6789|
102, Jane Smith, 987-65-4321
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- CustomerID
- CustomerName
- SSN

Creare le seguenti dimensioni calcolate:

- =Hash160(CustomerName)
- =Hash160(SSN)

Tabella dei risultati

| CustomerID | CustomerName | SSN         | Hash160(CustomerName)        | Hash160(SSN)                  |
|------------|--------------|-------------|------------------------------|-------------------------------|
| 101        | John Doe     | 123-45-6789 | GMD(*B9*!PQ&MTY2@\$4A\KCL?[0 | N&[PSH:IS\$M?OF<5/9LV&N*;'CZ( |
| 102        | Jane Smith   | 987-65-4321 | EP@9]GSI)5G\I*,B?H,C?A\$%VG: | HCFXMR'2=\3WF6?_\6LN;.*6:K?   |

L'output genera chiavi univoche che presentano i seguenti vantaggi:

- Privacy dei dati: i dati sensibili originali non sono direttamente visibili, ma il valore hash rimane coerente tra i diversi record. Ciò consente di effettuare analisi, come l'aggregazione delle attività dei clienti, senza esporre l'identità effettiva del cliente.
- Conformità: grazie all'hashing dei campi sensibili, l'azienda garantisce la conformità alle normative sulla privacy dei dati.
- Sicurezza: la funzione hash produce un output di dimensioni fisse che non è reversibile, aggiungendo un ulteriore livello di protezione per i dati sensibili.

### Hash256

**Hash256()** restituisce un valore hash a 256 bit dei valori di espressione di input combinati. Il risultato è una stringa da 43 caratteri. I valori hash sono utili per mascherare le informazioni personali identificabili (PII), come i nomi dei clienti, i numeri di previdenza sociale o i numeri di conto.

#### Sintassi:

```
Hash256(expr{, expression})
```

Tipo di dati restituiti: stringa

| Argomenti         |                         |
|-------------------|-------------------------|
| Argomento         | Descrizione             |
| <code>expr</code> | La stringa da valutare. |

Esempio: espressioni del grafico

| Esempio                                                                                            | Risultato                                                                     |
|----------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|
| <code>Hash256( 'abc', 'xyz', '123' )</code>                                                        | Restituisce<br>MA&5]6+3=:>;>G%S<U*S2I: `=X*A.IO*8N\%Y7Q;YEJ                   |
| <code>Hash256( Region, Year, Month )</code><br><br>Note: Region, Year, and Month are table fields. | Restituisce <code>G7*=6GKPJ(Z+)^KM?&lt;\$'AI.)?U\$#X2RB</code><br>[:0ZP=+Z`F: |

## Esempio: principi fondamentali della funzione Hash256

Espressione del grafico

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi saranno utilizzati per creare un valore hash a 256 bit:
  - `Region`
  - `Year`
  - `Month`

### Script di caricamento

```
Example:
Load *
inline [
Region, Year, Month
abc, xyz, 123
EU, 2022, 01
UK, 2022, 02
US, 2022, 02
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- Region
- Year
- Month

Creare la seguente dimensione calcolata:

- =Hash256(Region, Year, Month), per calcolare il valore hash a 256 bit.

Tabella dei risultati

| Area geografica | Anno | Mese | Hash256(Region, Year, Month)                 |
|-----------------|------|------|----------------------------------------------|
| abc             | xyz  | 123  | MA&5]6+3=:>;>G%S<U*S2I:.`=X*A.IO*8N\%Y7Q;YEJ |
| EU              | 2022 | 01   | B40^K&[T@!;VB'XR]<5=//_F853?BE6'G&,YH*T'MF)  |
| UK              | 2022 | 02   | O5T;+1?[B&"F&1//MA[MN!T"FWZT=4\#V'M%6_\0C>4  |
| US              | 2022 | 02   | C6@[#]4#_G-(J7EQY#KRW`@KF+W-0)`[Z8R+#'")=+0  |

Per ogni riga, le stringhe hash per Region, Year e Month vengono unite e viene restituita una stringa di 43 caratteri.

### Script di caricamento

Il codice seguente mostra come utilizzare la funzione in uno script di caricamento.

```
Hash_256:
Load *,
Hash256(Region, Year, Month) as Hash256_Region_Year_Month;
Load * inline [
Region, Year, Month
abc, xyz, 123
EU, 2022, 01
UK, 2022, 02
US, 2022, 02 ];
```

Tabella dei risultati

| Area geografica | Anno | Mese | Hash256_Region_Year_Month                    |
|-----------------|------|------|----------------------------------------------|
| abc             | xyz  | 123  | MA&5]6+3=:>;>G%S<U*S2I:.`=X*A.IO*8N\%Y7Q;YEJ |
| EU              | 2022 | 01   | B40^K&[T@!;VB'XR]<5=//_F853?BE6'G&,YH*T'MF)  |
| UK              | 2022 | 02   | O5T;+1?[B&"F&1//MA[MN!T"FWZT=4\#V'M%6_\0C>4  |
| US              | 2022 | 02   | C6@[#]4#_G-(J7EQY#KRW`@KF+W-0)`[Z8R+#'")=+0  |

### Esempio: scenario per l'applicazione della funzione Hash256 con voci duplicate

Espressione del grafico

#### Panoramica

In questo esempio si mostra come creare un valore hash univoco per ogni voce del prodotto. Il valore hash funge da identificatore univoco, consentendo all'azienda di individuare e gestire efficacemente i duplicati.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `TransactionID`
  - `ProductID`
  - `ProductName`
  - `SupplierID`

#### Script di caricamento

```
Example:
Load *
inline [
TransactionID, ProductID, ProductName, SupplierID
1, 101, widget A, S001, 100
2, 102, widget B, S002, 200
3, 101, widget A, S001, 50
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `Transaction ID`
- `ProductID`
- `ProductName`
- `SupplierID`

Creare la seguente dimensione calcolata:

- `=Hash256(ProductID, ProductName, SupplierID)` per calcolare una chiave di 43 caratteri.

Tabella dei risultati

| TransactionID | ProductID | ProductName | SupplierID | Hash256(ProductID, ProductName, SupplierID) |
|---------------|-----------|-------------|------------|---------------------------------------------|
| 1             | 101       | Widget A    | S001       | CY`&^(N,E/#`TJT4&]/7N1Q+*T3K=35VDVQ>>;%+A0] |
| 2             | 102       | Widget B    | S002       | O7(1;V1%MG;Z+A[/H)G5&`6RA0(`%6T?YZQ.+>*&\JF |
| 3             | 101       | Widget A    | S001       | CY`&^(N,E/#`TJT4&]/7N1Q+*T3K=35VDVQ>>;%+A0] |

Confrontare l'output della funzione `hash256`. Notare che il primo e il terzo record di `TransactionID` hanno valori hash duplicati perché la combinazione delle stringhe `ProductID`, `ProductName` e `SupplierID` è la stessa per queste righe.

### Esempio: scenario per l'applicazione della funzione Hash256 per l'anonimizzazione dei dati

Espressione del grafico

#### Panoramica

Una società di servizi finanziari gestisce le informazioni sensibili sui clienti, come carte di credito, numeri di previdenza sociale e i nomi dei clienti. Per proteggere la privacy e garantire la conformità a normative come il General Data Protection Regulation (GDPR) o l'Health Insurance Portability and Accountability Act (HIPAA), hanno bisogno di rendere anonimi i dati sensibili quando generano report per l'analisi. Tuttavia, devono anche mantenere la coerenza, in modo da poter rintracciare lo stesso cliente in diversi set di dati senza rivelarne l'identità.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `CustomerID`
  - `CustomerName`
  - `SSN`

#### Script di caricamento

Example:

```
Load *
```

```
inline [
```

```
CustomerID, CustomerName, SSN
```

```
101, John Doe, 123-45-6789]
```

```
102, Jane Smith, 987-65-4321
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- CustomerID
- CustomerName
- SSN

Creare le seguenti dimensioni calcolate:

- =Hash256(CustomerName)
- =Hash256(SSN)

Tabella dei risultati

| CustomerID | CustomerName | SSN         | Hash256(CustomerName)                            | Hash256(SSN)                                            |
|------------|--------------|-------------|--------------------------------------------------|---------------------------------------------------------|
| 101        | John Doe     | 123-45-6789 | GMD(*B9*!PQ&MTY2@\$4A\KCL?<br>[0@T%D@8C#6&UR8/95 | N&<br>[PSH:\$M?OF<5/9LV&<br>N*;'CZ<br>((X""9Y2FKJC6W04[ |
| 102        | Jane Smith   | 987-65-4321 | EP@9]GSI)5G\I*,B?H,C?A\$%VG:SL#6I<br>>Q1:L<0O7UW | HCFXMR'2=\3WF6?_<br>\6LN;.*6:K?Y:%MZ#*P<br>PJWI`YS<     |

L'output genera chiavi univoche che presentano i seguenti vantaggi:

- Privacy dei dati: i dati sensibili originali non sono direttamente visibili, ma il valore hash rimane coerente tra i diversi record. Ciò consente di effettuare analisi, come l'aggregazione delle attività dei clienti, senza esporre l'identità effettiva del cliente.
- Conformità: grazie all'hashing dei campi sensibili, l'azienda garantisce la conformità alle normative sulla privacy dei dati.
- Sicurezza: la funzione hash produce un output di dimensioni fisse che non è reversibile, aggiungendo un ulteriore livello di protezione per i dati sensibili.

## Index

**Index()** ricerca una stringa per individuare la posizione iniziale della n-esima occorrenza di una sottostringa fornita. Un terzo argomento opzionale fornisce il valore di n, che corrisponde a 1 se omesso. Un valore negativo esegue la ricerca dalla fine della stringa. Le posizioni nella stringa sono numerate da sinistra a destra a partire dal valore 1.

### Sintassi:

```
Index(text, substring[, count])
```

Tipo di dati restituiti: numero intero

#### Argomenti

| Argomento        | Descrizione                                                                                                                                                                                                                           |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>text</b>      | La stringa originale.                                                                                                                                                                                                                 |
| <b>substring</b> | Una stringa di caratteri da ricercare in <b>text</b> .<br><br><div>  Se la sottostringa non esiste nel testo, l'Indicizzazione restituisce 0. </div> |
| <b>count</b>     | Definisce l'occorrenza di <b>substring</b> da ricercare. Ad esempio, un valore di 2 ricerca per la seconda occorrenza. Un valore negativo inizia dalla fine della stringa.                                                            |

#### Esempio: espressioni del grafico

| Esempio                                                                        | Risultato                                                       |
|--------------------------------------------------------------------------------|-----------------------------------------------------------------|
| <code>Index( 'abcdefg', 'cd' )</code>                                          | Restituisce 3                                                   |
| <code>Index( 'abcdabcd', 'b', 2 )</code>                                       | Restituisce 6 (la seconda occorrenza di b)                      |
| <code>Index( 'abcdabcd', 'b', -2 )</code>                                      | Restituisce 2 (la seconda occorrenza di b iniziando dalla fine) |
| <code>Left( Date, Index( Date, '-' ) -1 ) where Date = 1997-07-14</code>       | Restituisce 1997                                                |
| <code>Mid( Date, Index( Date, '-' , 2 ) -2, 2 ) where Date = 1997-07-14</code> | Restituisce 07                                                  |
| <code>Index( 'abc', 'x' )</code>                                               | Restituisce 0 (x non esiste nella stringa abc)                  |
| <code>Index( 'abc', 'a', 2 )</code>                                            | Restituisce 0 (non è presente la seconda occorrenza di a)       |

## Esempio: principi fondamentali della funzione Index

Espressione del grafico

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Un campo della tabella dati denominata `InputString`.

### Script di caricamento

Example:

```
Load * inline [  
  InputString  
  abcdefg  
  abcdabcd ];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- InputString

Creare le seguenti dimensioni calcolate:

- =Index(InputString, 'cd')
- =Index(InputString, 'e')
- =Index(InputString, 'b', -1)

Tabella dei risultati

| InputString | Index(InputString, 'cd') | Index(InputString, 'e') | Index(InputString, 'b', -1) |
|-------------|--------------------------|-------------------------|-----------------------------|
| abcdabcd    | 3                        | 0                       | 6                           |
| abcdefg     | 3                        | 5                       | 2                           |

La prima riga con la stringa abcdabcd restituisce:

- 3: la prima occorrenza di cd
- 0: perché e non si trova nella stringa
- 6: la prima occorrenza di b a partire dalla fine della stringa

La seconda riga con la stringa abcdefg restituisce:

- 3: la prima occorrenza di cd
- 5: la prima occorrenza di e
- 2: la prima occorrenza di b a partire dalla fine della stringa



*Un parametro count negativo esegue la scansione della stringa al contrario. Tuttavia, la posizione restituita è sempre indicata da sinistra a destra, anche se la scansione avviene al contrario.*

### Script di caricamento

Il codice seguente mostra come utilizzare la funzione in uno script di caricamento.

Example:

```
Load  
*,
```

```
index(String, 'cd') as Index_CD,           // returns 3 in Index_CD
index(String, 'b') as Index_B,            // returns 2 in Index_B
index(String, 'b', -1) as Index_B2;       // returns 2 or 6 in Index_B2
Load * inline [
String
abcdefg
abcdabcd ];
```

Tabella dei risultati

| Stringa  | Index_CD | Index_B | Index_B2 |
|----------|----------|---------|----------|
| abcdefg  | 3        | 2       | 2        |
| abcdabcd | 3        | 2       | 6        |

### Esempio: scenario per l'applicazione della funzione Index

Espressione del grafico

#### Panoramica

Un centro di assistenza clienti ha un set di dati che contiene le righe dell'oggetto dei messaggi e-mail dei clienti. Il centro di assistenza vuole classificare le e-mail in base alle parole chiave che compaiono nell'oggetto, ad esempio Refund, Complaint e Technical Issue. Le righe dell'oggetto sono memorizzate in un campo del set di dati denominato EmailSubject.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata Example.
- Un campo della tabella dati denominata EmailSubject.

#### Script di caricamento

```
Example:
Load * inline [
EmailSubject
Request for a Refund - Order 12345
Technical Issue with Product XYZ
Complaint about delivery service
Follow-up on Refund Request
Technical Issue not resolved
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- EmailSubject

Creare la seguente dimensione calcolata:

- `=If(Index(EmailSubject, 'Refund') > 0, 'Refund', 'other')`, per calcolare il testo dell'oggetto delle e-mail che si riferisce ai rimborsi.

| Risultati                                |                                                          |
|------------------------------------------|----------------------------------------------------------|
| EmailSubject                             | If(Index(EmailSubject, 'Refund') > 0, 'Refund', 'Other') |
| Reclamo relativo al servizio di consegna | Altro                                                    |
| Follow-up sulla richiesta di rimborso    | Rimborso                                                 |
| Richiesta di rimborso - Ordine 12345     | Rimborso                                                 |
| Problema tecnico non risolto             | Altro                                                    |
| Problema tecnico con il prodotto XYZ     | Altro                                                    |

L'output della funzione `Index` ha identificato con successo tutte le righe dell'oggetto che contengono la stringa `Refund`. Questo è un esempio di come è possibile analizzare e categorizzare i dati utilizzando funzioni annidate e fornire valore agli utenti.

## IndexRegEx

**IndexRegEx()** effettua una ricerca nella stringa di input e restituisce la posizione iniziale dell'occorrenza n-esima del modello di espressione regolare specificato. Un terzo argomento opzionale **count** fornisce il valore di n, che corrisponde a 1 se omissso. Le posizioni nella stringa sono numerate da sinistra a destra a partire dal valore 1 in poi. Se non viene trovata alcuna corrispondenza, la funzione restituisce 0.

Questa funzione esegue operazioni regex che distinguono tra maiuscole e minuscole. In alternativa, è possibile utilizzare la variante **IndexRegExI()** per eseguire operazioni regex senza distinzione tra maiuscole e minuscole.

### Sintassi:

```
IndexRegEx (text, regex [, count])
```

**Tipo di dati restituiti:** numero intero

| Argomenti    |                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Argomento    | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>text</b>  | Il testo della stringa di input in cui si desidera cercare un'espressione regolare.                                                                                                                                                                                                                                                                                                                                                  |
| <b>regex</b> | L'espressione regolare da utilizzare per la ricerca della stringa di input.                                                                                                                                                                                                                                                                                                                                                          |
| <b>count</b> | <p>Il numero della corrispondenza. Questo è utile quando nel testo si possono trovare più corrispondenze per l'espressione regolare. Per esempio, specificare il valore 4 per estrarre la posizione della quarta corrispondenza.</p> <p>Questo argomento è facoltativo. Se non viene specificata, l'impostazione predefinita è 1. È possibile specificare un valore negativo per cercare le corrispondenze da destra a sinistra.</p> |

## Esempi di funzioni

| Esempio                                             | Risultato                                                                                                                      |
|-----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| <code>IndexRegEx('abc123', '[a-z][0-9]+')</code>    | Restituisce 3 (la posizione iniziale della prima corrispondenza).                                                              |
| <code>IndexRegEx('abc123', '[a-z][0-9]+', 2)</code> | Restituisce 0 (la regex non ha una seconda corrispondenza).                                                                    |
| <code>IndexRegEx('ABC123', '[a-z][0-9]+')</code>    | Restituisce 0, perché <b>IndexRegEx()</b> fa distinzione tra maiuscole e minuscole.                                            |
| <code>IndexRegExI('ABC123', '[a-z][0-9]+')</code>   | Restituisce 3. Viene utilizzata la variante senza distinzione tra maiuscole e minuscole della funzione, <b>IndexRegExI()</b> . |

## Casi di utilizzo

I casi d'uso di questa funzione includono:

- Identificazione di dove si verificano modelli testuali specifici all'interno di corpi di testo più ampi. Ad esempio, potrebbe essere necessario sapere dove un modello di indirizzo e-mail viene utilizzato in una serie di lunghi messaggi e-mail.
- La funzione **IndexRegEx()** è particolarmente utile per l'elaborazione avanzata dei dati e viene comunemente utilizzata nella prima fase di una trasformazione più lunga e complessa. In genere, è più facile risolvere i problemi con altre funzioni regex come **ExtractRegEx()**, **MatchRegEx()** e **CountRegEx()**, ma a volte **IndexRegEx()** offre soluzioni che queste funzioni non possono fornire.

## Esempio 1 - script di caricamento per identificare i libri in base all'ISBN

Script di caricamento e risultati

## Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- La tabella **ISBN**, che include il contenuto di un messaggio e-mail in un campo denominato **EmailBody**.

Bisogna trovare le posizioni di tutti i codici ISBN a 13 cifre dal messaggio e-mail in linguaggio naturale.

## Script di caricamento

```
SET ISBN_RegEx = 'ISBN[ ]*([0-9]{3})-([0-9]{1})-([0-9]{4})-([0-9]{4})-([0-9]{1})';
```

ISBN:

```
LOAD IndexRegEx(EmailText,'$(ISBN_RegEx)',1) AS ISBNCode1,
IndexRegEx(EmailText,'$(ISBN_RegEx)',2) AS ISBNCode2,
IndexRegEx(EmailText,'$(ISBN_RegEx)',3) AS ISBNCode3,
* INLINE `
EmailText
```

Hi there! I have some books that I'm interested in ordering from your distribution center. Just wanted to make sure they were in stock before I put the order in on the system - do you think you could tell me whether you have these in stock, and how many you might be able to ship us? Thanks! Item 1: ISBN 123-3-1234-1234-0. This is one I've had a really hard time getting a hold of lately. Do you think you would have 5 in stock? If not, can I put a hold on the next 5 you get? Item 2: ISBN 012-2-0123-0123-4 ... This one is not high-priority, but if you have one, I'll take it! Item 3: ISBN 000-1-0123-0123-2. Customers have been requesting this one for a long time, and we haven't had it in stock for years due to it being out of print. Any chance you might have 7 of them? Thanks!

` (delimiter is '/t/';

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- ISBNCode1
- ISBNCode2
- ISBNCode3

Tabella dei risultati

| ISBNCode1 | ISBNCode2 | ISBNCode3 |
|-----------|-----------|-----------|
| 301       | 487       | 588       |

### Esempio 2 - espressione del grafico per identificare i contatti con numeri di telefono secondari

Espressione del grafico e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Una tabella denominata `BusinessContactInfo`, che contiene informazioni sulle aziende trovate sul sito web di ciascuna azienda.
- `ContactInfo` è un campo che contiene i numeri di telefono dell'azienda. Bisogna determinare quali aziende hanno più numeri di telefono disponibili.

Requisiti:

- I numeri di telefono devono seguire il sistema NANP (piano di numerazione nordamericano), con 10 cifre.
- Bisogna consentire che il prefisso sia racchiuso tra parentesi.
- Bisogna consentire la presenza di un singolo spazio vuoto o di un trattino tra ogni segmento di un numero di telefono.

### Script di caricamento

BusinessContactInfo:

```
Load * Inline `
ID      CompanyName      ContactInfo
1      Company A        (123) 456-7890 (023) 123-4567
2      Company B        0123456790 1357913579 0246802468
3      Company C        234-567-8901

` (delimiter is '\t');
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- CompanyName

Aggiungere la seguente dimensione calcolata alla tabella:

```
=if(IndexRegex(ContactInfo, '\({0,1}[0-9]{3}\){0,1}[ -]{0,1}[0-9]{3}[ -]{0,1}[0-9]{4}', 2)=0,
'No secondary phone number(s)', 'Has secondary phone number(s)')
```

Tabella dei risultati

| CompanyName | =if(IndexRegex(ContactInfo, '\({0,1}[0-9]{3}\){0,1}[ -]{0,1}[0-9]{3}[ -]{0,1}[0-9]{4}', 2)=0, 'No secondary phone number(s)', 'Has secondary phone number(s)') |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Company A   | Has secondary phone number(s)                                                                                                                                  |
| Company B   | Has secondary phone number(s)                                                                                                                                  |
| Company C   | No secondary phone number(s)                                                                                                                                   |

Company A ha un numero di telefono secondario. Company B ha due numeri di telefono secondari. Company C ha tre numeri di telefono secondari.



La funzione **MatchRegex()** è utile anche per questi tipi di casi d'uso di convalida.

## IndexRegexGroup

**IndexRegexGroup()** effettua una ricerca nella stringa di input e restituisce la posizione iniziale dell'occorrenza n-esima del modello di espressione regolare composta specificato. Un quarto argomento opzionale **count** fornisce il valore di n, che corrisponde a 1 se omesso. Le posizioni nella stringa sono numerate da sinistra a destra a partire dal valore 1 in poi. Se non viene trovata alcuna corrispondenza, la funzione restituisce 0.

Questa funzione esegue operazioni regex che distinguono tra maiuscole e minuscole. In alternativa, è possibile utilizzare la variante **IndexRegexGroupI()** per eseguire operazioni regex senza distinzione tra maiuscole e minuscole.

#### Sintassi:

```
IndexRegexGroup (text, regex, group [, count])
```

**Tipo di dati restituiti:** numero intero

#### Argomenti

| Argomento    | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>text</b>  | Il testo della stringa di input in cui si desidera cercare un'espressione regolare.                                                                                                                                                                                                                                                                                                                                                 |
| <b>regex</b> | L'espressione regolare da utilizzare per la ricerca della stringa di input.                                                                                                                                                                                                                                                                                                                                                         |
| <b>group</b> | <p>Il numero del gruppo, nel caso di un'espressione regolare composta.</p> <p>Un valore <b>group</b> pari a 0 restituisce l'indice dell'intera regex. Tuttavia, se l'espressione regolare deve restituire solo l'indice dell'intera corrispondenza, utilizzare invece la funzione <b>IndexRegex()</b>.</p> <p>È possibile specificare un valore <b>group</b> negativo per cercare le corrispondenze da destra a sinistra.</p>       |
| <b>count</b> | <p>Il numero della corrispondenza. Questo è utile quando nel testo si possono trovare più corrispondenze per l'espressione regolare. Per esempio, specificare il valore 4 per cercare la posizione della quarta corrispondenza.</p> <p>Questo argomento è facoltativo. Se non viene specificata, l'impostazione predefinita è 1. È possibile specificare un valore negativo per cercare le corrispondenze da destra a sinistra.</p> |

#### Esempi di funzioni

| Esempio                                                        | Risultato                                                                                                                                                  |
|----------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>IndexRegexGroup ('abc123', '([a-z])([0-9]+)', 0)</code>  | Restituisce 3 (posizione iniziale della regex completa).                                                                                                   |
| <code>IndexRegexGroup ('abc123', '([a-z])([0-9]+)', 1)</code>  | Restituisce 3 (posizione iniziale del primo gruppo).                                                                                                       |
| <code>IndexRegexGroup ('abc123', '([a-z])([0-9]+)', 2)</code>  | Restituisce 4, che è la posizione iniziale del secondo gruppo. Nel modello regex di input, la stringa <code>([0-9]+)</code> corrisponde al secondo gruppo. |
| <code>IndexRegexGroup ('ABC123', '([a-z])([0-9]+)', 1)</code>  | Restituisce 0, perché <b>IndexRegexGroup()</b> fa distinzione tra maiuscole e minuscole.                                                                   |
| <code>IndexRegexGroupI ('ABC123', '([a-z])([0-9]+)', 1)</code> | Restituisce 3. Viene utilizzata la variante senza distinzione tra maiuscole e minuscole della funzione, <b>IndexRegexI()</b> .                             |

### Casi di utilizzo

I casi d'uso di questa funzione includono:

- Identificazione di dove si verificano modelli testuali specifici all'interno di corpi di testo più ampi. Ad esempio, potrebbe essere necessario sapere dove viene utilizzato il dominio di un indirizzo e-mail in una serie di lunghi messaggi e-mail.
- La funzione **IndexRegExGroup()** è particolarmente utile per l'elaborazione avanzata dei dati e viene comunemente utilizzata nella prima fase di una trasformazione più lunga e complessa. In genere, è più facile risolvere i problemi con altre funzioni regex come **ExtractRegExGroup()**, **MatchRegEx()** e **CountRegEx()**, ma a volte **IndexRegExGroup()** offre soluzioni che queste funzioni non possono fornire.

### Esempio 1 - script di caricamento per trovare le posizioni dei componenti ISBN

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- La creazione di una variabile, `ISBN_RegEx`, per memorizzare l'espressione regolare che si desidera utilizzare.
- Un set di codici ISBN per i libri che una libreria vuole ordinare da diversi fornitori.

Per ogni codice ISBN, è necessario estrarre la posizione iniziale di ciascuno dei seguenti elementi:

- EAN
- Gruppo
- Soggetto registrato
- Pubblicazione
- Checksum

#### Script di caricamento

```
SET ISBN_RegEx = 'ISBN[ ]*([0-9]{3})-([0-9]{1})-([0-9]{4})-([0-9]{4})-([0-9]{1})';

ISBN:
LOAD Supplier,
IndexRegExGroup(Books, '$(ISBN_RegEx)', 1) AS EAN,
IndexRegExGroup(Books, '$(ISBN_RegEx)', 2) AS Group,
IndexRegExGroup(Books, '$(ISBN_RegEx)', 3) AS Registrant,
IndexRegExGroup(Books, '$(ISBN_RegEx)', 4) AS Publication,
IndexRegExGroup(Books, '$(ISBN_RegEx)', 5) AS Checksum;
// Split the ISBN with the Group function in a preceding load to avoid generating a cartesian
product
LOAD *,
ExtractRegEx(SupplierBooks, '$(ISBN_RegEx)') AS Books
```

```
INLINE [  
Supplier, SupplierBooks  
Supplier 1, ISBN 123-3-1234-1234-0 ISBN 012-2-0123-0123-4 ISBN 000-1-0123-0123-2 ISBN 234-5-  
2345-2345-1 ISBN 555-2-5555-5555-3 ISBN 222-4-2222-2222-2  
Supplier 2, ISBN 000-0-3333-3333-3 ISBN 333-3-3333-3333-3 ISBN 555-1-5151-5151-3 ISBN 232-1-  
2323-2323-1 ISBN 008-0-7777-7777-3 ISBN 888-0-9999-0000-0  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- Supplier
- EAN
- Group
- Registrant
- Publication
- Checksum

Tabella dei risultati

| Supplier   | EAN | Group | Registrant | Publication | Checksum |
|------------|-----|-------|------------|-------------|----------|
| Supplier 1 | 6   | 10    | 12         | 17          | 22       |
| Supplier 1 | 6   | 10    | 12         | 17          | 22       |
| Supplier 1 | 6   | 10    | 12         | 17          | 22       |
| Supplier 1 | 6   | 10    | 12         | 17          | 22       |
| Supplier 1 | 6   | 10    | 12         | 17          | 22       |
| Supplier 1 | 6   | 10    | 12         | 17          | 22       |
| Supplier 2 | 6   | 10    | 12         | 17          | 22       |
| Supplier 2 | 6   | 10    | 12         | 17          | 22       |
| Supplier 2 | 6   | 10    | 12         | 17          | 22       |
| Supplier 2 | 6   | 10    | 12         | 17          | 22       |
| Supplier 2 | 6   | 10    | 12         | 17          | 22       |
| Supplier 2 | 6   | 10    | 12         | 17          | 22       |

In questo caso, ogni libro contiene componenti nelle stesse posizioni.

### Esempio 2 - espressioni del grafico per trovare le posizioni dell'URL (con confronto con **IndexRegEx()** )

Espressione del grafico e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Una tabella denominata `Correspondence`, che contiene il testo del messaggio e-mail in un campo denominato `EmailBody`.
- Contenuto del messaggio e-mail che contiene URL web.

Bisogna trovare la posizione del secondo URL, se presente, e le posizioni dei componenti seguenti:

- Dominio
- Percorso

Bisogna utilizzare un'unica espressione regolare per eseguire tutte le operazioni.

#### Script di caricamento

Correspondence:

```
Load * Inline `
ID      EmailBody
1      Thanks again for this morning's meeting! You can find the meeting minutes posted here:
https://example.com/resourceexample. If you still have any questions, always feel free to ask
me or one of the other team members. Here are a few learning resources that might help you:
http://www.example.ca/training1.pptx http://www.example.ca/training2.pptx
http://www.example.ca/training3.pptx Thanks again!
2      Hi, you'll want to visit our company website for that, it's available at
https://www.example.se.
3      Hello all, I just wanted to let you know that our online stores are now up and running! I
couldn't be more excited. We are already seeing quite a bit of traffic and volume sold, which
is very promising! For Product A, go to https://www.examplestore1.com/products. For Product B,
you'll want go to https://www.examplestore2.com/products. Product C, go check out
https://www.examplestore3.com/products. Cheers!

` (delimiter is '\t');
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- ID

Aggiungere le seguenti dimensioni calcolate:

- **Posizione URL 2:**  
=IndexRegExGroup(EmailBody, '([a-zA-Z0-9]+):\\/(\\/([a-zA-Z0-9]+\\.)\*([a-zA-Z0-9]+)\\.([a-zA-Z0-9]+)))(\\/[a-zA-Z0-9]+)\*(\\/[a-zA-Z0-9]+)(\\.([a-zA-Z0-9]+)?))?', 2)
- **Posizione dominio URL 2:**  
=IndexRegExGroup(EmailBody, '([a-zA-Z0-9]+):\\/(\\/([a-zA-Z0-9]+\\.)\*([a-zA-Z0-9]+)\\.([a-zA-Z0-9]+)))(\\/[a-zA-Z0-9]+)\*(\\/[a-zA-Z0-9]+)(\\.([a-zA-Z0-9]+)?))?', 2, 2)
- **Posizione percorso URL 2:**  
=IndexRegExGroup(EmailBody, '([a-zA-Z0-9]+):\\/(\\/([a-zA-Z0-9]+\\.)\*([a-zA-Z0-9]+)\\.([a-zA-Z0-9]+)))(\\/[a-zA-Z0-9]+)\*(\\/[a-zA-Z0-9]+)(\\.([a-zA-Z0-9]+)?))?', 6, 2)

Tabella dei risultati

| ID | Posizione URL 2 | Posizione dominio URL 2 | Posizione percorso URL 2 |
|----|-----------------|-------------------------|--------------------------|
| 1  | 274             | 281                     | 295                      |
| 2  | 0               | 0                       | 0                        |
| 3  | 298             | 306                     | 327                      |

**IndexRegEx()** restituisce la posizione iniziale dell'intero URL, mentre **IndexRegExGroup()** restituisce le posizioni iniziali di singole parti dell'URL, corrispondenti al valore **group** che è stato utilizzato. Il record con un valore per **ID** uguale a 2 contiene solo un URL, quindi vengono restituiti i valori 0.

La stessa espressione regolare viene utilizzata in tutte le espressioni dei grafici per restituire diverse informazioni. La scomposizione dei gruppi definiti nell'espressione regolare è la seguente.

Gruppi nell'espressione regolare

| Numero del gruppo | Regex                                                             | Parte URL                                  |
|-------------------|-------------------------------------------------------------------|--------------------------------------------|
| 1                 | ([a-zA-Z0-9]+)                                                    | Protocollo                                 |
| 2                 | ((([a-zA-Z0-9]+\\.)*([a-zA-Z0-9]+)\\.([a-zA-Z0-9]+))              | Dominio (gruppi 3, 4 e 5)                  |
| 3                 | ([a-zA-Z0-9]+\\.)*                                                | Dominio radice                             |
| 4                 | ([a-zA-Z0-9]+)                                                    | Dominio di primo livello                   |
| 5                 | ([a-zA-Z0-9]+)                                                    | Percorso (gruppi 7, 8, 9 e 10) (opzionale) |
| 6                 | ((\\([a-zA-Z0-9]+)\\(\\([a-zA-Z0-9]+)\\(\\.[a-zA-Z0-9]+)\\)\\)?)? | Sottodirectory                             |
| 7                 | (\\([a-zA-Z0-9]+)\\)*                                             | File (gruppi 9 e 10)                       |
| 8                 | (\\([a-zA-Z0-9]+)\\(\\.[a-zA-Z0-9]+)\\)?)                         | File                                       |
| 9                 | ([a-zA-Z0-9]+)                                                    | Nome file                                  |
| 10                | (\\.[a-zA-Z0-9]+)?                                                | Estensione del file (opzionale)            |

## IsJson

**IsJson()** verifica se una stringa specificata contiene dati JSON (JavaScript Object Notation) validi. È anche possibile convalidare un tipo specifico di dati JSON.

### Sintassi:

```
value IsJson(json [, type])
```

**Tipo di dati restituiti:** duale

### Argomenti

| Argomento   | Descrizione                                                                                                                                                                                                                                                         |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>json</b> | Stringa da testare. Può contenere spazi extra o nuove righe.                                                                                                                                                                                                        |
| <b>type</b> | Argomento opzionale che specifica il tipo di dati JSON da testare. <ul style="list-style-type: none"> <li>• 'value' (predefinito)</li> <li>• 'object'</li> <li>• 'array'</li> <li>• 'stringa'</li> <li>• 'number'</li> <li>• 'Boolean'</li> <li>• 'null'</li> </ul> |

### Esempio: espressioni del grafico valide e non valide

| Esempio                      | Risultato                                                   |
|------------------------------|-------------------------------------------------------------|
| IsJson( 'null' )             | Restituisce -1 (true)                                       |
| IsJson( '"abc"', 'value' )   | Restituisce -1 (true)                                       |
| IsJson( '"abc"', 'string' )  | Restituisce -1 (true)                                       |
| IsJson( 123, 'number' )      | Restituisce -1 (true)                                       |
| IsJson( 'text' )             | Restituisce 0 (false), 'text' non è un valore JSON valido   |
| IsJson( '"text"', 'number' ) | Restituisce 0 (false), '"text"' non è un numero JSON valido |
| IsJson( '"text"', 'text' )   | Restituisce 0 (false), 'text' non è un tipo JSON valido     |

## Esempio: principi fondamentali della funzione IsJson

Espressione del grafico

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Un campo della tabella dati denominata `API_Response`.
- Un campo derivato denominato `ID`. Questo campo utilizza la funzione `Rec()` per enumerare i record di input.

### Script di caricamento

Example:

Load

`Recno()` AS `ID`, `API_Response`

inline [

`API_Response`

`'{"id": 1, "name": "Alice"}'`

`'{"id": 2, "name": "Bob"}'`

`'{invalid json string}'`

`'{"id": 4, "name": "Charlie"}'`

`'{"id": 5, name: "David"}'`

];

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `ID`
- `API_Response`

Creare la seguente dimensione calcolata:

- `=IsJson(API_Response)`, per calcolare se i valori in `API_Response` sono JSON validi.

Tabella dei risultati

| ID | API_Response                                | IsJson(API_Response) |
|----|---------------------------------------------|----------------------|
| 1  | <code>'{"id": 1, "name": "Alice"}'</code>   | -1 (true)            |
| 2  | <code>'{"id": 2, "name": "Bob"}'</code>     | -1 (true)            |
| 3  | <code>{stringa json non valida}</code>      | 0 (false)            |
| 4  | <code>'{"id": 4, "name": "Charlie"}'</code> | -1 (true)            |
| 5  | <code>'{"id": 5, name: "David"}'</code>     | 0 (false)            |

L'output restituisce -1 o true, per i valori con sintassi JSON valida.

L'output restituisce 0 o false, per i seguenti record ID (3 e 5) con JSON non valido:

- `{invalid json string}`: la stringa di testo non rappresenta una struttura di risposta JSON valida.
- `'{"id": 5, name: "David"}'`: la seconda coppia chiave-valore non ha le virgolette intorno alla chiave `"name"`.

### Esempio: scenario per l'applicazione della funzione IsJson

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Un campo della tabella dati denominata `orderDetails`.

#### Script di caricamento

Example:

Load \*

INLINE [OrderDetails

```
{ "order_id": "12345", "customer": { "name": "John Doe", "email": "john.doe@example.com"},  
"items": { "product": "Laptop", "quantity": 2, "price": 1200 }, "total_price": 2400 }  
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `orderDetails`

Creare le seguenti misure:

- `IsJson( orderDetails)`, per calcolare se i valori in `orderDetails` sono JSON validi.
- `IsJson( JsonGet ( OrderDetails, '/items/price' ), 'number' )`, che utilizza la funzione `JsonGet`, recupera il testo JSON per la chiave `price` e convalida che il valore di `price` sia un numero.

Tabella dei risultati

| OrderDetails                                                                                                                                                                           | IsJson<br>(OrderDetails) | IsJson( JsonGet ( OrderDetails,<br>'/items/price' ),<br>'number' ) |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|--------------------------------------------------------------------|
| { "order_id": "12345", "customer": { "name": "John Doe",<br>"email": "john.doe@example.com"}, "items": { "product":<br>"Laptop", "quantity": 2, "price": 1200 }, "total_price": 2400 } | -1 (true)                | -1 (true)                                                          |

La prima misura restituisce -1 (true) perché `orderDetails` contiene una sintassi JSON valida.

La seconda misura restituisce -1 (true) perché il valore della chiave `price` è un numero valido, 1200.

## IsRegEx

**IsRegEx()** restituisce se il testo specificato è un'espressione regolare valida.

Questa funzione esegue operazioni regex che distinguono tra maiuscole e minuscole. In alternativa, è possibile utilizzare la variante **IsRegExI()** per eseguire operazioni regex senza distinzione tra maiuscole e minuscole.

### Sintassi:

```
IsRegEx (expr [, debug])
```

Tipo di dati restituiti: duale

### Argomenti

| Argomento    | Descrizione                                                                                                                                                                                                         |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>expr</b>  | Espressione stringa contenente l'espressione regolare da valutare.                                                                                                                                                  |
| <b>debug</b> | Questo argomento è facoltativo. Se viene specificato un valore per questo argomento, l'espressione restituisce un suggerimento di testo che indica come potrebbe essere risolta un'espressione regolare non valida. |

### Esempi di funzioni

| Esempio                          | Risultato                                                                                                                                   |
|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| <code>IsRegEx(' [a-z]')</code>   | Restituisce -1 (true).                                                                                                                      |
| <code>IsRegEx(']0-1[', 1)</code> | Questo esempio restituisce un messaggio di errore con un suggerimento per correggere l'espressione regolare non valida <code>]0-1[</code> . |
| <code>IsRegEx(' [a-z]')</code>   | Restituisce 0 (false). Non viene mostrato alcun messaggio di errore perché l'argomento <b>debug</b> non è stato specificato.                |

## Casi di utilizzo

È possibile utilizzare **IsRegEx()** per verificare se le espressioni regolari scritte dall'utente sono valide in Qlik Sense. Ad esempio, è possibile caricare una tabella di input nell'app con un testo di input e una valutazione della validità del testo come regex.

## Esempio 1 - caricamento della validità della regex nel modello dati

Script di caricamento e risultati

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un campo, `reg_exp`, contenente stringhe di input che possono essere o meno espressioni regolari valide.
- Due campi calcolati per verificare se ogni valore in `reg_exp` è un'espressione regolare valida o meno:
  - `validity_no_hint`: visualizza -1 (true) se la regex è valida, e 0 (false) se non è valida.
  - `validity_with_hint`: visualizza -1 (true) se la regex è valida. Se la regex non è valida, un messaggio di errore generato automaticamente evidenzia i problemi nell'input, con possibili suggerimenti per risolvere il problema.

### Script di caricamento

```
[Expression validity]:
    Load
        *,
        IsRegex(reg_exp) as validity_no_hint,
        IsRegex(reg_exp,1) as validity_with_hint
    ;

Load
*
Inline
`
reg_exp
[a-z]+
([a-z)(0-9)+
abcdefg|[0-7
\$|¢(0-9)
`
;
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `reg_exp`
- `validity_no_hint`
- `validity_with_hint`

Tabella dei risultati

| reg_exp                   | validity_no_hint | validity_with_hint                               |
|---------------------------|------------------|--------------------------------------------------|
| <code>([a-z)(0-9)+</code> | 0                | -1                                               |
| <code>[a-z]+</code>       | -1               | Testo di suggerimento (generato automaticamente) |
| <code>\\$ ¢(0-9)</code>   | -1               | Testo di suggerimento (generato automaticamente) |
| <code>abcdefg [0-7</code> | 0                | -1                                               |

### Esempio 2 - caricamento solo della regex valida nel modello dati

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un campo, `reg_exp`, contenente stringhe di input che possono essere o meno espressioni regolari valide. Questo campo è contenuto in una tabella denominata `Regular Expressions`.
- La clausola **where** che carica i record nel modello dati solo quando l'input è un'espressione regolare valida.

#### Script di caricamento

[Regular Expressions]:

```
Load
*
Inline
`
reg_exp
[a-z]+
([a-z](0-9)+
abcdefg|[0-7
\$\|?(0-9)
`
WHERE
IsRegEx(reg_exp)=True()
;
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `reg_exp`

Tabella dei  
risultati

| <code>reg_exp</code>     |
|--------------------------|
| <code>[a-z]+</code>      |
| <code>\\$\ ?(0-9)</code> |

`[a-z]+` e `\$\|?(0-9)` sono espressioni regolari valide, quindi sono state caricate nel modello dati. `([a-z](0-9)+` e `abcdefg|[0-7` non sono espressioni regolari valide, quindi non sono state caricate nel modello dati.

### Esempio 3 - espressioni dei grafici

Espressione del grafico e risultati

#### Panoramica

In questo esempio, bisogna caricare una tabella di input nel modello dati e aggiungerla una tabella, insieme a due dimensioni calcolate che valutano la validità della regex. Le due dimensioni calcolate trasmettono le stesse informazioni in modi diversi.

Aprire Editor caricamento dati, quindi aggiungere il seguente script di caricamento in una nuova sezione.

Lo script di caricamento include un campo, `reg_exp`, contenente stringhe di input che possono essere o meno espressioni regolari valide. Questo campo è contenuto in una tabella denominata `Regular Expressions`.

#### Script di caricamento

```
[Regular Expressions]:
```

```
Load
*
Inline
`
reg_exp
[a-z]+
([a-z](0-9)+
abcdefg|[0-7
\$\|¢(0-9)
`;
```

#### Risultati

Procedere come indicato di seguito:

1. Caricare i dati e aprire un foglio. Creare una nuova tabella. Aggiungere `reg_exp` come dimensione.
2. Quindi, aggiungere la seguente dimensione calcolata:  
`=IsRegex(reg_exp)`  
Si tratta di un'espressione del grafico di base che mostra -1 (true) se l'input è un'espressione regolare valida, e 0 (false) se non lo è.
3. Aggiungere un'altra dimensione calcolata:  
`=if(IsRegex(reg_exp), 'Valid regex', 'Invalid regex')`  
Si tratta di un'espressione condizionale che offre maggiori dettagli sulla validità dell'input.

Tabella dei risultati

| reg_exp                   | =IsRegex(reg_exp) | =if(IsRegex(reg_exp), 'Valid regex', 'Invalid regex') |
|---------------------------|-------------------|-------------------------------------------------------|
| <code>([a-z](0-9)+</code> | 0                 | Invalid regex                                         |

| reg_exp      | =IsRegEx(reg_exp) | =if(IsRegEx(reg_exp), 'Valid regex', 'Invalid regex') |
|--------------|-------------------|-------------------------------------------------------|
| [a-z]+       | -1                | Valid regex                                           |
| \\$ ¢(0-9)   | -1                | Valid regex                                           |
| abcdefg [0-7 | 0                 | Invalid regex                                         |

## JsonArray

JsonArray aggrega e concatena i dati JSON.

Sintassi:

```
JsonArray([DISTINCT][TOTAL] value [, sort_weight])
```

Tipo di dati restituiti: duale

### Argomenti

| Argomento   | Descrizione                                                                                                                                                                                                                                                                                                                                           |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DISTINCT    | Se la parola <b>DISTINCT</b> è riportata prima degli argomenti della funzione, i duplicati risultanti dalla valutazione degli argomenti della funzione vengono ignorati.                                                                                                                                                                              |
| TOTAL       | Se la parola <b>TOTAL</b> viene riportata prima degli argomenti della funzione, il calcolo verrà effettuato su tutti i valori possibili dati dalle selezioni correnti e non solo su quelli relativi al valore dimensionale attuale, vale a dire che verranno ignorate le dimensioni del grafico.                                                      |
| value       | Campo di input. L'espressione contenente i dati da ordinare. Viene trovato il primo valore (il più basso) di <b>sort_weight</b> dal quale viene determinato il valore corrispondente dell'espressione <b>value</b> . Inserendo un segno meno davanti a <b>sort_weight</b> , la funzione restituisce invece l'ultimo valore ordinato (il più elevato). |
| sort_weight | Campo di input. L'espressione contenente i dati da ordinare. Viene trovato il primo valore (il più basso) di <b>sort_weight</b> dal quale viene determinato il valore corrispondente dell'espressione <b>value</b> . Inserendo un segno meno davanti a <b>sort_weight</b> , la funzione restituisce invece l'ultimo valore ordinato (il più elevato). |

Esempio:

Il seguente script di caricamento carica e aggrega i dati con **JsonArray**.

Data:

```
LOAD Id,
      JsonArray(ValueOrNull) AS Json,
      JsonArray(DISTINCT ValueOrNull) AS JsonDistinct,
      JsonArray(ValueOrNull, Order) AS JsonSorted
GROUP BY Id;

LOAD *, If(Value='- ',Null(),Value) AS ValueOrNull;

LOAD * INLINE '
      Id | Value          | Order
```

```

1 | 123          | 1
2 | "json text"  | 2
2 | normal text  | 1
3 | [1,2,3]      | 3
3 | {"name":"abc"} | 2
3 | [1,2,3]      | 1
4 | 1            | 1
4 | 2 and text    | 2
4 |              | 3
4 | -            | 4
' (delimiter is '|');

```

Il risultato è la seguente tabella dati:

| Risultati |                                  |                                |                                  |
|-----------|----------------------------------|--------------------------------|----------------------------------|
| ID        | JSON                             | JSON distinto                  | JSON ordinato                    |
| 1         | [123]                            | [123]                          | [123]                            |
| 2         | ["testo json","testo normale"]   | ["testo json","testo normale"] | ["testo normale","testo json"]   |
| 3         | [[1,2,3],[1,2,3],{"name":"abc"}] | [[1,2,3],{"name":"abc"}]       | [[1,2,3],{"name":"abc"},[1,2,3]] |
| 4         | [1,"2 e testo",""]               | [1,"2 e testo",""]             | [1,"2 e testo",""]               |

## JsonGet

**JsonGet()** restituisce il percorso di una stringa dati JSON (JavaScript Object Notation). I dati devono essere dati JSON validi ma possono contenere spazi extra o righe nuove.

### Sintassi:

```
value JsonGet(json, path)
```

**Tipo di dati restituiti:** duale

### Argomenti

| Argomento   | Descrizione                                                                                                                                                                                             |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>json</b> | Stringa contenente dati JSON.                                                                                                                                                                           |
| <b>path</b> | Il percorso deve essere specificato in base a <a href="#">RFC 6901</a> . Ciò consentirà la ricerca di proprietà all'interno dei dati JSON senza utilizzare funzioni complesse di sottostringa o indice. |

### Esempio: espressioni del grafico valide e non valide

| Esempio                                                       | Risultato                                              |
|---------------------------------------------------------------|--------------------------------------------------------|
| JsonGet( '{"a":{"xyz":"XYZ"},"b": [123,"abc","ABC"]}', '' )   | Restituisce {"a":{"xyz":"XYZ"},"b": [123,"abc","ABC"]} |
| JsonGet( '{"a":{"xyz":"XYZ"},"b": [123,"abc","ABC"]}', '/a' ) | Restituisce {"xyz":"XYZ"}                              |

| Esempio                                                                               | Risultato                                                                                  |
|---------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|
| <code>JsonGet( '{ "a": {"xyz": "XYZ"}, "b": [123, "abc", "ABC"] }', '/a/xyz' )</code> | Restituisce "XYZ"                                                                          |
| <code>JsonGet( '{ "a": {"xyz": "XYZ"}, "b": [123, "abc", "ABC"] }', '/b' )</code>     | Restituisce [123, "abc", "ABC"]                                                            |
| <code>JsonGet( '{ "a": {"xyz": "XYZ"}, "b": [123, "abc", "ABC"] }', '/b/0' )</code>   | Restituisce 123                                                                            |
| <code>JsonGet( '{ "a": {"xyz": "XYZ"}, "b": [123, "abc", "ABC"] }', '/b/1' )</code>   | Restituisce abc                                                                            |
| <code>JsonGet( '{ "a": {"xyz": "XYZ"}, "b": [123, "abc", "ABC"] }', '/b/2' )</code>   | Restituisce ABC                                                                            |
| <code>JsonGet( '{ "a": "b" }', '/b' )</code>                                          | Restituisce null. Il percorso non punta a una parte valida dei dati JSON.                  |
| <code>JsonGet( '{ "a" }', '/a' )</code>                                               | Restituisce null. I dati JSON non sono valori JSON validi (il membro "a" non ha un valore) |

## Esempio: principi fondamentali della funzione JsonGet

Espressione del grafico

### Panoramica

Si ricevono i dati JSON con i dettagli dell'ordine come una stringa di testo in un'unica riga. Si vuole recuperare le seguenti informazioni:

- prezzo dell'articolo e la convalida che il valore sia un numero
- indirizzo e-mail del cliente
- prezzo totale (prezzo \* quantità)

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Un campo della tabella dati denominata `orderDetails`.

### Script di caricamento

Example:

```
Load *
INLINE [OrderDetails
'{ "order_id": "12345", "customer": { "name": "John Doe", "email": "john.doe@example.com"},
"items": { "product": "Laptop", "quantity": 2, "price": 1200 }, "total_price": 2400 }'
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `OrderDetails`

Creare le seguenti misure:

- `IsJson( JsonGet ( OrderDetails, '/items/price' ), 'number' )`, combinato con la funzione `IsJson`, recupera il testo JSON per `price` e convalida che il valore di `price` sia un numero.
- `JsonGet( OrderDetails, '/customer/email' )`, per recuperare l'indirizzo e-mail del cliente utilizzando il percorso corretto.
- `JsonGet( OrderDetails, '/total_price' )`, per recuperare il prezzo totale utilizzando il percorso corretto.

Tabella dei risultati

| OrderDetails                                                                                                                                                                     | IsJson( JsonGet ( OrderDetails, '/items/price' ), 'number' ) | JsonGet( OrderDetails, '/customer/email' ) | JsonGet( OrderDetails, '/total_price' ), ) |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------|--------------------------------------------|--------------------------------------------|
| { "order_id": "12345", "customer": { "name": "John Doe", "email": "john.doe@example.com"}, "items": { "product": "Laptop", "quantity": 2, "price": 1200 }, "total_price": 2400 } | -1 (true)                                                    | john.doe@example.com                       | 2400                                       |

La prima misura restituisce `-1 true` perché la chiave `price` ha il valore `1200`, che è un numero.

La seconda misura restituisce l'indirizzo e-mail del cliente, `john.doe@example.com`, da `OrderDetails`, utilizzando il percorso corretto della chiave `email` nella stringa JSON.

La terza misura restituisce `2400`, che è il prezzo totale dell'ordine dal campo `OrderDetails`, utilizzando il percorso corretto della chiave `total_price` nella stringa JSON.

### JsonObject

`JsonObject` viene utilizzato per creare oggetti JSON.



*È possibile aggiungere un numero qualsiasi di coppie nome-valore con `JsonObject`.*

#### Sintassi:

```
JsonObject (name, value [name, value, ...])
```

Tipo di dati restituiti: duale

Argomenti

| Argomento | Descrizione                              |
|-----------|------------------------------------------|
| json      | Stringa contenente dati JSON.            |
| nome      | Il nome del campo in formato JSON.       |
| value     | Il valore della stringa in formato JSON. |

### Esempio:

Il seguente script di caricamento carica e imposta i dati come un oggetto JSON.

```
Data:
Load *,
    JsonObject('id', Id,
               'name', Name,
               'address', Address,
               'phone', Phone,
               'fax', FaxOrNull) AS Json;

LOAD *, If(Fax='-',Null(),Fax) AS FaxOrNull;

LOAD * INLINE [
    Id, Name,      Address,      Phone,      Fax
    1, John Doe,   Oak Way,      1 234 567, 1 234 568
    2, Jane Doe,   Maple Way,  123456,   -
    3, Mr Xavier,  Spruce Way,  1-800-MRX
];
```

Il risultato è la seguente tabella dati:

```
{ "id":1, "name":"John Doe", "address":"Oak Way", "phone":"1 234 567", "fax":"1 234 568" }
{ "id":2, "name":"Jane Doe", "address":"Maple Way", "phone":123456 }
{ "id":3, "name":"Mr Xavier", "address":"Spruce Way", "phone":"1-800-MRX", "fax":"" }
```

## JsonSet

**JsonSet()** modifica una stringa contenente dati JSON (JavaScript Object Notation). Può impostare o inserire un valore JSON con la nuova posizione specificata dal percorso. I dati devono essere dati JSON validi ma possono contenere spazi extra o righe nuove.



*È possibile aggiungere un numero qualsiasi di coppie nome-valore con JsonSet.*

### Sintassi:

```
value JsonSet(json, path, value)
```

Tipo di dati restituiti: duale

#### Argomenti

| Argomento    | Descrizione                                                                                                                                                                                                                |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>json</b>  | Stringa contenente dati JSON.                                                                                                                                                                                              |
| <b>path</b>  | Il percorso deve essere specificato in base a <a href="#">RFC 6901</a> . Ciò consentirà la creazione di proprietà all'interno dei dati JSON senza l'utilizzo di funzione complesse sottostringa o indice e concatenazione. |
| <b>value</b> | Il nuovo valore della stringa in formato JSON.                                                                                                                                                                             |

#### Esempio: espressioni del grafico valide e non valide

| Esempio                                                   | Risultato                                                                                           |
|-----------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| <code>JsonSet( '{}', '/a', '"b"' )</code>                 | Restituisce <code>{"a": "b"}</code>                                                                 |
| <code>JsonSet( '[]', '/0', '"x"' )</code>                 | Restituisce <code>["x"]</code>                                                                      |
| <code>JsonSet( '"abc"', '', '123' )</code>                | Restituisce 123                                                                                     |
| <code>JsonSet( '"abc"', '/x', '123' )</code>              | Restituisce null. Il percorso non punta a una parte valida dei dati JSON.                           |
| <code>JsonSet( '{"a": {"b": "c"}}', 'a/b', '"x"' )</code> | Restituisce null. Il percorso non è valido.                                                         |
| <code>JsonSet( '{"a": "b"}', '/a', 'abc' )</code>         | Restituisce null. Il valore non è un JSON valido. Una stringa deve essere racchiusa tra virgolette. |

## Esempio: principi fondamentali della funzione JsonSet

Espressione del grafico

### Panoramica

I dati JSON vengono ricevuti come una stringa di testo in un'unica riga. Si desidera aggiornare il prezzo dell'articolo e il prezzo totale dell'ordine di conseguenza.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Un campo della tabella dati denominata `orderDetails`.

### Script di caricamento

Example:

Load \*

```
INLINE [OrderDetails  
{ "order_id": "12345", "customer": { "name": "John Doe", "email": "john.doe@example.com"},  
  "items": { "product": "Laptop", "quantity": 2, "price": 1200 }, "total_price": 2400 }  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- OrderDetails

Creare la seguente misura:

- `JsonSet(JsonSet(OrderDetails, '/items/price', 1100), '/total_price', 2200)`, per impostare il prezzo su 1100 e il prezzo totale su 2200 utilizzando il percorso corretto delle rispettive chiavi nella stringa JSON.

Tabella dei risultati

| OrderDetails                                                                                                                                                                                 | JsonSet(JsonSet<br>(OrderDetails, '/items/price', 1100), '/total_<br>price', 2200)                                                                                      |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| { "order_id": "12345", "customer": { "name":<br>"John Doe", "email":<br>"john.doe@example.com"}, "items":<br>{ "product": "Laptop", "quantity": 2, "price":<br>1200 }, "total_price": 2400 } | {"order_id":"12345","customer":{"name":"John<br>Doe","email":"john.doe@example.com"},"items":<br>{"product":"Laptop","quantity":2,"price":1100},"total_<br>price":2200} |

L'output della misura aggiorna le chiavi price e total price a 1100 e 2200, rispettivamente.

## JsonSetEx

**JsonSetEx()** modifica le stringhe contenenti dati JSON (JavaScript Object Notation). Può impostare o inserire i valori JSON con la nuova posizione specificata dal percorso. I dati devono essere dati JSON validi ma possono contenere spazi extra o righe nuove.



*È possibile aggiungere un numero qualsiasi di coppie nome-valore con JsonSetEx.*

### Sintassi:

```
JsonSetEx(json, path, value, ...)
```

Tipo di dati restituiti: duale

Argomenti

| Argomento | Descrizione                                                                                                                                                                                                                |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| json      | Stringa contenente dati JSON.                                                                                                                                                                                              |
| path      | Il percorso deve essere specificato in base a <a href="#">RFC 6901</a> . Ciò consentirà la creazione di proprietà all'interno dei dati JSON senza l'utilizzo di funzione complesse sottostringa o indice e concatenazione. |
| value     | Il nuovo valore della stringa in formato JSON.                                                                                                                                                                             |

### Esempio:

Il seguente script di caricamento carica e formatta i dati in formato JSON.

```
Data:
Load *,
    JsonSetEx('{ "fax":123}',
        '/name', Name,
        '/address', Address,
        '/phone', Phone,
        '/fax', FaxOrNull) AS Json;

LOAD *, If(Fax='- ',Null(),Fax) AS FaxOrNull;

LOAD * INLINE [
    Id, Name,      Address,      Phone,      Fax
    1,  John Doe,   Oak Way,      1 234 567,  1 234 568
    2,  Jane Doe,   Maple Way,  123456,      -
    3,  Mr Xavier,   Spruce Way,  1-800-MRX
];
```

Il risultato è la seguente tabella dati:

```
{ "name": "John Doe", "address": "Oak Way", "phone": "1 243 567", "fax": "1 234 568" }
{ "name": "Jane Doe", "address": "Maple Way", "phone": "123456" }
{ "fax": "", "name": "Mr Xavier", "address": "Spruce Way", "phone": "1-800-MRX" }
```

## KeepChar

**KeepChar()** restituisce una stringa costituita da qualsiasi carattere nella prima stringa che corrisponde ai caratteri nella seconda stringa. Questa funzione è sensibile alle maiuscole e alle minuscole.

### Sintassi:

```
KeepChar (text, keep_chars)
```

Tipo di dati restituiti: stringa

### Argomenti

| Argomento         | Descrizione                                                                                                            |
|-------------------|------------------------------------------------------------------------------------------------------------------------|
| <b>text</b>       | La stringa originale.                                                                                                  |
| <b>keep_chars</b> | Una stringa contenente i caratteri in <b>text</b> da conservare. Questo argomento distingue tra maiuscole e minuscole. |

### Esempio: espressioni del grafico

| Esempio                        | Risultato        |
|--------------------------------|------------------|
| KeepChar ( 'a1b2c3', '123' )   | Restituisce 123  |
| KeepChar ( 'a1b2c3', '1234' )  | Restituisce 123  |
| KeepChar ( 'a1b22c3', '1234' ) | Restituisce 1223 |
| KeepChar( 'a1b2c3', '312' )    | Restituisce 123  |

## Esempio: principi fondamentali della funzione KeepChar

Espressione del grafico

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `InputText` che contiene la stringa di testo originale da elaborare.
  - `CharsToKeep` che contiene l'elenco dei caratteri da conservare dalla stringa di testo originale dopo l'elaborazione.

### Script di caricamento

Example:

```
Load * inline [  
  InputText, CharsToKeep  
  'a1b2c3', '123'  
  'a^b^c', 'abc'  
  'A^b^c', 'abc'  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- InputText
- CharsToKeep

Creare la seguente dimensione calcolata:

- `=KeepChar(InputText, CharsToKeep)`, per calcolare i caratteri in `InputText` che corrispondono ai caratteri in `CharsToKeep`.

Tabella dei risultati

| InputText | CharsToKeep | KeepChar(InputText, CharsToKeep) |
|-----------|-------------|----------------------------------|
| a1b2c3    | 123         | 123                              |
| a^b^c     | abc         | abc                              |
| A^b^c     | abc         | bc                               |

La colonna `keepchar(InputText, charsToKeep)` restituisce solo i caratteri in `InputText` che corrispondono esattamente ai caratteri in `charsToKeep`. Per esempio, la terza riga restituisce `bc`, poiché questi sono gli unici caratteri in `InputText` che corrispondono ai caratteri in `charsToKeep`. Il carattere in maiuscolo `A` non viene restituito perché gli elementi in `charsToKeep` sono solo in minuscolo.

### Script di caricamento

Il codice seguente mostra come utilizzare la funzione in uno script di caricamento.

Example:

```
Load
*,
keepchar(String1, String2) as keepChar;
Load * inline [
String1, String2
'a1b2c3', '123'
];
```

Tabella dei risultati

| String1 | String2 | KeepChar |
|---------|---------|----------|
| a1b2c3  | 123     | 123      |

### Esempio: scenario per l'applicazione della funzione KeepChar per analizzare il testo non strutturato

Espressione del grafico

#### Panoramica

In questo esempio si mostra come utilizzare la funzione per analizzare il testo non strutturato (lettere, numeri e simboli) e fornire un output strutturato che conserva solo i caratteri numerici.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `InputText` che contiene la stringa di testo originale da elaborare.
  - `CharsToKeep` che contiene l'elenco dei caratteri da conservare dalla stringa di testo originale dopo l'elaborazione.

### Script di caricamento

Example:

```
Load * inline [  
InputText, InputCategory, CharsToKeep  
'The numeric code for input processing is 123-456|789.', 'Code', '0123456789'  
'Delivery of batch number: 333*456789.', 'Batch', '0123456789'  
'Find the goods received in bay 16a', 'Bay', '0123456789'  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `InputText`
- `CharsToKeep`

Creare la seguente dimensione calcolata:

- `=KeepChar(InputText, CharsToKeep)`, per calcolare i caratteri in `InputText` che corrispondono ai caratteri in `CharsToKeep`.

Tabella dei risultati

| InputText                                                      | CharsToKeep | KeepChar(InputText, CharsToKeep) |
|----------------------------------------------------------------|-------------|----------------------------------|
| Consegna del numero di lotto: 333*456789                       | 0123456789  | 333456789                        |
| Cerca le merci ricevute a molo 16a                             | 0123456789  | 16                               |
| Il codice numerico per l'elaborazione dell'input è 123-456 789 | 0123456789  | 123456789                        |

Confrontare l'output della funzione `keepchar` con i valori delle stringhe originali di `InputText` caricate nello script. La funzione `keepChar` ha rimosso con successo tutti i caratteri non numerici dal testo di input.

### Vedere anche:

p [PurgeChar \(page 1817\)](#)

## Left

**Left()** restituisce una stringa costituita dai primi caratteri (posizionati più a sinistra) della stringa di input, in cui il numero di caratteri viene stabilito dal secondo argomento.

### Sintassi:

```
Left(text, count)
```

Tipo di dati restituiti: stringa

#### Argomenti

| Argomento    | Descrizione                                                                                          |
|--------------|------------------------------------------------------------------------------------------------------|
| <b>text</b>  | La stringa originale.                                                                                |
| <b>count</b> | Definisce il numero di caratteri da includere partendo dal lato sinistro della stringa <b>text</b> . |

#### Esempio: espressione del grafico

| Esempio             | Risultato       |
|---------------------|-----------------|
| Left( 'abcdef', 3 ) | Restituisce abc |

### Esempio: principi fondamentali della funzione Left

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `InputText`
  - `NoOfChars`

#### Script di caricamento

Example:

```
Load * inline [  
  InputText, NoOfChars  
  'abcdef', 3  
  '2021-07-14', 4  
  '2021-07-14', 2  
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- InputText
- NoOfChars

Creare la seguente dimensione calcolata:

- =Left(InputText,NoOfChars)

Tabella dei risultati

| InputText  | NoOfChars | Left(InputText,NoOfChars) |
|------------|-----------|---------------------------|
| abcdef     | 3         | abc                       |
| 2021-07-14 | 2         | 20                        |
| 2021-07-14 | 4         | 2021                      |

La prima riga restituisce abc, in quanto questi sono i primi 3 caratteri da sinistra della stringa InputText.

La seconda riga restituisce 20, in quanto questi sono i primi 2 caratteri da sinistra della stringa InputText.

La terza riga restituisce 2021, in quanto questi sono i primi 4 caratteri da sinistra della stringa InputText.

### Script di caricamento

Il codice seguente mostra come utilizzare la funzione in uno script di caricamento.

Example:

```
Load
*,
left(Text,NoOfChars) as Left;
Load * inline [
Text, NoOfChars
'abcdef', 3
'2021-07-14', 4
'2021-07-14', 2
];
```

Tabella dei risultati

| Testo      | NoOfChars | Sinistra |
|------------|-----------|----------|
| abcdef     | 3         | abc      |
| 2021-07-14 | 4         | 2021     |
| 2021-07-14 | 2         | 20       |

### Esempio: scenario per l'applicazione della funzione Left

Espressione del grafico

#### Panoramica

Nel seguente esempio si analizzano i codici degli ordini in un set di dati, che contengono il codice della regione, l'anno e il numero dell'ordine, per creare un nuovo campo per l'analisi dei dati.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `OrderID`
  - `CustomerName`
  - `SalesAmount`

#### Script di caricamento

Example:

```
Load * inline [  
OrderID, CustomerName, SalesAmount  
US2024001, John Doe, 1500  
EU2024002, Jane Smith, 2300  
AS2024003, Mike Davis, 1800  
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `OrderID`
- `CustomerName`
- `SalesAmount`

Creare la seguente dimensione calcolata:

- `=Left(OrderID, 2)` per calcolare la regione.

Tabella dei risultati

| OrderID   | CustomerName | SalesAmount | Left(OrderID, 2) |
|-----------|--------------|-------------|------------------|
| AS2024003 | Mike Davis   | 1800        | AS               |
| EU2024002 | Jane Smith   | 2300        | UE               |
| US2024001 | John Doe     | 1500        | US               |

L'output della funzione `Left` analizza il campo `OrderID` per estrarre la regione, creando un nuovo campo da utilizzare per ulteriori analisi dei dati.

### Esempio: scenario per l'applicazione della funzione `Left` avanzata

Espressione del grafico

#### Panoramica

Nel seguente esempio si analizzano i codici dei prodotti in un set di dati, che contengono il codice della categoria del reparto e un identificatore univoco, per creare un nuovo campo per l'analisi dei dati.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Un campo della tabella dati denominata `ProductCode`.

#### Script di caricamento

```
Example:
Load * inline [
ProductCode
ELC12345
HOM67890
FAS09876
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `ProductCode`

Creare la seguente dimensione calcolata:

- `=Pick(Match(Left(ProductCode,3),'ELC','HOM','FAS'),'Electronics','Home Appliances','Fashion')`, per calcolare il reparto.

Tabella dei risultati

| ProductCode | Pick(Match(left(ProductCode,3),'ELC','HOM','FAS'),'Electronics','Home Appliances','Fashion') |
|-------------|----------------------------------------------------------------------------------------------|
| ELC12345    | Elettronica                                                                                  |
| FAS09876    | Moda                                                                                         |
| HOM67890    | Elettrodomestici per la casa                                                                 |

L'output utilizza le funzioni `Pick`, `Match` e `Left` per estrarre il reparto dal codice del prodotto, creando un nuovo campo per l'analisi dei dati.

p Vedere anche [Index \(page 1756\)](#), che consente un'analisi della stringa più complessa.

### Len

**Len()** restituisce la lunghezza della stringa di input.

**Sintassi:**

```
Len (text)
```

**Tipo di dati restituiti:** numero intero

| Argomenti |                         |
|-----------|-------------------------|
| Argomento | Descrizione             |
| text      | La stringa da valutare. |

Esempio: espressione del grafico

| Esempio      | Risultato     |
|--------------|---------------|
| Len('Peter') | Restituisce 5 |

### Esempio: principi fondamentali della funzione Len

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Un campo della tabella dati denominata `customerComment`. Il campo contiene la stringa di testo originale da valutare.

#### Script di caricamento

Example:

```
Load * inline [  
  CustomerComment  
  Please deliver after 5 PM.  
  Thank you for the quick service!  
  Can you add a gift wrap?  
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `CustomerComment`

Creare la seguente misura:

- `=Len(CustomerComment)`

Tabella dei risultati

| CustomerComment                       | Len(CustomerComment) |
|---------------------------------------|----------------------|
| Può aggiungere una confezione regalo? | 24                   |
| Si prega di consegnare dopo le 17:00. | 26                   |
| Grazie per il servizio rapido!        | 32                   |

L'output della misura `Len(CustomerComment)` restituisce la lunghezza della stringa di input `CustomerComment`.

### Esempio: scenario per l'applicazione della funzione Len

Espressione del grafico

#### Panoramica

Nel seguente esempio si analizza il feedback dei clienti prendendo in considerazione la lunghezza di ogni commento come indicatore dell'importanza o del livello di dettaglio.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `TicketID`
  - `Description`

#### Script di caricamento

Example:

```
Load * inline [
TicketID, Description
1001, "I received the wrong product."
1002, "The package arrived damaged, and I would like a replacement."
1003, "I've been trying to reset my password for two days, and I haven't received an email."
1004, "My order status shows 'Delivered' but I have not received my package yet."
1005, "Is product XYZ available in size Large?"
1006, "I need help with an exchange for a faulty product, and I've attached photos."
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `TicketID`
- `Description`

Creare la seguente dimensione calcolata:

- `=If(Len(Description) < 30, 'Short',If(Len(Description) <= 50, 'Medium', 'Long'))`, per calcolare e assegnare una categoria (Breve, Medio, Lungo) al ticket in base alla lunghezza della descrizione.

Tabella dei risultati

| TicketID | Descrizione                                                                                | If(Len(Description) < 30, 'Short',If(Len(Description) <= 50, 'Medium', 'Long')) |
|----------|--------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| 1001     | Ho ricevuto il prodotto sbagliato.                                                         | Breve                                                                           |
| 1002     | Il pacco è arrivato danneggiato e vorrei una sostituzione.                                 | Lungo                                                                           |
| 1003     | Sto cercando di reimpostare la mia password da due giorni e non ho ricevuto alcuna e-mail. | Lungo                                                                           |
| 1004     | Il prodotto XYZ è disponibile nella taglia Large?                                          | Medio                                                                           |
| 1005     | Lo stato del mio ordine indica "Consegnato", ma non ho ancora ricevuto il mio pacco.       | Lungo                                                                           |
| 1006     | Ho bisogno di aiuto per la sostituzione di un prodotto difettoso e ho allegato delle foto. | Lungo                                                                           |

L'output della dimensione calcolata mostra come è possibile utilizzare la funzione `Len` per classificare i dati interpretando la lunghezza delle stringhe di testo.

### Esempio: scenario per l'applicazione della funzione `Len` con manipolazione delle stringhe

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Un campo della tabella dati denominata `InputText`.

#### Script di caricamento

Example:

```
Load * inline [  
InputText  
this is a sample text string  
capitalize first letter only  
];
```

## Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- InputText

Creare le seguenti dimensioni calcolate:

- =Upper(Left(InputText,1)) , per convertire la prima lettera della stringa di testo in maiuscolo.
- =Mid(InputText,Len(upper(Left(InputText,1)))+1), per rimuovere la prima lettera dalla stringa di testo.
- =Upper(Left(InputText,1)) & Mid(InputText,Len(upper(Left(InputText,1)))+1), per combinare l'output della prima dimensione calcolata con l'output della seconda dimensione calcolata.

Tabella dei risultati

| InputText                                         | Upper(Left<br>(InputText,1)) | Mid(InputText,Len<br>(upper(Left<br>(InputText,1)))+1) | Upper(left(InputText,1)) & mid<br>(InputText,len(upper(left<br>(InputText,1)))+1) |
|---------------------------------------------------|------------------------------|--------------------------------------------------------|-----------------------------------------------------------------------------------|
| questo è un<br>esempio di stringa<br>di testo     | Questo                       | è un esempio di stringa<br>di testo                    | Questo è un esempio di stringa di<br>testo                                        |
| scrivere solo la<br>prima lettera in<br>maiuscolo | Scrivere                     | solo la prima lettera in<br>maiuscolo                  | Scrivere solo la prima lettera in<br>maiuscolo                                    |

Nella prima dimensione calcolata, le funzioni `upper` e `Left` vengono combinate per restituire la prima lettera di `InputText` in maiuscolo. Nella seconda dimensione calcolata, la funzione `Mid` utilizza la funzione `Len` per restituire una stringa di testo che rimuove il primo carattere da `InputText`. La terza dimensione calcolata combina la prima e la seconda dimensione calcolata e restituisce la stringa `InputText` con in primo carattere in maiuscolo.

## Script di caricamento

In questo esempio si utilizzano le stesse funzioni (`upper`, `Mid` e `Len`) dello scenario dell'espressione del grafico. Lo script di caricamento crea un nuovo campo, `NewInputText`, che restituisce `InputText` con il primo carattere in maiuscolo.

Example:

```
Load InputText, First&Second as NewInputText;
Load *, mid(InputText,len(First)+1) as Second;
Load *, upper(left(InputText,1)) as First;
Load * inline [
InputText
this is a sample text string
capitalize first letter only ];
```

Tabella dei risultati

| InputText                    | NewInputText                 |
|------------------------------|------------------------------|
| this is a sample text string | This is a sample text string |
| capitalize first letter only | Capitalize first letter only |

## LevenshteinDist

**LevenshteinDist()** restituisce la distanza Levenshtein tra due stringhe. Viene definita come il numero minimo di modifiche a un singolo carattere (inserimenti, eliminazioni o sostituzioni) richiesto per cambiare una stringa con un'altra. La funzione è utile per i confronti tra stringhe fuzzy.

### Sintassi:

```
LevenshteinDist(text1, text2)
```

Tipo di dati restituiti: numero intero

Argomenti

| Argomento    | Descrizione                                                                                                               |
|--------------|---------------------------------------------------------------------------------------------------------------------------|
| <b>text1</b> | La prima stringa.                                                                                                         |
| <b>text2</b> | La seconda stringa che sarà confrontata con la prima per calcolare il numero minimo di modifiche di un singolo carattere. |

Esempio: espressione del grafico

| Esempio                                | Risultato     |
|----------------------------------------|---------------|
| LevenshteinDist( 'Kitten', 'Sitting' ) | Restituisce 3 |

## Esempio: principi fondamentali della funzione LevenshteinDist

Espressione del grafico

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Un campo della tabella dati denominata `InputText`.

### Script di caricamento

Example:

```
Load * inline [
InputText
```

```
Sliver  
SSiver  
SSiveer  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- InputText

Creare la seguente misura:

- =LevenshteinDist('Silver', InputText), per calcolare il numero minimo di modifiche di un singolo carattere necessarie per cambiare il valore della stringa per InputText nella parola silver.

Tabella dei risultati

| InputText | LevenshteinDist('Silver', InputText) |
|-----------|--------------------------------------|
| Sliver    | 2                                    |
| SSiveer   | 3                                    |
| SSiver    | 2                                    |

L'output della funzione LevenshteinDist restituisce il numero di modifiche necessarie per cambiare InputText nel testo previsto, silver. Ad esempio, la prima riga richiede due modifiche per modificare la parola Sliver in silver. La seconda riga richiede 3 modifiche: 1) Eliminazione del carattere extra s. 2) Eliminazione del carattere extra e. 3) Inserimento di un nuovo carattere l.

### Esempio: scenario per l'applicazione della funzione LevenshteinDist

Espressione del grafico

#### Panoramica

In questo esempio si mostra come consolidare i nomi dei prodotti generati da sistemi diversi. I nomi dei prodotti non sono sempre scritti nello stesso modo a causa di errori di battitura, abbreviazioni, spaziatura o altre variazioni. Utilizzando la funzione LevenshteinDist, è possibile misurare la somiglianza tra due nomi di prodotti e identificare quali si riferiscono probabilmente allo stesso prodotto, anche se i nomi non sono identici.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata Example.
- I seguenti campi nella tabella dati:
  - ProductA
  - ProductB

### Script di caricamento

Example:

```
Load * inline [
ProductA, ProductB
Coca Cola 330ml, CocaCola 330 ml
Pepsi 500 ml, Pepsi 500ml
Sprite Zero 600 ml, SpriteZero600ml
Red Bull 250ml, Redbull 250ml
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- ProductA
- ProductB

Creare la seguente misura:

- =LevenshteinDist(ProductA, ProductB), per calcolare il numero minimo di modifiche di un singolo carattere necessarie per cambiare il valore della stringa per ProductB in modo da corrispondere a ProductA.

Tabella dei risultati

| ProductA           | ProductB        | LevenshteinDist(ProductA, ProductB) |
|--------------------|-----------------|-------------------------------------|
| Coca Cola 330ml    | CocaCola 330 ml | 2                                   |
| Pepsi 500 ml       | Pepsi 500ml     | 1                                   |
| Red Bull 250ml     | Redbull 250ml   | 2                                   |
| Sprite Zero 600 ml | SpriteZero600ml | 3                                   |

La distanza di Levenshtein è un tipo di corrispondenza parziale ampiamente utilizzato nei correttori ortografici, per il riconoscimento ottico dei caratteri e nei sistemi di correzione per aree come la gestione dei dati dei clienti, i sistemi di inventario e l'elaborazione dei documenti, dove si verificano frequentemente lievi variazioni nel testo.

### Script di caricamento

Il codice seguente mostra come utilizzare la funzione in uno script di caricamento.

### Script di caricamento

Example:

```
Load *, recno() as ID;
Load 'Silver' as String_1,* inline [
String_2
Sliver
SSiver
```

```
SSiveer ];
```

Example:

```
Load *, recno()+3 as ID;
Load 'Gold' as String_1,* inline [
String_2
Bold
Bool
Bond ];
```

Example:

```
Load *, recno()+6 as ID;
Load 'Ove' as String_1,* inline [
String_2
Ove
Uve
Üve ];
```

Example:

```
Load *, recno()+9 as ID;
Load 'ABC' as String_1,* inline [
String_2
DEFG
abc
𐀀𐀀𐀀 ];
```

```
set nullinterpret = '<NULL>';
```

Example:

```
Load *, recno()+12 as ID;
Load 'X' as String_1,* inline [
String_2
,,
<NULL>
1 ];
```

R1:

```
Load
ID,
String_1,
String_2,
LevenshteinDist(String_1, String_2) as LevenshteinDistance
resident Example;
```

```
Drop table Example;
```

Tabella dei risultati

| ID | String_1 | String_2 | LevenshteinDistance |
|----|----------|----------|---------------------|
| 1  | Silver   | Sliver   | 2                   |
| 2  | Silver   | SSiver   | 2                   |
| 3  | Silver   | SSiveer  | 3                   |

| ID | String_1 | String_2  | LevenshteinDistance |
|----|----------|-----------|---------------------|
| 4  | Gold     | Grassetto | 1                   |
| 5  | Gold     | Bool      | 3                   |
| 6  | Gold     | Bond      | 2                   |
| 7  | Ove      | Ove       | 0                   |
| 8  | Ove      | Uve       | 1                   |
| 9  | Ove      | Üve       | 1                   |
| 10 | ABC      | DEFG      | 4                   |
| 11 | ABC      | abc       | 3                   |
| 12 | ABC      | ビビビ       | 3                   |
| 13 | X        |           | 1                   |
| 14 | X        | -         | 1                   |
| 15 | X        | 1         | 1                   |

## Lower

**Lower()** applica il formato minuscolo a tutti i caratteri della stringa di input.

**Sintassi:**

**Lower** (text)

**Tipo di dati restituiti:** stringa

### Argomenti

| Argomento | Descrizione             |
|-----------|-------------------------|
| text      | La stringa da valutare. |

### Esempio: espressione del grafico

| Esempio         | Risultato        |
|-----------------|------------------|
| Lower( 'abcd' ) | Restituisce abcd |

## Esempio: principi fondamentali della funzione Lower

Espressione del grafico

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Un campo della tabella dati denominata `InputText`.

### Script di caricamento

```
Example:  
Load * inline [  
InputText  
rHode iSland  
washingTon d.C.  
new york  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `InputText`

Creare la seguente dimensione calcolata:

- `=Lower(InputText)` per convertire tutti i caratteri del campo `InputText` in minuscolo.

Tabella dei risultati

| <b>InputText</b> | <b>Lower(InputText)</b> |
|------------------|-------------------------|
| new york         | new york                |
| rHode iSland     | rhode island            |
| washingTon d.C.  | washington d.c.         |

L'output della funzione `Lower` converte tutti i valori in minuscolo.

### Script di caricamento

Il codice seguente mostra come utilizzare la funzione in uno script di caricamento.

```
Load  
String,  
Lower(String)  
Inline  
[String  
rHode iSland  
washingTon d.C.  
new york];
```

Tabella dei risultati

| <b>Stringa</b>  | <b>Lower(String)</b> |
|-----------------|----------------------|
| rHode iSland    | rhode island         |
| washingTon d.C. | washington d.c.      |
| new york        | new york             |

### Esempio: scenario per l'applicazione della funzione Lower

Espressione del grafico

#### Panoramica

Un set di dati con i dati dei clienti contiene gli indirizzi e-mail per il dominio, scritti in lettere maiuscole e minuscole. L'azienda vuole standardizzare gli indirizzi in modo da utilizzare unicamente le minuscole.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `CustomerID`
  - `EmailAddress`
  - `Inquiry`

#### Script di caricamento

Example:

```
Load * inline [
CustomerID,    EmailAddress, Inquiry
1001, john.doe@Gmail.com, Inquiry about product return
1002, mary.jane@yahoo.com, Inquiry about order status
1003, a.smith@GMAIL.COM, Inquiry about shipping details
1004, lisa.black@outlook.com, General product inquiry
1005, james.wong@Yahoo.COM, Question about pricing
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `CustomerID`
- `EmailAddress`

Creare la seguente dimensione calcolata:

- `=Lower(EmailAddress)` per separare la parte del dominio dell'e-mail come una nuova colonna e convertire i caratteri in minuscolo.

Tabella dei risultati

| CustomerID | EmailAddress        | Lower(EmailAddress) |
|------------|---------------------|---------------------|
| 1001       | john.doe@Gmail.com  | john.doe@gmail.com  |
| 1002       | MARY.JANE@YAHOO.COM | mary.jane@yahoo.com |

| CustomerID | EmailAddress           | Lower(EmailAddress)    |
|------------|------------------------|------------------------|
| 1003       | a.smith@GMAIL.COM      | a.smith@gmail.com      |
| 1004       | lisa.black@Outlook.Com | lisa.black@outlook.com |
| 1005       | james.wong@Yahoo.COM   | james.won@yahoo.com    |

L'output della funzione `Lower` restituisce tutti gli indirizzi e-mail in minuscolo.

## LTrim

`LTrim()` restituisce la stringa di input senza spazi iniziali.

**Sintassi:**

**`LTrim(text)`**

**Tipo di dati restituiti:** stringa

### Argomenti

| Argomento   | Descrizione             |
|-------------|-------------------------|
| <b>text</b> | La stringa da valutare. |

### Esempio: espressioni del grafico

| Esempio                      | Risultato       |
|------------------------------|-----------------|
| <code>LTrim( ' abc' )</code> | Restituisce abc |
| <code>LTrim( 'abc ' )</code> | Restituisce abc |

## Esempio: principi fondamentali della funzione LTrim

Espressione del grafico

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Un campo della tabella dati denominata `InputText`.

### Script di caricamento

```
Set verbatim=1;
Example:
Load * inline [
InputText
'   abc   '
' def '];
```



L'istruzione "set verbatim=1" è inclusa nell'esempio per assicurare che gli spazi non vengano automaticamente tagliati prima della dimostrazione della funzione `LTrim`. Vedere [Verbatim \(page 223\)](#) per ulteriori informazioni.

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `InputText`

Creare la seguente dimensione calcolata:

- `=LTrim(InputText)` per rimuovere gli spazi iniziali da `InputText`.

Tabella dei risultati

| <code>InputText</code> | <code>LTrim(InputText)</code> |
|------------------------|-------------------------------|
| ' abc '                | 'abc '                        |
| 'def '                 | 'def '                        |

L'output della funzione `LTrim` ha rimosso tutti gli spazi iniziali a sinistra del testo originale, ma ha mantenuto tutti gli spazi finali.

### Esempio: scenario per l'applicazione della funzione `LTrim`

Espressione del grafico

#### Panoramica

Un sistema di gestione delle relazioni con i clienti (CRM) contiene record con i dati immessi in modo incoerente che includono degli spazi iniziali extra. Ai fini della creazione di report, i dati devono essere puliti per rimuovere tali spazi e garantire un ordinamento e un raggruppamento corretto dei nomi dei clienti.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Un campo della tabella dati denominata `CustomerName`.

#### Script di caricamento

```
set verbatim=1;
Example:
Load * inline [
CustomerName
' John Doe'
'Jane Smith'
' Michael Johnson'
```

```
'Emily Davis'  
];
```



L'istruzione "set verbatim=1" è inclusa nell'esempio per assicurare che gli spazi non vengano automaticamente tagliati prima della dimostrazione della funzione `LTrim`. Vedere [Verbatim \(page 223\)](#) per ulteriori informazioni.

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- CustomerName

Creare la seguente dimensione calcolata:

- `=LTrim(CustomerName)` per rimuovere tutti gli spazi iniziali da CustomerName.

Tabella dei risultati

| CustomerName       | LTrim(CustomerName) |
|--------------------|---------------------|
| ' Michael Johnson' | 'Michael Johnson'   |
| ' John Doe'        | 'John Doe'          |
| 'Emily Davis'      | 'Emily Davis'       |
| 'Jane Smith'       | 'Jane Smith'        |

L'output mostra che la funzione `LTrim` ha rimosso tutti gli spazi iniziali dai valori della stringa originale in CustomerName.

### Esempio: scenario per l'applicazione della funzione LTrim avanzata

Espressione del grafico

#### Panoramica

In questo esempio si mostra come rimuovere tutti gli spazi iniziali dalla stringa di testo originale. L'espressione del grafico include misure che utilizzano la funzione `Len` per contare i caratteri nella stringa prima e dopo aver utilizzato la funzione `LTrim`.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Un campo della tabella dati denominata `string`.

#### Script di caricamento

```
set verbatim=1;  
Example:
```

```
Load * inline [
String
'   abc   '
' def '];
```



L'istruzione "Set verbatim=1" è inclusa nell'esempio per assicurare che gli spazi non vengano automaticamente tagliati prima della dimostrazione della funzione `LTrim`. Vedere [Verbatim \(page 223\)](#) per ulteriori informazioni.

## Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- String

Creare la seguente dimensione calcolata:

- `=LTrim(String)` per rimuovere tutti gli spazi iniziali extra.

Creare le seguenti misure:

- `=Len(String)` per calcolare la lunghezza della stringa originale.
- `=Len(LTrim(String))` per calcolare la lunghezza della stringa dopo la rimozione degli spazi iniziali.

Tabella dei risultati

| Stringa | LTrim(String) | Len(String) | Len(LTrim(String)) |
|---------|---------------|-------------|--------------------|
| ' abc ' | 'abc '        | 10          | 7                  |
| ' def ' | 'def '        | 6           | 5                  |

Confrontando l'output della funzione `LTrim` con i valori delle stringhe originali nello script, si può notare come tutti gli spazi iniziali siano stati rimossi.

## Script di caricamento

Il codice seguente mostra come utilizzare la funzione in uno script di caricamento.

Set verbatim=1;

Example:

```
Load *,
len(LtrimString) as LtrimStringLength;
Load *,
ltrim(String) as LtrimString;
Load *,
len(String) as StringLength;
Load * Inline [
String
'   abc   '
' def '];
```



L'istruzione "Set verbatim=1" è inclusa nell'esempio per assicurare che gli spazi non vengano automaticamente tagliati prima della dimostrazione della funzione `Ltrim`. Vedere [Verbatim \(page 223\)](#) per ulteriori informazioni.

Tabella dei risultati

| Stringa | StringLength | LtrimString | LtrimStringLength |
|---------|--------------|-------------|-------------------|
| abc     | 10           | abc         | 7                 |
| def     | 6            | def         | 5                 |

**Vedere anche:**

p [RTrim \(page 1842\)](#)

## MatchRegEx

**MatchRegEx()** confronta la stringa di input con uno o più modelli di espressione regolare specificati e restituisce la posizione numerica dei modelli di espressione regolare corrispondenti. Se non vengono trovate corrispondenze, la funzione restituisce 0. Questa funzione cerca solo le corrispondenze esatte.

Questa funzione esegue operazioni regex che distinguono tra maiuscole e minuscole. In alternativa, è possibile utilizzare la variante **MatchRegExI()** per eseguire operazioni regex senza distinzione tra maiuscole e minuscole.

**Sintassi:**

**MatchRegEx** (text, regex1 [ , regex2, ...regexN] )

**Tipo di dati restituiti:** numero intero

Argomenti

| Argomento    | Descrizione                                                                                                                                                                                       |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>text</b>  | Il testo della stringa di input a cui si desidera far corrispondere una o più espressioni regolari.                                                                                               |
| <b>regex</b> | L'espressione regolare che bisogna cercare di abbinare alla stringa di input. È possibile specificare più di un'espressione regolare in un'unica espressione, separata come argomenti successivi. |

Esempi di funzioni

| Esempio                                                                   | Risultato                                                                 |
|---------------------------------------------------------------------------|---------------------------------------------------------------------------|
| <code>MatchRegEx('abc123', '[a-z]', '[0-9]', '[a-z]+[0-9]+', '.*')</code> | Restituisce 3. Il terzo modello regex è il primo che corrisponde ai dati. |

| Esempio                                            | Risultato                                                                                                                      |
|----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| <code>MatchRegex('ABC', '[abc]+', '[123]')</code>  | Restituisce 0, perché <b>MatchRegex()</b> fa distinzione tra maiuscole e minuscole.                                            |
| <code>MatchRegexI('ABC', '[abc]+', '[123]')</code> | Restituisce 1. Viene utilizzata la variante senza distinzione tra maiuscole e minuscole della funzione, <b>MatchRegexI()</b> . |

### Casi di utilizzo

I casi d'uso di questa funzione includono:

- Convalidare il testo per soddisfare gli standard di formattazione e di conformità e per identificare gli errori nei dati.
- Identificare le differenze nel modo in cui il testo viene formattato. Ad esempio, è possibile identificare quanti numeri di telefono utilizzano il formato A, il formato B e così via.

### Esempio 1 - script di caricamento per analizzare i formati dei numeri di telefono

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- La tabella `BusinessContactInfo` contenente i numeri di telefono di contatto di varie aziende. I numeri di telefono seguono il sistema NANP (piano di numerazione nordamericano) e hanno 10 cifre.
- Un campo calcolato, `PhoneNumberFormat`, per valutare se ogni numero di telefono corrisponde a uno di una serie specifica di formati di numeri di telefono.

Vi sono tre diversi formati che bisogna verificare: `(###) ###-####`, `###-###-####` e `#####`.

#### Script di caricamento

`BusinessContactInfo:`

```
Load MatchRegex(ContactInfo, '\({0,1}[0-9]{3}\) [0-9]{3}-[0-9]{4}', '[0-9]{3}-[0-9]{3}-[0-9]{4}', '[0-9]{10}') as PhoneNumberFormat,
* Inline `
ID      CompanyName      ContactInfo
1       Company A       (123) 456-7890
2       Company B       0123456790
3       Company C       234-567-8901
4       Company D       (024) 680-2456
5       Company E       (135) 791-3579
6       Company F       8901234567
7       Company G       235-235-2352
8       Company H       (555) (555) (5555)
```

```

9      Company I      2342342343
10     Company J      444-444-4444
` (delimiter is '\t');

```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- CompanyName
- ContactInfo
- PhoneNumberFormat

Tabella dei risultati

| CompanyName | ContactInfo        | PhoneNumberFormat |
|-------------|--------------------|-------------------|
| Company A   | (123) 456-7890     | 1                 |
| Company B   | 0123456790         | 3                 |
| Company C   | 234-567-8901       | 2                 |
| Company D   | (024) 680-2456     | 1                 |
| Company E   | (135) 791-3579     | 1                 |
| Company F   | 8901234567         | 3                 |
| Company G   | 235-235-2352       | 2                 |
| Company H   | (555) (555) (5555) | 0                 |
| Company I   | 2342342343         | 3                 |
| Company J   | 444-444-4444       | 2                 |

Da qui, ad esempio, è possibile creare dei KPI per calcolare il numero totale per il formato di ogni numero di telefono, nonché il numero di numeri di telefono che non hanno un formato riconosciuto:

Esempi di espressioni di aggregazione per contare le occorrenze di ogni formato

| Formato             | Numero di istanze | Espressione del grafico                                  |
|---------------------|-------------------|----------------------------------------------------------|
| (###) ###-####      | 3                 | =Count({<PhoneNumberFormat={'1'}>}<br>PhoneNumberFormat) |
| ###-###-####        | 3                 | =Count({<PhoneNumberFormat={'2'}>}<br>PhoneNumberFormat) |
| #####               | 3                 | =Count({<PhoneNumberFormat={'3'}>}<br>PhoneNumberFormat) |
| Unrecognized format | 1                 | =Count({<PhoneNumberFormat={'0'}>}<br>PhoneNumberFormat) |

### Esempio 2 - espressione del grafico per identificare il formato ISBN

Espressione del grafico e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- La tabella `books` che elenca i codici ISBN di vari libri.

Da questi dati, bisogna identificare se ogni codice ISBN è un codice valido di 10 o 13 cifre, in base alla sintassi specificata in un modello regex.

#### Script di caricamento

Books:

```
Load * Inline [
Book
ISBN 123-3-1234-1234-0
ISBN 012-2-0123-0123-4
ISBN 000-1-0123-0123-2
ISBN 0-111-23456-7
ISBN 555-2-5555-5555-3
ISBN 222-4-2222-2222-2
ISBN 1-901-23456-8
ISBN 333-3-3333-3333-3
ISBN 555-1-5151-5151-3
ISBN 232-1-2323-2323-1
ISBN 2-444-44444-4
ISBN 888-0-9999-0000-0
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- Book

Aggiungere questa dimensione calcolata e assegnarle l'etichetta *ISBN Format*:

```
=if(MatchRegEx(Book, 'ISBN[ ]*([0-9]{3})-([0-9]{1})-([0-9]{4})-([0-9]{4})-([0-9]{1})', 'ISBN[ ]*([0-9]{1})-([0-9]{3})-([0-9]{5})-([0-9]{1})')=1, '13-digit',
    if(MatchRegEx(Book, 'ISBN[ ]*([0-9]{3})-([0-9]{1})-([0-9]{4})-([0-9]{4})-([0-9]{1})', 'ISBN[ ]*([0-9]{1})-([0-9]{3})-([0-9]{5})-([0-9]{1})')=2, '10-digit',
        'Invalid ISBN'))
```

Tabella dei risultati

| Book                   | ISBN Format |
|------------------------|-------------|
| ISBN 000-1-0123-0123-2 | 13-digit    |
| ISBN 0-111-23456-7     | 10-digit    |
| ISBN 1-901-23456-8     | 10-digit    |
| ISBN 2-444-44444-4     | 10-digit    |
| ISBN 012-2-0123-0123-4 | 13-digit    |
| ISBN 123-3-1234-1234-0 | 13-digit    |
| ISBN 222-4-2222-2222-2 | 13-digit    |
| ISBN 232-1-2323-2323-1 | 13-digit    |
| ISBN 333-3-3333-3333-3 | 13-digit    |
| ISBN 555-1-5151-5151-3 | 13-digit    |
| ISBN 555-2-5555-5555-3 | 13-digit    |
| ISBN 888-0-9999-0000-0 | 13-digit    |

## Mid

**Mid()** restituisce la parte della stringa di input che inizia nella posizione del carattere definito dal secondo argomento, 'start', e che restituisce il numero di caratteri definito dal terzo argomento, 'count'. Se viene omesso 'count', viene restituita la parte rimanente della stringa di input. Il primo carattere nella stringa di input viene contrassegnato con il numero 1.

### Sintassi:

```
Mid(text, start[, count])
```

**Tipo di dati restituiti:** stringa

### Argomenti

| Argomento    | Descrizione                                                                                                                                        |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>text</b>  | La stringa originale.                                                                                                                              |
| <b>start</b> | Numero intero che definisce la posizione del primo carattere di <b>text</b> da includere.                                                          |
| <b>count</b> | Definisce la lunghezza della stringa di output. Se omesso, verranno inclusi tutti i caratteri a partire dalla posizione definita da <b>start</b> . |

### Esempio: espressioni del grafico

| Esempio                            | Risultato        |
|------------------------------------|------------------|
| <code>Mid( 'abcdef', 3 )</code>    | Restituisce cdef |
| <code>Mid( 'abcdef', 3, 2 )</code> | Restituisce cd   |

### Esempio: principi fondamentali della funzione Mid

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `InputText`, la stringa di testo originale.
  - `StartPosition`, la posizione iniziale del primo carattere da includere dalla stringa di testo originale dopo l'elaborazione.
  - `CharacterCount`, il numero di caratteri da restituire (parametro facoltativo).

#### Script di caricamento

Example:

```
Load * inline [  
InputText, StartPosition, CharacterCount  
abcdef, 3, 2  
abcdef, 2, 3  
210714, 3, 2  
210714, 2, 3  
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `InputText`
- `StartPosition`
- `CharacterCount`

Creare le seguenti dimensioni calcolate:

- `=mid(InputText,StartPosition)`, per calcolare i caratteri da estrarre da `InputText` a partire dal valore del campo `StartPosition`. La funzione restituirà tutti i caratteri dal valore `StartPosition` alla fine di `InputString` perché non è stato fornito alcun parametro `Count`.
- `=mid(InputText,StartPosition,CharacterCount)`, per calcolare i caratteri da estrarre da `InputText` a partire dal valore del campo `StartPosition`. Il valore di `CharacterCount` determina il numero di caratteri restituiti, a partire dal valore del campo `StartPosition`.

Tabella dei risultati

| TestoInput | PosizioneIniziale | ConteggioCaratteri | Mid (TestoInput, PosizioneIniziale) | Mid (TestoInput, PosizioneIniziale, NumeroCaratteri) |
|------------|-------------------|--------------------|-------------------------------------|------------------------------------------------------|
| abcdef     | 2                 | 3                  | bcdef                               | bcd                                                  |
| abcdef     | 3                 | 2                  | cdef                                | cd                                                   |
| 20210714   | 2                 | 3                  | 10714                               | 107                                                  |
| 20210714   | 3                 | 2                  | 0714                                | 07                                                   |

La prima riga restituisce il valore bcdef per la prima espressione. L'espressione inizia dalla posizione 2 e restituisce tutti i caratteri perché l'argomento count non è impostato. Anche la seconda espressione inizia dalla posizione 2 ma restituisce solo i caratteri bcd perché l'argomento count è uguale a 2.

La seconda riga restituisce il valore cdef per la prima espressione. L'espressione inizia dalla posizione 3 e restituisce tutti i caratteri perché l'argomento count non è impostato. Anche la seconda espressione inizia dalla posizione 3 ma restituisce solo i caratteri cd perché l'argomento count è uguale a 2.

La terza riga restituisce il valore 10714 per la prima espressione. L'espressione inizia dalla posizione 2 e restituisce tutti i caratteri perché l'argomento count non è impostato. Anche la seconda espressione inizia dalla posizione 2 ma restituisce solo i caratteri 107 perché l'argomento count è uguale a 3.

La quarta riga restituisce il valore 0714 per la prima espressione. L'espressione inizia dalla posizione 3 e restituisce tutti i caratteri perché l'argomento count non è impostato. Anche la seconda espressione inizia dalla posizione 2 ma restituisce solo i caratteri 07 perché l'argomento count è uguale a 2.

### Script di caricamento

Il codice seguente mostra come utilizzare la funzione in uno script di caricamento.

```
Example:
Load *,
mid(Text,StartPosition) as Mid1,
mid(Text,StartPosition,Count) as Mid2;
Load * inline [
Text, StartPosition, Count
'abcdef', 3, 2
'abcdef', 2, 3
'210714', 3, 2
'210714', 2, 3
];
```

Tabella dei risultati

| Testo  | PosizioneIniziale | Mid1  | Conteggio | Mid2 |
|--------|-------------------|-------|-----------|------|
| abcdef | 3                 | cdef  | 2         | cd   |
| abcdef | 2                 | bcdef | 3         | bcd  |

| Testo  | PosizioneIniziale | Mid1  | Conteggio | Mid2 |
|--------|-------------------|-------|-----------|------|
| 210714 | 3                 | 0714  | 2         | 07   |
| 210714 | 2                 | 10714 | 3         | 107  |

### Esempio: scenario per l'applicazione della funzione Mid

Espressione del grafico

#### Panoramica

Una sorgente dati contiene una colonna chiamata `orderID` che utilizza il seguente formato strutturato: `ORDYYYY-COUNTRYCODE-ORDERNUMBER`. Per soddisfare un requisito per la creazione di report, è necessario estrarre e presentare il codice del Paese in una colonna separata.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `orders`.
- Un campo della tabella dati denominata `orderID`.

#### Script di caricamento

```
orders:
Load * inline [
orderID
ORD2024-US-123456
ORD2024-UK-654321
ORD2024-CA-789123
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `orderID`

Creare la seguente dimensione calcolata:

- `=Mid(orderID, 9, 2)` per estrarre il codice di due lettere per il Paese dal sito `orderID`.

Tabella dei risultati

| IDOrdine          | Mid(IDOrdine, 9, 2) |
|-------------------|---------------------|
| ORD2024-CA-789123 | CA                  |
| ORD2024-UK-654321 | GB                  |
| ORD2024-US-123456 | US                  |

L'output della funzione `mid` ha estratto con successo i codici di due lettere del Paese dai valori della stringa `orderID`.

**Vedere anche:**

p [Index \(page 1756\)](#)

### Ord

**Ord()** restituisce il valore numerico (ASCII o Unicode) del primo carattere di una stringa. Questa funzione è utile per valutare o confrontare le stringhe in base ai codici dei caratteri sottostanti, ad esempio quando si ordinano o si filtrano stringhe con caratteri non standard.

**Sintassi:**

**Ord** (*text*)

**Tipo di dati restituiti:** numero intero

| Argomenti   |                       |
|-------------|-----------------------|
| Argomento   | Descrizione           |
| <b>text</b> | La stringa originale. |

Esempio: espressioni del grafico

| Esempio                  | Risultato                       |
|--------------------------|---------------------------------|
| <code>ord( 'A' )</code>  | Restituisce il numero intero 65 |
| <code>ord( 'Ab' )</code> | Restituisce il numero intero 65 |

### Esempio: principi fondamentali della funzione Ord

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Chinese`
  - `western`

#### Script di caricamento

Example:

```
Load * inline [
```

Chinese, Western  
古琴, Guqin  
];

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- Chinese
- Western

Creare le seguenti dimensioni calcolate:

- =ord(Chinese) per restituire il valore numerico del valore nel campo denominato Chinese.
- =ord(Western) per restituire il valore numerico del valore nel campo denominato Western.

Tabella dei risultati

| Cinese | Occidentale | ord(Chinese) | ord(Western) |
|--------|-------------|--------------|--------------|
| 古琴     | Guqin       | 21476        | 71           |

### Script di caricamento

Il codice seguente mostra come utilizzare la funzione in uno script di caricamento.

```
Example:  
Load *,  
ord(Chinese) as OrdUnicode,  
ord(Western) as OrdASCII;  
Load * inline [  
Chinese, Western  
古琴, Guqin ];
```

Tabella dei risultati

| Cinese | Occidentale | OrdUnicode | OrdASCII |
|--------|-------------|------------|----------|
| 古琴     | Guqin       | 21476      | 71       |

## Esempio: scenario per l'applicazione della funzione Ord

Espressione del grafico

### Panoramica

Un set di dati con i codici prodotto comprende alcuni codici che sono considerati non validi perché iniziano con un carattere speciale invece che con un carattere alfanumerico. Utilizzando la funzione ord per identificare i codici prodotto non validi, viene creata una nuova misura per etichettare i record come Validi o Non validi.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Products`.
- Un campo della tabella dati denominata `ProductCode`.

### Script di caricamento

```
Products:
Load * inline [
ProductCode
AB123
#EX45
12XY9
DEF456
*KLM78
GH789
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `ProductCode`

Creare le seguenti dimensioni calcolate:

- `=Ord(ProductCode)` per restituire il valore numerico del primo carattere.
- `=If(Ord(ProductCode) < 48 OR (Ord(ProductCode) > 57 AND Ord(ProductCode) < 65) OR Ord(ProductCode) > 122, 'Invalid Product Code', 'Valid Product Code')` per verificare se il valore ordinale del codice è alfanumerico. In caso contrario, il record viene contrassegnato come `Not valid`.

Tabella dei risultati

| Codice prodotto | Ord(ProductCode) | If(Ord(ProductCode) < 48 OR (Ord(ProductCode) > 57 AND Ord(ProductCode) < 65) OR Ord(ProductCode) > 122, 'Invalid Product Code', 'Valid Product Code') |
|-----------------|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| *KLM78          | 42               | Invalid Product Code                                                                                                                                   |
| #EX45           | 35               | Invalid Product Code                                                                                                                                   |
| 12XY9           | 49               | Valid Product Code                                                                                                                                     |
| AB123           | 65               | Valid Product Code                                                                                                                                     |
| DEF456          | 68               | Valid Product Code                                                                                                                                     |
| GH789           | 71               | Valid Product Code                                                                                                                                     |

Confrontando l'output della funzione `ord` con una serie di valori alfanumerici validi, è possibile determinare quali codici prodotto non sono validi.

## PurgeChar

**PurgeChar()** restituisce una stringa costituita dai caratteri contenuti nella stringa di input ('text'), ad eccezione di qualsiasi carattere presente nel secondo argomento ('remove\_chars').

### Sintassi:

```
PurgeChar(text, remove_chars)
```

Tipo di dati restituiti: stringa

### Argomenti

| Argomento           | Descrizione                                                     |
|---------------------|-----------------------------------------------------------------|
| <b>text</b>         | La stringa originale.                                           |
| <b>remove_chars</b> | Una stringa contenente i caratteri in <b>text</b> da rimuovere. |

### Esempio: espressioni del grafico

| Esempio                        | Risultato        |
|--------------------------------|------------------|
| PurgeChar( 'a1b2c3', '123' )   | Restituisce abc  |
| PurgeChar( 'a1b2c3', '1234' )  | Restituisce abc  |
| PurgeChar( 'a1b22c3', '1234' ) | Restituisce abc  |
| PurgeChar( 'a1b2c3', '312' )   | Restituisce abc  |
| PurgeChar ( 'a1b2c3', 'Abc' )  | Restituisce a123 |

## Esempio: principi fondamentali della funzione PurgeChar

Espressione del grafico

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `InputText`
  - `CharsToPurge`

### Script di caricamento

Example:

```
Load * inline [
```

```
InputText, CharstoPurge
'a1b2c3', '123'
'a^b^c', '^'
'A^b^c', 'A^'
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- InputText
- CharstoPurge

Creare la seguente dimensione calcolata:

- =PurgeChar(InputText,CharstoPurge), per calcolare i caratteri in InputText che corrispondono ai caratteri in CharstoPurge e restituire solo i caratteri che non corrispondono.

Results table

| InputText | CharsToPurge | PurgeChar(InputText,CharsToPurge) |
|-----------|--------------|-----------------------------------|
| A^b^c     | A^           | bc                                |
| a^b^c     | ^            | abc                               |
| a1b2c3    | 123          | abc                               |

L'output della funzione PurgeChar restituisce solo i caratteri di InputText che non corrispondono ai caratteri di CharstoPurge. Ad esempio, la prima riga restituisce bc e rimuove gli altri caratteri.

### Script di caricamento

Il codice seguente mostra come utilizzare la funzione in uno script di caricamento.

```
Example:
Load
*,
purgechar(String1, String2) as PurgeChar;
Load * inline [
String1, String2
'a1b2c3', '123'
];
```

Tabella dei risultati

| Stringa1 | Stringa2 | Carattere di eliminazione |
|----------|----------|---------------------------|
| a1b2c3   | 123      | abc                       |

### Esempio: scenario per l'applicazione della funzione PurgeChar

Espressione del grafico

#### Panoramica

Un set di dati contiene i nomi e i numeri di telefono dei clienti. In questo esempio si mostra come analizzare le stringhe dei numeri di telefono per mantenere solo i caratteri numerici e scartare tutti gli altri caratteri, come parentesi, trattini, punti, segni più e spazi.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `CustomerName`
  - `PhoneNumber`

#### Script di caricamento

Example:

```
Load * inline [  
  CustomerName, PhoneNumber  
  John Doe, (123) 456-7890  
  Jane Smith, 123-456-7890  
  Bob Johnson, 123.456.7890  
  Alice Brown, +1 (123) 456 7890  
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `CustomerName`
- `PhoneNumber`

Creare la seguente dimensione calcolata:

- `=PurgeChar(PhoneNumber, '()- .+ ')`, per estrarre i caratteri non numerici dal campo `PhoneNumber` e restituire solo i caratteri numerici.

Results table

| CustomerName | PhoneNumber       | PurgeChar(PhoneNumber,'()- .+ ') |
|--------------|-------------------|----------------------------------|
| Alice Brown  | +1 (123) 456 7890 | 11234567890                      |
| Bob Johnson  | 123.456.7890      | 1234567890                       |

| CustomerName | PhoneNumber    | PurgeChar(PhoneNumber,'()-+. ') |
|--------------|----------------|---------------------------------|
| Jane Smith   | 123-456-7890   | 1234567890                      |
| John Doe     | (123) 456-7890 | 1234567890                      |

L'output della funzione `PurgeChar` ha rimosso con successo tutti i caratteri non numerici dalla stringa `PhoneNumber` e ha restituito solo i numeri.

**Vedere anche:**

p [KeepChar \(page 1783\)](#)

## Repeat

**Repeat()** crea una stringa costituita dalla stringa di input ripetuta il numero di volte stabilito dal secondo argomento.

**Sintassi:**

```
Repeat (text[, repeat_count])
```

**Tipo di dati restituiti:** stringa

Argomenti

| Argomento           | Descrizione                                                                                                               |
|---------------------|---------------------------------------------------------------------------------------------------------------------------|
| <b>text</b>         | La stringa originale.                                                                                                     |
| <b>repeat_count</b> | Definisce il numero di volte in cui i caratteri della stringa <b>text</b> devono essere ripetuti nella stringa di output. |

Esempio: espressione del grafico

| Esempio                         | Risultato        |
|---------------------------------|------------------|
| <code>Repeat( ' * ', 4 )</code> | Restituisce **** |

## Esempio: principi fondamentali della funzione Repeat

Espressione del grafico

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Un campo della tabella dati denominata `InputText`.

### Script di caricamento

```
Example:
Load * inline [
InputText
hello world!
hOw aRe you?
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `InputText`

Creare la seguente dimensione calcolata:

- `=Repeat(InputText,2)`, per calcolare il numero di volte in cui ripetere il valore in `InputText`. In questo esempio, ogni stringa sarà ripetuta due volte.

Tabella dei risultati

| <b>InputText</b> | <b>Repeat(InputText,2)</b> |
|------------------|----------------------------|
| hello world!     | hello world!hello world!   |
| hOw aRe you?     | hOw aRe you?hOw aRe you?   |

L'output della funzione `Repeat` include due istanze della stringa `InputText`, ad esempio, la riga 1 restituisce `hello world!hello world!`.

### Script di caricamento

Il codice seguente mostra come utilizzare la funzione in uno script di caricamento.

```
Example:
Load *,
repeat(String,2) as Repeat;
Load * inline [
String
hello world!
hOw aRe you? ];
```

Tabella dei risultati

| <b>Stringa</b> | <b>Ripeti</b>            |
|----------------|--------------------------|
| hello world!   | hello world!hello world! |
| hOw aRe you?   | hOw aRe you?hOw aRe you? |

### Esempio: scenario per l'applicazione della funzione Repeat

Espressione del grafico

#### Panoramica

In questo esempio si mostra come utilizzare i dati sulla soddisfazione dei clienti per convertire un punteggio di soddisfazione numerico in una rappresentazione visiva utilizzando le emoji 😊.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `Customer`
  - `SatisfactionScore`

#### Script di caricamento

Example:

```
Load * inline [  
Customer, SatisfactionScore  
John,4  
Mary,2  
Peter,5  
Susan,3  
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `Customer`
- `SatisfactionScore`

Creare la seguente dimensione calcolata:

- `=Repeat('😊',(SatisfactionScore))`, per calcolare e visualizzare il punteggio in modo visivo con le emoji.

Tabella dei risultati

| Cliente | SatisfactionScore | Repeat<br>( '😊',SatisfactionScore) |
|---------|-------------------|------------------------------------|
| John    | 4                 | 😊😊😊😊                               |
| Mary    | 2                 | 😊😊                                 |
| Peter   | 5                 | 😊😊😊😊😊                              |
| Susan   | 3                 | 😊😊😊                                |

L'output della funzione `repeat` mostra una rappresentazione visiva della soddisfazione del cliente, utilizzando le emoji di faccine sorridenti per rappresentare il punteggio numerico di soddisfazione.

### Esempio: scenario per l'applicazione della funzione `Repeat` avanzata

Espressione del grafico

#### Panoramica

Un set di dati con i conti dei clienti contiene i numeri di conto con lunghezze di caratteri variabili. In questo esempio si mostra come utilizzare la funzione `Repeat` per standardizzare la lunghezza dei numeri di conto.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Un campo della tabella dati denominata `AccountNo`.

#### Script di caricamento

```
Example:  
Load * inline [  
AccountNo  
123  
45125783  
41523  
489654  
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `AccountNo`

Creare la seguente dimensione calcolata:

- `=Repeat(0,10 - Len(AccountNo)) & AccountNo`, per aggiungere degli zeri al numero di conto per ottenere una lunghezza totale di 10 caratteri.

Tabella dei risultati

| AccountNo | Repeat(0, 10 - Len(AccountNo)) & AccountNo |
|-----------|--------------------------------------------|
| 123       | 0000000123                                 |
| 41523     | 0000041523                                 |
| 489654    | 0000489654                                 |
| 45125783  | 0045125783                                 |

L'output della funzione `repeat` mostra una lunghezza standardizzata di 10 caratteri per tutti i numeri di conto.

### Replace

**Replace()** restituisce una stringa dopo la sostituzione di tutte le occorrenze di una sottostringa fornita all'interno della stringa di input con un'altra sottostringa. La funzione non è ricorrente e viene applicata da sinistra verso destra.

Sintassi:

```
Replace(text, from_str, to_str)
```

Tipo di dati restituiti: stringa

Argomenti

| Argomento       | Descrizione                                                                                              |
|-----------------|----------------------------------------------------------------------------------------------------------|
| <b>text</b>     | La stringa originale.                                                                                    |
| <b>from_str</b> | Una stringa che può ricorrere una o più volte all'interno della stringa di input <b>text</b> .           |
| <b>to_str</b>   | La stringa che sostituirà tutte le occorrenze di <b>from_str</b> all'interno della stringa <b>text</b> . |

Esempio: espressione del grafico

| Esempio                                       | Risultato                        |
|-----------------------------------------------|----------------------------------|
| <code>Replace( 'abccde', 'cc', 'xyz' )</code> | Restituisce <code>abxyzde</code> |

### Esempio: principi fondamentali della funzione Replace

Espressione del grafico

#### Panoramica

Un set di dati con i nomi dei clienti contiene alcuni errori di ortografia. In questo esempio si mostra come utilizzare la funzione `Replace` per elaborare e sostituire qualsiasi occorrenza errata di "Jhon" con "John".

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `CustomerID`
  - `CustomerName`

### Script di caricamento

Example:

```
Load * inline [  
CustomerID, CustomerName  
1, Jhon Smith  
2, Jhon Doe  
3, John Williams  
4, Jhonathan Harris  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- CustomerID
- CustomerName

Creare la seguente dimensione calcolata:

- `=Replace(CustomerName, 'Jhon', 'John')`, per sostituire tutte le istanze di Jhon con John.

Tabella dei risultati

| CustomerID | NomeCliente         | Sostituisci<br>(NomeCliente,'Jhon','John') |
|------------|---------------------|--------------------------------------------|
| 1          | Jhon Smith          | John Smith                                 |
| 2          | Jhon Doe            | John Doe                                   |
| 3          | John Williams       | John Williams                              |
| 4          | Jhonathan<br>Harris | Johnathan Harris                           |

L'output della funzione `Repeat` trova e corregge tutte le istanze di Jhon con John.

### Esempio: scenario per l'applicazione della funzione Replace

Espressione del grafico

#### Panoramica

Un set di dati che elenca i prodotti utilizza un'etichettatura incoerente per i codici di categoria. Per esempio, i codici dei prodotti elettrici hanno più varianti, come `ELEC`, `ELC` e `e1c`. Per standardizzare i codici, la funzione `Replace` viene utilizzata per correggere tutti i codici di categoria incoerenti per i prodotti elettrici e di arredamento.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `ProductID`
  - `CategoryCode`

### Script di caricamento

Example:

```
Load * inline [  
ProductID, CategoryCode  
1, ELEC  
2, ELC  
3, FURN  
4, FRN  
5, ELEC  
6, ELC  
7, elc  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `ProductID`
- `CategoryCode`

Creare la seguente dimensione calcolata:

- `=Replace(Replace(Upper(CategoryCode), 'ELC', 'ELEC'),'FRN', 'FURN')`, per sostituire qualsiasi istanza di ELC con ELEC e qualsiasi istanza di FRN con FURN.

Tabella dei risultati

| ID prodotto | CodiceCategoria | Sostituisci(Sostituisci (Maiuscolo (CodiceCategoria), 'ELC', 'ELEC'),'FRN', 'FURN') |
|-------------|-----------------|-------------------------------------------------------------------------------------|
| 1           | ELEC            | ELEC                                                                                |
| 2           | ELC             | ELEC                                                                                |
| 3           | FURN            | FURN                                                                                |
| 4           | FRN             | Arredamento                                                                         |
| 5           | Elettronica     | Elettronica                                                                         |
| 6           | ELC             | ELEC                                                                                |
| 7           | elc             | ELEC                                                                                |

L'output della funzione `replace` ha sostituito tutte le istanze di `ELC` con `ELEC`, e di `FRN` con `FURN`. Notare che anche il codice `e1c` è stato sostituito. Sebbene la funzione `replace` distingua tra maiuscole e minuscole, la formula della dimensione utilizza la funzione `upper` per conformarne l'utilizzo in tutte le occorrenze in `categorycodes` prima di sostituire la stringa. La funzione rende la sostituzione delle stringhe in modo che non distinguano tra maiuscole e minuscole.

**Vedere anche:**

### ReplaceRegEx

**ReplaceRegEx()** restituisce una stringa dopo aver sostituito una o più corrispondenze tra una stringa di input e un modello di espressione regolare specificato. Il testo che sostituisce il testo corrispondente è specificato nell'argomento `to_str`. La funzione viene applicata da sinistra verso destra, ma se si specifica un valore **occurrence** negativo, le funzioni leggono i valori da destra a sinistra.

Questa funzione esegue operazioni regex che distinguono tra maiuscole e minuscole. In alternativa, è possibile utilizzare la variante **ReplaceRegExI()** per eseguire operazioni regex senza distinzione tra maiuscole e minuscole.

**Sintassi:**

```
ReplaceRegEx (text, regex, to_str [, occurrence])
```

**Tipo di dati restituiti:** stringa

#### Argomenti

| Argomento         | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>text</b>       | Il testo della stringa di input che si desidera sostituire, completamente o parzialmente, con il nuovo testo di <code>to_str</code> .                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>regex</b>      | L'espressione regolare che definisce quando sostituire il testo. Le corrispondenze tra questo argomento e l'argomento <code>text</code> vengono sostituite.                                                                                                                                                                                                                                                                                                                                                                            |
| <b>to_str</b>     | Il nuovo testo che si desidera utilizzare per sostituire il contenuto esistente di <code>text</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>occurrence</b> | <p>Il numero della corrispondenza (tra il testo di input e l'espressione regolare) da sostituire con il nuovo testo.</p> <p>Questo argomento è facoltativo. Se viene omissso, il valore predefinito è 0. Quando viene utilizzato il valore 0 o se l'argomento viene omissso, tutte le corrispondenze tra <code>text</code> e <code>regex</code> vengono sostituite con <code>to_str</code>.</p> <p>È possibile specificare un valore negativo per <b>occurrence</b> se si desidera cercare le corrispondenze da destra a sinistra.</p> |

## Esempi di funzioni

| Esempio                                              | Risultato                                                                                                                                                                                       |
|------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>ReplaceRegex('abc123', '[0-9]', 'x')</code>    | Restituisce <code>abcxxx</code> . In questo esempio, tutto il testo della stringa di input che corrisponde al modello regex viene sostituito con il nuovo testo.                                |
| <code>ReplaceRegex('abc123', '[0-9]', 'x', 1)</code> | Restituisce <code>abcx23</code> . Viene sostituita solo la prima occorrenza di una corrispondenza tra la stringa di input e il modello regex.                                                   |
| <code>ReplaceRegex('abc123', '[0-9]', 'x', 4)</code> | Restituisce <code>abc123</code> . Vi sono solo tre possibili corrispondenze tra la stringa di input e il modello regex, quindi non viene eseguita alcuna modifica.                              |
| <code>ReplaceRegex('ABC123', '[a-c]', '5')</code>    | Restituisce <code>ABC123</code> . Non viene eseguita alcuna modifica perché <b>ReplaceRegex()</b> fa distinzione tra maiuscole e minuscole e la variante che non le distingue non è utilizzata. |
| <code>ReplaceRegexI('ABC123', '[a-c]', '5')</code>   | Restituisce <code>555123</code> . Viene utilizzata la variante senza distinzione tra maiuscole e minuscole della funzione, <b>ReplaceRegexI()</b> .                                             |

## Casi di utilizzo

È possibile utilizzare **ReplaceRegex()** per modificare il testo per soddisfare gli standard di formattazione e di conformità, soprattutto quando esistono modelli complessi nei dati. Per esempio, è possibile riformattare le stringhe dei numeri di telefono in modo da includere solo i valori numerici.

Se necessario, è possibile anche utilizzare questa funzione per mascherare le informazioni sensibili, come le informazioni di identificazione personale (PII), in modo che non vengano mostrate agli utenti che analizzano l'app.

## Esempio 1 - script di caricamento per aggiornare gli indirizzi e-mail dei dipendenti

Script di caricamento e risultati

## Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- La tabella `Employees` contenente gli indirizzi e-mail dei dipendenti di un'organizzazione.
- Creazione di un nuovo campo, `Email_Updated`, per eseguire un'operazione di ricerca e sostituzione sui valori del campo `Email_Original`.

Dopo che l'azienda ha cambiato nome, tutti gli indirizzi e-mail sono stati aggiornati per riflettere il nuovo nome.

Requisiti:

- Converti tutti i riferimenti al dominio di test.com in newcompanyname.com.
- È necessario utilizzare le operazioni regex senza distinzione tra maiuscole e minuscole.

### Script di caricamento

Employees:

```
Load
ReplaceRegExI(Email_Original, '[a-zA-Z0-9.-]*\.?test\.com', '@newcompanyname.com') as Email_Updated,
* Inline [
ID, Email_Original
1, john_doe@test.com
2, jane_doe@test.com
3, Michael_Andrews@test.com
4, charlie_kent@test.com
5, frances_johnson@test.com
6, rebecca_smith@test.com
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- ID
- Email\_Original
- Email\_Updated

Tabella dei risultati

| ID | Email_Original                | Email_Updated                      |
|----|-------------------------------|------------------------------------|
| 1  | john_doe@test.com             | john_doe@newcompanyname.com        |
| 2  | jane_doe@test.com             | jane_doe@newcompanyname.com        |
| 3  | Michael_Andrews@test.com      | Michael_Andrews@newcompanyname.com |
| 4  | charlie_kent@support.test.com | charlie_kent@newcompanyname.com    |
| 5  | frances_johnson@test.com      | frances_johnson@newcompanyname.com |
| 6  | rebecca_smith@hr.test.com     | rebecca_smith@newcompanyname.com   |

## Esempio 2 - script di caricamento per formattare i codici ISBN

Script di caricamento e risultati

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Le variabili `ISBN_13` e `ISBN_10`, che memorizzano le espressioni regolari da utilizzare per i modelli di stringhe ISBN a 13 e 10 cifre, rispettivamente.
- Una tabella `Employees` contenente i valori numerici non elaborati dei codici ISBN dei libri. La tabella contiene sia ISBN a 13 cifre che a 10 cifre.
- Diverse operazioni per estrarre ogni parte dell'ISBN, per poi restituire una stringa che mostra l'intero codice nel formato standard previsto. È necessario utilizzare una combinazione di **ReplaceRegex()**, **MatchRegex()**, **ExtractRegexGroup()** e **Pick()**.

Requisiti:

- Bisogna formattare i numeri in un testo più leggibile, utilizzando la formattazione standardizzata:
  - ISBN ###-#-####-####-#
  - ISBN #-###-####-#

### Script di caricamento

```
Set ISBN_13 = '^([0-9]{3})([0-9]{1})([0-9]{4})([0-9]{4})([0-9]{1})$';
Set ISBN_10 = '^([0-9]{1})([0-9]{3})([0-9]{5})([0-9]{1})$';
```

Formatted\_Codes:

```
Load
Pick(MatchRegex(ISBN_Unformatted, '$(ISBN_13)', '$(ISBN_10)'), ReplaceRegex(ISBN_Unformatted,
'$(ISBN_13)', 'ISBN ' & ISBN_Component1 & '-' & ISBN_Component2 & '-' & ISBN_Component3 & '-'
& ISBN_Component4 & '-' & ISBN_Component5), ReplaceRegex(ISBN_Unformatted, '$(ISBN_10)', 'ISBN
' & ISBN_Component1 & '-' & ISBN_Component2 & '-' & ISBN_Component3 & '-' & ISBN_Component4))
as ISBN_Formatted;
```

```
Load *,
Pick(MatchRegex(ISBN_Unformatted, '$(ISBN_10)', '$(ISBN_13)'), ExtractRegexGroup(ISBN_
Unformatted, '$(ISBN_10)', 1, 1), ExtractRegexGroup(ISBN_Unformatted, '$(ISBN_13)', 1, 1)) as
ISBN_Component1,
Pick(MatchRegex(ISBN_Unformatted, '$(ISBN_10)', '$(ISBN_13)'), ExtractRegexGroup(ISBN_
Unformatted, '$(ISBN_10)', 2, 1), ExtractRegexGroup(ISBN_Unformatted, '$(ISBN_13)', 2, 1)) as
ISBN_Component2,
Pick(MatchRegex(ISBN_Unformatted, '$(ISBN_10)', '$(ISBN_13)'), ExtractRegexGroup(ISBN_
Unformatted, '$(ISBN_10)', 3, 1), ExtractRegexGroup(ISBN_Unformatted, '$(ISBN_13)', 3, 1)) as
ISBN_Component3,
Pick(MatchRegex(ISBN_Unformatted, '$(ISBN_10)', '$(ISBN_13)'), ExtractRegexGroup(ISBN_
Unformatted, '$(ISBN_10)', 4, 1), ExtractRegexGroup(ISBN_Unformatted, '$(ISBN_13)', 4, 1)) as
ISBN_Component4,
```

```
Pick(MatchRegex(ISBN_Unformatted, '$(ISBN_10)', '$(ISBN_13)'), ExtractRegexGroup(ISBN_
Unformatted, '$(ISBN_10)', 5, 1), ExtractRegexGroup(ISBN_Unformatted, '$(ISBN_13)', 5, 1)) as
ISBN_Component5
inline `
ISBN_Unformatted
1233123412340
0122012301234
0001012301232
0111234567
5552555555553
2224222222222
1901234568
3333333333333
5551515151513
2321232323231
24444444444
8880999900000
`;
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- ISBN\_Formatted

Tabella dei risultati

| ISBN_Formatted         |
|------------------------|
| ISBN 000-1-0123-0123-2 |
| ISBN 0-111-23456-7     |
| ISBN 1-901-23456-8     |
| ISBN 2-444-44444-4     |
| ISBN 012-2-0123-0123-4 |
| ISBN 123-3-1234-1234-0 |
| ISBN 222-4-2222-2222-2 |
| ISBN 232-1-2323-2323-1 |
| ISBN 333-3-3333-3333-3 |
| ISBN 555-1-5151-5151-3 |
| ISBN 555-2-5555-5555-3 |
| ISBN 888-0-9999-0000-0 |

### Esempio 3 - espressione del grafico per aggiornare gli indirizzi e-mail secondari

Espressione del grafico e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- La tabella `Employees` contenente gli indirizzi e-mail dei dipendenti di un'organizzazione. Questi sono memorizzati in una singola stringa per ogni dipendente, in cui sono memorizzati più indirizzi e-mail.

Requisiti:

- Gli indirizzi e-mail secondari devono essere aggiornati per utilizzare il dominio `newcompanyname.com`. Non bisogna aggiornare alcun indirizzo e-mail primario.
- È necessario utilizzare le operazioni regex senza distinzione tra maiuscole e minuscole.

#### Script di caricamento

`Employees:`

```
Load * Inline [
ID, Emails
1, Primary: john_doe@test.com Secondary: john_michael_doe@example.com
2, Primary: jane_doe@test.com Secondary: jane_tabitha_doe_doe@example.com
3, Primary: Michael_Andrews@test.com Secondary: Michael_John_Andrews@example.com
4, Primary: charlie_kent@test.com Secondary: charlie_peter_kent@example.com
5, Primary: frances_johnson@test.com Secondary: frances_johnson@example.com
6, Primary: rebecca_smith@test.com Secondary: rebecca_danielle_smith@example.com
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `ID`
- `Emails`

Aggiungere la seguente dimensione calcolata:

```
=ReplaceRegExI(Emails, '@[a-zA-Z0-9.-]*\.[a-zA-Z0-9]*\.com', '@newcompanyname.com', 2)
```

Tabella dei risultati

| ID | Emails                                                                           | =ReplaceRegExI(Emails,'@[a-zA-Z0-9.-]*\.?example\.com','@newcompanyname.com',2)      |
|----|----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| 1  | Primary: john_doe@test.com<br>Secondary: john_michael_doe@example.com            | Primary: john_doe@test.com Secondary: john_michael_doe@newcompanyname.com            |
| 2  | Primary: jane_doe@test.com<br>Secondary: jane_tabitha_doe_doe@example.com        | Primary: jane_doe@test.com Secondary: jane_tabitha_doe_doe@newcompanyname.com        |
| 3  | Primary: Michael_Andrews@test.com Secondary: Michael_John_Andrews@example.com    | Primary: Michael_Andrews@test.com Secondary: Michael_John_Andrews@newcompanyname.com |
| 4  | Primary: charlie_kent@test.com<br>Secondary: charlie_peter_kent@example.com      | Primary: charlie_kent@test.com Secondary: charlie_peter_kent@newcompanyname.com.com  |
| 5  | Primary: frances_johnson@test.com<br>Secondary: frances_johnson@example.com      | Primary: frances_johnson@test.com Secondary: frances_johnson@newcompanyname.com      |
| 6  | Primary: rebecca_smith@test.com<br>Secondary: rebecca_danielle_smith@example.com | Primary: rebecca_smith@test.com Secondary: rebecca_danielle_smith@newcompanyname.com |

Specificando il valore 2 per l'argomento **occurrence**, è possibile aggiornare solo la seconda istanza che corrisponde al modello regex.

## ReplaceRegExGroup

**ReplaceRegExGroup()** restituisce una stringa dopo aver sostituito una o più corrispondenze tra una stringa di input e il modello di espressione regolare composta specificato. Il testo che sostituisce il testo corrispondente è specificato nell'argomento **to\_str**. La funzione viene applicata da sinistra verso destra, ma se si specifica un valore **occurrence** negativo, le funzioni leggono i valori da destra a sinistra.

Questa funzione esegue operazioni regex che distinguono tra maiuscole e minuscole. In alternativa, è possibile utilizzare la variante **ReplaceRegExGroupI()** per eseguire operazioni regex senza distinzione tra maiuscole e minuscole.

### Sintassi:

```
ReplaceRegExGroup (text, regex, to_str, group [, occurrence])
```

Tipo di dati restituiti: stringa

#### Argomenti

| Argomento         | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>text</b>       | Il testo della stringa di input che si desidera sostituire, completamente o parzialmente, con il nuovo testo di <b>to_str</b> .                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>regex</b>      | L'espressione regolare che definisce quando sostituire il testo. Le corrispondenze tra questo argomento e l'argomento <b>text</b> vengono sostituite.                                                                                                                                                                                                                                                                                                                                                                |
| <b>to_str</b>     | Il nuovo testo che si desidera utilizzare per sostituire il contenuto esistente di <b>text</b> .                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>group</b>      | <p>Il numero del gruppo, nel caso di un'espressione regolare composta.</p> <p>Se l'espressione regolare contiene solo un gruppo, utilizzare invece la funzione <b>ReplaceRegEx()</b>. In alternativa, utilizzare <b>ReplaceRegExGroup()</b> con un valore <b>group</b> di 0.</p> <p>È possibile specificare un valore negativo per <b>group</b> per cercare i gruppi da destra a sinistra.</p>                                                                                                                       |
| <b>occurrence</b> | <p>Il numero della corrispondenza (tra il testo di input e l'espressione regolare) da sostituire con il nuovo testo.</p> <p>Questo argomento è facoltativo. Se viene omissso, il valore predefinito è 0. Quando viene utilizzato il valore 0 o se l'argomento viene omissso, tutte le corrispondenze tra <b>text</b> e <b>regex</b> vengono sostituite con <b>to_str</b>.</p> <p>È possibile specificare un valore negativo per <b>occurrence</b> se si desidera cercare le corrispondenze da destra a sinistra.</p> |

#### Esempi di funzioni

| Esempio                                                                      | Risultato                                                                                                                                                              |
|------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>ReplaceRegExGroup('abc123', '([a-z])([0-9]+)', 'x', 0)</code>          | Restituisce abx (sostituisce la corrispondenza completa).                                                                                                              |
| <code>ReplaceRegExGroup('abc123', '([a-z])([0-9]+)', 'x', 1)</code>          | Restituisce abx123 (sostituisce solo il primo gruppo nella corrispondenza).                                                                                            |
| <code>ReplaceRegExGroup('abc123', '([a-z])([0-9]+)', 'x', 2)</code>          | Restituisce abcx (sostituisce solo il secondo gruppo nella corrispondenza).                                                                                            |
| <code>ReplaceRegExGroup('abc123abc123', '([a-z])([0-9]+)', 'x', 1, 2)</code> | Restituisce abc123abx123. Vi sono due corrispondenze tra il primo gruppo del testo di input e il modello regex. Tuttavia, solo la seconda occorrenza viene sostituita. |

| Esempio                                                                     | Risultato                                                                                                                                                                                                                      |
|-----------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>ReplaceRegexGroup ('ABC123ABC123', '([a-z])([0-9]+)', 'x', 1)</code>  | Restituisce ABC123ABC123. In altre parole, non viene sostituito alcun testo. Ciò si deve al fatto che la funzione fa distinzione tra maiuscole e minuscole e la variante della funzione che non le distingue non è utilizzata. |
| <code>ReplaceRegexGroupI ('ABC123ABC123', '([a-z])([0-9]+)', 'x', 1)</code> | Restituisce ABx123ABx123. In questo esempio, la sostituzione del testo avviene perché viene utilizzata la variante senza distinzione di maiuscole e minuscole della funzione, <b>ReplaceRegexGroupI()</b> .                    |

## Casi di utilizzo

È possibile utilizzare **ReplaceRegexGroup()** per modificare il testo per soddisfare gli standard di formattazione e di conformità, soprattutto quando bisogna identificare modelli complessi nei dati. L'argomento **group** aggiuntivo per questa funzione permette di aggiornare solo alcune porzioni di un modello di testo più ampio. Ad esempio, è possibile aggiornare parti specifiche di un URL.

Se necessario, è possibile anche utilizzare questa funzione per mascherare le informazioni sensibili, come le informazioni di identificazione personale (PII), in modo che non vengano mostrate agli utenti che analizzano l'app.

## Esempio 1 - script di caricamento per aggiornare il dominio negli URL

Script di caricamento e risultati

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- La tabella **URLs** che inizialmente contiene un elenco di stringhe di input in cui bisogna identificare e aggiornare URL specifici. Bisogna aggiornare il dominio in questi URL.
- La creazione di un campo, **ResourceURL\_Updated**, che contiene un testo di input modificato con gli URL elaborati.
- Una variabile **URL\_RegEx** per memorizzare l'espressione regolare utilizzata per identificare gli URL da elaborare.

Requisiti:

- Ogni stringa di input contiene due URL, con l'URL della risorsa specificato nel primo URL. Non bisogna aggiornare nessun altro URL oltre al primo dell'elenco.
- L'URL da aggiornare deve essere nel formato `<https o https>://<dominio>.com/<percorso risorsa>`. Il percorso della risorsa è opzionale, ma non viene utilizzato in questo esempio.

### Script di caricamento

```
Set URL_RegEx = '(https?):\\\/((([a-zA-Z0-9]+)\\.([a-zA-Z0-9]*)\\.??([a-zA-Z0-9]*))\\\/?((([a-zA-Z0-9]*)*)((\\\/?([a-zA-Z0-9]*)*)*)\\\/{0,1}([\\.,;]+(=?\\s))*?'))';
```

URLs:

Load

```
ReplaceRegExGroup(URLsList, '$(URL_RegEx)', 'replacement-server', 2,1) as ResourceURL_Updated;
```

Load \* Inline `

URLsList

The resource is located at <https://testserver.com/files>. For more help, see:

<https://support.company.com>.

The resource is located at <https://testserver.com/files/worksheet>. For more help, see:

<https://support.company.com>.

The resource is located at <https://testserver.com/files/book>. For more help, see:

<https://support.company.com>.

The resource is located at <https://testserver.com/files/form>. For more help, see:

<https://support.company.com>.

The resource is located at <https://testserver.com/datamodel>. For more help, see:

<https://support.company.com>.

The resource is located at <https://testserver.com/resourcenew>. For more help, see:

<https://support.company.com>.

The resource is located at <https://testserver.com/page>. For more help, see:

<https://support.company.com>.

The resource is located at <https://testserver.com/page/overview>. For more help, see:

<https://support.company.com>.

` (delimiter is '\t');

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- ResourceURL\_Updated

Tabella dei risultati

| ResourceURL_Updated                                                                                                                                                                                                             |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| The resource is located at <a href="https://replacement-server/datamodel">https://replacement-server/datamodel</a> . For more help, see:<br><a href="https://support.company.com">https://support.company.com</a> .             |
| The resource is located at <a href="https://replacement-server/files">https://replacement-server/files</a> . For more help, see:<br><a href="https://support.company.com">https://support.company.com</a> .                     |
| The resource is located at <a href="https://replacement-server/files/book">https://replacement-server/files/book</a> . For more help, see:<br><a href="https://support.company.com">https://support.company.com</a> .           |
| The resource is located at <a href="https://replacement-server/files/form">https://replacement-server/files/form</a> . For more help, see:<br><a href="https://support.company.com">https://support.company.com</a> .           |
| The resource is located at <a href="https://replacement-server/files/worksheet">https://replacement-server/files/worksheet</a> . For more help, see:<br><a href="https://support.company.com">https://support.company.com</a> . |
| The resource is located at <a href="https://replacement-server/page">https://replacement-server/page</a> . For more help, see:<br><a href="https://support.company.com">https://support.company.com</a> .                       |

### ResourceURL\_Updated

The resource is located at <https://replacement-server/page/overview>. For more help, see: <https://support.company.com>.

The resource is located at <https://replacement-server/resourcenew>. For more help, see: <https://support.company.com>.

## Esempio 2 - espressione del grafico per aggiornare il protocollo negli URL

### Espressione del grafico e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- La tabella `URLSLIST` che contiene un elenco di stringhe di input in cui bisogna identificare e aggiornare URL specifici. Bisogna aggiornare il protocollo da `http` a `https` nei collegamenti al sito web di supporto.

Requisiti:

- Ogni stringa di input contiene due URL, con il collegamento del sito web di supporto che appare come secondo URL. Non bisogna aggiornare nessun altro URL oltre al secondo dell'elenco.
- L'URL da aggiornare deve essere nel formato `<https o https>://<dominio>.com/<percorso risorsa>`. Il percorso della risorsa è opzionale, ma non viene utilizzato in questo esempio.

#### Script di caricamento

URLS:

```
Load * Inline `
URLSLIST
The resource is located at https://testserver.com/files. For more help, see:
http://support.company.com.
The resource is located at https://testserver.com/files/worksheet. For more help, see:
http://support.company.com.
The resource is located at https://testserver.com/files/book. For more help, see:
http://support.company.com.
The resource is located at https://testserver.com/files/form. For more help, see:
http://support.company.com.
The resource is located at https://testserver.com/datamodel. For more help, see:
http://support.company.com.
The resource is located at https://testserver.com/resourcenew. For more help, see:
http://support.company.com.
The resource is located at https://testserver.com/page. For more help, see:
http://support.company.com.
The resource is located at https://testserver.com/page/overview. For more help, see:
http://support.company.com.
` (delimiter is '\t');
```

## Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- URLsList

Aggiungere la seguente dimensione calcolata:

```
=ReplaceRegExGroup(URLsList, '(https?):\\\/((([a-zA-Z0-9]+)\.([a-zA-Z0-9]*))\.\?([a-zA-Z0-9]*))\?\/((([a-zA-Z0-9]*)(\?\/([a-zA-Z0-9]*))*))\{0,1\}([\.,;]+(=?\s))*?', 'https', 1,2)
```

Tabella dei risultati

| URLsList                                                                                                                                                                                                           | =ReplaceRegExGroup(URLsList, '(https?):\\\/((([a-zA-Z0-9]+)\.([a-zA-Z0-9]*))\.\?([a-zA-Z0-9]*))\?\/((([a-zA-Z0-9]*)(\?\/([a-zA-Z0-9]*))*))\{0,1\}([\.,;]+(=?\s))*?', 'https', 1,2)                                   |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| The resource is located at <a href="https://testserver.com/files">https://testserver.com/files</a> . For more help, see: <a href="http://support.company.com">http://support.company.com</a> .                     | The resource is located at <a href="https://testserver.com/datamodel">https://testserver.com/datamodel</a> . For more help, see: <a href="https://support.company.com">https://support.company.com</a> .             |
| The resource is located at <a href="https://testserver.com/files/worksheet">https://testserver.com/files/worksheet</a> . For more help, see: <a href="http://support.company.com">http://support.company.com</a> . | The resource is located at <a href="https://testserver.com/files">https://testserver.com/files</a> . For more help, see: <a href="https://support.company.com">https://support.company.com</a> .                     |
| The resource is located at <a href="https://testserver.com/files/book">https://testserver.com/files/book</a> . For more help, see: <a href="http://support.company.com">http://support.company.com</a> .           | The resource is located at <a href="https://testserver.com/files/book">https://testserver.com/files/book</a> . For more help, see: <a href="https://support.company.com">https://support.company.com</a> .           |
| The resource is located at <a href="https://testserver.com/files/form">https://testserver.com/files/form</a> . For more help, see: <a href="http://support.company.com">http://support.company.com</a> .           | The resource is located at <a href="https://testserver.com/files/form">https://testserver.com/files/form</a> . For more help, see: <a href="https://support.company.com">https://support.company.com</a> .           |
| The resource is located at <a href="https://testserver.com/datamodel">https://testserver.com/datamodel</a> . For more help, see: <a href="http://support.company.com">http://support.company.com</a> .             | The resource is located at <a href="https://testserver.com/files/worksheet">https://testserver.com/files/worksheet</a> . For more help, see: <a href="https://support.company.com">https://support.company.com</a> . |
| The resource is located at <a href="https://testserver.com/resourcenew">https://testserver.com/resourcenew</a> . For more help, see: <a href="http://support.company.com">http://support.company.com</a> .         | The resource is located at <a href="https://testserver.com/page">https://testserver.com/page</a> . For more help, see: <a href="https://support.company.com">https://support.company.com</a> .                       |
| The resource is located at <a href="https://testserver.com/page">https://testserver.com/page</a> . For more help, see: <a href="http://support.company.com">http://support.company.com</a> .                       | The resource is located at <a href="https://testserver.com/page/overview">https://testserver.com/page/overview</a> . For more help, see: <a href="https://support.company.com">https://support.company.com</a> .     |

|                                                                                                                           |                                                                                                                                                                    |
|---------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| URLsList                                                                                                                  | =ReplaceRegExGroup(URLsList, '(https?):VW([a-zA-Z0-9]+)\.([a-zA-Z0-9]*).?([a-zA-Z0-9]*)V?([a-zA-Z0-9]*)((V?([a-zA-Z0-9]*)*)V{0,1}([\.,;]+(?=\s))*?', 'https', 1,2) |
| The resource is located at<br>https://testserver.com/page/overview.<br>For more help, see:<br>http://support.company.com. | The resource is located at<br>https://testserver.com/resourcenew. For more help, see:<br>https://support.company.com.                                              |

## Right

**Right()** restituisce una stringa costituita dagli ultimi caratteri (posizionati più a destra) della stringa di input, in cui il numero di caratteri viene stabilito dal secondo argomento.

**Sintassi:**

**Right**(text, count)

**Tipo di dati restituiti:** stringa

Argomenti

| Argomento | Descrizione                                                                                        |
|-----------|----------------------------------------------------------------------------------------------------|
| text      | La stringa originale.                                                                              |
| count     | Definisce il numero di caratteri da includere partendo dal lato destro della stringa <b>text</b> . |

Esempio: espressione del grafico

| Esempio              | Risultato       |
|----------------------|-----------------|
| Right( 'abcdef', 3 ) | Restituisce def |

## Esempio: principi fondamentali della funzione Right

Espressione del grafico

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `InputText`
  - `NoOfChars`

### Script di caricamento

Example:

```
Load * inline [
InputText, NoOfChars
'abcdef', 3
'2021-07-14', 5
'2021-07-14', 2
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- InputText
- NoOfChars

Creare la seguente dimensione calcolata:

- =Right(InputText,NoOfChars), per calcolare i caratteri in InputText che devono essere estratti più a destra.

Tabella dei risultati

| InputText  | NoOfChars | Right(InputText,NoOfChars) |
|------------|-----------|----------------------------|
| 2021-07-14 | 5         | 07-14                      |
| abcdef     | 3         | def                        |
| 2021-07-14 | 2         | 14                         |

La prima riga restituisce 07-14, in quanto questi sono i primi 5 caratteri a destra della stringa InputText.

La seconda riga restituisce def, in quanto questi sono i primi 3 caratteri a destra della stringa InputText.

La terza riga restituisce 14, in quanto questi sono i primi 2 caratteri a destra della stringa InputText.

### Script di caricamento

Il codice seguente mostra come utilizzare la funzione in uno script di caricamento.

Example:

```
Load
*,
right(Text,NoOfChars) as Right;
Load * inline [
Text, NoOfChars
'abcdef', 3
'2021-07-14', 5
'2021-07-14', 2
];
```

Tabella dei risultati

| Testo      | NoOfChars | Right |
|------------|-----------|-------|
| abcdef     | 3         | def   |
| 2021-07-14 | 5         | 7-14  |
| 2021-07-14 | 2         | 14    |

### Esempio: scenario per l'applicazione della funzione Right

Espressione del grafico

#### Panoramica

Nel seguente esempio si analizzano i codici degli ordini in un set di dati, che contengono il codice della regione e il numero dell'ordine, per creare un nuovo campo per l'analisi dei dati.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `OrderID`
  - `CustomerName`
  - `SalesAmount`

#### Script di caricamento

Example:

```
Load * inline [  
OrderID, CustomerName, SalesAmount  
US1000001, John Doe, 1500  
EU3000002, Jane Smith, 2300  
AS2000003, Mike Davis, 1800  
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `OrderID`
- `CustomerName`
- `SalesAmount`

Creare la seguente dimensione calcolata:

- `=Right(OrderID, 7)`, per dividere il numero dell'ordine.

Tabella dei risultati

| OrderID   | CustomerName | SalesAmount | Right<br>(OrderID, 7) |
|-----------|--------------|-------------|-----------------------|
| AS2000003 | Mike Davis   | 1800        | 2000003               |
| EU3000002 | Jane Smith   | 2300        | 3000002               |
| US1000001 | John Doe     | 1500        | 1000001               |

Confrontare l'output della funzione `Right` con l'input `orderID`. La dimensione calcolata deriva un valore numerico che ora può essere utilizzato per attività come l'ordinamento dell'output in ordine numerico anziché alfabetico.

## RTrim

**RTrim()** restituisce la stringa di input senza spazi finali.

**Sintassi:**

```
RTrim(text)
```

**Tipo di dati restituiti:** stringa

Argomenti

| Argomento   | Descrizione             |
|-------------|-------------------------|
| <b>text</b> | La stringa da valutare. |

Esempio: espressioni del grafico

| Esempio                      | Risultato       |
|------------------------------|-----------------|
| <code>RTrim( ' abc' )</code> | Restituisce abc |
| <code>RTrim( 'abc ' )</code> | Restituisce abc |

## Esempio: principi fondamentali della funzione RTrim

Espressione del grafico

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Un campo della tabella dati denominata `InputText`.

### Script di caricamento

```
Set verbatim=1;
Example:
Load * inline [
InputText
'   abc   '
' def '];
```



L'istruzione `set verbatim=1` è inclusa nell'esempio per assicurare che gli spazi non vengano automaticamente tagliati prima della dimostrazione della funzione `RTrim`. Vedere [Verbatim \(page 223\)](#) per ulteriori informazioni.

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- InputText

Creare la seguente dimensione calcolata:

- `=RTrim(InputText)` per rimuovere gli spazi finali da InputText.

Tabella dei risultati

| InputText   | RTrim(InputText) |
|-------------|------------------|
| '   abc   ' | '   abc'         |
| ' def '     | ' def'           |

L'output della funzione `RTrim` ha rimosso tutti gli spazi finali a destra del testo originale, ma ha mantenuto tutti gli spazi iniziali.

### Esempio: scenario per l'applicazione della funzione RTrim

Espressione del grafico

#### Panoramica

Un sistema di gestione delle relazioni con i clienti (CRM) contiene record con i dati immessi in modo incoerente che includono degli spazi finali extra. Ai fini della creazione di report, i dati devono essere puliti per rimuovere tali spazi e garantire un ordinamento e un raggruppamento corretto dei nomi dei clienti.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Un campo della tabella dati denominata `CustomerName`.

### Script di caricamento

```
Set verbatim=1;
Example:
Load * inline [
CustomerName
'John Doe '
'Jane Smith'
'Michael Johnson '
'Emily Davis'
];
```



L'istruzione "Set verbatim=1" è inclusa nell'esempio per assicurare che gli spazi non vengano automaticamente tagliati prima della dimostrazione della funzione trim. Vedere [Verbatim \(page 223\)](#) per ulteriori informazioni.

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- CustomerName

Creare la seguente dimensione calcolata:

- =RTrim(CustomerName) per rimuovere tutti gli spazi finali da CustomerName.

Tabella dei risultati

| CustomerName       | RTrim(CustomerName) |
|--------------------|---------------------|
| 'Emily Davis'      | 'Emily Davis'       |
| 'Jane Smith'       | 'Jane Smith'        |
| 'John Doe '        | 'John Doe'          |
| 'Michael Johnson ' | 'Michael Johnson'   |

L'output mostra che la funzione RTrim ha rimosso tutti gli spazi finali dai valori della stringa originale in CustomerName.

### Esempio: scenario per l'applicazione della funzione RTrim avanzata

Espressione del grafico

#### Panoramica

In questo esempio si mostra come rimuovere tutti gli spazi finali dalla stringa di testo originale. L'espressione del grafico include misure che utilizzano la funzione Len per contare i caratteri nella stringa prima e dopo aver utilizzato la funzione RTrim.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Un campo della tabella dati denominata `String`.

### Script di caricamento

```
Set verbatim=1;
Example:
Load * inline [
String
'   abc   '
' def '];
```



L'istruzione "Set verbatim=1" è inclusa nell'esempio per assicurare che gli spazi non vengano automaticamente tagliati prima della dimostrazione della funzione `RTrim`. Vedere [Verbatim \(page 223\)](#) per ulteriori informazioni.

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `String`

Creare la seguente dimensione calcolata:

- `=RTrim(String)`, per rimuovere tutti gli spazi finali extra.

Creare le seguenti misure:

- `=Len(String)` per calcolare la lunghezza della stringa originale.
- `=Len(RTrim(String))`, per calcolare la lunghezza della stringa dopo la rimozione degli spazi finali.

Tabella dei risultati

| Stringa | RTrim(String) | Len(String) | Len(RTrim(String)) |
|---------|---------------|-------------|--------------------|
| ' abc ' | ' abc'        | 10          | 6                  |
| ' def ' | ' def'        | 6           | 4                  |

Confrontando l'output della funzione `RTrim` con i valori delle stringhe originali nello script, si può notare come tutti gli spazi finali siano stati rimossi.

### Script di caricamento

Il codice seguente mostra come utilizzare la funzione in uno script di caricamento.

```
Set verbatim=1;
Example:
Load *, len(RtrimString) as RtrimStringLength;
Load *, rtrim(String) as RtrimString;
Load *, len(String) as StringLength;
```

```
Load * Inline [
String
'  abc  '
' def '];
```



L'istruzione "set verbatim=1" è inclusa nell'esempio per assicurare che gli spazi non vengano automaticamente tagliati prima della dimostrazione della funzione `rtrim`. Vedere [Verbatim \(page 223\)](#) per ulteriori informazioni.

Tabella dei risultati

| Stringa   | StringLength | RtrimString | RtrimStringLength |
|-----------|--------------|-------------|-------------------|
| '  abc  ' | 10           | '  abc'     | 6                 |
| ' def '   | 6            | ' def'      | 4                 |

#### Vedere anche:

► [LTrim \(page 1802\)](#)

## SubField

**SubField()** consente di estrarre i componenti della sottostringa da un campo della stringa padre, in cui i campi del record originali sono costituiti da due o più parti separate da un delimitatore.

La funzione **Subfield()**, ad esempio, consente di estrarre il nome di battesimo e il cognome da un elenco di record composto da nomi completi, le parti del componente di un nome del percorso o di estrarre dati da tabelle separate da virgole.

Se si utilizza la funzione **Subfield()** in un'istruzione **LOAD** escludendo il parametro `field_no` opzionale, per ogni sottostringa verrà generato un record completo. Se diversi campi vengono caricati utilizzando **Subfield()**, vengono creati i prodotti cartesiani di tutte le combinazioni.

#### Sintassi:

```
SubField(text, delimiter[, field_no ])
```

Tipo di dati restituiti: stringa

Argomenti

| Argomento        | Descrizione                                                                                                                                            |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>text</b>      | La stringa originale. Può trattarsi di un testo codificato in forma rigida, una variabile, un'espansione del segno del dollaro o un'altra espressione. |
| <b>delimiter</b> | Un carattere all'interno del <b>text</b> di input che divide la stringa in parti del componente.                                                       |

| Argomento       | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>field_no</b> | <p>Il terzo argomento opzionale è un numero intero che specifica la sottostringa della stringa principale <b>text</b> che verrà restituita. Utilizzare il valore 1 per restituire la prima sottostringa, 2 per restituire la seconda sottostringa e così via.</p> <ul style="list-style-type: none"> <li>Se <b>field_no</b> è un valore positivo, le sottostringhe vengono estratte da sinistra a destra.</li> <li>Se <b>field_no</b> è un valore negativo, le sottostringhe vengono estratte da destra a sinistra.</li> </ul> |



*È possibile utilizzare `SubField()` invece di utilizzare combinazioni di funzioni complesse, come ad esempio `Len()`, `Right()`, `Left()`, `Mid()` e altre funzioni delle stringhe.*

#### Esempio: espressioni del grafico

| Esempio                                                                      | Risultato                     |
|------------------------------------------------------------------------------|-------------------------------|
| <code>SubField( 'abc;cde;efg', ';' ,2 )</code>                               | Restituisce cde               |
| <code>SubField( Null, ';' ,1 )</code>                                        | Restituisce una stringa vuota |
| <code>SubField( ';', ';' ,1 )</code>                                         | Restituisce una stringa vuota |
| <code>SubField( '\Users\ext_jrb\Documents\Qlik\Sense\Apps;', '\',-3 )</code> | Restituisce Qlik              |

## Esempio: principi fondamentali della funzione SubField

### Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Un campo della tabella dati denominata `Name`.

#### Script di caricamento

```
Example:
Load * inline [
Name
Dave Owen
Joe Tem
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- Name

Creare le seguenti dimensioni calcolate:

- `=SubField(Name, ' ', 1)`, per estrarre la prima sottostringa che appare prima del delimitatore spaziale ' '.
- `=SubField(Name, ' ', -1)`, per estrarre la prima sottostringa che appare prima del delimitatore spaziale ' ', iniziando dalla destra della stringa.

Results table

| Name      | SubField(Name, ' ', 1) | SubField(Name, ' ', -1) |
|-----------|------------------------|-------------------------|
| Dave Owen | Dave                   | Owen                    |
| Joe Tem   | Joe                    | Tem                     |

La funzione `SubField()` estrae la prima sottostringa `Name` impostando l'argomento `field_no` su 1. Dato che il valore di `field_no` è positivo, per l'estrazione della sottostringa viene seguito l'ordine da sinistra a destra. Una seconda chiamata di funzione estrae la seconda sottostringa impostando l'argomento `field_no` su -1, in modo da estrarre la sottostringa seguendo un ordine da destra a sinistra.

### Script di caricamento

### Script di caricamento

Aprire Editor caricamento dati, quindi aggiungere il seguente script di caricamento in una nuova sezione.

```
FullName:
LOAD * inline [
Name
'Dave Owen'
'Joe Tem'
];

SepNames:
Load Name,
SubField(Name, ' ', 1) as FirstName,
SubField(Name, ' ', -1) as Surname
Resident FullName;
Drop Table FullName;
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- Name
- FirstName
- SurName

| Results table |           |         |
|---------------|-----------|---------|
| Name          | FirstName | SurName |
| Dave Owen     | Dave      | Owen    |
| Joe Tem       | Joe       | Tem     |

### Spiegazione

La funzione `subField` estrae la prima sottostringa, `name`, impostando l'argomento `field_no` su 1. Dato che il valore di `field_no` è positivo, per l'estrazione della sottostringa viene seguito l'ordine da sinistra a destra. Una seconda chiamata di funzione estrae la seconda sottostringa impostando l'argomento `field_no` su -1, in modo da estrarre la sottostringa seguendo un ordine da destra a sinistra.

### Esempio: scenario per l'applicazione della funzione SubField

Espressione del grafico

#### Panoramica

Un set di dati di prodotti contiene una tabella con i prodotti. Ogni prodotto nella tabella ha un campo con il tag che identifica l'area del prodotto. I prodotti possono avere più valori per il campo del tag. Ad esempio, il widget A ha i seguenti tag: Elettronica, Gadget, Casa. I valori dei tag sono separati da una barra verticale (|) come delimitatore. In questo esempio si mostra come estrarre i valori di tag specifici utilizzando la funzione `subField`.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `ProductID`
  - `ProductName`
  - `Tags`
  - `Sales`

#### Script di caricamento

```
Example:
Load *
INLINE [
ProductID, ProductName, Tags, Sales
1, widget A, Electronics|Gadgets|Home, 150
2, widget B, Electronics|Accessories, 200
3, widget C, Furniture|Home, 100
4, widget D, Gadgets|Accessories, 250
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- ProductID
- ProductName
- Tags

Creare le seguenti dimensioni calcolate:

- =SubField(Tags, '|', 1), per estrarre il primo elemento nel campo Tags.
- =SubField(Tags, '|', 2), per estrarre il secondo elemento nel campo Tags.
- =SubField(Tags, '|', 3), per estrarre il terzo elemento nel campo Tags.

Results table

| ProductID | ProductName | Tags                     | SubField<br>(Tags, ' ', 1) | SubField<br>(Tags, ' ', 2) | SubField<br>(Tags, ' ', 3) |
|-----------|-------------|--------------------------|----------------------------|----------------------------|----------------------------|
| 1         | Widget A    | Electronics Gadgets Home | Electronics                | Gadgets                    | Home                       |
| 2         | Widget B    | Electronics Accessories  | Electronics                | Accessories                | -                          |
| 3         | Widget C    | Furniture Home           | Furniture                  | Home                       | -                          |
| 4         | Widget D    | Gadgets Accessories      | Gadgets                    | Accessories                | -                          |

L'output delle funzioni di SubField ha restituito con successo i tag dalle posizioni rilevanti nella stringa originale di Tags.

### Script di caricamento

#### Script di caricamento

Aprire Editor caricamento dati, quindi aggiungere il seguente script di caricamento in una nuova sezione.

```
LOAD DISTINCT
Instrument,
SubField(Player,',' ) as Player,
SubField(Project,',' ) as Project;
```

```
Load * inline [
Instrument|Player|Project
Guitar|Neil,Mike|Music,Video
Guitar|Neil|Music,OST
Synth|Neil,Jen|Music,Video,OST
Synth|Jo|Music
Guitar|Neil,Mike|Music,OST
] (delimiter is '|');
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- Instrument
- Player
- Project

| Results table |        |         |
|---------------|--------|---------|
| Instrument    | Player | Project |
| Guitar        | Mike   | Music   |
| Guitar        | Mike   | Video   |
| Guitar        | Mike   | OST     |
| Guitar        | Neil   | Music   |
| Guitar        | Neil   | Video   |
| Guitar        | Neil   | OST     |
| Synth         | Jen    | Music   |
| Synth         | Jen    | Video   |
| Synth         | Jen    | OST     |
| Synth         | Jo     | Music   |
| Synth         | Neil   | Music   |
| Synth         | Neil   | Video   |
| Synth         | Neil   | OST     |

### Spiegazione

Questo esempio mostra come l'utilizzo di più istanze della funzione `subfield()`, ciascuna senza il parametro `field_no`, crea prodotti cartesiani di tutte le combinazioni dalla stessa istruzione `LOAD`. L'opzione `DISTINCT` viene utilizzata per evitare la creazione di record duplicati.

## SubFieldRegEx

**SubFieldRegEx()** estrae il testo da un'espressione di stringa di input utilizzando il modello di espressione regolare specificato come delimitatore. La funzione restituisce un valore null se non vengono trovate corrispondenze.

Questa funzione esegue operazioni regex che distinguono tra maiuscole e minuscole. In alternativa, è possibile utilizzare la variante **SubFieldRegExI()** per eseguire operazioni regex senza distinzione tra maiuscole e minuscole.

### Sintassi:

```
SubFieldRegEx (text, regex_delimiter [, field_no])
```

Tipo di dati restituiti: stringa

### Argomenti

| Argomento              | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>text</b>            | Il testo della stringa di input in cui si desidera cercare il modello di un'espressione regolare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>regex_delimiter</b> | Espressione stringa contenente l'espressione regolare da utilizzare come delimitatore. È possibile specificare più di un delimitatore nello stesso modello di espressione regolare, separandoli con un carattere   (barra verticale).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>field_no</b>        | <p>Valore intero che indica il numero della suddivisione che si desidera estrarre. Per esempio, specificare il valore 4 per estrarre la quarta suddivisione. Questo argomento è facoltativo.</p> <p>Le seguenti indicazioni si applicano indipendentemente dal fatto che la funzione sia utilizzata in uno script di caricamento o in un'espressione del grafico:</p> <ul style="list-style-type: none"><li>• Se <b>field_no</b> è un valore positivo, la funzione restituirà un valore, identificando le corrispondenze da sinistra a destra.</li><li>• Se <b>field_no</b> è un valore negativo, la funzione restituirà un valore, identificando le corrispondenze da destra a sinistra.</li></ul> <p>Quando si utilizza la funzione in uno script di caricamento:</p> <ul style="list-style-type: none"><li>• Se si utilizza la funzione <b>SubFieldRegEx()</b> in un'istruzione <b>LOAD</b> e il valore <b>field_no</b> viene omissso, la funzione genera tanti record quante sono le corrispondenze.</li><li>• Se diversi campi vengono caricati utilizzando <b>SubFieldRegEx()</b> e nessuno di essi specifica un argomento <b>field_no</b>, vengono creati i prodotti cartesiani per tutte le combinazioni.</li></ul> <p>Quando utilizzare la funzione in un'espressione del grafico:</p> <ul style="list-style-type: none"><li>• Se <b>field_no</b> viene omissso, il valore predefinito è 1.</li></ul> |

### Esempi di funzioni

| Esempio                                             | Risultato                                                                                                             |
|-----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|
| <code>SubFieldRegEx('a,b c;1 2,3','  ,;', 1)</code> | Restituisce a. In questo esempio, vengono specificati tre caratteri delimitatori (spazio, virgola e punto e virgola). |

| Esempio                                 | Risultato                                                                                          |
|-----------------------------------------|----------------------------------------------------------------------------------------------------|
| SubFieldRegEx('a;b;c;1;2;3',';', 2)     | Restituisce b. In questo caso, viene specificato solo un carattere delimitatore (punto e virgola). |
| SubFieldRegEx('a,b c;1 2,3','  , ;', 4) | Restituisce 1.                                                                                     |

## Casi di utilizzo

È possibile utilizzare **SubFieldRegEx()** per estrarre più valori singoli da un elenco delimitato. Per esempio, se si dispone di un elenco con i valori separati da virgole all'interno di un blocco più grande di linguaggio naturale, è possibile utilizzare questa funzione per isolare ogni singolo valore dell'elenco in un nuovo record del modello dati.

## Esempio 1 - script di caricamento per separare i tag in base alla transazione

Script di caricamento e risultati

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- La tabella `Transaction_By_Tag`, che inizialmente contiene i dati della transazione. Questi dati vengono elaborati per estrarre i valori dei singoli tag in nuovi record.
- La creazione di un campo, `TransactionTag`, per memorizzare i valori dei tag estratti.

Nel campo contenente i tag della transazione, più tag sono contenuti in stringhe di testo, separate da delimitatori. Il delimitatore può essere uno dei seguenti caratteri: `,` `|` `;`;

### Script di caricamento

`Transaction_By_Tag:`

```
Load SubFieldRegEx(Tags, ',|;|\|') as TransactionTag,
* Inline [
ID      Product Tags      Amount
1      Product A      Special order,Eligible for return,warranty included      100.93
2      Product B      Ineligible for return|No warranty available      51.11
3      Product C      No warranty available      12.83
4      Product B      Special order;Liquidation sale;warranty included      209.48
] (delimiter is '\t');

drop fields Tags;
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- ID
- Product
- TransactionTag

Tabella dei risultati

| ID | Product   | TransactionTag        |
|----|-----------|-----------------------|
| 1  | Product A | Eligible for return   |
| 1  | Product A | Warranty included     |
| 1  | Product A | Special order         |
| 2  | Product B | No warranty available |
| 2  | Product B | Ineligible for return |
| 3  | Product C | No warranty available |
| 4  | Product B | Liquidation sale      |
| 4  | Product B | Special order         |
| 4  | Product B | Warranty included     |

Questo esempio dimostra che, se utilizzata in uno script di caricamento, la funzione **SubFieldRegEx()** restituisce una riga separata per ogni corrispondenza trovata nella stringa di input.

### Esempio 2 - espressioni dei grafici per separare i tag in base alla transazione

Script di caricamento e risultati

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- La tabella `Transaction_By_Tag`, che contiene i dati della transazione.
- Un campo, `Tags`, che contiene stringhe di testo per memorizzare tutti i tag associati a una transazione.

Requisiti:

- Nel campo contenente i tag della transazione, più tag sono contenuti in stringhe di testo, separate da delimitatori. Il delimitatore può essere uno dei seguenti caratteri: `,` `|` `;`
- Bisogna utilizzare le espressioni dei grafici per separare ogni tag nel proprio campo, con un massimo di tre tag da aggiungere a una transazione.

#### Script di caricamento

`Transaction_By_Tag:`

```
Load * Inline [
ID      Product Tags      Amount
1      Product A      Special order, Eligible for return, Warranty included      100.93
2      Product B      Ineligible for return | No warranty available      51.11
3      Product C      No warranty available      12.83
4      Product B      Special order;Liquidation sale;Warranty included      209.48
] (delimiter is '\t');
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- ID
- Tags

Aggiungere le seguenti dimensioni calcolate:

- =SubFieldRegEx(Tags, ',|;|\|', 1)
- =SubFieldRegEx(Tags, ',|;|\|', 2)
- =SubFieldRegEx(Tags, ',|;|\|', 3)

Tabella dei risultati

| ID | Tags                                                  | =SubFieldRegEx(Tags, ', ; \ ', 1) | =SubFieldRegEx(Tags, ', ; \ ', 2) | =SubFieldRegEx(Tags, ', ; \ ', 3) |
|----|-------------------------------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|
| 1  | Special order, Eligible for return, Warranty included | Special order                     | Eligible for return               | Warranty included                 |
| 2  | Ineligible for return   No warranty available         | Ineligible for return             | No warranty available             | -                                 |
| 3  | No warranty available                                 | No warranty available             | -                                 | -                                 |
| 4  | Special order;Liquidation sale;Warranty included      | Special order                     | Liquidation sales                 | Warranty included                 |

## SubStringCount

**SubStringCount()** restituisce il numero di occorrenze della sottostringa specificata nel testo della stringa di input. In mancanza di corrispondenze, viene restituito 0.

**Sintassi:**

```
SubStringCount(text, sub_string)
```

**Tipo di dati restituiti:** numero intero

Argomenti

| Argomento         | Descrizione                                                                                    |
|-------------------|------------------------------------------------------------------------------------------------|
| <b>text</b>       | La stringa originale.                                                                          |
| <b>sub_string</b> | Una stringa che può ricorrere una o più volte all'interno della stringa di input <b>text</b> . |

Esempio: espressioni del grafico

| Esempio                                             | Risultato     |
|-----------------------------------------------------|---------------|
| <code>SubStringCount( 'abcdefgcdxyz', 'cd' )</code> | Restituisce 2 |
| <code>SubStringCount( 'abcdefgcdxyz', 'dc' )</code> | Restituisce 0 |

### Esempio: principi fondamentali della funzione SubStringCount

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Un campo della tabella dati denominata `FruitSentence`.

#### Script di caricamento

Example:

```
Load * inline [  
FruitSentence  
"I love eating an apple every day.Apple juice is great too!"  
"Apple pie is my favorite dessert. I also enjoy apple jam."  
"There is an apple and a banana on the table."  
"I don't like apples as much as oranges."  
"An apple a day keeps the doctor away, but two apples are even better!"  
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `FruitSentence`

Creare la seguente dimensione calcolata:

- `=SubStringCount(FruitSentence, 'apple')`, per calcolare quante volte la parola `apple` è presente in `FruitSentence`.

Results table

| <code>FruitSentence</code>                                            | <code>SubStringCount(FruitSentence, 'apple')</code> |
|-----------------------------------------------------------------------|-----------------------------------------------------|
| An apple a day keeps the doctor away, but two apples are even better! | 2                                                   |

| FruitSentence                                               | SubStringCount(FruitSentence, 'apple') |
|-------------------------------------------------------------|----------------------------------------|
| Apple pie is my favorite dessert. I also enjoy apple jam.   | 1                                      |
| I don't like apples as much as oranges.                     | 1                                      |
| I love eating an apple every day. Apple juice is great too! | 1                                      |
| There is an apple and a banana on the table.                | 1                                      |

La tabella è ordinata in ordine crescente in base alla dimensione `FruitSentence`. La funzione `substringcount` restituisce il numero di volte in cui la parola `apple` ricorre in ogni frase. Ad esempio, la prima frase restituisce 2, perché la parola ricorre due volte nella frase. Nella seconda e quarta frase, la funzione ignora la parola `Apple` perché l'utilizzo delle maiuscole e minuscole non corrisponde al termine di ricerca `apple`.



*Per effettuare una ricerca senza distinzione tra maiuscole e minuscole, è possibile convertire prima la stringa di ricerca in caratteri maiuscoli o minuscoli, e quindi utilizzare la funzione `substringcount` sul risultato. Per esempio, per cercare tutte le occorrenze della parola `apple`, indipendentemente dall'utilizzo delle maiuscole e minuscole, utilizzare la seguente espressione di misura: `=SubStringCount(Upper(FruitSentence), 'APPLE')`. L'espressione utilizza la funzione `substringcount` con la funzione `upper` per convertire l'intera stringa in caratteri maiuscoli, quindi cerca la parola `APPLE`.*

### Script di caricamento

Il codice seguente mostra come utilizzare la funzione in uno script di caricamento. In questo esempio è incluso un carico precedente che utilizza la funzione `substringcount` con la funzione `upper` per convertire l'intera stringa in caratteri in maiuscolo, e poi cerca la parola `APPLE`.

Example:

```
Load *,
SubStringCount(Upper(FruitSentence), 'APPLE') as SubStringCount_APPLE;
Load * inline [
FruitSentence
"I love eating an apple every day. Apple juice is great too!"
"Apple pie is my favorite dessert. I also enjoy apple jam."
"There is an apple and a banana on the table."
"I don't like apples as much as oranges."
"An apple a day keeps the doctor away, but two apples are even better!"
];
```

Tabella dei risultati

| FruitSentence                                               | SubStringCount_APPLE |
|-------------------------------------------------------------|----------------------|
| I love eating an apple every day. Apple juice is great too! | 2                    |

| FruitSentence                                                         | SubStringCount_APPLE |
|-----------------------------------------------------------------------|----------------------|
| Apple pie is my favorite dessert. I also enjoy apple jam.             | 2                    |
| There is an apple and a banana on the table.                          | 1                    |
| I don't like apples as much as oranges.                               | 1                    |
| An apple a day keeps the doctor away, but two apples are even better! | 2                    |

## Esempio: scenario per l'applicazione della funzione SubStringCount

Espressione del grafico

### Panoramica

Un set di dati con i prodotti a base di frutta contiene un campo con le descrizioni dei prodotti. Nel seguente esempio si calcola il numero di volte in cui i seguenti nomi di frutta ricorrono in ogni descrizione: mela, banana o arancia.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Un campo della tabella dati denominata `ProductDescription`.

### Script di caricamento

Example:

```
Load * inline [
ProductDescription
"Fresh apple and banana smoothie."
"Organic apple, banana, and orange juice. Great for breakfast."
"A refreshing orange soda made with real orange juice."
"Banana chips with apple and cinnamon flavor."
"Delicious apple pie with a hint of cinnamon and vanilla."
"Tropical banana and orange mix for smoothies."
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- `ProductDescription`

Creare le seguenti dimensioni calcolate:

- `=SubStringCount(ProductDescription, 'apple')`, per calcolare il numero di occorrenze della parola `apple` nella stringa `ProductDescription`.
- `=SubStringCount(ProductDescription, 'banana')`, per calcolare il numero di occorrenze della parola `banana` nella stringa `ProductDescription`.
- `=SubStringCount(ProductDescription, 'orange')`, per calcolare il numero di occorrenze della parola `orange` nella stringa `ProductDescription`.

Results table

| ProductDescription                                                  | SubStringCount<br>(ProductDescription,<br>'apple') | SubStringCount<br>(ProductDescription,<br>'banana') | SubStringCount<br>(ProductDescription,<br>'orange') |
|---------------------------------------------------------------------|----------------------------------------------------|-----------------------------------------------------|-----------------------------------------------------|
| A refreshing orange<br>soda made with real<br>orange juice.         | 0                                                  | 0                                                   | 2                                                   |
| Banana chips with<br>apple and cinnamon<br>flavor.                  | 1                                                  | 0                                                   | 0                                                   |
| Delicious apple pie with<br>a hint of cinnamon and<br>vanilla.      | 1                                                  | 0                                                   | 0                                                   |
| Fresh apple and banana<br>smoothie.                                 | 1                                                  | 1                                                   | 0                                                   |
| Organic apple, banana,<br>and orange juice. Great<br>for breakfast. | 1                                                  | 1                                                   | 1                                                   |
| Tropical banana and<br>orange mix for<br>smoothies.                 | 0                                                  | 1                                                   | 1                                                   |

L'output della funzione `substringcount` restituisce con successo il numero di istanze in cui ogni sottostringa (`apple`, `banana`, `orange`) è presente nella descrizione del prodotto. Per esempio, nella prima descrizione, la funzione restituisce 2 perché ci sono due istanze della parola `orange`. L'argomento `sub_string` distingue tra le maiuscole e minuscole quindi, nella seconda descrizione, la funzione non conteggia la parola `banana` perché non corrisponde alla sottostringa `banana` nell'espressione di misura.



*Per effettuare una ricerca senza distinzione tra maiuscole e minuscole, è possibile convertire prima la stringa di ricerca in caratteri maiuscoli o minuscoli, e quindi utilizzare la funzione `substringCount` sul risultato. Per esempio, per cercare tutte le occorrenze della parola `apple`, indipendentemente dall'utilizzo delle maiuscole e minuscole, utilizzare la seguente espressione di misura: `=substringCount(Upper(ProductDescription), 'APPLE')`. L'espressione utilizza la funzione `substringCount` con la funzione `upper` per convertire l'intera stringa in caratteri maiuscoli, quindi cerca la parola `APPLE`.*

### Esempio: scenario per l'applicazione della funzione `SubStringCount` scenario con analisi dei dati annidati

Espressione del grafico

#### Panoramica

Un set di dati con i nomi di prodotti IT contiene un elenco di numeri ID e categorie di modifica. Nella tabella dati, il campo `Categories` viene derivato da una gerarchia multilivello che elenca la categoria principale seguita da tutte le sottocategorie, ad esempio `Software > Programma > Patch`. Questo scenario mostra come scegliere l'ultimo valore, o valore leaf, da questa gerarchia irregolare di categorie, utilizzando una combinazione delle funzioni `substringCount` e `subField`.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `changeID` contiene l'ID della modifica originale da elaborare.
  - `Categories` contiene un elenco di categorie di modifica. Le categorie sono separate da punti e virgola e sono elencate in ordine gerarchico da sinistra a destra, partendo dalla categoria principale superiore fino alla sottocategoria più dettagliata.

#### Script di caricamento

```
Example:
Load * inline [
ChangeID, Categories
CHG00100, Software;Program;Patch
CHG00101, Hardware;Server
CHG00102, Hardware;Network;Router
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `ChangeID`
- `Categories`

Creare le seguenti dimensioni calcolate:

- `=SubStringCount(Categories, ';')+1`, per calcolare il numero di livelli di categoria nella gerarchia delle modifiche.
- `=SubField(Categories, ';', SubStringCount(Categories, ';')+1)`, per estrarre la categoria più a destra nella gerarchia delle modifiche.

Results table

| ChangeID | Categories              | SubStringCount<br>(Categories, ';')+1 | SubField(Categories, ';',<br>SubStringCount<br>(Categories, ';')+1) |
|----------|-------------------------|---------------------------------------|---------------------------------------------------------------------|
| CHG00100 | Software;Program;Patch  | 3                                     | Patch                                                               |
| CHG00101 | Hardware;Server         | 2                                     | Server                                                              |
| CHG00102 | Hardware;Network;Router | 3                                     | Router                                                              |

In questo caso, la funzione `subStringCount` è annidata all'interno della funzione `subField`. La prima dimensione calcolata, `subStringCount()`, calcola il numero di valori annidati nel campo `Categories`. Questo viene determinato contando il numero di punti e virgole `' ; '` e aggiungendo 1 al risultato. L'output viene quindi inserito come terzo parametro della funzione `subField` per estrarre la categoria più a destra annidata nel campo `Categories`.

Per esempio, rivedere i risultati della voce `changeIDCHG00100`. Nella prima dimensione calcolata, `subStringCount` è 2. L'espressione aggiunge quindi 1 a questo risultato per determinare che il campo `Categories` ha tre valori di categoria annidati, `Software`, `Program` e `Patch`. La funzione `subField`, nella seconda dimensione calcolata, utilizza poi questo risultato per estrarre la terza categoria, `Patch`.

## TextBetween

**TextBetween()** restituisce il testo nella stringa di input che si trova tra i caratteri specificati come delimitatori.

**Sintassi:**

```
TextBetween (text, delimiter1, delimiter2[, n])
```

**Tipo di dati restituiti:** stringa

Argomenti

| Argomento         | Descrizione                                                                               |
|-------------------|-------------------------------------------------------------------------------------------|
| <b>text</b>       | La stringa originale.                                                                     |
| <b>delimiter1</b> | Specifica il primo carattere di delimitazione (o stringa) da ricercare in <b>text</b> .   |
| <b>delimiter2</b> | Specifica il secondo carattere di delimitazione (o stringa) da ricercare in <b>text</b> . |

| Argomento | Descrizione                                                                                                                                                                                                                       |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>n</b>  | Definisce l'occorrenza della coppia di delimitazione in cui eseguire la ricerca. Ad esempio, un valore di 2 restituisce i caratteri tra la seconda occorrenza di <b>delimiter1</b> e la seconda occorrenza di <b>delimiter2</b> . |

Esempio: espressioni del grafico

| Esempio                                                                | Risultato                                                                                                               |
|------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| <code>TextBetween( '&lt;abc&gt;', '&lt;', '&gt;' )</code>              | Restituisce abc                                                                                                         |
| <code>TextBetween( '&lt;abc&gt;&lt;de&gt;', '&lt;', '&gt;', 2 )</code> | Restituisce de                                                                                                          |
| <code>TextBetween( 'abc', '&lt;', '&gt;' )</code>                      | Restituisce NULL<br>Se uno qualsiasi dei delimitatori non viene trovato nella stringa, viene restituito il valore NULL. |
| <code>TextBetween( '&lt;a&lt;b', '&lt;', '&gt;' )</code>               | Restituisce NULL<br>Se uno qualsiasi dei delimitatori non viene trovato nella stringa, viene restituito NULL.           |
| <code>TextBetween( '&lt;&gt;', '&lt;', '&gt;' )</code>                 | Restituisce una stringa con lunghezza zero.                                                                             |
| <code>TextBetween( '&lt;abc&gt;', '&lt;', '&gt;', 2 )</code>           | Restituisce NULL, in quanto n è superiore al numero di occorrenze dei delimitatori.                                     |

## Esempio: principi fondamentali della funzione TextBetween

Espressione del grafico

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Un campo della tabella dati denominata `InputText`.

### Script di caricamento

```
Example:
Load * inline [
InputText,
<abc><de>
<def><ghi><jk>
];
```

## Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- InputText

Creare le seguenti dimensioni calcolate:

- =TextBetween(InputText, '<', '>'), per calcolare la prima istanza dei caratteri che appaiono tra i delimitatori '<' e '>' nel campo InputText.
- =TextBetween(InputText, '<', '>', 2), per calcolare la seconda istanza dei caratteri che appaiono tra i delimitatori '<' e '>' nel campo InputText.

Results table

| InputText       | TextBetween(InputText,'<','>') | TextBetween(InputText,'<','>',2) |
|-----------------|--------------------------------|----------------------------------|
| <abc><de>       | abc                            | de                               |
| <def><ghi><jkl> | def                            | ghi                              |

Nella prima dimensione calcolata, l'output della funzione TextBetween restituisce la prima istanza dei caratteri che appaiono tra i delimitatori '<' e '>'. Ad esempio, la riga 1 restituisce abc. La seconda dimensione calcolata ha un argomento n uguale a 2, quindi l'output restituisce la seconda istanza dei caratteri che appaiono tra i delimitatori '<' e '>'. Ad esempio, la riga 1 restituisce de.

## Script di caricamento

Il codice seguente mostra come utilizzare la funzione in uno script di caricamento.

```
Load *,
textbetween(Text, '<', '>') as TextBetween,
textbetween(Text, '<', '>', 2) as SecondTextBetween;
Load * inline [
Text
<abc><de>
<def><ghi><jkl>];
```

Tabella dei risultati

| Testo           | TestoIntermedio | SecondoTestoIntermedio |
|-----------------|-----------------|------------------------|
| <abc><de>       | abc             | de                     |
| <def><ghi><jkl> | def             | ghi                    |

### Esempio: scenario per l'applicazione della funzione TextBetween

Espressione del grafico

#### Panoramica

Un set di dati contiene una tabella dati con gli indirizzi e-mail. In questo esempio si mostra come utilizzare la funzione `TextBetween` per estrarre il dominio di secondo livello dall'indirizzo.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- I seguenti campi nella tabella dati:
  - `UserID`
  - `EmailAddress`

#### Script di caricamento

Example:

```
Load * inline [  
UserID, EmailAddress  
1, alice@example.com  
2, bob@sample.org  
3, charlie@domain.net  
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `UserID`
- `EmailAddress`

Creare la seguente dimensione calcolata:

- `=TextBetween(EmailAddress, '@', '.')`, per restituire il testo tra i caratteri '@' e '.' all'interno del campo `EmailAddress`.

Results table

| UserID | EmailAddress       | TextBetween(EmailAddress, '@', '.') |
|--------|--------------------|-------------------------------------|
| 1      | alice@example.com  | example                             |
| 2      | bob@sample.org     | sample                              |
| 3      | charlie@domain.net | domain                              |

L'output della funzione `TextBetween` restituisce il dominio di secondo livello per ogni ID utente, cercando il contenuto tra i delimitatori specificati: '@' e '.'.

### Trim

`Trim()` restituisce la stringa di input senza spazi iniziali e finali.

#### Sintassi:

```
Trim(text)
```

Tipo di dati restituiti: stringa

| Argomenti |                         |
|-----------|-------------------------|
| Argomento | Descrizione             |
| text      | La stringa da valutare. |

#### Esempio: espressione del grafico

| Esempio                      | Risultato       |
|------------------------------|-----------------|
| <code>Trim( ' abc' )</code>  | Restituisce abc |
| <code>Trim( 'abc ' )</code>  | Restituisce abc |
| <code>Trim( ' abc ' )</code> | Restituisce abc |

### Esempio: principi fondamentali della funzione Trim

Espressione del grafico

#### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata `Example`.
- Un campo della tabella dati denominata `InputText`.

#### Script di caricamento

```
Set verbatim=1;
Example:
Load * inline [
InputText
'   abc   '
' def '];
```



L'istruzione "Set verbatim=1" è inclusa nell'esempio per assicurare che gli spazi non vengano automaticamente tagliati prima della dimostrazione della funzione trim. Vedere [Verbatim \(page 223\)](#) per ulteriori informazioni.

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- InputText

Creare la seguente dimensione calcolata:

- =Trim(InputText), per rimuovere tutti gli spazi iniziali o finali extra da InputText.

Tabella dei risultati

| InputText | Trim(InputText) |
|-----------|-----------------|
| ' abc '   | 'abc'           |
| 'def '    | 'def'           |

L'output della funzione Trim rimuove tutti gli spazi iniziali e finali.

### Esempio: scenario per l'applicazione della funzione Trim

Espressione del grafico

#### Panoramica

Un sistema di gestione delle relazioni con i clienti (CRM) contiene record con i dati immessi in modo incoerente che includono degli spazi iniziali extra. Ai fini della creazione di report, i dati devono essere puliti per rimuovere tali spazi e garantire un ordinamento e un raggruppamento corretto dei nomi dei clienti.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata Example.
- Un campo della tabella dati denominata CustomerName.

#### Script di caricamento

```
Set verbatim=1;
Example:
Load * inline [
CustomerName
' John Doe '
'Jane Smith '
' Michael Johnson'
```

```
'Emily Davis'  
];
```



L'istruzione "Set verbatim=1" è inclusa nell'esempio per assicurare che gli spazi non vengano automaticamente tagliati prima della dimostrazione della funzione trim. Vedere [Verbatim \(page 223\)](#) per ulteriori informazioni.

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- CustomerName

Creare la seguente dimensione calcolata:

- =Trim(CustomerName), per rimuovere tutti gli spazi iniziali o finali extra da CustomerName.

Tabella dei risultati

| CustomerName       | Trim(CustomerName) |
|--------------------|--------------------|
| ' Michael Johnson' | 'Michael Johnson'  |
| ' John Doe '       | 'John Doe'         |
| 'Emily Davis'      | 'Emily Davis'      |
| 'Jane Smith '      | 'Jane Smith'       |

L'output mostra che la funzione trim ha rimosso tutti gli spazi iniziali e finali dai valori della stringa originale in CustomerName.

### Esempio: scenario per l'applicazione della funzione Trim avanzata

Espressione del grafico

#### Panoramica

In questo esempio si mostra come rimuovere tutti gli spazi iniziali e finali dalla stringa di testo originale. L'espressione del grafico include misure che utilizzano la funzione Len per contare i caratteri nella stringa prima e dopo aver utilizzato la funzione Trim.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

- Un set di dati che viene caricato in una tabella dati chiamata Example.
- Un campo della tabella dati denominata string.

#### Script di caricamento

```
Set verbatim=1;  
Example:
```

```
Load * inline [  
String  
'   abc   '  
' def '];
```



L'istruzione "Set verbatim=1" è inclusa nell'esempio per assicurare che gli spazi non vengano automaticamente tagliati prima della dimostrazione della funzione trim. Vedere [Verbatim \(page 223\)](#) per ulteriori informazioni.

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- String

Creare la seguente dimensione calcolata:

- =Trim(String), per rimuovere tutti gli spazi iniziali o finali extra.

Creare le seguenti misure:

- =Len(String), per calcolare la lunghezza della stringa originale.
- =Len(Trim(String)) per calcolare la lunghezza della stringa dopo la rimozione degli spazi.

Tabella dei risultati

| Stringa     | Trim(String) | Len(String) | Len(Trim(String)) |
|-------------|--------------|-------------|-------------------|
| '   abc   ' | 'abc'        | 10          | 3                 |
| ' def '     | 'def'        | 6           | 3                 |

Confrontando l'output della funzione trim con i valori delle stringhe originali nello script, si può notare come tutti gli spazi iniziali e finali siano stati rimossi.

### Script di caricamento

Il codice seguente mostra come utilizzare la funzione in uno script di caricamento.

```
Set verbatim=1;  
Example:  
Load *, len(TrimString) as TrimStringLength;  
Load *, trim(String) as TrimString;  
Load *, len(String) as StringLength;  
Load * inline [  
String  
'   abc   '  
' def '](delimiter is '\t');
```



L'istruzione "Set verbatim=1" è inclusa nell'esempio per assicurare che gli spazi non vengano automaticamente tagliati prima della dimostrazione della funzione trim. Vedere [Verbatim \(page 223\)](#) per ulteriori informazioni.

Tabella dei risultati

| Stringa | StringLength | TrimString | TrimStringLength |
|---------|--------------|------------|------------------|
| abc     | 10           | abc        | 3                |
| def     | 6            | def        | 3                |

## Upper

**Upper()** applica il carattere maiuscolo a tutti i caratteri della stringa di input per tutti i caratteri di testo nell'espressione. I numeri e i simboli vengono ignorati.

### Sintassi:

**Upper** (text)

Tipo di dati restituiti: stringa

Argomenti

| Argomento | Descrizione             |
|-----------|-------------------------|
| text      | La stringa da valutare. |

Esempio: espressione del grafico

| Esempio          | Risultato        |
|------------------|------------------|
| Upper( ' abcd' ) | Restituisce ABCD |

## Esempio: principi fondamentali della funzione Upper

Espressione del grafico

### Panoramica

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

Un set di dati che viene caricato in una tabella dati chiamata Example.

- Un campo della tabella dati denominata InputText.

### Script di caricamento

Example:

```
Load * inline [
```

```
InputText  
rHode iSland  
washingTon d.C.  
new york  
];
```

### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere questo campo come dimensione:

- InputText

Creare la seguente dimensione calcolata:

- =Upper(InputText), per convertire tutti i caratteri del campo InputText in maiuscolo.

Tabella dei risultati

| InputText       | Upper(InputText) |
|-----------------|------------------|
| new york        | NEW YORK         |
| rHode iSland    | RHODE ISLAND     |
| washingTon d.C. | WASHINGTON D.C.  |

L'output della funzione upper converte tutti i valori in maiuscolo.

### Script di caricamento

Il codice seguente mostra come utilizzare la funzione in uno script di caricamento.

```
Load  
String,Upper(String)  
Inline  
[String  
rHode iSland  
washingTon d.C.  
new york];
```

Tabella dei risultati

| Stringa         | Upper(String)   |
|-----------------|-----------------|
| rHode iSland    | RHODE ISLAND    |
| washingTon d.C. | WASHINGTON D.C. |
| new york        | NEW YORK        |

### Esempio: scenario per l'applicazione della funzione Upper

Espressione del grafico

#### Panoramica

Un set di dati contiene i codici ISO dei Paesi e i nomi dei Paesi in formati diversi, ad esempio tutti in minuscolo, tutti in maiuscolo o con maiuscole e minuscole. L'obiettivo è pulire i dati in modo che ogni codice e nome dei Paesi sia formattato in modo da utilizzare le maiuscole.

Aprire l'Editor caricamento dati e aggiungere lo script di caricamento sotto in una nuova sezione.

Lo script di caricamento contiene:

Un set di dati che viene caricato in una tabella dati chiamata `Example`.

- I seguenti campi nella tabella dati:
  - `ISO_CountryCode`
  - `CountryName`

#### Script di caricamento

```
Country:
Load * inline [
ISO_CountryCode, CountryName
gbr, United kingdom
Ind, india
CAN, Canada
Are, United arab emirates
];
```

#### Risultati

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- `ISO_CountryCode`
- `CountryName`

Creare le seguenti dimensioni calcolate:

- `=Upper(ISO_CountryCode)`, per creare una colonna con i codici dei Paesi che utilizzano tutti i caratteri in maiuscolo.
- `=Upper(CountryName)`, per creare una colonna con i nomi dei Paesi che utilizzano tutti i caratteri in maiuscolo.

Tabella dei risultati

| ISO_CountryCode | CountryName          | Upper(ISO_CountryCode) | Upper(CountryName)   |
|-----------------|----------------------|------------------------|----------------------|
| Are             | United arab emirates | ARE                    | UNITED ARAB EMIRATES |
| CAN             | Canada               | CAN                    | CANADA               |
| gbr             | United Kingdom       | GBR                    | UNITED KINGDOM       |
| Ind             | india                | IND                    | INDIA                |

La funzione upper ha convertito tutti i codici paese e i nomi dei paesi in caratteri maiuscoli.

### Esempio: scenario per l'applicazione della funzione Upper avanzata

Espressione del grafico

#### Panoramica

Utilizzando lo stesso set di dati dello scenario precedente, questo esempio combina il nome e il codice dei Paesi e formatta queste stringhe utilizzando le funzioni capitalize e upper.

#### Risultato

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- ISO\_CountryCode
- CountryName

Creare la seguente dimensione calcolata:

- =`Capitalize(CountryName) & ' (' & Upper(ISO_CountryCode) & ')'`, per creare una nuova dimensione che concatena CountryName e ISO\_CountryCode e formatta i nomi dei Paesi in modo da usare le iniziali in maiuscolo e i codici dei Paesi il maiuscolo.

Tabella dei risultati

| ISO_CountryCode | CountryName          | Capitalize(CountryName) & ' (' & Upper(ISO_CountryCode) & ')' |
|-----------------|----------------------|---------------------------------------------------------------|
| Are             | United arab emirates | United Arab Emirates (ARE)                                    |
| CAN             | Canada               | Canada (CAN)                                                  |
| gbr             | United Kingdom       | United Kingdom (GBR)                                          |
| Ind             | india                | India (IND)                                                   |

## 8.25 Funzioni di sistema

Le funzioni di sistema forniscono funzioni per accedere alle proprietà del sistema, del dispositivo e delle app Qlik Sense.

### Prospetto delle funzioni di sistema

Alcune funzioni vengono ulteriormente descritte dopo la panoramica. Per tali funzioni, è inoltre possibile fare clic sul nome della funzione nella sintassi per accedere immediatamente ai dettagli per tale funzione specifica.

#### Author()

Questa funzione restituisce una stringa contenente la proprietà Author dell'app attuale. Può essere utilizzato sia nello script di caricamento dei dati che in un'espressione grafica.



*Non è possibile impostare la proprietà Author nella versione attuale di Qlik Sense. Se si esegue la migrazione di un documento QlikView, la proprietà Author verrà conservata.*

#### CalcDim()

Quando si utilizza in un'espressione di dimensione, **CalcDim()** sovrascrive l'interpretazione automatica del motore su come gestire internamente l'espressione. Nello specifico, la funzione impedisce che venga creato un nuovo campo interno nel modello dati. Invece, le selezioni effettuate nell'app vengono considerate come selezioni nei campi applicabili dal modello dati.

**CalcDim - funzione per grafici** (expr)

#### ClientPlatform()

Questa funzione restituisce la stringa dell'agente utente del browser client. Può essere utilizzato sia nello script di caricamento dei dati che in un'espressione grafica.

#### Esempio:

Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)  
Chrome/35.0.1916.114 Safari/537.36

#### ComputerName

Questa funzione restituisce una stringa contenente il nome del computer come restituito dal sistema operativo. Può essere utilizzato sia nello script di caricamento dei dati che in un'espressione grafica.



*Se il nome del computer ha più di 15 caratteri, la stringa conterrà solo i primi 15 caratteri.*

**ComputerName** ( )

### DocumentName

Questa funzione di script restituisce una stringa contenente il nome dell'app Qlik Sense attuale, senza percorso ma con l'estensione. Può essere utilizzato sia nello script di caricamento dei dati che in un'espressione grafica.

```
DocumentName ( )
```

### DocumentPath

Questa funzione restituisce una stringa contenente il percorso completo dell'app Qlik Sense attuale. Può essere utilizzato sia nello script di caricamento dei dati che in un'espressione grafica.

```
DocumentPath ( )
```



*Questa funzione non è supportata in modalità standard. .*

### DocumentTitle

Questa funzione restituisce una stringa contenente il titolo dell'app Qlik Sense attuale. Può essere utilizzato sia nello script di caricamento dei dati che in un'espressione grafica.

```
DocumentTitle ( )
```

### EngineVersion

Questa funzione restituisce la versione completa dell'engine di Qlik Sense sotto forma di stringa.

```
EngineVersion ( )
```

### GetCollationLocale

Questa funzione di script restituisce il nome della lingua delle impostazioni locali delle regole di confronto utilizzato. Se la variabile CollationLocale non è stata impostata, vengono restituite le impostazioni locali effettive del computer dell'utente.

```
GetCollationLocale ( )
```

### GetObjectField

**GetObjectField()** restituisce l'espressione di campo della dimensione. **Index** è un numero intero opzionale che indica la dimensione che deve essere restituita.

```
GetObjectField - funzione per grafici ([index])
```

### GetSysAttr

La funzione **GetSysAttr()** viene utilizzata per restituire informazioni sul tenant e sullo spazio in cui si trova un'app. Con la funzione **GetSysAttr()**, è possibile valutare l'ID e il nome del tenant, l'ID e il nome dello spazio e altri dettagli sull'ambiente in cui risiede l'app. Questa funzione semplifica l'identificazione dello spazio e del tenant di un'app Qlik Sense che può essere spostata tra ambienti differenti. La funzione consente di identificare facilmente l'ambiente specifico nel quale si lavora. Questa è particolarmente utile durante le fasi di sviluppo, testing, approvazione e produzione (DTAP) del ciclo di vita di un'app.

```
GetSysAttr (name)
```



*Se si usa la funzione in Qlik Sense Client-Managed, restituisce unicamente valori dati vuoti.*

### GetUserAttr

La funzione **GetUserAttr()** restituisce le informazioni per l'utente che accede a un'app. Può essere utilizzato sia nello script di caricamento dei dati che in un'espressione grafica.

GetUserAttr (name)



*Se si usa la funzione in Qlik Sense Client-Managed, restituisce unicamente valori dati vuoti.*

### GroupDimensionIndex

Questa funzione restituisce l'indice del campo attivo per la dimensione ciclica specificata. Nel primo campo il valore di indice è 1. Inserire il nome della dimensione ciclica tra virgolette singole. Questo può essere utilizzato solo nelle espressioni del grafico.

GroupDimensionIndex(dim\_name Expression)

### GroupDimensionLabel

Questa funzione restituisce l'etichetta del campo della fase corrente nella dimensione ciclica specificata. Inserire il nome della dimensione ciclica tra virgolette singole. Questo può essere utilizzato solo nelle espressioni del grafico.

GroupDimensionLabel(dim\_name Expression)

### IsPartialReload

Questa funzione restituisce- 1 (True) se l'operazione di ricaricamento attuale è parziale, altrimenti restituisce 0 (False).

IsPartialReload ()

### InObject

La funzione grafico **InObject()** valuta se l'oggetto corrente è contenuto o meno all'interno di un altro oggetto con l'ID specificato nell'argomento funzione. L'oggetto può essere un foglio o una visualizzazione.

InObject - funzione per grafici (id\_str)

### ObjectId

La funzione del grafico **ObjectId()** restituisce l'ID dell'oggetto nel quale è valutata l'espressione. La funzione accetta un argomento opzionale che specifica il tipo di oggetto a cui si riferisce la funzione. L'oggetto può essere un foglio o una visualizzazione. La funzione è disponibile solo nelle espressioni del grafico.

ObjectId - funzione per grafici ([object\_type\_str])

### OSUser

Questa funzione restituisce una stringa contenente il nome dell'utente attualmente connesso. Può essere utilizzato sia nello script di caricamento dei dati che in un'espressione grafica.

### OSUser ( )



*In Qlik Sense Desktop e Qlik Sense Client-Managed Mobile questa funzione restituisce sempre 'Personal\Me'.*

### ProductVersion

Questa funzione restituisce la versione e il numero di build completi di Qlik Sense sotto forma di stringa.

Questa funzione è deprecata e sostituita da **EngineVersion()**.

### ProductVersion ( )

### ReloadTime

Questa funzione restituisce un'indicazione di data/ora relativa al completamento dell'ultimo caricamento di dati. Può essere utilizzato sia nello script di caricamento dei dati che in un'espressione grafica.

### ReloadTime ( )

### StateName

**StateName()** restituisce il nome dello stato alternato della visualizzazione in cui è utilizzata. StateName può essere utilizzata, ad esempio, per creare visualizzazioni con colori e testo dinamici che mostrano i cambiamenti di stato di una visualizzazione. Questa funzione può essere utilizzata nelle espressioni dei grafici, ma non può essere utilizzata per determinare lo stato a cui si riferisce l'espressione.

### StateName - funzione per grafici ( )

## CalcDim - funzione per grafici

Quando si utilizza in un'espressione di dimensione, **CalcDim()** sovrascrive l'interpretazione automatica del motore su come gestire internamente l'espressione. Nello specifico, la funzione impedisce che venga creato un nuovo campo interno nel modello dati. Invece, le selezioni effettuate nell'app vengono considerate come selezioni nei campi applicabili dal modello dati.

Questa funzione non può essere utilizzata nello script di caricamento. Inoltre, una best practice per le prestazioni del sistema è caricare i campi utilizzati nella dimensione calcolata come campi logici tramite lo script di caricamento quando possibile. **CalcDim()** offre un modo per controllare il comportamento del motore in scenari in cui il caricamento del campo calcolato dallo script non è possibile.

In Qlik Sense, l'espressione della dimensione è per impostazione predefinita realizzata come un campo al volo, ovvero come un campo effettivo, visibile nella barra di selezione corrente, costruito su richiesta. Le selezioni vengono applicate e inferite a questo campo, in modo simile a qualsiasi campo logico dallo script.

Un'espressione racchiusa in **CalcDim()** si comporterà allo stesso modo in cui funzionano tutte le espressioni di dimensione calcolata in QlikView, ovvero una mappatura ad hoc dei valori calcolati ai valori del campo di origine sottostante. Tutti i calcoli field-on-the-fly possono essere trasferiti in dimensioni

calcolate utilizzando la funzione **CalcDim()** poiché la dimensione calcolata deve soddisfare meno requisiti basati sul modello dati rispetto al corrispondente calcolo field-on-the-fly. Questa trasformazione può essere desiderabile per garantire un comportamento simile per tutte le dimensioni calcolate, laddove il modello dati potrebbe impedire l'uso di un calcolo field-on-the-fly in alcuni casi, ma non in altri.

≤ [Campi calcolati](#)

### Sintassi:

```
CalcDim(expr)
```

Tipo di dati restituiti: duale

#### Argomenti

| Argomento | Descrizione                                                                                            |
|-----------|--------------------------------------------------------------------------------------------------------|
| expr      | Un'espressione utilizzata come dimensione o come segmento all'interno di un'espressione di dimensione. |

### Casi di utilizzo

**CalcDim()** è utile quando è necessario un maggiore controllo su come il motore Qlik gestisce i campi richiamati in un'espressione di dimensione.

Per impostazione predefinita, Qlik Sense decide automaticamente se creare e fare riferimento a campi interni nel modello dati quando si utilizza un'espressione di dimensione personalizzata. Qlik Sense crea e utilizza questi campi interni ogni volta che è possibile. Quando non è possibile, l'espressione viene gestita con riferimento ai campi del modello dati esistenti su cui si basa.

A seconda del caso d'uso, questa elaborazione automatica potrebbe non essere sempre desiderabile. Ad esempio, se si sta utilizzando una variabile in un'espressione di dimensione, la creazione automatica di nuovi campi può diventare problematica. In questo caso, utilizzare **CalcDim()** per forzare la gestione dell'espressione senza questo metodo.

Di seguito sono riportati alcuni casi d'uso comuni:

- È stata eseguita la migrazione del contenuto QlikView in Qlik Sense e si desidera conservare il metodo di elaborazione QlikView per le dimensioni calcolate.
- Si desidera semplificare il comportamento di selezione nella propria app. Ad esempio, si preferisce mostrare le selezioni di campo come selezioni nei campi del modello dati originale, piuttosto che selezioni in nuovi campi.
- Il tuo caso d'uso richiede un'espressione ad hoc per coerenza e prestazioni. Ad esempio, se l'espressione della tua dimensione contiene parametri di visualizzazione dinamici controllabili dall'utente, come formattazione controllabile, valori di cut-off o periodi.

### Esempi di funzioni

| Esempio                                                  | Risultato                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|----------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>=CalcDim(if(Year &lt; 2023, 'Older', Year))</code> | <p>Effettuare una selezione in una visualizzazione che utilizza questa dimensione verrà elaborato e visualizzato come una selezione nel valore del campo <i>Year</i> associato dal modello dati.</p> <p>Al contrario, supponiamo che tu non racchiuda l'espressione in <b>CalcDim()</b>. In questo caso, apparirebbe così:</p> <pre>=if(Year &lt; 2023, 'Older', Year)</pre> <p>Effettuare una selezione nell'espressione precedente verrebbe elaborata e visualizzata come una selezione nel campo interno <code>=if(Year &lt; 2023, 'Older', Year)</code>, che non è presente nel modello dati.</p> |

### Esempio 1 - Confronto semplice

Espressione del grafico e risultati

#### Panoramica

In questo esempio, si procederà come segue:

- Aggiungere una tabella inline e caricare i dati in un'app.
- Creare una tabella per visualizzare i dati ed eseguire calcoli.
- Effettua selezioni nella tabella per osservare la differenza di comportamento.

Aprire Editor caricamento dati, quindi aggiungere il seguente script di caricamento in una nuova sezione.

Lo script di caricamento carica una tabella inline. Questa tabella include due campi: un campo per il nome dell'individuo e un campo per il cognome dell'individuo.

#### Script di caricamento

Names:

```
Load * inline [
FirstName, LastName
John, Doe
Amanda, Honda
Amalia, Craig
Cart, Lynch
];
```

#### Creazione tabella

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- *FirstName*
- *LastName*

Successivamente, aggiungi ciascuna delle seguenti espressioni come dimensioni nella tabella:

- *=FirstName & ' ' & LastName*
- *=CalcDim(FirstName & ' ' & LastName)*

La tabella dovrebbe apparire come segue.

Qlik Sense tabella con risultati di esempio

| FirstName | LastName | =FirstName & ' ' & LastName | =CalcDim(FirstName & ' ' & LastName) |
|-----------|----------|-----------------------------|--------------------------------------|
| John      | Doe      | John Doe                    | John Doe                             |
| Amanda    | Honda    | Amanda Honda                | Amanda Honda                         |
| Amalia    | Craig    | Amalia Craig                | Amalia Craig                         |
| Cart      | Lynch    | Cart Lynch                  | Cart Lynch                           |

### Confronto: Effettuare selezioni

Nella colonna *=FirstName & ' ' & LastName* della tabella, selezionare il valore *John Doe*. Si noti che nella barra degli strumenti di selezione, il valore è selezionato come valore da un campo denominato *=FirstName&' '&LastName*.

Cancella la selezione.

Nella colonna *=CalcDim(FirstName & ' ' & LastName)* della tabella, selezionare il valore *John Doe*. Nella barra di selezione, il valore è selezionato come valori da due campi diversi: *FirstName* e *LastName*.

Queste due espressioni presentano le stesse informazioni: una concatenazione dei campi *FirstName* e *LastName*. La differenza è che senza **CalcDim()**, viene creato un campo sintetico per memorizzare il calcolo. Con **CalcDim()**, i campi originali dal modello dati vengono usati come base per la dimensione calcolata.

### Esempio 2 - Caso d'uso pratico per modificare il comportamento predefinito

Espressione del grafico e risultati

#### Panoramica

In questo esempio, creiamo espressioni di dimensione dinamiche che fanno riferimento a una variabile, anziché a valori statici.

In questo esempio, si procederà come segue:

- Aggiungere una tabella inline e caricare i dati nell'app.
- Crea una variabile per archiviare dinamicamente un valore.
- Crea una tabella per visualizzare i dati ed eseguire calcoli.
- Effettua selezioni nella tabella per osservare la differenza di comportamento.

Aprire Editor caricamento dati, quindi aggiungere il seguente script di caricamento in una nuova sezione.

Lo script di caricamento carica una tabella inline. Questa tabella include un campo data e un campo calcolato, *Year*, che è derivato dalle date grezze.

### Script di caricamento

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load *,
```

```
RowNo() as ID,  
Year(date) as Year  
;
```

```
Load * Inline  
[  
date,amount  
'1/19/2022',37.23  
'2/28/2022',88.27  
'3/16/2022',53.80  
'4/1/2022',82.06  
'5/16/2022',87.21  
'6/26/2022',45.89  
'7/9/2022',36.23  
'8/2/2022',76.11  
'9/26/2022',84.21  
'10/29/2022',67.67  
'1/1/2023',100.49  
'2/24/2023',99.47  
'3/6/2023',50.02  
'4/10/2023',2.05  
'5/20/2023',75.64  
'6/25/2023',98.11  
'7/19/2023',30.00  
'9/27/2023',76.11  
'10/26/2023',43.21  
'11/12/2023',135.41  
'1/15/2024',67.24  
];
```

### Creazione di variabili

Caricare i dati e aprire un foglio. Crea una variabile denominata *vCutoffYear* e assegnale la seguente **Definizione:** 2023

### Creazione di tabelle

Caricare i dati e aprire un foglio. Creare una nuova tabella e aggiungere tali campi come dimensioni:

- *date*
- *Year*
- *amount*

Successivamente, aggiungi ciascuna delle seguenti espressioni come dimensioni nella tabella:

- `=if(Year < $(vCutoffYear), 'Older', Year)`
- `=CalcDim(if(Year < $(vCutoffYear), 'Older', Year))`

La tabella dovrebbe apparire come segue.

Qlik Sense tabella con risultati di esempio

| date       | Year | amount | if(Year < \$(vCutoffYear), 'Older', Year) | =CalcDim(if(Year < \$(vCutoffYear), 'Older', Year)) |
|------------|------|--------|-------------------------------------------|-----------------------------------------------------|
| 1/19/2022  | 2022 | 37.23  | Older                                     | Older                                               |
| 2/28/2022  | 2022 | 88.27  | Older                                     | Older                                               |
| 3/16/2022  | 2022 | 53.80  | Older                                     | Older                                               |
| 4/1/2022   | 2022 | 82.06  | Older                                     | Older                                               |
| 5/16/2022  | 2022 | 87.21  | Older                                     | Older                                               |
| 6/26/2022  | 2022 | 45.89  | Older                                     | Older                                               |
| 7/9/2022   | 2022 | 36.23  | Older                                     | Older                                               |
| 8/2/2022   | 2022 | 76.11  | Older                                     | Older                                               |
| 9/26/2022  | 2022 | 84.21  | Older                                     | Older                                               |
| 10/29/2022 | 2022 | 67.67  | Older                                     | Older                                               |
| 1/1/2023   | 2023 | 100.49 | 2023                                      | 2023                                                |
| 2/24/2023  | 2023 | 99.47  | 2023                                      | 2023                                                |
| 3/6/2023   | 2023 | 50.02  | 2023                                      | 2023                                                |
| 4/10/2023  | 2023 | 2.05   | 2023                                      | 2023                                                |
| 5/20/2023  | 2023 | 75.64  | 2023                                      | 2023                                                |
| 6/25/2023  | 2023 | 98.11  | 2023                                      | 2023                                                |
| 7/19/2023  | 2023 | 30.00  | 2023                                      | 2023                                                |
| 9/27/2023  | 2023 | 76.11  | 2023                                      | 2023                                                |
| 10/26/2023 | 2023 | 43.21  | 2023                                      | 2023                                                |

| date       | Year | amount | if(Year < \$(vCutoffYear), 'Older', Year) | =CalcDim(if(Year < \$(vCutoffYear), 'Older', Year)) |
|------------|------|--------|-------------------------------------------|-----------------------------------------------------|
| 11/12/2023 | 2023 | 135.41 | 2023                                      | 2023                                                |
| 1/15/2024  | 2024 | 67.24  | 2024                                      | 2024                                                |

### Confronti, parte 1: Comportamento senza CalcDim()

Nella tabella precedente, non vi è alcuna differenza tra le due colonne di dimensione calcolata.

Nella colonna *if(Year < \$(vCutoffYear), 'Older', Year)* della tabella, selezionare il valore *2023*. Si noti che nella barra degli strumenti di selezione, il valore è selezionato come valore da un campo denominato *=if(Year<2023,'Older',Year)*.

Successivamente, modificare la **Definizione** della *vCutoffYear* variabile in *2024*. Si noti che mentre il campo di selezione *=if(Year<2023,'Older',Year)* è lo stesso e mantiene il valore selezionato *2023*, i dati nella tabella sono cambiati. Il valore *2023* è cambiato nel valore *Older*. Se si seleziona *Older* nella *if(Year < \$(vCutoffYear), 'Older', Year)* colonna, la selezione *Older* viene effettuata in un nuovo campo: *=if(Year<2024,'Older',Year)*.

In sostanza, viene creato internamente un nuovo campo per ogni valore della *vCutoffYear* variabile. Quando si lavora con selezioni concorrenti e mutevoli, questo comportamento potrebbe non essere desiderabile e può aggiungere complessità all'interpretazione dei dati.

### Confronti, parte 2: Comportamento con CalcDim()

Ora che hai visto il comportamento dell'espressione quando non è racchiusa in **CalcDim()**, cancella tutte le selezioni nell'app. Reimposta la variabile *vCutoffYear* sul valore *2023*.

Nella colonna *=CalcDim(if(Year < \$(vCutoffYear), 'Older', Year))* della tabella, seleziona il valore *2023*. Nota che nella barra degli strumenti di selezione, il valore è selezionato come valore dal campo sottostante, *Year*.

Successivamente, modificare la **Definizione** della *vCutoffYear* variabile in *2024*. Il campo di selezione *Year* è lo stesso, mantiene il valore selezionato *2023* e i dati nella tabella sono cambiati, come si vede senza **CalcDim()**. Tuttavia, se ora si seleziona *Older* nella colonna *=CalcDim(if(Year < \$(vCutoffYear), 'Older', Year))*, il valore *2024* viene selezionato in *Year*.

Tutte le selezioni effettuate in una dimensione calcolata racchiusa con **CalcDim()** verranno riflesse come selezioni nei campi sottostanti a cui si fa riferimento dal modello dati originale. Ciò può semplificare l'interpretazione dei dati e impedisce che espressioni complesse e disordinate appaiano nella barra degli strumenti di selezione quando gli utenti utilizzano il contenuto analitico.

## EngineVersion

Questa funzione restituisce la versione completa dell'engine di Qlik Sense sotto forma di stringa.

### Sintassi:

```
EngineVersion()
```

### Funzioni di sistema

Le funzioni di sistema forniscono funzioni per accedere alle proprietà del sistema, del dispositivo e delle app Qlik Sense.

#### Prospetto delle funzioni di sistema

Alcune funzioni vengono ulteriormente descritte dopo la panoramica. Per tali funzioni, è inoltre possibile fare clic sul nome della funzione nella sintassi per accedere immediatamente ai dettagli per tale funzione specifica.

#### Author()

Questa funzione restituisce una stringa contenente la proprietà Author dell'app attuale. Può essere utilizzato sia nello script di caricamento dei dati che in un'espressione grafica.



*Non è possibile impostare la proprietà Author nella versione attuale di Qlik Sense. Se si esegue la migrazione di un documento QlikView, la proprietà Author verrà conservata.*

#### CalcDim()

Quando si utilizza in un'espressione di dimensione, **CalcDim()** sovrascrive l'interpretazione automatica del motore su come gestire internamente l'espressione. Nello specifico, la funzione impedisce che venga creato un nuovo campo interno nel modello dati. Invece, le selezioni effettuate nell'app vengono considerate come selezioni nei campi applicabili dal modello dati.

[CalcDim - funzione per grafici](#) (expr)

#### ClientPlatform()

Questa funzione restituisce la stringa dell'agente utente del browser client. Può essere utilizzato sia nello script di caricamento dei dati che in un'espressione grafica.

#### Esempio:

Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)  
Chrome/35.0.1916.114 Safari/537.36

#### ComputerName

Questa funzione restituisce una stringa contenente il nome del computer come restituito dal sistema operativo. Può essere utilizzato sia nello script di caricamento dei dati che in un'espressione grafica.



*Se il nome del computer ha più di 15 caratteri, la stringa conterrà solo i primi 15 caratteri.*

**ComputerName ( )**

### DocumentName

Questa funzione di script restituisce una stringa contenente il nome dell'app Qlik Sense attuale, senza percorso ma con l'estensione. Può essere utilizzato sia nello script di caricamento dei dati che in un'espressione grafica.

```
DocumentName ( )
```

### DocumentPath

Questa funzione restituisce una stringa contenente il percorso completo dell'app Qlik Sense attuale. Può essere utilizzato sia nello script di caricamento dei dati che in un'espressione grafica.

```
DocumentPath ( )
```



*Questa funzione non è supportata in modalità standard. .*

### DocumentTitle

Questa funzione restituisce una stringa contenente il titolo dell'app Qlik Sense attuale. Può essere utilizzato sia nello script di caricamento dei dati che in un'espressione grafica.

```
DocumentTitle ( )
```

### EngineVersion

Questa funzione restituisce la versione completa dell'engine di Qlik Sense sotto forma di stringa.

```
EngineVersion ( )
```

### GetCollationLocale

Questa funzione di script restituisce il nome della lingua delle impostazioni locali delle regole di confronto utilizzato. Se la variabile CollationLocale non è stata impostata, vengono restituite le impostazioni locali effettive del computer dell'utente.

```
GetCollationLocale ( )
```

### GetObjectField

**GetObjectField()** restituisce l'espressione di campo della dimensione. **Index** è un numero intero opzionale che indica la dimensione che deve essere restituita.

```
GetObjectField - funzione per grafici ([index])
```

### GetSysAttr

La funzione **GetSysAttr()** viene utilizzata per restituire informazioni sul tenant e sullo spazio in cui si trova un'app. Con la funzione **GetSysAttr()**, è possibile valutare l'ID e il nome del tenant, l'ID e il nome dello spazio e altri dettagli sull'ambiente in cui risiede l'app. Questa funzione semplifica l'identificazione dello spazio e del tenant di un'app Qlik Sense che può essere spostata tra ambienti differenti. La funzione consente di identificare facilmente l'ambiente specifico nel quale si lavora. Questa è particolarmente utile durante le fasi di sviluppo, testing, approvazione e produzione (DTAP) del ciclo di vita di un'app.

```
GetSysAttr (name)
```



*Se si usa la funzione in Qlik Sense Client-Managed, restituisce unicamente valori dati vuoti.*

### GetUserAttr

La funzione **GetUserAttr()** restituisce le informazioni per l'utente che accede a un'app. Può essere utilizzato sia nello script di caricamento dei dati che in un'espressione grafica.

GetUserAttr (name)



*Se si usa la funzione in Qlik Sense Client-Managed, restituisce unicamente valori dati vuoti.*

### GroupDimensionIndex

Questa funzione restituisce l'indice del campo attivo per la dimensione ciclica specificata. Nel primo campo il valore di indice è 1. Inserire il nome della dimensione ciclica tra virgolette singole. Questo può essere utilizzato solo nelle espressioni del grafico.

GroupDimensionIndex(dim\_name Expression)

### GroupDimensionLabel

Questa funzione restituisce l'etichetta del campo della fase corrente nella dimensione ciclica specificata. Inserire il nome della dimensione ciclica tra virgolette singole. Questo può essere utilizzato solo nelle espressioni del grafico.

GroupDimensionLabel(dim\_name Expression)

### IsPartialReload

Questa funzione restituisce- 1 (True) se l'operazione di ricaricamento attuale è parziale, altrimenti restituisce 0 (False).

IsPartialReload ()

### InObject

La funzione grafico **InObject()** valuta se l'oggetto corrente è contenuto o meno all'interno di un altro oggetto con l'ID specificato nell'argomento funzione. L'oggetto può essere un foglio o una visualizzazione.

InObject - funzione per grafici (id\_str)

### ObjectId

La funzione del grafico **ObjectId()** restituisce l'ID dell'oggetto nel quale è valutata l'espressione. La funzione accetta un argomento opzionale che specifica il tipo di oggetto a cui si riferisce la funzione. L'oggetto può essere un foglio o una visualizzazione. La funzione è disponibile solo nelle espressioni del grafico.

ObjectId - funzione per grafici ([object\_type\_str])

### OSUser

Questa funzione restituisce una stringa contenente il nome dell'utente attualmente connesso. Può essere utilizzato sia nello script di caricamento dei dati che in un'espressione grafica.

### OSUser ( )



*In Qlik Sense Desktop e Qlik Sense Client-Managed Mobile questa funzione restituisce sempre 'Personal\Me'.*

### ProductVersion

Questa funzione restituisce la versione e il numero di build completi di Qlik Sense sotto forma di stringa.

Questa funzione è deprecata e sostituita da **EngineVersion()**.

### ProductVersion ( )

### ReloadTime

Questa funzione restituisce un'indicazione di data/ora relativa al completamento dell'ultimo caricamento di dati. Può essere utilizzato sia nello script di caricamento dei dati che in un'espressione grafica.

### ReloadTime ( )

### StateName

**StateName()** restituisce il nome dello stato alternato della visualizzazione in cui è utilizzata. StateName può essere utilizzata, ad esempio, per creare visualizzazioni con colori e testo dinamici che mostrano i cambiamenti di stato di una visualizzazione. Questa funzione può essere utilizzata nelle espressioni dei grafici, ma non può essere utilizzata per determinare lo stato a cui si riferisce l'espressione.

### StateName - funzione per grafici ( )

## GetSysAttr

La funzione **GetSysAttr()** viene utilizzata per restituire informazioni sul tenant e sullo spazio in cui si trova un'app. Con la funzione **GetSysAttr()**, è possibile valutare l'ID e il nome del tenant, l'ID e il nome dello spazio e altri dettagli sull'ambiente in cui risiede l'app. Questa funzione semplifica l'identificazione dello spazio e del tenant di un'app Qlik Sense che può essere spostata tra ambienti differenti. La funzione consente di identificare facilmente l'ambiente specifico nel quale si lavora. Questa è particolarmente utile durante le fasi di sviluppo, testing, approvazione e produzione (DTAP) del ciclo di vita di un'app.

È possibile utilizzarla nello script di caricamento o nelle espressioni dei grafici, direttamente come un'espressione o in un'espressione memorizzata in una variabile. Se si prevede che la propria app verrà spostata tra diversi spazi o tenant durante il suo ciclo di vita, usare la funzione in un'espressione del grafico per ottenere in modo dinamico informazioni sullo spazio e sul tenant senza ricaricare l'app. Quando si esegue questa operazione, memorizzare l'espressione della funzione in una variabile esterna allo script di caricamento o usare la funzione direttamente in un'espressione di un grafico.

Utilizzare **GetSysAttr()** per determinare le seguenti informazioni sull'app:

- L'ID del tenant in cui si trova l'app.
- Il nome del tenant in cui si trova l'app.
- L'URL del tenant in cui si trova l'app.
- L'URL alias che è stato definito per il tenant in cui si trova l'app.
- La regione del tenant in cui si trova l'app.
- L'ID dello spazio in cui si trova l'app.
- Il nome dello spazio in cui si trova l'app.
- Il tipo di spazio (condiviso o gestito) in cui si trova l'app.

Se si usa la funzione in Qlik Sense Client-Managed, restituisce valori dati vuoti. Pertanto, è possibile utilizzare la funzione per sviluppare script di caricamento in Qlik Sense Client-Managed senza rilevare errori, per caricare successivamente le app in Qlik Cloud.

Per accedere alla documentazione completa per la funzionalità Qlik Cloud, vedere [GetSysAttr -funzione di script grafico](#).

### GetUserAttr

Questa funzione restituisce gli attributi dell'utente, tra cui l'indirizzo e-mail, l'argomento e i gruppi, per un'app selezionata. Può essere utilizzato sia nello script di caricamento dei dati che in un'espressione grafica.

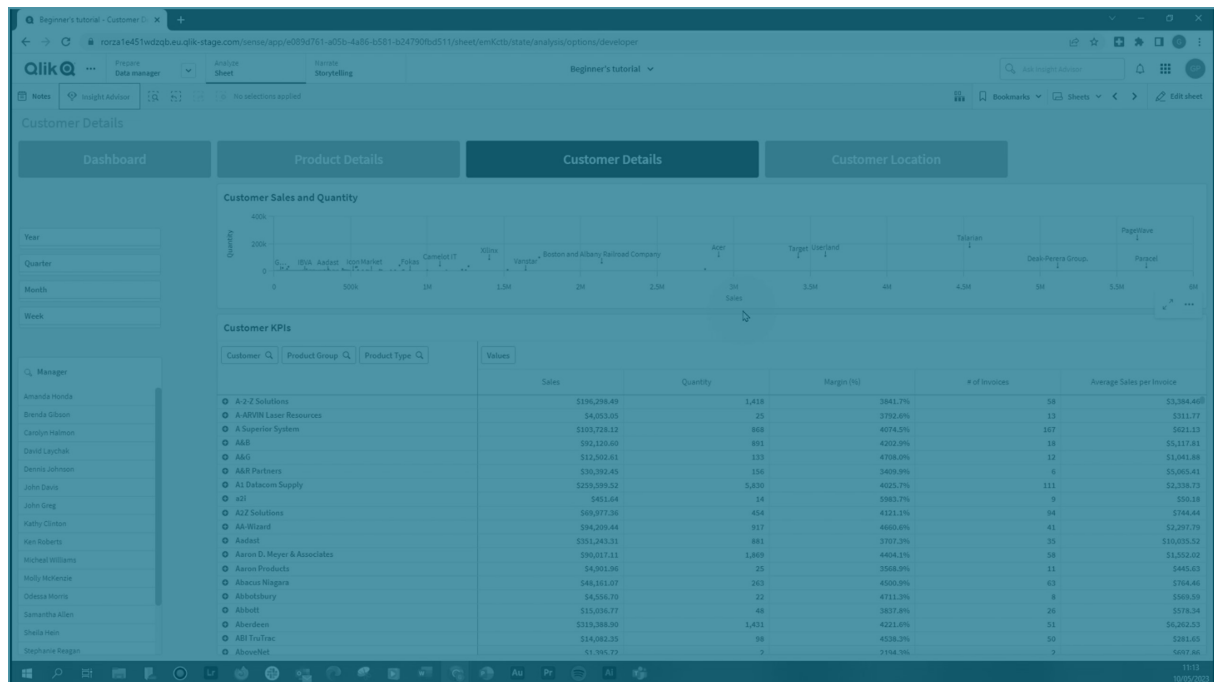
Se si usa la funzione in Qlik Sense Client-Managed, restituisce valori dati vuoti. Pertanto, è possibile utilizzare la funzione per sviluppare script di caricamento in Qlik Sense Client-Managed senza rilevare errori, per caricare successivamente le app in Qlik Cloud.

Per accedere alla documentazione completa per la funzione in Qlik Cloud, vedere [GetUserAttr - funzione di script grafico](#).

### InObject - funzione per grafici

La funzione grafico **InObject()** valuta se l'oggetto corrente è contenuto o meno all'interno di un altro oggetto con l'ID specificato nell'argomento funzione. L'oggetto può essere un foglio o una visualizzazione.

Questa funzione può essere utilizzata per mostrare la gerarchia degli oggetti in un foglio, dall'oggetto di livello superiore del foglio alle visualizzazioni nidificate all'interno di altre visualizzazioni. Questa funzione può essere utilizzata insieme alle funzioni **if** e **ObjectId** per creare una navigazione personalizzata nelle app.



### Sintassi:

```
InObject(id_str)
```

### Tipo di dati restituiti: Booleano

In Qlik Sense, il valore booleano vero è rappresentato da -1 e il valore falso è rappresentato da 0.

### Argomenti

| Argomento | Descrizione                                                      |
|-----------|------------------------------------------------------------------|
| id_str    | Un valore stringa che rappresenta l'ID dell'oggetto da valutare. |

L'ID del foglio può essere ottenuto dall'URL dell'app. Per le visualizzazioni, utilizzare le opzioni **Sviluppatore** per identificare l'ID dell'oggetto e la stringa di testo del tipo di oggetto.

### Procedere come indicato di seguito:

1. In modalità analisi, aggiungere il seguente testo all'URL:  
`/options/developer`
2. Fare clic con il pulsante destro del mouse su una visualizzazione e fare clic su **Sviluppatore**.
3. Sotto **Proprietà**, ottenere l'ID oggetto dall'installazione finestra di dialogo, oltre al tipo di oggetto dalla proprietà **"qType"**.

### Limiti:

Questa funzione può dare risultati imprevisti se richiamata in un oggetto (ad esempio, un pulsante) all'interno di un contenitore che rappresenta una voce principale. Questa limitazione si applica anche alle voci principali della casella di filtro, che rappresentano contenitori di diverse caselle di elenco. Ciò si

verifica a causa del modo in cui le voci principali utilizzano la gerarchia oggetti.

**InObject()** è spesso utilizzato in combinazione con le seguenti funzioni:

### Funzioni correlate

| Funzione                                                    | Interazione                                                                                                                                                                                                                                        |
|-------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">if (page 649)</a>                               | Le funzioni <b>if</b> e <b>ObjectId</b> possono essere utilizzate insieme per creare espressioni condizionali. Ad esempio, le visualizzazioni possono ottenere una colorazione condizionale attraverso espressioni che utilizzano queste funzioni. |
| <a href="#">ObjectId - funzione per grafici (page 1892)</a> | In modo simile a <b>if</b> , <b>ObjectId</b> è utilizzato anche con <b>InObject</b> per creare espressioni condizionali.                                                                                                                           |

### Esempio 1 - Funzionalità di base

#### Espressione del grafico e risultati

Il seguente esempio di base mostra come determinare se un oggetto è contenuto in un altro oggetto. In questo caso, controlleremo se un oggetto **Testo e immagine** risiede in un oggetto foglio, utilizzando l'ID del foglio come argomento.

#### Procedere come indicato di seguito:

1. Aprire un nuovo foglio e trascinare un grafico **Testo e immagine** sul foglio.
2. Nel pannello proprietà, fare clic su **Aggiungi misura**.
3. Fare clic su  $f_x$  per visualizzare l'editor delle espressioni.

4. Incollare la seguente espressione nella finestra di dialogo:  
`=Inobject()`
5. Modificare l'espressione per includere l'ID del foglio come stringa tra le parentesi.  
Ad esempio, per un foglio con ID 1234-5678, si utilizza quanto segue:  
`=Inobject('1234-5678')`
6. Fare clic su **Applica**.

Il valore -1 viene visualizzato nel grafico, a indicare che l'espressione è stata valutata come vera.

### Esempio 2 - Oggetti con colori condizionati

Espressione del grafico e risultati

#### Panoramica

L'esempio seguente mostra come creare pulsanti di navigazione personalizzati con colorazioni diverse per indicare il foglio attualmente aperto.

Si inizia creando una nuova app e aprendo l'editor caricamento dati. Incollare il seguente script di caricamento in una nuova scheda. Si noti che i dati stessi sono un segnaposto e non saranno utilizzati nel contenuto dell'esempio.

#### Script di caricamento

Transactions:

```
Load
*
Inline
[
id,date,amount
8188,'1/19/2022',37.23
8189,'1/7/2022',17.17
8190,'2/28/2022',88.27
8191,'2/5/2022',57.42
8192,'3/16/2022',53.80
8193,'4/1/2022',82.06
8194,'4/7/2022',40.39
8195,'5/16/2022',87.21
8196,'6/15/2022',95.93
8197,'7/26/2022',45.89
8198,'8/9/2022',36.23
8199,'9/22/2022',25.66
8200,'11/23/2022',82.77
8201,'12/27/2022',69.98
8202,'1/1/2023',76.11
8203,'2/8/2022',25.12
8204,'3/19/2022',46.23
8205,'6/26/2022',84.21
8206,'9/14/2022',96.24
8207,'11/29/2022',67.67
];
```

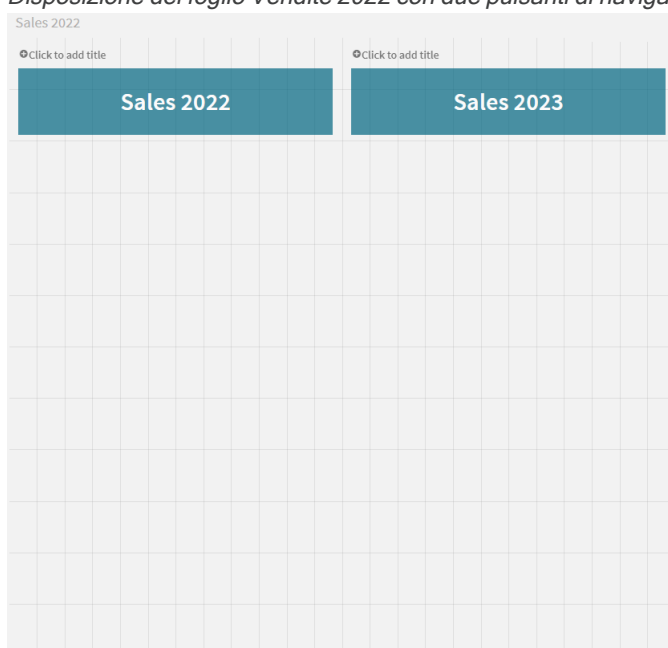
### Creazione delle visualizzazioni

Caricare i dati e creare due nuovi fogli. Intitolarli rispettivamente *Vendite 2022* e *Vendite 2023*.

Successivamente, costruire due oggetti pulsante che verranno utilizzati per navigare tra i due fogli.

**Procedere come indicato di seguito:**

1. Aggiungere due oggetti **Pulsante** al foglio.
2. Sotto **Aspetto > Generale**, impostare l'**Etichetta** di ciascun pulsante rispettivamente su *Vendite 2022* e *Vendite 2023*.
3. Disporre i pulsanti in modo che corrispondano all'immagine seguente.  
*Disposizione del foglio Vendite 2022 con due pulsanti di navigazione*



4. Selezionare il pulsante *Vendite 2022* ed espandere **Azioni e navigazione** nel pannello delle proprietà.
5. Fare clic su **Aggiungi azione** e sotto **Navigazione**, selezionare **Vai in un foglio**.
6. Sotto **Foglio**, selezionare *Vendite 2022*.
7. Ripetere la configurazione dell'azione di questo pulsante per collegare il pulsante **Vendite 2023** al foglio *Vendite 2023*.
8. Per convertire i pulsanti in voci principali, fare clic con il pulsante destro del mouse e selezionare  **Aggiungi a voci principali**.

Ora è possibile copiare ogni pulsante e incollarlo nel foglio *Vendite 2023*, utilizzando le stesse dimensioni e la stessa disposizione sul foglio.

### Creazione di colori condizionali

Quindi, configurare i pulsanti in modo che siano blu se sono collegati al foglio attualmente aperto e grigio chiaro se sono collegati al foglio non aperto.

Procedere come indicato di seguito:

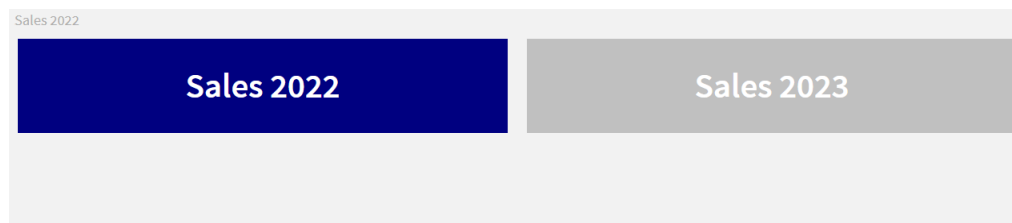
1. Aprire il foglio *Vendite 2022* e ottenere l'ID foglio dall'URL. Mantenere aperto il foglio *Vendite 2022*.
2. Fare clic sulla voce principale del pulsante **Vendite 2022** e selezionare **Modifica** nel pannello proprietà.
3. Sotto **Aspetto** > **Sfondo**, selezionare per colorare il pulsante **Per espressione**.
4. In **Espressione**, incollare il testo seguente:  
`=if(InObject(""), Blue(), LightGray())`
5. Tra le parentesi dell'espressione precedente, incollare l'ID del foglio *Vendite 2022*.

Il pulsante è ora configurato per diventare blu quando il foglio *Vendite 2022* è aperto e grigio chiaro quando non è aperto.

Ripetere le istruzioni di cui sopra per il foglio *Vendite 2023*, collegando la voce principale del pulsante **Vendite 2023** all'ID foglio *Vendite 2023*.

Ogni foglio dovrebbe ora avere due pulsanti che indicano il foglio attualmente aperto con il colore blu.

*Foglio Vendite 2022 con colorazione blu per indicare che Vendite 2022 risulta attualmente visualizzato*



### IsPartialReload

Questa funzione restituisce- 1 (True) se l'operazione di ricaricamento attuale è parziale, altrimenti restituisce 0 (False).

**Sintassi:**

```
IsPartialReload()
```

### ObjectId - funzione per grafici

La funzione del grafico **ObjectId()** restituisce l'ID dell'oggetto nel quale è valutata l'espressione. La funzione accetta un argomento opzionale che specifica il tipo di oggetto a cui si riferisce la funzione. L'oggetto può essere un foglio o una visualizzazione. La funzione è disponibile solo nelle espressioni del grafico.

**Sintassi:**

```
ObjectId([object_type_str])
```

**Tipo di dati restituiti:** stringa

L'unico argomento della funzione, **object\_type\_str**, è opzionale e si riferisce a un valore stringa che rappresenta il tipo dell'oggetto.

Argomenti

| Argomento              | Descrizione                                                         |
|------------------------|---------------------------------------------------------------------|
| <b>object_type_str</b> | Un valore stringa che rappresenta il tipo dell'oggetto da valutare. |

Se nell'espressione della funzione non è specificato alcun argomento, **ObjectId()** restituisce l'ID dell'oggetto in cui viene utilizzata l'espressione. Per restituire l'ID dell'oggetto foglio all'interno del quale appare la visualizzazione, usare *ObjectId('sheet')*.

Nel caso di oggetti di visualizzazione nidificati all'interno di altri oggetti di visualizzazione, specificare il tipo di oggetto desiderato nell'argomento della funzione per ottenere risultati diversi. Ad esempio, per un grafico **Testo e immagine** con un contenitore, utilizzare *'text-image'* per restituire l'oggetto **Testo e immagine** e *'contenitore'* per restituire l'ID del contenitore.

**Procedere come indicato di seguito:**

1. In modalità analisi, aggiungere il seguente testo all'URL:  
*/options/developer*
2. Fare clic con il pulsante destro del mouse su una visualizzazione e fare clic su  **Sviluppatore**.
3. Sotto **Proprietà**, ottenere l'ID oggetto dall'intestazione finestra di dialogo, oltre al tipo di oggetto dalla proprietà **"qType"**.

**Limiti:**

Questa funzione può dare risultati imprevisti se richiamata in un oggetto (ad esempio, un pulsante) all'interno di un contenitore che rappresenta una voce principale. Questa limitazione si applica anche alle voci principali della casella di filtro, che rappresentano contenitori di diverse caselle di elenco. Ciò si verifica a causa del modo in cui le voci principali utilizzano la gerarchia oggetti.

L'espressione grafico *ObjectId('sheet')* restituirà una stringa vuota in tali casi, mentre *ObjectId('masterobject')* mostrerà l'identificatore della voce principale posseduta.

**ObjectId()** viene spesso utilizzato in combinazione con le seguenti funzioni:

## Funzioni correlate

| Funzione                                                    | Interazione                                                                                                                                                                                                                                        |
|-------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">if (page 649)</a>                               | Le funzioni <b>if</b> e <b>ObjectId</b> possono essere utilizzate insieme per creare espressioni condizionali. Ad esempio, le visualizzazioni possono ottenere una colorazione condizionale attraverso espressioni che utilizzano queste funzioni. |
| <a href="#">InObject - funzione per grafici (page 1887)</a> | In modo simile a <b>if</b> , <b>InObject</b> viene utilizzato anche con <b>ObjectId</b> per creare espressioni condizionali.                                                                                                                       |

## Esempio 1 - Restituzione ID oggetto grafico

## Espressione del grafico e risultati

Il seguente esempio di base dimostra come restituire l'ID di una visualizzazione.

## Procedere come indicato di seguito:

1. Aprire un nuovo foglio e trascinare un grafico **Testo e immagine** sul foglio.
2. Nel pannello proprietà, fare clic su **Aggiungi misura**.
3. Fare clic su  $f_x$  per visualizzare l'editor delle espressioni.
4. Incollare la seguente espressione nella finestra di dialogo:  
=ObjectId()
5. Fare clic su **Applica**.

L'ID dell'oggetto **Testo e immagine** viene mostrato nella visualizzazione.

Lo stesso risultato può essere ottenuto con la seguente espressione:

```
=ObjectId('text-image')
```

### Esempio 2 - ID del foglio di ritorno

Espressione del grafico e risultati

Il seguente esempio di base dimostra come restituire l'ID del foglio in cui appare una visualizzazione.

**Procedere come indicato di seguito:**

1. Aprire un nuovo foglio e trascinare un grafico **Testo e immagine** sul foglio.
2. Nel pannello proprietà, fare clic su **Aggiungi misura**.
3. Fare clic su  $f_x$  per visualizzare l'editor delle espressioni.
4. Incollare la seguente espressione nella finestra di dialogo:  
`=ObjectId('sheet')`
5. Fare clic su **Applica**.

L'ID del foglio viene visualizzato nella visualizzazione.

### Esempio 3 - Espressione nidificata

Espressione del grafico e risultati

L'esempio seguente mostra come la funzione **ObjectId()** possa essere nidificata all'interno di altre espressioni.

**Procedere come indicato di seguito:**

1. Aprire un nuovo foglio e trascinare un grafico **Testo e immagine** sul foglio.
2. Nel pannello proprietà, fare clic su **Aggiungi misura**.
3. Fare clic su  $f_x$  per visualizzare l'editor delle espressioni.
4. Incollare la seguente espressione nella finestra di dialogo:  
`=if(InObject(ObjectId('text-image')), 'In Text & image', 'Not in Text & image')`
5. Fare clic su **Applica**.

Il testo *In Testo e immagine* appare nel grafico, indicando che l'oggetto a cui si fa riferimento nell'espressione è un grafico **Testo e immagine**.

Per un esempio più dettagliato di utilizzo della colorazione condizionale, vedere l'esempio su [InObject - funzione per grafici \(page 1887\)](#).

## ProductVersion

Questa funzione restituisce la versione e il numero di build completi di Qlik Sense sotto forma di stringa. Questa funzione è deprecata e sostituita da **EngineVersion()**.

**Sintassi:**

```
ProductVersion()
```

## StateName - funzione per grafici

**StateName()** restituisce il nome dello stato alternato della visualizzazione in cui è utilizzata. StateName può essere utilizzata, ad esempio, per creare visualizzazioni con colori e testo dinamici che mostrano i cambiamenti di stato di una visualizzazione. Questa funzione può essere utilizzata nelle espressioni dei grafici, ma non può essere utilizzata per determinare lo stato a cui si riferisce l'espressione.

### Sintassi:

```
StateName ()
```

### Example 1:

```
Testo dinamico
='Region - ' & if(StateName() = '$', 'Default', StateName())
```

### Example 2:

```
Colori dinamici
if(StateName() = 'Group 1', rgb(152, 171, 206),
    if(StateName() = 'Group 2', rgb(187, 200, 179),
        rgb(210, 210, 210)
    )
)
```

## 8.26 Funzioni di tabella

Le funzioni di tabella restituiscono informazioni relative alla tabella dati in fase di caricamento. Se non viene specificato alcun nome di tabella e la funzione viene utilizzata all'interno di un'istruzione **LOAD**, viene utilizzata la tabella attuale.

Nello script di caricamento dei dati è possibile utilizzare tutte le funzioni, mentre in un'espressione grafica è possibile utilizzare solo **NoOfRows**.

## Panoramica sulle funzioni di tabella

Alcune funzioni vengono ulteriormente descritte dopo la panoramica. Per tali funzioni, è inoltre possibile fare clic sul nome della funzione nella sintassi per accedere immediatamente ai dettagli per tale funzione specifica.

### FieldName

La funzione di script **FieldName** restituisce il nome del campo con il numero specificato all'interno di una tabella caricata in precedenza. Se la funzione viene utilizzata all'interno di un'istruzione **LOAD**, non deve fare riferimento alla tabella in corso di caricamento.

```
FieldName (field_number ,table_name)
```

### FieldName

La funzione di script **FieldName** restituisce il numero di un campo specifico all'interno di una tabella caricata in precedenza. Se la funzione viene utilizzata all'interno di un'istruzione **LOAD**, non deve fare riferimento alla tabella in corso di caricamento.

```
FieldName (field_name ,table_name)
```

### NoOfFields

La funzione di script **NoOfFields** restituisce il numero di campi all'interno di una tabella caricata in precedenza. Se la funzione viene utilizzata all'interno di un'istruzione **LOAD**, non deve fare riferimento alla tabella in corso di caricamento.

```
NoOfFields (table_name)
```

### NoOfRows

La funzione **NoOfRows** restituisce il numero di righe (record) all'interno di una tabella caricata in precedenza. Se la funzione viene utilizzata all'interno di un'istruzione **LOAD**, non deve fare riferimento alla tabella in corso di caricamento.

```
NoOfRows (table_name)
```

### NoOfTables

Questa funzione di script restituisce il numero di tabelle caricate in precedenza.

```
NoOfTables ()
```

### TableName

Questa funzione di script restituisce il nome della tabella con il numero specificato.

```
TableName (table_number)
```

### TableNumber

Questa funzione di script restituisce il numero della tabella specificata. La tabella ripetizione ha il numero 0.

Se table\_name non esiste, viene restituito il valore NULL.

```
TableNumber (table_name)
```

### Esempio:

In questo esempio si desidera creare una tabella con informazioni sulle tabelle e sui campi caricati.

Innanzitutto si caricheranno alcuni dati semplici. Questa operazione crea le due tabelle che verranno utilizzate per illustrare le funzioni di tabella descritte in questa sezione.

Characters:

```
Load Chr(RecNo()+Ord('A')-1) as Alpha, RecNo() as Num autogenerate 26;
```

ASCII:

```
Load
```

```

if(RecNo()>=65 and RecNo()<=90,RecNo()-64) as Num,
Chr(RecNo()) as AsciiAlpha,
RecNo() as AsciiNum
autogenerate 255
where (RecNo()>=32 and RecNo()<=126) or RecNo()>=160 ;

```

In seguito, si ripete l'operazione sulle tabelle caricate utilizzando la funzione **NoOfTables**, quindi sui campi di ogni tabella utilizzando la funzione **NoOfFields** e si caricheranno le informazioni utilizzando le funzioni di tabella.

```

//Iterate through the loaded tables
For t = 0 to NoOfTables() - 1

//Iterate through the fields of table
For f = 1 to NoOfFields(TableName$(t))
Tables:
Load
    TableName$(t) as Table,
    TableNumber(TableName$(t)) as TableNo,
    NoOfRows(TableName$(t)) as TableRows,
    FieldName$(f),TableName$(t)) as Field,
    FieldNumber(FieldName$(f),TableName$(t)),TableName$(t)) as FieldNo
    Autogenerate 1;
Next f
Next t;

```

La tabella Tables risultante avrà il seguente aspetto:

Load table

| Table      | TableNo | TableRows | Field      | FieldNo |
|------------|---------|-----------|------------|---------|
| Characters | 0       | 26        | Alpha      | 1       |
| Characters | 0       | 26        | Num        | 2       |
| ASCII      | 1       | 191       | Num        | 1       |
| ASCII      | 1       | 191       | AsciiAlpha | 2       |
| ASCII      | 1       | 191       | AsciiNum   | 3       |

### FieldName

La funzione di script **FieldName** restituisce il nome del campo con il numero specificato all'interno di una tabella caricata in precedenza. Se la funzione viene utilizzata all'interno di un'istruzione **LOAD**, non deve fare riferimento alla tabella in corso di caricamento.

#### Sintassi:

```
FieldName(field_number ,table_name)
```

**Argomenti:**

Argomenti

| Argomento    | Descrizione                                                        |
|--------------|--------------------------------------------------------------------|
| field_number | Il numero del campo a cui si desidera fare riferimento.            |
| table_name   | La tabella contenente il campo a cui si desidera fare riferimento. |

**Esempio:**

```
LET a = FieldName(4,'tab1');
```

## FieldNumber

La funzione di script **FieldNumber** restituisce il numero di un campo specifico all'interno di una tabella caricata in precedenza. Se la funzione viene utilizzata all'interno di un'istruzione **LOAD**, non deve fare riferimento alla tabella in corso di caricamento.

**Sintassi:**

```
FieldNumber(field_name ,table_name)
```

**Argomenti:**

Argomenti

| Argomento  | Descrizione                                |
|------------|--------------------------------------------|
| field_name | Il nome del campo.                         |
| table_name | Il nome della tabella contenente il campo. |

Se il campo field\_name non esiste in table\_name o table\_name non esiste, la funzione restituisce 0.

**Esempio:**

```
LET a = FieldNumber('Customer','tab1');
```

## NoOfFields

La funzione di script **NoOfFields** restituisce il numero di campi all'interno di una tabella caricata in precedenza. Se la funzione viene utilizzata all'interno di un'istruzione **LOAD**, non deve fare riferimento alla tabella in corso di caricamento.

**Sintassi:**

```
NoOfFields(table_name)
```

**Argomenti:**

| Argomenti  |                        |
|------------|------------------------|
| Argomento  | Descrizione            |
| table_name | Il nome della tabella. |

**Esempio:**

```
LET a = NoOfFields('tab1');
```

**NoOfRows**

La funzione **NoOfRows** restituisce il numero di righe (record) all'interno di una tabella caricata in precedenza. Se la funzione viene utilizzata all'interno di un'istruzione **LOAD**, non deve fare riferimento alla tabella in corso di caricamento.

**Sintassi:**

```
NoOfRows (table_name)
```

**Argomenti:**

| Argomenti  |                        |
|------------|------------------------|
| Argomento  | Descrizione            |
| table_name | Il nome della tabella. |

**Esempio:**

```
LET a = NoOfRows('tab1');
```

## 8.27 Funzioni trigonometriche e iperboliche

In questa sezione vengono descritte le funzioni per l'esecuzione delle operazioni trigonometriche e iperboliche. In tutte le funzioni gli argomenti sono espressioni che restituiscono angoli misurati in radianti, in cui **x** deve essere interpretato come un numero reale.

Tutti gli angoli sono misurati in radianti.

Tutte le funzioni possono essere utilizzate sia nello script di caricamento dei dati che nelle espressioni grafiche.

**cos**

Coseno di **x**. Il risultato è un numero compreso tra -1 e 1.

```
cos ( x )
```

### **acos**

Coseno inverso di **x**. La funzione è definita solo se  $-1 \leq x \leq 1$ . Il risultato è un numero compreso tra 0 e  $\pi$ .

```
acos( x )
```

### **sin**

Seno di **x**. Il risultato è un numero compreso tra -1 e 1.

```
sin( x )
```

### **asin**

Seno inverso di **x**. La funzione è definita solo se  $-1 \leq x \leq 1$ . Il risultato è un numero compreso tra  $-\pi/2$  e  $\pi/2$ .

```
asin( x )
```

### **tan**

Tangente di **x**. Il risultato è un numero reale.

```
tan( x )
```

### **atan**

Tangente inversa di **x**. Il risultato è un numero compreso tra  $-\pi/2$  e  $\pi/2$ .

```
atan( x )
```

### **atan2**

Generalizzazione bidimensionale della funzione della tangente inversa. Restituisce l'angolo compreso tra l'origine e il punto rappresentato dalle coordinate **x** e **y**. Il risultato è un numero compreso tra  $-\pi$  e  $+\pi$ .

```
atan2( y, x )
```

### **cosh**

Coseno iperbolico di **x**. Il risultato è un numero reale positivo.

```
cosh( x )
```

### **sinh**

Seno iperbolico di **x**. Il risultato è un numero reale.

```
sinh( x )
```

### **tanh**

Tangente iperbolica di **x**. Il risultato è un numero reale.

```
tanh( x )
```

### **acosh**

Coseno iperbolico inverso di **x**. Il risultato è un numero reale positivo.

```
acosh( x )
```

### **asinh**

Seno iperbolico inverso di **x**. Il risultato è un numero reale.

```
asinh( x )
```

### atanh

Tangente iperbolica inversa di **x**. Il risultato è un numero reale.

```
atanh( x )
```

### Esempi:

Il codice di script seguente carica un tabella campione, quindi carica una tabella contenente le operazioni trigonometriche e iperboliche calcolate sui valori.

```
SampleData:
LOAD * Inline
[Value
-1
0
1];

Results:
Load *,
cos(Value),
acos(Value),
sin(Value),
asin(Value),
tan(Value),
atan(Value),
atan2(Value, Value),
cosh(Value),
sinh(Value),
tanh(Value)
RESIDENT SampleData;

Drop Table SampleData;
```

## 8.28 Funzioni Window

Le funzioni Window eseguono calcoli utilizzando valori di più righe per produrre un valore per ciascuna riga separatamente. Le funzioni Window possono essere calcolate solo una volta letta l'intera tabella.

È possibile utilizzare le funzioni Window per eseguire operazioni come:

- Confrontare un numero individuale in una riga con quello medio, massimo o minimo all'interno di una colonna.
- Calcolare la classificazione di un valore individuale, all'interno della colonna o dell'intera tabella.

Le funzioni Window non cambiano il numero di record nella tabella ma possono comunque eseguire attività simili a funzioni di aggregazione, relazionali e di scala.

Ciascuna funzione viene descritta ulteriormente dopo la panoramica. È inoltre possibile fare clic sul nome della funzione nella sintassi per accedere immediatamente ai dettagli per tale funzione specifica.

### Window

La funzione **Window** esegue calcoli da più righe, producendo un valore per ciascuna riga separatamente.

```
Window (main_expr, [partition1, partition2, ...], [sort_type, [sort_expr],  
[filter_expr], [start_expr, end_expr]])
```

### WRank

La funzione **WRank** esegue calcoli di classificazione all'interno di **Window**.

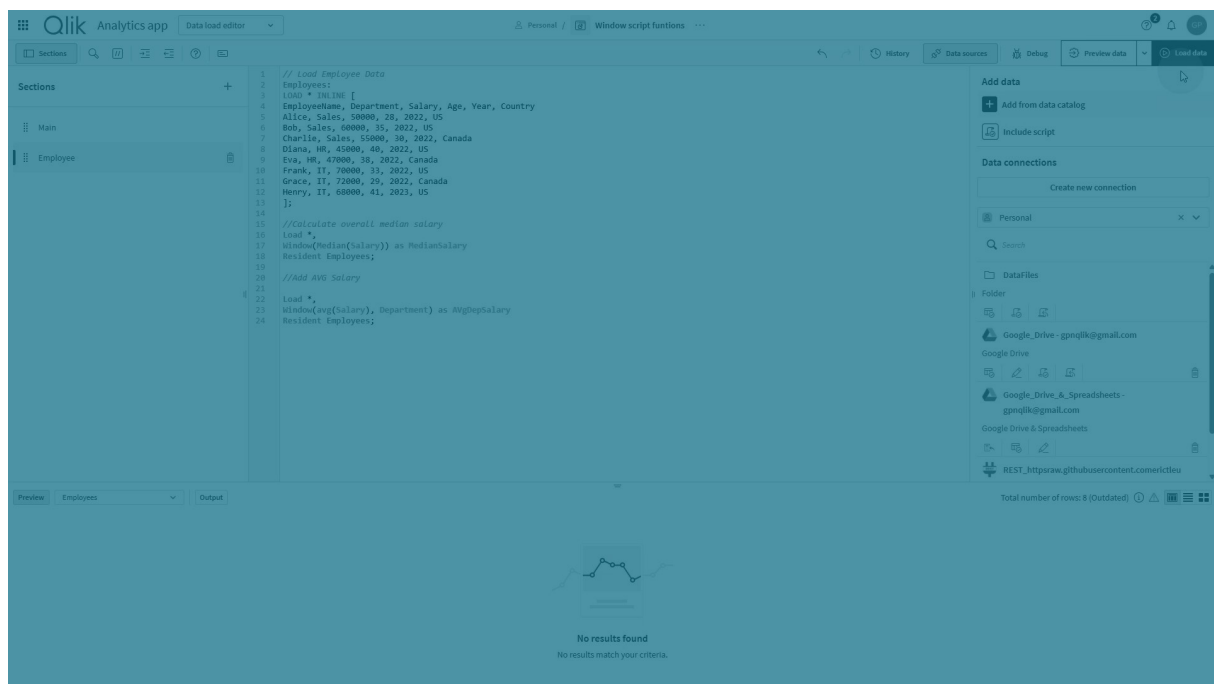
```
WRank ([mode[, fmt]])
```

## Window

**Window()** esegue calcoli da più righe di una sezione (o "finestra") di una tabella completa e produce un singolo valore per ogni riga.

È possibile utilizzare le funzioni **Window** per eseguire operazioni come:

- Confrontare il valore di un numero individuale in una riga con quello medio, massimo o minimo in una finestra della tabella.
- Calcolare la classificazione di un valore individuale, nella finestra o nell'intera tabella.



La funzione **Window** non cambia il numero di record nella tabella, ma può comunque eseguire attività simili a funzioni di aggregazione, relazionali e di scala.

La funzione **Window** deve essere inclusa nell'istruzione LOAD della tabella con cui si lavora, oppure fare riferimento alla tabella, come ad esempio con Resident LOAD. Per esempio, vedere [Esempio - Aggiunta di un campo contenente un'aggregazione \(page 1907\)](#).

Il nome può essere omissso, ma questo comporta che il nome del campo venga impostato sull'intera espressione della funzione **Window**. Ad esempio, *AvgDepSalary* nell'esempio seguente:

```
[AvgSalary]:  
Load  
    *,  
    window(avg(Salary), Department) as AvgDepSalary  
Resident Employees;
```



*Si consiglia di assegnare alla funzione **Window** un nome di campo.*

**Window** supporta funzioni generali, come l'arrotondamento o operazioni numeriche di base. Ad esempio:

```
Load *, Round(window(Sum(Salary),Department)) as SumDepSalary  
Load *, window(Sum(Salary),Department) + 5 as SumDepSalary
```



*La funzione **Window** esegue la scansione dell'intera tabella. In termini di consumo di memoria, il costo di questa scansione è proporzionale al numero di campi nella tabella. Una tabella con pochi campi, oltre a quelli su cui opera la funzione **Window**, è sempre da preferire a una tabella con molti campi.*

*In genere, una funzione **Window** consuma 2-4 volte la memoria e richiede un tempo 2-4 volte superiore per il ricaricamento rispetto a un'operazione di script simile contenente la stessa aggregazione, una clausola *where* (filtro), una clausola *group by* (partizione) e una clausola *order by* (ordinamento).*

### Sintassi:

```
Window (main_expr, [partition1, partition2, ...], [sort_type, [sort_expr],  
[filter_expr], [start_expr, end_expr]])
```

Tipo di dati restituiti: un valore per riga, che può essere combinato o utilizzato in un'altra espressione, oppure aggiunto direttamente alla tabella risultante creata dall'istruzione LOAD.

### Argomenti:

| Argomenti               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Argomento               | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| main_expr               | <p>L'espressione di input principale calcolata e restituita dalla funzione. Dev'essere un'espressione qualsiasi basata su un'aggregazione, come <code>Median(Salary)</code>. Ad esempio:</p> <pre>window(Median(Salary)) as MedianSalary</pre> <p>L'espressione può essere anche un nome di campo senza alcuna aggregazione applicata. In questo caso, <b>Window</b> lo tratta come se la funzione <b>Only()</b> fosse applicata a quel campo. Ad esempio:</p> <pre>window(Salary) as wSalary</pre> <p>Le espressioni che includono aggregazioni annidate o funzioni finestra annidate, come <code>window(Sum(Avg(Salary)))</code> o <code>Window(Sum(window(Salary)))</code>, non sono consentite.</p>                                                                                                                                                                                                                                                                                                               |
| partition 1, partition2 | <p>Dopo <code>main_expr</code>, è possibile definire un numero qualsiasi di partizioni. Le partizioni sono campi che definiscono con quali combinazioni si applicano le aggregazioni. Ad esempio:</p> <pre>window(Avg(Salary), Department) as AvgDepSalary</pre> <p>questa operazione calcola il salario medio per ogni reparto.</p> <p>Quando vengono forniti più parametri di partizione, l'aggregazione verrà applicata a ciascuna riga con una combinazione unica di campi di partizione. Ad esempio:</p> <pre>window(Avg(Salary), Department, Country) as AvgDepSalaryPerCountry</pre> <p>questa operazione calcola il salario medio per ogni reparto in ogni Paese. I reparti duplicati in Paesi diversi, come <i>Marketing in Svezia</i> e <i>Marketing in Canada</i>, saranno trattati come partizioni diverse.</p> <p>Il partizionamento è simile al raggruppamento ottenuto con la clausola <b>group by</b>. La partizione non riduce il numero di record in confronto alla tabella di input, tuttavia.</p> |

| Argomento                     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| sort_type,<br>[sort_<br>expr] | <p>Facoltativamente, specificare il tipo di ordinamento e l'espressione di ordinamento. Questa operazione definisce l'ordine in cui viene eseguita l'aggregazione. sort_type è una stringa costante e può avere uno dei tre valori:</p> <ul style="list-style-type: none"> <li>• ASC: ordinamento crescente.</li> <li>• DESC: ordinamento decrescente.</li> <li>• NONE</li> </ul> <p>Se si definisce il valore ASC o DESC per sort_type, è necessario definire anche un'espressione di ordinamento. Con sort_type NONE, non è necessario fornire alcuna espressione di ordinamento. L'espressione di ordinamento stabilisce l'ordine delle righe all'interno di una partizione.</p> <p>Ad esempio:</p> <pre>window(RecNo(), Department, 'ASC', Year)</pre> <p>Nell'esempio qui sopra, l'aggregazione all'interno della partizione viene eseguita in ordine crescente in base al campo <i>Anno</i>.</p> <p>I parametri di ordinamento influenzano anche l'ordine in cui una finestra scorrevole elabora una tabella. Per un esempio, vedere <a href="#">Esempio: aggiunta di un campo con una finestra scorrevole con ordinamento (page 1910)</a>.</p>               |
| filter_expr                   | <p>Facoltativamente, aggiungere un'espressione di filtro. Si tratta di un'espressione booleana che stabilisce se il record dovrebbe essere incluso nel calcolo o meno.</p> <p>Se viene fornito il valore filter_expr, è necessario includere un tipo di ordinamento. Se non è definito alcun tipo di ordinamento, l'argomento può essere interpretato come un parametro di partizione piuttosto che come un'espressione di ordinamento. Il parametro filter_expr può essere omissso completamente o è possibile includere un valore vero. In entrambi i casi non viene utilizzato alcun filtro, quindi non viene esclusa alcuna riga.</p> <p>Ad esempio:</p> <pre>window(avg(Salary), Department, 'ASC', Age, Country='US' Or Country='Canada') as AvgDepSalaryInUSOrCanada</pre> <p>L'istruzione <b>Window</b> di cui sopra filtrerà tutte le righe in cui il campo <i>Paese</i> non è <i>US</i> o <i>Canada</i>, escludendo queste righe dall'aggregazione. Continueranno a esserci valori di output per le righe che sono state filtrate, ma come tutte le altre righe di output, conterranno solo i risultati aggregati delle righe non escluse dal filtro.</p> |

| Argomento                     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| start_<br>expr,end_<br>d_expr | <p>Facoltativamente, impostare gli argomenti per la funzionalità della finestra scorrevole. Questa operazione limita ulteriormente il numero di righe da aggregare insieme. Ogni riga di output includerà solo i risultati (aggregati) di un certo numero di righe precedenti e successive alla riga corrente. Una finestra scorrevole richiede due argomenti (start_expr,end_expr). Se questi argomenti vengono forniti, il tipo di ordinamento non deve essere ignorato. Altrimenti, gli argomenti possono essere interpretati come un parametro di partizione anziché una finestra scorrevole. Il parametro start_expr,end_expr comprende i seguenti elementi:</p> <ul style="list-style-type: none"> <li>• Espressione inizio: il numero di righe prima della riga corrente da includere nella finestra.</li> <li>• Espressione fine: il numero di righe dopo la riga corrente da includere nella finestra.</li> </ul> <p>Ad esempio, se si vuole includere le 3 righe precedenti, la riga corrente e la riga successiva:</p> <pre>window(concat(Text(Salary),'-'), Department, 'NONE', 1, -3, 1) as wSlidingSalaryDepartment</pre> <p>Per indicare tutte le righe precedenti o tutte le righe successive, è possibile usare la funzione <b>Unbounded()</b>. Ad esempio, per includere tutte le righe precedenti, la riga corrente e la riga successiva:</p> <pre>window(concat(Text(Salary),'-'), Department, 'NONE', 1, UNBOUNDED(), 1) as wSlidingSalaryDepartment</pre> <p>Ad esempio, per includere la terza riga da quella corrente e tutte le righe successive:</p> <pre>window(concat(Text(Salary),'-'), Department, 'NONE', 1, 3, UNBOUNDED()) as wSlidingSalaryDepartment</pre> <p>Negli esempi precedenti, il tipo di ordinamento NONE è fornito in modo che lo script possa distinguere tra gli argomenti della partizione e i parametri che vengono dopo il tipo di ordinamento.</p> |

## Esempio - Aggiunta di un campo contenente un'aggregazione

Esempio: aggiunta di un campo contenente un'aggregazione

### Script di caricamento

Creare una nuova scheda nell'editor caricamento dati, quindi caricare i seguenti dati come caricamento inline. Creare la tabella in basso in Qlik Sense per visualizzare i risultati.

```
[Employees]:
Load
    *,
    window(avg(Salary), Department) as AvgDepSalary
Inline [
    Name, Department, Salary, Age, Year, Country
    David, Sales, 11, 32, 2019, Canada
```

```
Sara, Sales, 22, 25, 2022, Sweden
Nils, Marketing, 33, 18, 2022, Sweden
Patrick, Marketing, 44, 43, 2022, Canada
Anna, Sales, 55, 33, 2023, Canada
Johan, Sales, 66, 26, 2025, Sweden
Olle, Marketing, 77, 19, 2023, Sweden
Bonnie, Marketing, 88, 44, 2023, US
];
```

### Risultati

Risultati per l'aggiunta di un campo contenente un'aggregazione

| Name    | Department | Salary | Age | Year | Country | AvgDepSalary |
|---------|------------|--------|-----|------|---------|--------------|
| David   | Sales      | 11     | 32  | 2019 | Canada  | 39           |
| Sara    | Sales      | 22     | 25  | 2022 | Sweden  | 39           |
| Nils    | Marketing  | 33     | 18  | 2022 | Sweden  | 61           |
| Patrick | Marketing  | 44     | 43  | 2022 | Canada  | 61           |
| Anna    | Sales      | 55     | 33  | 2023 | Canada  | 39           |
| Johan   | Sales      | 66     | 26  | 2025 | Sweden  | 39           |
| Olle    | Marketing  | 77     | 19  | 2023 | Sweden  | 61           |
| Bonnie  | Marketing  | 88     | 44  | 2023 | US      | 61           |

### Esempio - Aggiunta di un campo contenente un'aggregazione con un filtro per valori specifici

Esempio: aggiunta di un campo contenente un'aggregazione con un filtro per valori specifici

#### Script di caricamento

Creare una nuova scheda nell'editor caricamento dati, quindi caricare i seguenti dati come caricamento inline. Creare la tabella in basso in Qlik Sense per visualizzare i risultati.

```
[Employees]:
Load
*,
window(avg(Salary), Department, 'NONE', Country = 'Sweden') as AvgDepSalarySweden
Inline [
    Name, Department, Salary, Age, Year, Country
    David, Sales, 11, 32, 2019, Canada
    Sara, Sales, 22, 25, 2022, Sweden
    Nils, Marketing, 33, 18, 2022, Sweden
    Patrick, Marketing, 44, 43, 2022, Canada
    Anna, Sales, 55, 33, 2023, Canada
    Johan, Sales, 66, 26, 2025, Sweden
    Olle, Marketing, 77, 19, 2023, Sweden
```

```
Bonnie, Marketing, 88, 44, 2023, US  
];
```

### Risultati

Risultati per l'aggiunta di un campo contenente un'aggregazione con un filtro per valori specifici

| Name    | Department | Salary | Age | Year | Country | AvgDepSalarySweden |
|---------|------------|--------|-----|------|---------|--------------------|
| David   | Sales      | 11     | 32  | 2019 | Canada  | 44                 |
| Sara    | Sales      | 22     | 25  | 2022 | Sweden  | 44                 |
| Nils    | Marketing  | 33     | 18  | 2022 | Sweden  | 55                 |
| Patrick | Marketing  | 44     | 43  | 2022 | Canada  | 55                 |
| Anna    | Sales      | 55     | 33  | 2023 | Canada  | 44                 |
| Johan   | Sales      | 66     | 26  | 2025 | Sweden  | 44                 |
| Olle    | Marketing  | 77     | 19  | 2023 | Sweden  | 55                 |
| Bonnie  | Marketing  | 88     | 44  | 2023 | US      | 55                 |

### Esempio - Aggiunta di un campo con una finestra scorrevole

Esempio: aggiunta di un campo con una finestra scorrevole

#### Script di caricamento

Creare una nuova scheda nell'editor caricamento dati, quindi caricare i seguenti dati come caricamento inline. Creare la tabella in basso in Qlik Sense per visualizzare i risultati.

```
[Employees]:  
Load  
    *,  
    window(concat(Text(Salary),'-'), Department, 'NONE', -1, 1) as SlidingWindow  
Inline [  
    Name, Department, Salary, Age, Year, Country  
    David, Sales, 11, 32, 2019, Canada  
    Sara, Sales, 22, 25, 2022, Sweden  
    Nils, Marketing, 33, 18, 2022, Sweden  
    Patrick, Marketing, 44, 43, 2022, Canada  
    Anna, Sales, 55, 33, 2023, Canada  
    Johan, Sales, 66, 26, 2025, Sweden  
    Olle, Marketing, 77, 19, 2023, Sweden  
    Bonnie, Marketing, 88, 44, 2023, US  
];
```

### Risultati

Risultati per l'aggiunta di un campo contenente un'aggregazione con un filtro per valori specifici

| Name    | Department | Salary | Age | Year | Country | SlidingWindow |
|---------|------------|--------|-----|------|---------|---------------|
| David   | Sales      | 11     | 32  | 2019 | Canada  | 11-22         |
| Sara    | Sales      | 22     | 25  | 2022 | Sweden  | 11-22-55      |
| Nils    | Marketing  | 33     | 18  | 2022 | Sweden  | 33-44         |
| Patrick | Marketing  | 44     | 43  | 2022 | Canada  | 33-44-77      |
| Anna    | Sales      | 55     | 33  | 2023 | Canada  | 22-55-66      |
| Johan   | Sales      | 66     | 26  | 2025 | Sweden  | 55-66         |
| Olle    | Marketing  | 77     | 19  | 2023 | Sweden  | 44-77-88      |
| Bonnie  | Marketing  | 88     | 44  | 2023 | US      | 77-88         |

### Esempio: aggiunta di un campo con una finestra scorrevole con ordinamento

Esempio: aggiunta di un campo con una finestra scorrevole

#### Script di caricamento

Creare una nuova scheda nell'editor caricamento dati, quindi caricare i seguenti dati come caricamento inline. Creare la tabella in basso in Qlik Sense per visualizzare i risultati.

[Employees]:

Load

```
*,
    window(concat(Text(Salary),'-'), Department, 'ASC', Age, -1, 1) as SortedSlidingWindow
Inline [
    Name, Department, Salary, Age, Year, Country
    David, Sales, 11, 32, 2019, Canada
    Sara, Sales, 22, 25, 2022, Sweden
    Nils, Marketing, 33, 18, 2022, Sweden
    Patrick, Marketing, 44, 43, 2022, Canada
    Anna, Sales, 55, 33, 2023, Canada
    Johan, Sales, 66, 26, 2025, Sweden
    Olle, Marketing, 77, 19, 2023, Sweden
    Bonnie, Marketing, 88, 44, 2023, US
];
```

### Risultati

Risultati per l'aggiunta di un campo contenente un'aggregazione con un filtro per valori specifici

| Name  | Department | Salary | Age | Year | Country | SortedSlidingWindow |
|-------|------------|--------|-----|------|---------|---------------------|
| David | Sales      | 11     | 32  | 2019 | Canada  | 11-55-66            |

## 8 Funzioni per script e grafici

| Name    | Department | Salary | Age | Year | Country | SortedSlidingWindow |
|---------|------------|--------|-----|------|---------|---------------------|
| Sara    | Sales      | 22     | 25  | 2022 | Sweden  | 22-66               |
| Nils    | Marketing  | 33     | 18  | 2022 | Sweden  | 33-77-88            |
| Patrick | Marketing  | 44     | 43  | 2022 | Canada  | 44-77               |
| Anna    | Sales      | 55     | 33  | 2023 | Canada  | 11-55               |
| Johan   | Sales      | 66     | 26  | 2025 | Sweden  | 11-22-66            |
| Olle    | Marketing  | 77     | 19  | 2023 | Sweden  | 33-44-77            |
| Bonnie  | Marketing  | 88     | 44  | 2023 | US      | 33-88               |

Per capire meglio come le partizioni e l'ordinamento influenzano le aggregazioni in una finestra scorrevole, ordinare la tabella per nome in ordine ascendente.

Risultati per l'aggiunta di un campo contenente un'aggregazione con un filtro per valori specifici

| Name    | Department | Salary | Age | Year | Country | SortedSlidingWindow |
|---------|------------|--------|-----|------|---------|---------------------|
| Anna    | Sales      | 55     | 33  | 2023 | Canada  | 11-55               |
| Bonnie  | Marketing  | 88     | 44  | 2023 | US      | 33-88               |
| David   | Sales      | 11     | 32  | 2019 | Canada  | 11-55-66            |
| Johan   | Sales      | 66     | 26  | 2025 | Sweden  | 11-22-66            |
| Nils    | Marketing  | 33     | 18  | 2022 | Sweden  | 33-77-88            |
| Olle    | Marketing  | 77     | 19  | 2023 | Sweden  | 33-44-77            |
| Patrick | Marketing  | 44     | 43  | 2022 | Canada  | 44-77               |
| Sara    | Sales      | 22     | 25  | 2022 | Sweden  | 22-66               |

Nella riga di *Johan*, l'aggregazione ha incluso i anche i valori dei salari per *David* e *Sara*. Queste sono le righe precedenti e seguenti nella partizione del reparto *Vendite*, ordinate per età.

Per un altro esempio che mostra quali righe sono incluse, cambiare l'espressione **Windows** da `window(concat(Text(Salary), '-'), Department, 'ASC', Age, -1, 1)` a `window(concat(Text(Name), '-'), Department, 'ASC', Age, -1, 1)` e ricaricare. I valori in `SortedSlidingWindow` nella tabella sottostante indicano il nome delle righe precedenti e seguenti nella partizione *Vendite*.

Risultati per l'aggiunta di un campo contenente un'aggregazione con un filtro per valori specifici

| Name   | Department | Salary | Age | Year | Country | SortedSlidingWindow |
|--------|------------|--------|-----|------|---------|---------------------|
| Anna   | Sales      | 55     | 33  | 2023 | Canada  | Anna-David          |
| Bonnie | Marketing  | 88     | 44  | 2023 | US      | Bonnie-Nils         |
| David  | Sales      | 11     | 32  | 2019 | Canada  | Anna-David-Johan    |
| Johan  | Sales      | 66     | 26  | 2025 | Sweden  | David-Johan-Sara    |

| Name    | Department | Salary | Age | Year | Country | SortedSlidingWindow |
|---------|------------|--------|-----|------|---------|---------------------|
| Nils    | Marketing  | 33     | 18  | 2022 | Sweden  | Bonnie-Nils-Olle    |
| Olle    | Marketing  | 77     | 19  | 2023 | Sweden  | Nils-Olle-Patrick   |
| Patrick | Marketing  | 44     | 43  | 2022 | Canada  | Olle-Patrick        |
| Sara    | Sales      | 22     | 25  | 2022 | Sweden  | Johan-Sara          |

## Limitazioni

**Window** presenta le seguenti limitazioni:

- **Window** è una funzione che ha un forte impatto sulle risorse, soprattutto in termini di consumo di memoria.
- **Window** non è supportato in Qlik Sense Mobile.
- Le espressioni del grafico non supportano **Window**.
- Non è possibile nidificare funzioni **Window** in altre funzioni **Window**.
- **Window** non può essere utilizzata in una funzione di aggregazione.
- Non è possibile utilizzare **Window** in un'istruzione LOAD con una clausola "group by".
- **Window** deve essere in grado di scansionare l'intera tabella.
- **WRank()**, **RecNo()** e **RowNo()** non possono essere utilizzate con **Window** quando si utilizza la finestra scorrevole.
- Quando si utilizza la funzionalità della finestra scorrevole, il tempo di calcolo aumenta con la dimensione della finestra. Si consiglia di evitare di usare tabelle di grandi dimensioni combinate con finestre di grandi dimensioni.

## WRank

**WRank()** valuta le righe di una tabella nello script di caricamento, e per ciascuna riga, visualizza la posizione relativa del valore del campo valutato nello script di caricamento. Quando si valuta la tabella, la funzione confronta il risultato con il risultato di altre righe contenenti la partizione corrente e restituisce la classificazione per la riga corrente nel segmento.

*Partizioni in una tabella*

|                   | Region   | Country                  | Population  | Rank(Population) |
|-------------------|----------|--------------------------|-------------|------------------|
| Column segment #1 | Americas | Mexico                   | 128,932,753 | 2                |
|                   | Americas | Canada                   | 37,742,154  | 3                |
|                   | Americas | United States of America | 331,002,651 | 1                |
| Column segment #2 | Europe   | Sweden                   | 10,099,365  | 4                |
|                   | Europe   | United Kingdom           | 67,886,011  | 2                |
|                   | Europe   | France                   | 65,273,511  | 3                |
|                   | Europe   | Germany                  | 83,783,942  | 1                |

**WRank** può essere utilizzata solo in una funzione **Window**. La funzione **Window** deve includere un tipo di ordinamento e un'espressione di ordinamento. La classificazione viene applicata all'espressione di ordinamento.

**Sintassi:**

```
WRank ([mode[, fmt]])
```

**Tipo di dati restituiti:** duale

**Argomenti:**

Argomenti

| Argomento | Descrizione                                                                            |
|-----------|----------------------------------------------------------------------------------------|
| mode      | Facoltativamente, specifica la rappresentazione numerica del risultato della funzione. |
| fmt       | Facoltativamente, specifica la rappresentazione testuale del risultato della funzione. |

La classificazione viene restituita come valore duale che, nel caso in cui ogni riga presenti una classificazione univoca, sarà un numero intero compreso tra 1 e il numero di righe nella partizione attuale.

Nel caso in cui molte righe condividano la stessa classificazione, la rappresentazione testuale e numerica potrà essere controllata mediante i parametri **mode** e **fmt**.

### **mode**

Il primo argomento, **mode**, può assumere i seguenti valori:

Valori **mode**

| Valore          | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0 (predefinito) | <p>Se tutte le classificazioni all'interno del gruppo condiviso rientrano nella parte bassa del valore medio dell'intera classificazione, tutte le righe all'interno di tale gruppo riceveranno la classificazione più bassa.</p> <p>Se tutte le classificazioni all'interno del gruppo condiviso rientrano nella parte alta del valore medio dell'intera classificazione, tutte le righe all'interno di tale gruppo riceveranno la classificazione più alta.</p> <p>Se tutte le classificazioni all'interno del gruppo condiviso si distribuiscono lungo il valore medio dell'intera classificazione, tutte le righe all'interno di tale gruppo riceveranno il valore corrispondente alla media calcolata in base alla classificazione massima e minima dell'intera partizione.</p> |
| 1               | Classificazione minima su tutte le righe.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 2               | Classificazione media su tutte le righe.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 3               | Classificazione massima su tutte le righe.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 4               | Classificazione minima sulla prima riga, quindi incrementata di uno per ogni riga.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

### **fmt**

Il secondo argomento, **fmt**, può assumere i seguenti valori:

Valori **fmt**

| Valore          | Descrizione                                                       |
|-----------------|-------------------------------------------------------------------|
| 0 (predefinito) | Valore basso - valore alto su tutte le righe (ad esempio, 3 - 4). |
| 1               | Valore basso su tutte le righe.                                   |
| 2               | Valore basso sulla prima riga, vuoto sulle righe successive.      |

L'ordine delle righe per **mode 4** e **fmt 2** è determinato dall'ordine di caricamento dei campi della tabella.

### Esempio - Aggiunta di un campo classificato

Esempio: aggiunta di un campo classificato

#### Script di caricamento

Creare una nuova scheda nell'editor caricamento dati, quindi caricare i seguenti dati come caricamento inline. Creare la tabella in basso in Qlik Sense per visualizzare i risultati.

Transactions:

Load

\*,

Window(WRank(0),customer\_id, 'Desc', transaction\_amount) as TransactionRanking;

Load \* Inline [

transaction\_id, transaction\_date, transaction\_amount, transaction\_quantity, customer\_id, size, color\_code

3750, 20180830, 23.56, 2, 2038593, L, Red

3751, 20180907, 556.31, 6, 203521, M, Orange

3752, 20180916, 5.75, 1, 5646471, S, Blue

3753, 20180922, 125.00, 7, 3036491, L, Black

3754, 20180922, 484.21, 13, 049681, XS, Red

3756, 20180922, 59.18, 2, 2038593, M, Blue

3757, 20180923, 177.42, 21, 203521, XL, Black

3758, 20180924, 153.42, 14, 2038593, L, Red

3759, 20180925, 7.42, 5, 203521, M, Orange

3760, 20180925, 80.12, 18, 5646471, M, Blue

3761, 20180926, 3.42, 7, 3036491, XS, Black

3763, 20180926, 63.55, 12, 049681, S, Red

3763, 20180927, 177.56, 10, 2038593, L, Blue

3764, 20180927, 325.95, 8, 203521, XL, Black

];

## Risultati

Risultati per l'aggiunta di un campo classificato

| transactio<br>n_id | transactio<br>n_date | transactio<br>n_amount | transactio<br>n_quantity | custome<br>r_id | size | color_<br>code | transazioneRan<br>king |
|--------------------|----------------------|------------------------|--------------------------|-----------------|------|----------------|------------------------|
| 3750               | 20180830             | 23.56                  | 2                        | 2038593         | L    | Rosso          | 4-4                    |
| 3751               | 20180907             | 556.31                 | 6                        | 203521          | M    | Arancio<br>ne  | 1-1                    |
| 3752               | 20180916             | 5.75                   | 1                        | 5646471         | S    | Blu            | 2-2                    |
| 3754               | 20180922             | 484.21                 | 13                       | 049681          | XS   | Rosso          | 1-1                    |
| 3756               | 20180922             | 59.18                  | 2                        | 2038593         | M    | Blu            | 3-3                    |
| 3753               | 20180922             | 125.00                 | 7                        | 3036491         | L    | Nero           | 1-1                    |
| 3757               | 20180923             | 177.42                 | 21                       | 203521          | XL   | Nero           | 3-3                    |
| 3758               | 20180924             | 153.42                 | 14                       | 2038593         | L    | Rosso          | 2-2                    |
| 3759               | 20180925             | 7.42                   | 5                        | 203521          | M    | Arancio<br>ne  | 4-4                    |
| 3760               | 20180925             | 80.12                  | 18                       | 5646471         | M    | Blu            | 1-1                    |
| 3763               | 20180926             | 63.55                  | 12                       | 049681          | S    | Rosso          | 2-2                    |
| 3761               | 20180926             | 3.42                   | 7                        | 3036491         | XS   | Nero           | 2-2                    |
| 3764               | 20180927             | 325.95                 | 8                        | 203521          | XL   | Nero           | 2-2                    |
| 3763               | 20180927             | 177.56                 | 10                       | 2038593         | L    | Blu            | 1-1                    |

## Esempio - Aggiunta di un campo classificato utilizzando fmt per un risultato a cifra singola

Esempio: aggiunta di un campo classificato utilizzando fmt per un risultato a cifra singola

## Script di caricamento

Creare una nuova scheda nell'editor caricamento dati, quindi caricare i seguenti dati come caricamento inline. Creare la tabella in basso in Qlik Sense per visualizzare i risultati.

Transactions:

Load

```
*,window(WRank(0,1),customer_id, 'Desc', transaction_amount) as TransactionRanking;
```

Load \* Inline [

```
transaction_id, transaction_date, transaction_amount, transaction_quantity, customer_id, size, color_code
```

```
3750, 20180830, 23.56, 2, 2038593, L, Red
```

```
3751, 20180907, 556.31, 6, 203521, M, Orange
```

```
3752, 20180916, 5.75, 1, 5646471, S, Blue
3753, 20180922, 125.00, 7, 3036491, L, Black
3754, 20180922, 484.21, 13, 049681, XS, Red
3756, 20180922, 59.18, 2, 2038593, M, Blue
3757, 20180923, 177.42, 21, 203521, XL, Black
3758, 20180924, 153.42, 14, 2038593, L, Red
3759, 20180925, 7.42, 5, 203521, M, Orange
3760, 20180925, 80.12, 18, 5646471, M, Blue
3761, 20180926, 3.42, 7, 3036491, XS, Black
3763, 20180926, 63.55, 12, 049681, S, Red
3763, 20180927, 177.56, 10, 2038593, L, Blue
3764, 20180927, 325.95, 8, 203521, XL, Black
];
```

### Risultati

Risultati per l'aggiunta di un campo classificato utilizzando fmt per un risultato a cifra singola

| transactio<br>n_id | transactio<br>n_date | transactio<br>n_amount | transactio<br>n_quantity | custome<br>r_id | size | color_<br>code | transazioneRan<br>king |
|--------------------|----------------------|------------------------|--------------------------|-----------------|------|----------------|------------------------|
| 3750               | 20180830             | 23.56                  | 2                        | 2038593         | L    | Rosso          | 4                      |
| 3751               | 20180907             | 556.31                 | 6                        | 203521          | M    | Arancio<br>ne  | 1                      |
| 3752               | 20180916             | 5.75                   | 1                        | 5646471         | S    | Blu            | 2                      |
| 3754               | 20180922             | 484.21                 | 13                       | 049681          | XS   | Rosso          | 1                      |
| 3756               | 20180922             | 59.18                  | 2                        | 2038593         | M    | Blu            | 3                      |
| 3753               | 20180922             | 125.00                 | 7                        | 3036491         | L    | Nero           | 1                      |
| 3757               | 20180923             | 177.42                 | 21                       | 203521          | XL   | Nero           | 3                      |
| 3758               | 20180924             | 153.42                 | 14                       | 2038593         | L    | Rosso          | 2                      |
| 3759               | 20180925             | 7.42                   | 5                        | 203521          | M    | Arancio<br>ne  | 4                      |
| 3760               | 20180925             | 80.12                  | 18                       | 5646471         | M    | Blu            | 1                      |
| 3763               | 20180926             | 63.55                  | 12                       | 049681          | S    | Rosso          | 2                      |
| 3761               | 20180926             | 3.42                   | 7                        | 3036491         | XS   | Nero           | 2                      |
| 3764               | 20180927             | 325.95                 | 8                        | 203521          | XL   | Nero           | 2                      |
| 3763               | 20180927             | 177.56                 | 10                       | 2038593         | L    | Blu            | 1                      |

### Esempio: aggiunta di un campo classificato con più partizioni

Esempio: aggiunta di un campo classificato con più partizioni

#### Script di caricamento

Creare una nuova scheda nell'editor caricamento dati, quindi caricare i seguenti dati come caricamento inline. Creare la tabella in basso in Qlik Sense per visualizzare i risultati.

Transactions:

Load

```
*,window(wRank(0,1),customer_id, size, color_code, 'Desc', transaction_amount) as
```

TransactionRanking;

Load \* Inline [

```
transaction_id, transaction_date, transaction_amount, transaction_quantity, customer_id, size,
color_code
```

```
3750, 20180830, 23.56, 2, 2038593, L, Red
```

```
3751, 20180907, 556.31, 6, 203521, M, Orange
```

```
3752, 20180916, 5.75, 1, 5646471, S, Blue
```

```
3753, 20180922, 125.00, 7, 3036491, L, Black
```

```
3754, 20180922, 484.21, 13, 049681, XS, Red
```

```
3756, 20180922, 59.18, 2, 2038593, M, Blue
```

```
3757, 20180923, 177.42, 21, 203521, XL, Black
```

```
3758, 20180924, 153.42, 14, 2038593, L, Red
```

```
3759, 20180925, 7.42, 5, 203521, M, Orange
```

```
3760, 20180925, 80.12, 18, 5646471, M, Blue
```

```
3761, 20180926, 3.42, 7, 3036491, XS, Black
```

```
3763, 20180926, 63.55, 12, 049681, S, Red
```

```
3763, 20180927, 177.56, 10, 2038593, L, Blue
```

```
3764, 20180927, 325.95, 8, 203521, XL, Black
```

```
];
```

#### Risultati

Risultati per l'aggiunta di un campo classificato utilizzando fmt per un risultato a cifra singola

| transactio<br>n_id | transactio<br>n_date | transactio<br>n_amount | transactio<br>n_quantity | custome<br>r_id | size | color_<br>code | transazioneRan<br>king |
|--------------------|----------------------|------------------------|--------------------------|-----------------|------|----------------|------------------------|
| 3750               | 20180830             | 23.56                  | 2                        | 2038593         | L    | Rosso          | 2                      |
| 3751               | 20180907             | 556.31                 | 6                        | 203521          | M    | Arancio<br>ne  | 1                      |
| 3752               | 20180916             | 5.75                   | 1                        | 5646471         | S    | Blu            | 1                      |
| 3754               | 20180922             | 484.21                 | 13                       | 049681          | XS   | Rosso          | 1                      |
| 3756               | 20180922             | 59.18                  | 2                        | 2038593         | M    | Blu            | 1                      |
| 3753               | 20180922             | 125.00                 | 7                        | 3036491         | L    | Nero           | 1                      |
| 3757               | 20180923             | 177.42                 | 21                       | 203521          | XL   | Nero           | 2                      |

| transactio<br>n_id | transactio<br>n_date | transactio<br>n_amount | transactio<br>n_quantity | custome<br>r_id | size | color_<br>code | transazioneRan<br>king |
|--------------------|----------------------|------------------------|--------------------------|-----------------|------|----------------|------------------------|
| 3758               | 20180924             | 153.42                 | 14                       | 2038593         | L    | Rosso          | 1                      |
| 3759               | 20180925             | 7.42                   | 5                        | 203521          | M    | Arancio<br>ne  | 2                      |
| 3760               | 20180925             | 80.12                  | 18                       | 5646471         | M    | Blu            | 1                      |
| 3763               | 20180926             | 63.55                  | 12                       | 049681          | S    | Rosso          | 1                      |
| 3761               | 20180926             | 3.42                   | 7                        | 3036491         | XS   | Nero           | 1                      |
| 3764               | 20180927             | 325.95                 | 8                        | 203521          | XL   | Nero           | 1                      |
| 3763               | 20180927             | 177.56                 | 10                       | 2038593         | L    | Blu            | 1                      |

### Limitazioni

WRank presenta le seguenti limitazioni:

- Se il proprio valore fmt è 0 e si vuole utilizzare la parte testuale del doppio risultato per **WRank**, è necessario utilizzare **Text()** e **Window(WRank)**. Ad esempio: `Text(Window(WRank(0), Unit, 'DESC', Age)) as UnitWRankedByText`.

## 9 Restrizione dell'accesso al file system

Per motivi di sicurezza, in modalità standard Qlik Sense non supporta i percorsi nello script di caricamento dei dati o funzioni e variabili che espongono il file system.

Tuttavia, poiché il file system risulta supportato in QlikView, è possibile disabilitare la modalità standard e utilizzare la modalità legacy per poter riutilizzare gli script di caricamento di QlikView.



*La disabilitazione della modalità standard può determinare rischi per la sicurezza, in quanto viene esposto il file system.*

[Disabilitazione della modalità standard \(page 1926\)](#)

### 9.1 Aspetti relativi alla sicurezza quando si effettua la connessione alle connessioni dati ODBC e OLE DB basate su file

Le connessioni dati ODBC e OLE DB che utilizzano driver basati su file visualizzeranno il percorso del file di dati connesso nella stringa di connessione. Il percorso può essere visualizzato quando si modifica la connessione, nella finestra di dialogo di selezione dei dati o in alcune query SQL. Questo può avvenire sia nella modalità standard che nella modalità legacy.



*Se l'esposizione del percorso al file dati è un problema, si consiglia di connettersi al file dati mediante una connessione dati cartella, se possibile.*

### 9.2 Limitazioni nella modalità standard

Nella modalità standard molte istruzioni, variabili e funzioni non possono essere utilizzate oppure sono sottoposte a limitazioni. L'utilizzo di istruzioni non supportate nello script di caricamento dei dati genera errori quando lo script viene eseguito. I messaggi di errore vengono memorizzati nel file di registro dello script. L'utilizzo di variabili e funzioni non supportate non genera messaggi di errore o l'inserimento di voci nel file di registro, tuttavia, la funzione restituisce NULL.

Durante la modifica dello script di caricamento dei dati, non esiste alcuna indicazione del fatto che una variabile, un'istruzione o una funzione non è supportata.

### Variabili di sistema

Variabili di sistema

| Variabile  | Modalità standard | Modalità legacy | Definizione                                                                                                                               |
|------------|-------------------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| Floppy     | Non supportato    | Supportato      | Restituisce la lettera relativa alla prima unità floppy rilevata, in genere <i>a:</i> .                                                   |
| CD         | Non supportato    | Supportato      | Restituisce la lettera relativa alla prima unità CD-ROM rilevata. Se non viene rilevata alcuna unità CD-ROM, viene restituito <i>c:</i> . |
| QvPath     | Non supportato    | Supportato      | Restituisce la stringa costituita dal percorso del file eseguibile di Qlik Sense.                                                         |
| QvRoot     | Non supportato    | Supportato      | Restituisce la directory principale del file eseguibile di Qlik Sense.                                                                    |
| QvWorkPath | Non supportato    | Supportato      | Restituisce la stringa costituita dal percorso dell'app Qlik Sense attuale.                                                               |
| QvWorkRoot | Non supportato    | Supportato      | Restituisce la directory principale dell'app Qlik Sense attuale.                                                                          |
| WinPath    | Non supportato    | Supportato      | Restituisce la stringa costituita dal percorso di Windows.                                                                                |
| WinRoot    | Non supportato    | Supportato      | Restituisce la directory principale di Windows.                                                                                           |

## 9 Restrizione dell'accesso al file system

| Variabile       | Modalità standard                                                       | Modalità legacy                                                                          | Definizione                                                                                                                                                                                                                                                                                                                                             |
|-----------------|-------------------------------------------------------------------------|------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| \$(include=...) | Input supportato:<br>Percorso che utilizza la connessione alla libreria | Input supportato:<br>Percorso che utilizza la connessione alla libreria o al file system | La <b>Include/Must_Include</b> variabile specifica un file che contiene testo che deve essere incluso nello script e valutato come codice di script. Non è utilizzato per aggiungere dati. È possibile memorizzare parti del codice di script in un file di testo separato e riutilizzarlo in diverse app. Questa è una variabile definita dall'utente. |

### Istruzioni di script regolari

#### Istruzioni di script regolari

| Istruzione | Modalità standard                                                       | Modalità legacy                                                                          | Definizione                                                                                                                                                                                                                        |
|------------|-------------------------------------------------------------------------|------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Binary     | Input supportato:<br>Percorso che utilizza la connessione alla libreria | Input supportato:<br>Percorso che utilizza la connessione alla libreria o al file system | L'istruzione <b>binary</b> viene utilizzata per caricare i dati da un'altra app.                                                                                                                                                   |
| Connect    | Input supportato:<br>Percorso che utilizza la connessione alla libreria | Input supportato:<br>Percorso che utilizza la connessione alla libreria o al file system | L'istruzione <b>CONNECT</b> consente di definire l'accesso di Qlik Sense a un database generico mediante l'interfaccia OLE DB/ODBC. Per ODBC, occorre innanzitutto specificare la sorgente dati utilizzando l'amministratore ODBC. |

## 9 Restrizione dell'accesso al file system

| Istruzione     | Modalità standard                                                       | Modalità legacy                                                                          | Definizione                                                                                                                                                                                                                                 |
|----------------|-------------------------------------------------------------------------|------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Directory      | Input supportato:<br>Percorso che utilizza la connessione alla libreria | Input supportato:<br>Percorso che utilizza la connessione alla libreria o al file system | L'istruzione <b>Directory</b> definisce in quale directory ricercare i file dei dati nelle istruzioni <b>LOAD</b> successive finché non viene eseguita una nuova istruzione <b>Directory</b> .                                              |
| Execute        | Non supportato                                                          | Input supportato:<br>Percorso che utilizza la connessione alla libreria o al file system | L'istruzione <b>Execute</b> viene utilizzata per eseguire altri programmi, mentre Qlik Sense sta caricando i dati. Ad esempio, per effettuare le connessioni necessarie.                                                                    |
| LOAD from ...  | Input supportato:<br>Percorso che utilizza la connessione alla libreria | Input supportato:<br>Percorso che utilizza la connessione alla libreria o al file system | L'istruzione <b>LOAD</b> carica i campi da un file, dai dati definiti nello script, da una tabella caricata in precedenza, da una pagina Web, dal risultato di un'istruzione <b>SELECT</b> seguente o dalla generazione automatica di dati. |
| Store into ... | Input supportato:<br>Percorso che utilizza la connessione alla libreria | Input supportato:<br>Percorso che utilizza la connessione alla libreria o al file system | L'istruzione <b>Store</b> crea un file QVD, Parquet, CSV, o TXT.                                                                                                                                                                            |

### Istruzioni di controllo dello script

Istruzioni di controllo dello script

| Istruzione                                | Modalità standard                                                                                                              | Modalità legacy                                                                                                                                                                              | Definizione                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| For each...<br>filelist mask/dirlist mask | Input supportato:<br>Percorso che utilizza la connessione alla libreria<br><br>Output restituito:<br>connessione alla libreria | Input supportato:<br>Percorso che utilizza la connessione alla libreria o al file system<br><br>Output restituito:<br>Connessione alla libreria o percorso al file system, in base all'input | La sintassi filelist mask restituisce un elenco con valori separati da virgole di tutti i file presenti nella directory attuale che presentano una corrispondenza con <b>filelist mask</b> . La sintassi <b>dirlist mask</b> restituisce un elenco con valori separati da virgole di tutte le directory incluse nella directory attuale che presentano una corrispondenza con la maschera del nome della directory. |

### Funzioni file

Funzioni file

| Funzione           | Modalità standard                                                       | Modalità legacy                                                                              | Definizione                                                                                                                     |
|--------------------|-------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| Attribute()        | Input supportato:<br>Percorso che utilizza la connessione alla libreria | Input supportato:<br>Percorso che utilizza la connessione alla libreria o al file system     | Restituisce il valore dei metatag di file multimediali differenti come testo.                                                   |
| ConnectionString() | Output restituito: nome di connessione della libreria                   | Il nome di connessione della libreria o la connessione effettiva, a seconda dell'input       | Restituisce la stringa di connessione attiva per le connessioni ODBC o OLE DB.                                                  |
| FileDir()          | Output restituito:<br>connessione alla libreria                         | Output restituito:<br>Connessione alla libreria o percorso al file system, in base all'input | La funzione <b>FileDir</b> restituisce una stringa contenente il percorso della directory del file tabella in corso di lettura. |

## 9 Restrizione dell'accesso al file system

| Funzione        | Modalità standard                                                       | Modalità legacy                                                                              | Definizione                                                                                                                                                                                                                                                                          |
|-----------------|-------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FilePath()      | Output restituito:<br>connessione alla libreria                         | Output restituito:<br>Connessione alla libreria o percorso al file system, in base all'input | La funzione <b>FilePath</b> restituisce una stringa contenente il percorso completo del file tabella in corso di lettura.                                                                                                                                                            |
| FileSize()      | Input supportato:<br>Percorso che utilizza la connessione alla libreria | Input supportato:<br>Percorso che utilizza la connessione alla libreria o al file system     | La funzione <b>FileSize</b> restituisce un valore intero contenente le dimensioni in byte del file filename oppure, se non viene specificato alcun filename, del file tabella in corso di lettura.                                                                                   |
| FileTime()      | Input supportato:<br>Percorso che utilizza la connessione alla libreria | Input supportato:<br>Percorso che utilizza la connessione alla libreria o al file system     | La funzione <b>FileTime</b> restituisce data e ora in formato UTC dell'ultima modifica apportata al file specificato. Se non viene specificato nessun file, la funzione restituisce la data e l'ora in formato UTC dell'ultima modifica apportata al file tabella attualmente letto. |
| GetFolderPath() | Non supportato                                                          | Output restituito:<br>percorso assoluto                                                      | La funzione <b>GetFolderPath</b> restituisce il valore della funzione Microsoft Windows <i>SHGetFolderPath</i> . Questa funzione utilizza come input il nome di una cartella Microsoft Windows e restituisce il percorso completo della cartella.                                    |

## 9 Restrizione dell'accesso al file system

| Funzione         | Modalità standard                                                       | Modalità legacy                                                                          | Definizione                                                                                                                                                                              |
|------------------|-------------------------------------------------------------------------|------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| QvdCreateTime()  | Input supportato:<br>Percorso che utilizza la connessione alla libreria | Input supportato:<br>Percorso che utilizza la connessione alla libreria o al file system | Questa funzione di script restituisce l'intestazione XML relativa alla data e ora da un file QVD, se disponibile, altrimenti restituisce NULL. Nella data e ora, l'ora è fornita in UTC. |
| QvdFieldName()   | Input supportato:<br>Percorso che utilizza la connessione alla libreria | Input supportato:<br>Percorso che utilizza la connessione alla libreria o al file system | La funzione script restituisce il nome del numero campo <b>fieldno</b> in un file QVD. Se il campo non esiste, viene restituito NULL.                                                    |
| QvdNoOfFields()  | Input supportato:<br>Percorso che utilizza la connessione alla libreria | Input supportato:<br>Percorso che utilizza la connessione alla libreria o al file system | Questa funzione dello script restituisce il numero di campi all'interno di file QVD.                                                                                                     |
| QvdNoOfRecords() | Input supportato:<br>Percorso che utilizza la connessione alla libreria | Input supportato:<br>Percorso che utilizza la connessione alla libreria o al file system | Questa funzione dello script restituisce il numero di record attualmente presente in un file QVD.                                                                                        |
| QvdTableName()   | Input supportato:<br>Percorso che utilizza la connessione alla libreria | Input supportato:<br>Percorso che utilizza la connessione alla libreria o al file system | Questa funzione di script restituisce il nome della tabella memorizzata in un file QVD.                                                                                                  |

### Funzioni di sistema

#### Funzioni di sistema

| Funzione       | Modalità standard | Modalità legacy                         | Definizione                                                                                          |
|----------------|-------------------|-----------------------------------------|------------------------------------------------------------------------------------------------------|
| DocumentPath() | Non supportato    | Output restituito:<br>percorso assoluto | Questa funzione restituisce una stringa contenente il percorso completo dell'app Qlik Sense attuale. |

### 9.3 Disabilitazione della modalità standard

È possibile disabilitare la modalità standard o, in altre parole, impostare la modalità legacy, per poter riutilizzare gli script di caricamento di QlikView che fanno riferimento a percorsi di file assoluti o relativi, così come alle connessioni della libreria.



*La disabilitazione della modalità standard può determinare rischi per la sicurezza, in quanto viene esposto il file system.*

#### Qlik Sense

Per Qlik Sense, la modalità può essere disabilitata QMC mediante la proprietà **Modalità standard**.

#### Qlik Sense Desktop

In Qlik Sense Desktop è possibile impostare la modalità standard/legacy nel file *Settings.ini*.

Se si è installato Qlik Sense Desktop utilizzando il percorso di installazione predefinito, *Settings.ini* si trova nel percorso *C:\Users\{user}\Documents\Qlik\Sense\Settings.ini*. Se si è installato Qlik Sense Desktop in una cartella selezionata dall'utente, *Settings.ini* si trova nella cartella *Engine* del percorso di installazione.

**Procedere come indicato di seguito:**

1. Aprire il file *Settings.ini* in un editor di testo.
2. Modificare *StandardReload=1* in *StandardReload=0*.
3. Salvare il file e avviare Qlik Sense Desktop.

Qlik Sense Desktop sarà ora eseguito in modalità legacy.

#### Impostazioni

Le impostazioni disponibili per *StandardReload* sono:

- 1 (modalità standard)
- 0 (modalità legacy)

## 10 Scripting a livello di grafico

Durante la modifica dei dati sul grafico, l'utente utilizza un sottoinsieme di script Qlik Sense, costituito da una serie di istruzioni. Un'istruzione può essere un'istruzione di script regolare o un'istruzione di controllo dello script. Alcune istruzioni possono essere precedute da prefissi.

Le istruzioni regolari vengono generalmente utilizzate per la manipolazione dei dati. Queste istruzioni possono essere scritte su un qualsiasi numero di righe nello script e devono sempre terminare con un punto e virgola, ";".

In genere, le istruzioni di controllo vengono utilizzate per controllare il flusso di esecuzione dello script. Ogni clausola di un'istruzione di controllo deve essere mantenuta in una singola riga dello script e può terminare con un punto e virgola oppure con un fine riga.

I prefissi possono essere applicati alle istruzioni regolari pertinenti, ma mai a istruzioni di controllo.

Tutte le parole chiave dello script possono essere immesse con qualsiasi combinazione di caratteri maiuscoli e minuscoli. I nomi dei campi e delle variabili utilizzati nelle istruzioni possono essere immessi indipendentemente dal formato del carattere.

In questa sezione, è possibile trovare un elenco in ordine alfabetico di tutte le istruzioni script, istruzioni di controllo e prefissi disponibili nel sottoinsieme dello script utilizzato durante la modifica dei dati grafici.

### 10.1 Istruzione di controllo

Durante la modifica dei dati sul grafico, l'utente utilizza un sottoinsieme di script Qlik Sense, costituito da una serie di istruzioni. Un'istruzione può essere un'istruzione di script regolare o un'istruzione di controllo dello script.

In genere, le istruzioni di controllo vengono utilizzate per controllare il flusso di esecuzione dello script. Ogni clausola di un'istruzione di controllo deve essere inserita in una singola riga nello script e può terminare con un punto e virgola o con un carattere di fine riga.

I prefissi non vengono mai applicati alle istruzioni di controllo.

Tutte le parole chiave dello script possono essere immesse con qualsiasi combinazione di caratteri maiuscoli e minuscoli.

### Panoramica istruzioni di controllo modificatore grafico

Ciascuna funzione viene descritta ulteriormente dopo la panoramica. È inoltre possibile fare clic sul nome della funzione nella sintassi per accedere immediatamente ai dettagli per tale funzione specifica.

#### Call

L'istruzione di controllo **call** consente di chiamare una subroutine che deve essere definita da un'istruzione **sub** precedente.

```
Call name ( [ paramlist ] )
```

### Do..loop

L'istruzione di controllo **do..loop** è un costrutto per la ripetizione di script che esegue una o più istruzioni finché non incontra una condizione logica.

```
Do..ciclo [ ( while | until ) condition ] [statements]  
[exit do [ ( when | unless ) condition ] [statements]  
loop [ ( while | until ) condition ]
```

### End

La parola chiave dello script **End** viene utilizzata per chiudere le clausole **If**, **Sub** e **Switch**.

### Exit

La parola chiave dello script **Exit** fa parte dell'istruzione **Exit Script**, ma può essere utilizzata anche per uscire dalle clausole **Do**, **For** o **Sub**.

### Exit script

Questa istruzione di controllo interrompe l'esecuzione dello script. Può essere inserita in un punto qualsiasi dello script.

```
Exit script [ ( when | unless ) condition ]
```

### For..next

L'istruzione di controllo **for..next** è un costrutto per la ripetizione di script con un contatore. Le istruzioni all'interno del ciclo incluso tra **for** e **next** verranno eseguite per ogni valore del contatore in base ai limiti inferiore e superiore specificati.

```
For..next counter = expr1 to expr2 [ stepexpr3 ]
```

```
[statements]
```

```
[exit for [ ( when | unless ) condition ]
```

```
[statements]
```

```
Next [counter]
```

### For each ..next

L'istruzione di controllo **for each..next** è un costrutto per la ripetizione di script che esegue una o più istruzioni per ogni valore in un elenco le cui voci sono separate da virgole. Le istruzioni incluse nel ciclo fra **for** e **next** verranno eseguite per ogni valore nell'elenco.

```
For each..next var in list
```

```
[statements]
```

```
[exit for [ ( when | unless ) condition ]
```

```
[statements]
```

```
next [var]
```

### If..then

L'istruzione di controllo **if..then** è un costrutto per la selezione di script che forza l'esecuzione dello script su percorsi diversi in base a una o più condizioni logiche.



*Poiché **if..then** è un'istruzione di controllo e come tale termina con un punto e virgola o con un carattere di fine riga, ciascuna delle quattro possibili clausole corrispondenti (**if..then**, **elseif..then**, **else** e **end if**) deve essere contenuta in una sola riga.*

```
If..then..elseif..else..end if condition then
```

```
[ statements ]
```

```
{ elseif condition then
```

```
[ statements ] }
```

```
[ else
```

```
[ statements ] ]
```

```
end if
```

### Next

La parola chiave dello script **Next** consente di chiudere i loop **For**.

### Sub

L'istruzione di controllo **sub..end sub** definisce una subroutine che può essere richiamata da un'istruzione **call**.

```
Sub..end sub name [ ( paramlist ) ] statements end sub
```

### Switch

L'istruzione di controllo **switch** è un costrutto per la selezione di script che forza l'esecuzione dello script su percorsi diversi, in base al valore di un'espressione.

```
Switch..case..default..end switch expression {case valuelist [ statements ]}  
[default statements] end switch
```

### To

La parola chiave dello script **To** viene utilizzata in diverse istruzioni dello script.

### Call

L'istruzione di controllo **call** consente di chiamare una subroutine che deve essere definita da un'istruzione **sub** precedente.

#### Sintassi:

```
Call name ( [ paramlist ] )
```

**Argomenti:**

## Argomenti

| Argomento | Descrizione                                                                                                                                                                    |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| name      | Il nome della subroutine.                                                                                                                                                      |
| paramlist | Un elenco separato da virgole di parametri effettivi da inviare alla subroutine. Ogni voce dell'elenco può essere un nome di campo, una variabile o un'espressione arbitraria. |

La subroutine chiamata da un'istruzione **call** deve essere definita da un'istruzione **sub** rilevata precedentemente durante l'esecuzione dello script.

I parametri vengono copiati nella subroutine e, se il parametro nell'istruzione **call** è una variabile e non un'espressione, verranno copiati nuovamente all'uscita dalla subroutine.

**Limiti:**

- Poiché **call** è un'istruzione di controllo e come tale termina con un punto e virgola o con un carattere di fine riga, non deve superare un limite di riga.
- Quando si definisce una routine secondaria con `sub . . end sub` all'interno di un'istruzione di controllo, ad esempio `if . . then`, è possibile richiamare la routine secondaria solo dall'interno della stessa istruzione di controllo.

**Do..ciclo**

L'istruzione di controllo **do..loop** è un costrutto per la ripetizione di script che esegue una o più istruzioni finché non incontra una condizione logica.

**Sintassi:**

```
Do [ ( while | until ) condition ] [statements]
[exit do [ ( when | unless ) condition ] [statements]
loop[ ( while | until ) condition ]
```



*Poiché **do..loop** è un'istruzione di controllo che termina con un punto e virgola o con un carattere di fine riga, ciascuna delle tre possibili clausole corrispondenti (**do**, **exit do** e **loop**) deve essere contenuta in una sola riga.*

**Argomenti:**

## Argomenti

| Argomento     | Descrizione                                                                                                                                                                                                                                                                                                                       |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| condition     | Un'espressione logica che restituisce un valore True o False.                                                                                                                                                                                                                                                                     |
| statements    | Qualsiasi gruppo di una o più istruzioni dello script di Qlik Sense.                                                                                                                                                                                                                                                              |
| while / until | La clausola condizionale <b>while</b> o <b>until</b> deve comparire una sola volta in ciascuna istruzione <b>do..loop</b> , dopo <b>do</b> o dopo <b>loop</b> . Ogni espressione condition verrà interpretata solo al primo rilevamento, ma verrà valutata ogni volta che sarà rilevata nel ciclo.                                |
| exit do       | Se all'interno del ciclo è presente una clausola <b>exit do</b> , l'esecuzione dello script verrà trasferita alla prima istruzione dopo la clausola <b>loop</b> indicando quindi la fine del ciclo. Una clausola <b>exit do</b> può essere resa condizionale dall'utilizzo opzionale di un suffisso <b>when</b> o <b>unless</b> . |

**End**

La parola chiave dello script **End** viene utilizzata per chiudere le clausole **If**, **Sub** e **Switch**.

**Exit**

La parola chiave dello script **Exit** fa parte dell'istruzione **Exit Script**, ma può essere utilizzata anche per uscire dalle clausole **Do**, **For** o **Sub**.

**Exit script**

Questa istruzione di controllo interrompe l'esecuzione dello script. Può essere inserita in un punto qualsiasi dello script.

**Sintassi:**

```
Exit Script [ (when | unless) condition ]
```

Poiché **exit script** è un'istruzione di controllo e come tale termina con un punto e virgola o con un carattere di fine riga, non deve superare un limite di riga.

**Argomenti:**

## Argomenti

| Argomento     | Descrizione                                                                                                                        |
|---------------|------------------------------------------------------------------------------------------------------------------------------------|
| condition     | Un'espressione logica che restituisce un valore True o False.                                                                      |
| when / unless | Un'istruzione <b>exit script</b> può essere resa condizionale dall'utilizzo opzionale della clausola <b>when</b> o <b>unless</b> . |

**Esempi:**

```
//Exit script
Exit Script;
```

```
//Exit script when a condition is fulfilled
Exit Script when a=1
```

**For..next**

L'istruzione di controllo **for..next** è un costrutto per la ripetizione di script con un contatore. Le istruzioni all'interno del ciclo incluso tra **for** e **next** verranno eseguite per ogni valore del contatore in base ai limiti inferiore e superiore specificati.

**Sintassi:**

```
For counter = expr1 to expr2 [ step expr3 ]
```

```
[statements]
```

```
[exit for [ ( when | unless ) condition ]
```

```
[statements]
```

```
Next [counter]
```

Le espressioni *expr1*, *expr2* ed *expr3* vengono valutate solo la prima volta che il ciclo viene eseguito. Il valore della variabile di conteggio può essere modificato dalle istruzioni all'interno del ciclo, tuttavia l'utilizzo di questa procedura di programmazione non è consigliato.

Se all'interno del ciclo è presente una clausola **exit for**, l'esecuzione dello script verrà trasferita alla prima istruzione dopo la clausola **next** indicando quindi la fine del ciclo. Una clausola **exit for** può essere resa condizionale dall'utilizzo opzionale di un suffisso **when** o **unless**.



*Poiché **for..next** è un'istruzione di controllo che termina con un punto e virgola o con un carattere di fine riga, ciascuna delle tre possibili clausole corrispondenti (**for..to..step**, **exit for** e **next**) deve essere contenuta in una sola riga.*

**Argomenti:**

## Argomenti

| Argomento | Descrizione                                                                                                                                                               |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| counter   | Un nome di variabile. Se <i>counter</i> viene specificato dopo <b>next</b> , deve avere lo stesso nome di variabile rilevato dopo l'istruzione <b>for</b> corrispondente. |

| Argomento  | Descrizione                                                                                                                             |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| expr1      | Un'espressione che determina il primo valore della variabile <i>counter</i> per cui deve essere eseguito il ciclo.                      |
| expr2      | Un'espressione che determina l'ultimo valore della variabile <i>counter</i> per cui deve essere eseguito il ciclo.                      |
| expr3      | Un'espressione che determina il valore che indica l'incremento della variabile <i>counter</i> ogni volta che il ciclo è stato eseguito. |
| condition  | Un'espressione logica che restituisce un valore True o False.                                                                           |
| statements | Qualsiasi gruppo di una o più istruzioni dello script di Qlik Sense.                                                                    |

## For each..next

L'istruzione di controllo **for each..next** è un costrutto per la ripetizione di script che esegue una o più istruzioni per ogni valore in un elenco le cui voci sono separate da virgole. Le istruzioni incluse nel ciclo fra **for** e **next** verranno eseguite per ogni valore nell'elenco.

### Sintassi:

Una sintassi speciale consente di generare elenchi contenenti nomi di file e di directory nella directory attuale.

```
for each var in list
```

```
[statements]
```

```
[exit for [ ( when | unless ) condition ]
```

```
[statements]
```

```
next [var]
```

### Argomenti:

#### Argomenti

| Argomento | Descrizione                                                                                                                                                                                                                                                     |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| var       | Il nome di una variabile di script che acquisisce un nuovo valore dall'elenco a ogni esecuzione del ciclo. Se <b>var</b> viene specificato dopo <b>next</b> , deve avere lo stesso nome di variabile rilevato dopo l'istruzione <b>for each</b> corrispondente. |

Il valore della variabile **var** può essere modificato dalle istruzioni all'interno del ciclo, tuttavia l'utilizzo di questa procedura di programmazione non è consigliato.

Se all'interno del ciclo è presente una clausola **exit for**, l'esecuzione dello script verrà trasferita alla prima istruzione dopo la clausola **next** indicando quindi la fine del ciclo. Una clausola **exit for** può essere resa condizionale dall'utilizzo opzionale di un suffisso **when** o **unless**.



Poiché **for each...next** è un'istruzione di controllo che termina con un punto e virgola o con un carattere di fine riga, ciascuna delle tre possibili clausole corrispondenti (**for each**, **exit for** e **next**) deve essere contenuta in una sola riga.

**Sintassi:**

```
list := item { , item }
```

```
item := constant | (expression) | filelist mask | dirlist mask |  
fieldvaluelist mask
```

## Argomenti

| Argomento           | Descrizione                                                                                                                                                                                                                                                                                                                                                              |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| constant            | Qualsiasi numero o stringa. Tenere presente che una stringa inserita direttamente nello script deve essere racchiusa tra virgolette singole. Se la stringa non viene racchiusa tra virgolette singole, verrà interpretata come una variabile, pertanto verrà utilizzato il valore della variabile. Non è necessario che i numeri siano racchiusi tra virgolette singole. |
| expression          | Un'espressione arbitraria.                                                                                                                                                                                                                                                                                                                                               |
| mask                | Una maschera di un nome di file o di cartella che può includere un carattere qualsiasi di nome di file valido, così come i caratteri speciali standard, quali * e ?.<br><br>È possibile utilizzare percorsi di file assoluti o percorsi lib://.                                                                                                                          |
| condition           | Un'espressione logica che restituisce un valore True o False.                                                                                                                                                                                                                                                                                                            |
| statements          | Qualsiasi gruppo di una o più istruzioni dello script di Qlik Sense.                                                                                                                                                                                                                                                                                                     |
| filelist mask       | Questa sintassi restituisce un elenco con valori separati da virgole di tutti i file presenti nella directory attuale che presentano una corrispondenza con la maschera del nome di file.<br><br><div>  Questo argomento supporta esclusivamente le connessioni alla libreria in modalità standard. </div>                                                               |
| dirlist mask        | Questa sintassi restituisce un elenco con valori separati da virgole di tutte le cartelle incluse nella cartella attuale che presentano una corrispondenza con la maschera del nome di file.<br><br><div>  Questo argomento supporta esclusivamente le connessioni alla libreria in modalità standard. </div>                                                            |
| fieldvaluelist mask | Questa sintassi ripete i valori di un campo già caricato in Qlik Sense.                                                                                                                                                                                                                                                                                                  |



*Qlik Connettori a provider di archiviazione Web e altre connessioni DataFiles non supportano le maschere di filtro che utilizzano caratteri speciali (\* e ?).*

### Example 1: Caricamento di un elenco di file

```
// LOAD the files 1.csv, 3.csv, 7.csv and xyz.csv
for each a in 1,3,7,'xyz'
  LOAD * from file$(a).csv;
next
```

### Example 2: Creazione di un elenco di file sul disco

In questo esempio viene caricato un elenco di tutti i file correlati a Qlik Sense in una cartella.

```
sub DoDir (Root)
  for each Ext in 'qvw', 'qva', 'qvo', 'qvs', 'qvc', 'qvf', 'qvd'

    for each File in filelist (Root&'/*.' &Ext)

      LOAD
        '$(File)' as Name,
        FileSize( '$(File)' ) as Size,
        FileTime( '$(File)' ) as FileTime
      autogenerate 1;

    next File

  next Ext
  for each Dir in dirlist (Root&'/*' )

    call DoDir (Dir)

  next Dir
end sub

call DoDir ('lib://DataFiles')
```

### Example 3: Ripetizione dei valori di un campo

In questo esempio viene ripetuto l'elenco di valori caricati di FIELD e viene generato un nuovo campo NEWFIELD. Per ciascun valore di FIELD, verranno creati due record NEWFIELD.

```
load * inline [
FIELD
one
two
three
];

FOR Each a in FieldValueList('FIELD')
```

```
LOAD '$(a)' &'-'&RecNo() as NEWFIELD AutoGenerate 2;  
NEXT a
```

La tabella risultante avrà l'aspetto seguente:

Example table

| NEWFIELD |
|----------|
| one-1    |
| one-2    |
| two-1    |
| two-2    |
| three-1  |
| three-2  |

### If..then..elseif..else..end if

L'istruzione di controllo **if..then** è un costrutto per la selezione di script che forza l'esecuzione dello script su percorsi diversi in base a una o più condizioni logiche.

In genere, le istruzioni di controllo vengono utilizzate per controllare il flusso di esecuzione dello script. In un'espressione del grafico, utilizzare invece la funzione condizionale **if**.

#### Sintassi:

```
If condition then
```

```
[ statements ]
```

```
{ elseif condition then
```

```
[ statements ] }
```

```
[ else
```

```
[ statements ] ]
```

```
end if
```

Poiché **if..then** è un'istruzione di controllo e come tale termina con un punto e virgola o con un carattere di fine riga, ciascuna delle quattro possibili clausole corrispondenti (**if..then**, **elseif..then**, **else** e **end if**) deve essere contenuta in una sola riga.

**Argomenti:**

Argomenti

| Argomento  | Descrizione                                                          |
|------------|----------------------------------------------------------------------|
| condition  | Un'espressione logica che può restituire un valore True o False.     |
| statements | Qualsiasi gruppo di una o più istruzioni dello script di Qlik Sense. |

**Example 1:**

```
if a=1 then
    LOAD * from abc.csv;

    SQL SELECT e, f, g from tab1;
end if
```

**Example 2:**

```
if a=1 then; drop table xyz; end if;
```

**Example 3:**

```
if x>0 then
    LOAD * from pos.csv;
elseif x<0 then
    LOAD * from neg.csv;
else
    LOAD * from zero.txt;
end if
```

## Next

La parola chiave dello script **Next** consente di chiudere i loop **For**.

## Sub..end sub

L'istruzione di controllo **sub..end sub** definisce una subroutine che può essere richiamata da un'istruzione **call**.

**Sintassi:**

```
Sub name [ ( paramlist ) ] statements end sub
```

Gli argomenti vengono copiati nella subroutine e, se i relativi parametri reali nell'istruzione **call** corrispondono a un nome di variabile, vengono copiati nuovamente quando si chiude la subroutine.

Se una subroutine presenta più parametri formali di quelli effettivi passati da un'istruzione **call**, i parametri extra vengono inizializzati su NULL e possono essere utilizzati come variabili locali all'interno della subroutine.

### Argomenti:

Argomenti

| Argomento  | Descrizione                                                                                                                                                               |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| name       | Il nome della subroutine.                                                                                                                                                 |
| paramlist  | Un elenco separato da virgole di nomi di variabili per i parametri formali della subroutine. Può essere utilizzato come qualsiasi variabile all'interno della subroutine. |
| statements | Qualsiasi gruppo di una o più istruzioni dello script di Qlik Sense.                                                                                                      |

### Limiti:

- Poiché **sub** è un'istruzione di controllo che termina con un punto e virgola o con un carattere di fine riga, ciascuna delle due clausole corrispondenti (**sub** e **end sub**) deve essere contenuta in una sola riga.
- Quando si definisce una routine secondaria con **sub . . end sub** all'interno di un'istruzione di controllo, ad esempio **if . . then**, è possibile richiamare la routine secondaria solo dall'interno della stessa istruzione di controllo.

### Example 1:

```
Sub INCR (I,J)

I = I + 1

Exit Sub when I < 10

J = J + 1

End Sub

Call INCR (X,Y)
```

### Example 2: - trasferimento parametri

```
Sub ParTrans (A,B,C)

A=A+1

B=B+1

C=C+1

End Sub
```

```
A=1
```

```
X=1
```

```
C=1
```

```
Call ParTrans (A, (X+1)*2)
```

Dall'esempio precedente risulta che localmente, all'interno della subroutine, A verrà inizializzato su 1, B verrà inizializzato su 4 e C verrà inizializzato su NULL.

Quando si chiude la subroutine, la variabile globale A otterrà 2 come valore (ricopiato dalla subroutine). Il secondo parametro reale "(X+1)\*2" non verrà ricopiato dato che non si tratta di una variabile. Infine, la variabile globale C non verrà influenzata dalla chiamata della subroutine.

## Switch..case..default..end switch

L'istruzione di controllo **switch** è un costrutto per la selezione di script che forza l'esecuzione dello script su percorsi diversi, in base al valore di un'espressione.

### Sintassi:

```
Switch expression {case valuelist [ statements ]} [default statements] end
switch
```



Poiché **switch** è un'istruzione di controllo e come tale termina con un punto e virgola o con un carattere di fine riga, ciascuna delle quattro possibili clausole corrispondenti (**switch**, **case**, **default** e **end switch**) deve essere contenuta in una sola riga.

### Argomenti:

#### Argomenti

| Argomento  | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| expression | Un'espressione arbitraria.                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| valuelist  | Un elenco separato da virgole dei valori con i quali viene confrontato il valore dell'espressione. L'esecuzione dello script continua con le istruzioni del primo gruppo in cui il valore di valuelist è pari al valore nell'espressione. Ciascun valore in valuelist può essere un'espressione arbitraria. Se non viene individuata alcuna corrispondenza in nessuna clausola <b>case</b> , vengono eseguite le eventuali istruzioni della clausola <b>default</b> . |
| statements | Qualsiasi gruppo di una o più istruzioni dello script di Qlik Sense.                                                                                                                                                                                                                                                                                                                                                                                                  |

### Esempio:

```
Switch I
```

```
Case 1
```

```
LOAD '$(I): CASE 1' as case autogenerate 1;
```

Case 2

```
LOAD '$(I): CASE 2' as case autogenerate 1;
```

Default

```
LOAD '$(I): DEFAULT' as case autogenerate 1;
```

End Switch

## To

La parola chiave dello script **To** viene utilizzata in diverse istruzioni dello script.

## 10.2 Prefissi

I prefissi possono essere applicati alle istruzioni regolari pertinenti, ma mai a istruzioni di controllo.

Tutte le parole chiave dello script possono essere immesse con qualsiasi combinazione di caratteri maiuscoli e minuscoli. I nomi dei campi e delle variabili utilizzati nelle istruzioni possono essere immessi indipendentemente dal formato del carattere.

### Panoramica prefissi modificatore grafico

Ciascuna funzione viene descritta ulteriormente dopo la panoramica. È inoltre possibile fare clic sul nome della funzione nella sintassi per accedere immediatamente ai dettagli per tale funzione specifica.

#### Add

Il prefisso **Add** può essere aggiunto a qualsiasi istruzione **LOAD** o **SELECT** nello script per specificare che dovrebbe aggiungere record a un'altra tabella. Specifica anche che questa istruzione dovrebbe essere eseguita in un ricaricamento parziale. Il prefisso **Add** può essere usato anche in un'istruzione **Map**.

```
Add [only] [Concatenate[(tablename)]] (loadstatement | selectstatement)  
Add [ Only ] mapstatement
```

#### Replace

Il prefisso **Replace** può essere aggiunto a qualsiasi istruzione **LOAD** o **SELECT** nello script per specificare che la tabella caricata dovrebbe sostituire un'altra tabella. Specifica anche che questa istruzione dovrebbe essere eseguita in un ricaricamento parziale. Il prefisso **Replace** può essere usato anche in un'istruzione **Map**.

```
Replace [only] [Concatenate[(tablename)]] (loadstatement | selectstatement)  
Replace [only] mapstatement
```

#### Add

In un contesto di modifica del grafico, il prefisso **Add** è utilizzato con **LOAD** per aggiungere valori alla tabella *HC1*, che rappresenta l'ipercubo calcolato da Motore associativo di Qlik. È possibile specificare uno o più colonne. I valori mancanti vengono automaticamente completati da Motore associativo di Qlik.

**Sintassi:**

```
Add loadstatement
```

**Esempio:**

Questo esempio aggiunge due righe alle colonne *Date* e *Vendite* dall'istruzione inline.

```
Add Load
x as Dates,
y as Sales
inline
[
Dates,Sales
2001/09/1,1000
2001/09/10,-300
]
```

## Replace

In un contesto di modifica del grafico, il prefisso **Replace** modifica tutti i valori della tabella *HC1* con un valore calcolato definito dallo script.

**Sintassi:**

```
Replace loadstatement
```

**Esempio:**

Questo esempio sovrascrive tutti i valori nella colonna *z* con la somma di *x* e *y*.

```
Replace Load
x+y as z
Resident HC1;
```

## 10.3 Istruzioni regolari

Le istruzioni regolari vengono generalmente utilizzate per la manipolazione dei dati. Queste istruzioni possono essere scritte su un qualsiasi numero di righe nello script e devono sempre terminare con un punto e virgola, ";".

Tutte le parole chiave dello script possono essere immesse con qualsiasi combinazione di caratteri maiuscoli e minuscoli. I nomi dei campi e delle variabili utilizzati nelle istruzioni possono essere immessi indipendentemente dal formato del carattere.

### Panoramica istruzioni regolari modificatore grafico

Ciascuna funzione viene descritta ulteriormente dopo la panoramica. È inoltre possibile fare clic sul nome della funzione nella sintassi per accedere immediatamente ai dettagli per tale funzione specifica.

### LOAD

In un contesto di modifica del grafico, l'istruzione **LOAD** carica dati aggiuntivi sull'ipercubo, da dati definiti nello script o da una tabella precedentemente caricata. È anche possibile caricare dati da connessioni di analisi.



L'istruzione **LOAD** deve includere il prefisso **Replace** o **Add**, altrimenti verrà rifiutata.

```
Add | Replace Load [ distinct ] fieldlist
```

```
(
```

```
inline data [ format-spec ] |
```

```
resident table-label
```

```
) | extension pluginname.functionname([script] tabledescription)
```

```
[ where criterion | while criterion ]
```

```
[ group by groupbyfieldlist ]
```

```
[order by orderbyfieldlist ]
```

### Let

L'istruzione **let** è un complemento all'istruzione **set**, utilizzata per definire le variabili degli script.

L'istruzione **let**, a differenza dell'istruzione **set**, valuta l'espressione posta sul lato destro del simbolo '=' al tempo di esecuzione dello script prima dell'assegnazione alla variabile.

```
Let variablename=expression
```

### Set

L'istruzione **set** viene utilizzata per definire le variabili di script. Le variabili possono essere utilizzate per sostituire stringhe, percorsi, unità e così via.

```
Set variablename=string
```

### Put

L'istruzione **Put** viene utilizzata per impostare alcuni valori numerici nell'ipercubo.

### HCValue

L'istruzione **HCValue** è utilizzata per recuperare valori in una riga di una colonna specificata.

### Load

In un contesto di modifica del grafico, l'istruzione **LOAD** carica dati aggiuntivi sull'ipercubo, da dati definiti nello script o da una tabella precedentemente caricata. È anche possibile caricare dati da connessioni di analisi.



L'istruzione **LOAD** deve includere il prefisso **Replace** o **Add**, altrimenti verrà rifiutata.

### Sintassi:

```
Add | Replace LOAD fieldlist
```

```
(
```

```
inline data [ format-spec ] |
```

```
resident table-label
```

```
) | extension pluginname.functionname([script] tabledescription)
```

```
[ where criterion | while criterion ]
```

```
[ group by groupbyfieldlist ]
```

```
[order by orderbyfieldlist ]
```

Argomenti:

## Argomenti

| Argomento | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| fieldlist | <p><i>fieldlist</i> ::= ( *   <i>field</i>{, *   <i>field</i> } )</p> <p>Un elenco di campi da caricare. L'utilizzo del carattere * come elenco dei campi indica tutti i campi della tabella.</p> <p><i>field</i> ::= ( <i>fieldref</i>   <i>expression</i> ) [<b>as</b> <i>aliasname</i> ]</p> <p>La definizione di campo deve sempre contenere un valore letterale, un riferimento a un campo esistente o un'espressione.</p> <p><i>fieldref</i> ::= ( <i>fieldname</i>   @<i>fieldnumber</i>   @<i>startpos:endpos</i> [ <i>I</i>   <b>U</b>   <b>R</b>   <b>B</b>   <i>T</i> ] )</p> <p><i>fieldname</i> è un testo che è identico al nome di un campo nella tabella. Tenere presente che il nome di campo deve essere racchiuso da virgolette doppie o parentesi quadre se, ad esempio, contiene spazi. Talvolta i nomi dei campi non sono disponibili in modo esplicito. Verrà quindi utilizzata una notazione differente:</p> <p>@<i>fieldnumber</i> rappresenta il numero di campo di un file tabellare delimitato. Deve essere un intero positivo preceduto da "@". La numerazione viene sempre eseguita partendo da 1 fino al numero di campi presenti.</p> <p>@<i>startpos:endpos</i> rappresenta le posizioni di inizio e di fine di un campo in un file con record a lunghezza fissa. Le posizioni devono essere entrambe numeri interi positivi. I due valori numerici devono essere preceduti dal simbolo "@" e separati da due punti. La numerazione viene sempre eseguita partendo da 1 fino al numero di posizioni presenti. Nell'ultimo campo <b>n</b> viene utilizzato come posizione finale.</p> <ul style="list-style-type: none"> <li>• Se @<i>startpos:endpos</i> è seguito immediatamente dal carattere <b>I</b> o <b>U</b>, i byte letti verranno interpretati come un valore intero binario con segno (<b>I</b>) o senza segno (<b>U</b>), in base all'ordine dei byte Intel. Il numero di posizioni lette deve essere 1, 2 o 4.</li> <li>• Se @<i>startpos:endpos</i> è immediatamente seguito dal carattere <b>R</b>, la lettura dei byte verrà interpretata come un numero binario reale (a virgola mobile a 32 bit o a 64 bit conforme allo standard IEEE). Il numero di posizioni lette deve essere 4 o 8.</li> <li>• Se @<i>startpos:endpos</i> è immediatamente seguito dal carattere <b>B</b>, la lettura dei byte sarà interpretata come numeri BCD (Binary Coded Decimal), in base allo standard COMP-3. Può essere specificato un numero di byte qualsiasi.</li> </ul> <p><i>expression</i> può essere una funzione numerica o una funzione stringa basata su uno o molti altri campi della stessa tabella. Per ulteriori informazioni, vedere la sintassi delle espressioni.</p> <p><b>as</b> viene utilizzato per assegnare un nuovo nome al campo.</p> |

| Argomento | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| inline    | <p><b>inline</b> viene utilizzato quando i dati devono essere immessi direttamente nello script e non caricati da un file.</p> <p><i>data ::= [ text ]</i></p> <p>I dati immessi mediante una clausola <b>inline</b> devono essere racchiusi tra virgolette doppie o tra parentesi quadre. Il testo tra parentesi o virgolette viene interpretato allo stesso modo del contenuto di un file. Pertanto, quando si desidera modificare o immettere una nuova riga in un file di testo, è necessario eseguire questa operazione anche nel testo di una clausola <b>inline</b>, ad esempio premendo il tasto INVIO mentre si digita lo script. Il numero di colonne è definito nella prima riga.</p> <p><i>format-spec ::= ( fspec-item {, fspec-item } )</i></p> <p>La specifica del formato è costituita da un elenco di più voci di specifica del formato, racchiuse tra parentesi. Per ulteriori informazioni, vedere <a href="#">Voci per la specifica del formato (page 174)</a>.</p> |
| resident  | <p><b>resident</b> viene usato se i dati devono essere caricati da una tabella caricata in precedenza.</p> <p><i>table label</i> è un'etichetta che precede l'istruzione <b>LOAD</b> che ha creato la tabella originale. L'etichetta dovrà essere indicata con i due punti finali.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

| Argomento | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| extension | <p>È possibile caricare dati da connessioni di analisi. È necessario utilizzare la clausola <b>extension</b> per chiamare una funzione definita nel plug-in SSE (Server-Side Extension) o per valutare uno script.</p> <p>È possibile inviare una singola tabella al plug-in SSE, che restituisce una singola tabella di dati. Se il plug-in non specifica i nomi dei campi restituiti, i campi saranno denominati Field1, Field2 e così via.</p> <pre>Extension pluginname.functionname( tabledescription );</pre> <ul style="list-style-type: none"> <li>Caricamento di dati mediante una funzione in un plug-in SSE<br/> <i>tabledescription ::= (table { ,tablefield} )</i><br/>           Se i campi della tabella non vengono specificati, verranno utilizzati in ordine di caricamento.</li> <li>Caricamento di dati mediante valutazione di uno script in un plug-in SSE<br/> <i>tabledescription ::= ( script, table { ,tablefield} )</i></li> </ul> <p><b>Gestione dei tipi di dati nella definizione dei campi della tabella</b></p> <p>I tipi di dati sono riconosciuti automaticamente nelle connessioni di analisi. Se i dati non hanno valori numerici e hanno almeno una stringa di testo non NULL, il campo è considerato di tipo testo. In tutti gli altri casi è considerato di tipo numerico.</p> <p>È possibile forzare il tipo di dati racchiudendo il nome del campo in <b>String()</b> o <b>Mixed()</b>.</p> <ul style="list-style-type: none"> <li><b>String()</b> forza il tipo del campo su testo. Se il campo è numerico, viene estratta la parte di testo del valore duale, senza eseguire alcuna conversione.</li> <li><b>Mixed()</b> forza il tipo del campo su duale.</li> </ul> <p><b>String()</b> o <b>Mixed()</b> non possono essere utilizzati esternamente alle definizioni dei campi della tabella <b>extension</b> e non è possibile utilizzare altre funzioni di Qlik Sense nella definizione di un campo della tabella.</p> |
| where     | <p><b>where</b> è una clausola utilizzata per dichiarare se un record deve essere incluso o meno nella selezione. La selezione viene inclusa se <i>criterion</i> è True. <i>criterion</i> è un'espressione logica.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| while     | <p><b>while</b> è una clausola che specifica se un record deve essere letto ripetutamente. Viene letto lo stesso record finché <i>criterion</i> è True. Per risultare utile, una clausola <b>while</b> deve generalmente includere la funzione <b>IterNo( )</b>.</p> <p><i>criterion</i> è un'espressione logica.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

| Argomento | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| group by  | <p><b>group by</b> è una clausola usata per definire su quali campi devono essere aggregati (raggruppati) i dati. I campi di aggregazione devono essere inclusi in qualche modo nelle espressioni caricate. Nessun altro campo tranne i campi di aggregazione può essere usato al di fuori delle funzioni di aggregazione nelle espressioni caricate.</p> <p><i>groupbyfieldlist ::= (fieldname { ,fieldname } )</i></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| order by  | <p><b>order by</b> è una clausola utilizzata per ordinare i record di una tabella residente prima che vengano elaborati da un'istruzione <b>load</b>. La tabella residente può essere ordinata su uno o più campi, in modo crescente o decrescente. L'ordinamento viene eseguito innanzitutto in base ai valori numerici e secondariamente in base all'ordine di confronto nazionale. Questa clausola può essere utilizzata solamente quando la sorgente dati è una tabella residente. I campi di ordinamento specificano il campo in base al quale viene ordinata la tabella residente. Il campo può essere specificato da un nome o dal suo numero nella tabella residente (il primo campo è il numero 1).</p> <p><i>orderbyfieldlist ::= fieldname [ sortorder ] { , fieldname [ sortorder ] }</i></p> <p><i>sortorder</i> è <i>asc</i> per crescente o <i>desc</i> per decrescente. Se <i>sortorder</i> non è specificato, viene utilizzato <i>asc</i>.</p> <p><i>fieldname</i>, <i>path</i>, <i>filename</i> e <i>aliasname</i> sono stringhe di testo che rappresentano ciò che implica il loro rispettivo nome. Qualsiasi campo presente nella tabella sorgente può essere utilizzato come <i>fieldname</i>. Tuttavia, i campi creati mediante la clausola <i>as</i> (<i>aliasname</i>) non appartengono all'ambito e non possono essere utilizzati all'interno della stessa istruzione <b>load</b>.</p> |

## Let

L'istruzione **let** è un complemento all'istruzione **set**, utilizzata per definire le variabili degli script. L'istruzione **let**, a differenza dell'istruzione **set**, valuta l'espressione posta sul lato destro del simbolo '=' al tempo di esecuzione dello script prima dell'assegnazione alla variabile.

### Sintassi:

```
Let variablename=expression
```

Esempi e risultati:

| Esempio       | Risultato                                                                                                                                                                                                      |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Set x=3+4;    | \$(x) verrà valutato come '3+4 '                                                                                                                                                                               |
| Let y=3+4;    | \$(y) verrà valutato come '7 '                                                                                                                                                                                 |
| z=\$(y)+1;    | \$(z) verrà valutato come '8 '                                                                                                                                                                                 |
|               | Notare la differenza tra le istruzioni <b>Set</b> e <b>Let</b> . L'istruzione <b>Set</b> assegna la stringa '3+4' alla variabile, mentre l'istruzione <b>Let</b> valuta la stringa e assegna 7 alla variabile. |
| Let T=now( ); | \$(T) verrà restituito il valore dell'ora attuale.                                                                                                                                                             |

## Set

L'istruzione **set** viene utilizzata per definire le variabili di script. Le variabili possono essere utilizzate per sostituire stringhe, percorsi, unità e così via.

**Sintassi:**

```
Set variablename=string
```

**Example 1:**

```
Set FileToUse=Data1.csv;
```

**Example 2:**

```
Set Constant="My string";
```

**Example 3:**

```
Set BudgetYear=2012;
```

## Put

L'istruzione **put** viene utilizzata per impostare alcuni valori numerici nell'ipercubo.

È possibile accedere alle colonne tramite etichette. È possibile inoltre accedere a colonne e righe tramite ordine di dichiarazione. Per ulteriori informazioni, vedere gli esempi di seguito.

**Sintassi:**

```
put column(position)=value
```

**Example 1:**

È possibile accedere alle colonne tramite etichette.

Questo esempio imposterà un valore di 1 nella prima posizione della colonna con etichetta *Vendite*.

```
Put Sales(1) = 1;
```

### Example 2:

È possibile accedere alle colonne di misura tramite ordine di dichiarazione utilizzando il formato per misure `#hc1.measure`.

Questo esempio imposterà il valore 1000 nella decima posizione dell'ipercubo ordinato finale.

```
Put #hc1.measure.2(10) = 1000;
```

### Example 3:

È possibile accedere alle righe di dimensione tramite ordine di dichiarazione utilizzando il formato per dimensioni `#hc1.dimension`.

Questo esempio inserisce il valore della costante Pi nella quinta riga della terza dimensione dichiarata.

```
Put #hc1.dimension.3(5) = Pi();
```



*Se non sono presenti dimensioni o espressioni di questo tipo, in valore o etichette, viene restituito un errore che indica che la colonna non è stata trovata. Se l'indice della colonna non è compreso nell'intervallo, non viene visualizzato alcun errore.*

## HCValue

La funzione **HCValue** è utilizzata per recuperare valori in una riga di una colonna specificata.

### Sintassi:

```
HCValue(column, position)
```

### Example 1:

Questo esempio restituisce il valore alla prima posizione della colonna con etichetta "Vendite".

```
HCValue(Sales,1)
```

### Example 2:

Questo esempio restituisce il valore alla decima posizione dell'ipercubo ordinato.

```
HCValue(#hc1.measure2,10)
```

### Example 3:

Questo esempio restituisce il valore alla quinta riga della terza dimensione.

```
HCValue(#hc1.dimension.3,5)
```



*Se non sono presenti dimensioni o espressioni di questo tipo, in valore o etichette, viene restituito un errore che indica che la colonna non è stata trovata. Se l'indice della colonna non è compreso nell'intervallo, viene restituito NULL.*

## 11 Funzioni e istruzioni di QlikView non supportate in Qlik Sense

La maggior parte delle funzioni e delle istruzioni utilizzabili negli script di caricamento e nelle espressioni grafiche di QlikView è supportata anche in Qlik Sense, tuttavia vi sono alcune eccezioni che sono descritte di seguito.

### 11.1 Istruzioni di script non supportate in Qlik Sense

Istruzioni di script QlikView non supportate in Qlik Sense

| Istruzione | Commenti                       |
|------------|--------------------------------|
| Command    | Utilizzare invece <b>SQL</b> . |
| InputField |                                |

### 11.2 Funzioni non supportate in Qlik Sense

In questo elenco sono descritte le funzioni grafiche e di script di QlikView non supportate in Qlik Sense.

- **GetCurrentField**
- **GetExtendedProperty**
- **Input**
- **InputAvg**
- **InputSum**
- **MsgBox**
- **NoOfReports**
- **ReportComment**
- **ReportId**
- **ReportName**
- **ReportNumber**

### 11.3 Prefissi non supportati in Qlik Sense

Questo elenco presenta i prefissi di QlikView non supportati da Qlik Sense.

- **Bundle**
- **Image\_Size**
- **Info**

## 12 Funzioni e istruzioni non consigliate in Qlik Sense

La maggior parte delle funzioni e delle istruzioni che è possibile utilizzare negli script di caricamento e nelle espressioni grafiche di QlikView è supportata anche in Qlik Sense, tuttavia l'utilizzo di alcune funzioni non è consigliato in Qlik Sense. Vi sono inoltre funzioni e istruzioni disponibili in versioni precedenti di Qlik Sense che sono stati deprecati.

Per motivi di compatibilità, tali funzioni continueranno a funzionare come previsto, tuttavia è consigliabile aggiornare il codice come illustrato in questa sezione, in quanto esse potrebbero essere rimosse dalle versioni successive.

### 12.1 Istruzioni di script non consigliate in Qlik Sense

In questa tabella sono riportate le istruzioni di script di cui è sconsigliato l'utilizzo in Qlik Sense.

Istruzioni di script non consigliate

| Istruzione           | Consiglio                                 |
|----------------------|-------------------------------------------|
| <b>Command</b>       | Utilizzare invece <b>SQL</b> .            |
| <b>CustomConnect</b> | Utilizzare invece <b>Custom Connect</b> . |

### 12.2 Parametri dell'istruzione di script non consigliati in Qlik Sense

In questa tabella sono riportati i parametri dell'istruzione di script di cui è sconsigliato l'utilizzo in Qlik Sense.

Parametri dell'istruzione di script non consigliati

| Istruzione    | Parametri                                                                                                                                                       |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Buffer</b> | Utilizzare <b>Incremental</b> invece di: <ul style="list-style-type: none"><li>• <b>Inc</b> (non consigliato)</li><li>• <b>Incr</b> (non consigliato)</li></ul> |

## 12 Funzioni e istruzioni non consigliate in Qlik Sense

| Istruzione                                                            | Parametri                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| LOAD                                                                  | <p>Le seguenti parole chiave di parametri vengono generate dai wizard di trasformazione dei file di QlikView. La funzionalità viene mantenuta quando i dai vengono ricaricati, ma Qlik Sense non fornisce supporto guidato/wizard per la generazione dell'istruzione con questi parametri:</p> <ul style="list-style-type: none"><li>• Bottom</li><li>• Cellvalue</li><li>• Col</li><li>• Colmatch</li><li>• Colsplit</li><li>• Colxtr</li><li>• Compound</li><li>• Contain</li><li>• Equal</li><li>• Every</li><li>• Expand</li><li>• Filters</li><li>• Intarray</li><li>• Interpret</li><li>• Length</li><li>• Longer</li><li>• Numerical</li><li>• Pos</li><li>• Remove</li><li>• Rotate</li></ul> |
| Sintassi dello script e funzioni grafiche - Qlik Sense, November 2025 | <ul style="list-style-type: none"><li>• Row</li><li>• Rowcnd</li><li>• Shorter</li></ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

## 12.3 Funzioni non consigliate in Qlik Sense

In questa tabella sono riportate le funzioni grafiche e di script di cui è sconsigliato l'utilizzo in Qlik Sense.

Funzioni non consigliate

| Funzione        | Consiglio                                                                                                                                                                                                              |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NumAvg          | Utilizzare invece le funzioni di scala.                                                                                                                                                                                |
| NumCount        | <a href="#">Funzioni di scala (page 1584)</a>                                                                                                                                                                          |
| NumMax          |                                                                                                                                                                                                                        |
| NumMin          |                                                                                                                                                                                                                        |
| NumSum          |                                                                                                                                                                                                                        |
| Color()         | Utilizzare invece altre funzioni colore. <b>QliktechBlue()</b> può essere sostituito da <b>RGB(8, 18, 90)</b> e <b>QliktechGray</b> può essere sostituito da <b>RGB(158, 148, 137)</b> per ottenere gli stessi colori. |
| QliktechBlue    |                                                                                                                                                                                                                        |
| QliktechGray    |                                                                                                                                                                                                                        |
|                 | <a href="#">Funzioni colore (page 612)</a>                                                                                                                                                                             |
| QlikViewVersion | Utilizzare invece <b>EngineVersion</b> .<br><br><a href="#">EngineVersion (page 1882)</a>                                                                                                                              |
| ProductVersion  | Utilizzare invece <b>EngineVersion</b> .<br><br><a href="#">EngineVersion (page 1882)</a>                                                                                                                              |
| QVUser          |                                                                                                                                                                                                                        |
| Year2Date       | Utilizzare invece <b>YearToDate</b> .                                                                                                                                                                                  |
| Vrank           | Utilizzare invece <b>Rank</b> .                                                                                                                                                                                        |
| WildMatch5      | Utilizzare invece <b>WildMatch</b> .                                                                                                                                                                                   |

### Qualificatore ALL

In QlikView, il qualificatore **ALL** può essere posizionato prima di un'espressione. Ciò equivale a utilizzare **{1} TOTAL**. In tal caso, il calcolo verrà effettuato in base a tutti i valori del campo nel documento, ignorando le dimensioni del grafico e le selezioni correnti. Viene restituito sempre lo stesso valore, indipendentemente dallo stato logico nel documento. Se si utilizza il qualificatore **ALL**, non è possibile utilizzare un'espressione set, perché tale qualificatore **ALL** definisce di per sé un gruppo. Per motivi di legacy, il qualificatore **ALL** è ancora valido anche in questa versione di Qlik Sense, ma potrebbe essere rimosso nelle prossime versioni.