# Scriptsyntaxis en diagramfuncties

Qlik Sense®
November 2025
Copyright © 1993-jjjj} QlikTech International AB. Alle rechten voorbehouden.

© jjjj QlikTech International AB. All rights reserved. All company and/or product names may be trade names, trademarks and/or registered trademarks of the respective owners with which they are associated.

# Contents


1 Wat is Qlik Sense? 16
1.1 Wat kunt u doen in Qlik Sense? 16
1.2 Hoe werkt Qlik Sense? 16
Het app-model 16
De associatieve ervaring 16
Samenwerking en mobiliteit 16
1.3 Hoe implementeert u Qlik Sense? 16
Qlik Sense Desktop 17
Qlik Sense Enterprise 17
1.4 Een Qlik Sense-site besturen en beheren 17
1.5 Qlik Sense uitbreiden en aanpassen voor uw eigen doeleinden 17
Uitbreidingen en mashups bouwen 17
Clients bouwen 17
Serverhulpmiddelen bouwen 17
Verbinding maken met andere gegevensbronnen 17
2 Overzicht scriptsyntaxis 18
2.1 Inleiding tot scriptsyntaxis 18
2.2 Wat is het Backus-Naur-formalisme? 18
3 Scriptinstructies en sleutelwoorden 20
3.1 Scriptbesturingsopdrachten 20
Overzicht van scriptbesturingsopdrachten 20
Call 22
Do..loop 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 Scriptprefixen 34
Overzicht van scriptprefixen 34
Add 39
Buffer 40
Concatenate 42
Crosstable 47
First 57
Generic 59
Hierarchy 66
HierarchyBelongsTo 68
Inner 70
IntervalMatch 71
Join 75
Keep 85

Left ... 86
Mapping ... 87
Merge ... 88
NoConcatenate ... 93
Only ... 102
Outer ... 102
Gedeeltelijke lading ... 103
Replace ... 107
Right ... 108
Sample ... 109
Semantic ... 113
Unless ... 117
When ... 123
3.3 Reguliere scriptopdrachten ... 129
Overzicht van normale scriptopdrachten ... 129
Alias ... 136
AutoNumber ... 137
Binary ... 141
Comment field ... 142
Comment table ... 143
Connect ... 144
Declare ... 146
Derive ... 149
Direct Query ... 150
Directory ... 155
Disconnect ... 156
Drop ... 156
Drop table ... 158
Execute ... 159
Field/Fields ... 160
FlushLog ... 160
Force ... 160
From ... 163
Load ... 163
Let ... 187
Loosen Table ... 187
Map ... 188
NullAsNull ... 189
NullAsValue ... 189
Qualify ... 190
Rem ... 191
Rename ... 192
Search ... 194
Section ... 195
Select ... 195
Set ... 198
Sleep ... 198
SQL ... 198

SQLColumns 199
SQLTables 200
SQLTypes 200
Star 202
Store 203
Table/Tables 211
Tag 211
Trace 212
Unmap 212
Unqualify 213
Untag 214
3.4 Werkdirectory 215
Qlik Sense Desktop werkdirectory 215
Qlik Sense werkdirectory 215
**4 Werken met variabelen in de editor voor het laden van gegevens 216**
4.1 Overzicht 216
4.2 Een variabele definiëren 216
Variabelen een naam geven 217
4.3 Een variabele verwijderen 217
4.4 De waarde van een variabele laden als een veldwaarde 217
4.5 Berekeningen met variabelen 217
4.6 Systeemvariabelen 218
Overzicht van systeemvariabelen 219
CreateSearchIndexOnReload 221
HidePrefix 222
HideSuffix 222
Include 223
OpenUrlTimeout 224
StripComments 224
Verbatim 225
4.7 Variabelen voor afhandeling van waarden 225
Overzicht van variabelen voor afhandeling van waarden 225
NullDisplay 226
NullInterpret 226
NullValue 226
OtherSymbol 227
4.8 Variabelen voor getalinterpretatie 227
Valutanotatie 227
Getalnotatie 228
Tijdnotatie 228
BrokenWeeks 229
DateFormat 231
DayNames 237
DecimalSep 241
FirstWeekDay 244
LongDayNames 248
LongMonthNames 251

MoneyDecimalSep 255
MoneyFormat 259
MoneyThousandSep 264
MonthNames 268
NumericalAbbreviation 274
ReferenceDay 274
ThousandSep 279
TimeFormat 285
TimestampFormat 286
4.9 Direct Discovery-variabelen 289
Direct Discovery-systeemvariabelen 289
Teradata-variabelen voor query banding 290
Direct Discovery-tekenvariabelen 291
Direct Discovery-variabelen voor getalinterpretatie 292
4.10 Foutvariabelen 293
Overzicht van foutvariabelen 293
ErrorMode 294
ScriptError 294
ScriptErrorCount 295
ScriptErrorList 295
**5 Scriptuitdrukkingen 297**
**6 Diagramuitdrukkingen 298**
6.1 Het aggregatiebereik definiëren 298
Voorbeeld: TOTAL-kwalificatie 299
Voorbeeld: set-analyse 300
Voorbeeld: TOTAL-kwalificatie en set-analyse ({1}-id) 300
Voorbeeld: TOTAL-kwalificatie en set-analyse (gegevens vergelijken met een referentiewaarde) 301
Voorbeeld: TOTAL-kwalificatie met uitzondering, en set-analyse 302
Gegevens die worden gebruikt in voorbeelden op deze pagina 303
6.2 Set-analyse 303
Ingestelde uitdrukkingen 304
Voorbeelden 305
Natuurlijke sets 305
Set-Id's 307
Set-operatoren 309
Set-modificaties 310
Vlag voor behoud van lege set 332
Binnenste en buitenste set-uitdrukkingen 333
Buitenste set-uitdrukkingen en uitbreidingen met dollarteken 338
Zelfstudie - Een set-uitdrukking maken 341
Syntaxis voor set-uitdrukkingen 350
6.3 Algemene syntaxis voor diagramuitdrukkingen 350
6.4 Algemene syntaxis voor aggregaties 351
**7 Operatoren 352**
7.1 Bit-operatoren 352
7.2 Logische operatoren 353

7.3 Numerieke operatoren 354
7.4 Relationele operatoren 354
7.5 Tekenreeksoperatoren 356
& 356
like 356
**8 Script- en diagramfuncties 358**
8.1 Analytische verbindingen voor uitbreidingen aan de serverzijde (SSE) 358
8.2 Aggregatiefuncties 358
Aggregatiefuncties gebruiken in het script voor het laden van gegevens 359
Aggregatiefuncties gebruiken in diagramuitdrukkingen 359
Hoe aggregaties worden berekend 359
Aggregatie van sleutelvelden 360
Basisaggregatiefuncties 361
Teller-aggregatiefuncties 398
Financiële aggregatiefuncties 427
Statistische aggregatiefuncties 456
Statistische testfuncties 530
Aggregatiefuncties voor tekenreeksen 600
Functies voor synthetische dimensies 613
Geneste aggregaties 616
8.3 Aggr - diagramfunctie 616
Voorbeelden: diagramuitdrukkingen met Aggr 619
8.4 Kleurfuncties 622
Vooraf gedefinieerde kleurfuncties 625
ARGB 626
RGB 631
HSL 637
8.5 Voorwaardenfuncties 642
Overzicht voorwaardenfuncties 642
alt 644
class 651
coalesce 656
if 660
match 667
mixmatch 675
pick 683
wildmatch 688
8.6 Tellerfuncties 696
Overzicht tellerfuncties 696
autonumber 697
autonumberhash128 699
autonumberhash256 701
IterNo 703
RecNo 704
RowNo 706
RowNo - diagramfunctie 707
8.7 Datum- en tijdfuncties 709

# Contents


Overzicht van datum- en tijdfuncties ... 710
addmonths ... 718
addyears ... 728
age ... 736
converttolocaltime ... 737
day ... 741
dayend ... 747
daylightsaving ... 756
dayname ... 756
daynumberofquarter ... 758
daynumberofyear ... 765
daystart ... 771
firstworkdate ... 779
GMT ... 780
hour ... 784
inday ... 788
indaytotime ... 796
inlunarweek ... 806
inlunarweektodate ... 818
inmonth ... 830
inmonths ... 838
inmonthstodate ... 852
inmonthtodate ... 865
inquarter ... 875
inquartertodate ... 888
inweek ... 901
inweektodate ... 918
inyear ... 932
inyeartodate ... 945
lastworkdate ... 958
localtime ... 968
lunarweekend ... 972
lunarweekname ... 984
lunarweekstart ... 996
makedate ... 1008
maketime ... 1015
makeweekdate ... 1022
minute ... 1031
month ... 1036
monthend ... 1043
monthname ... 1052
monthsend ... 1060
monthsname ... 1073
monthsstart ... 1087
monthstart ... 1100
networkdays ... 1110
now ... 1120
quarterend ... 1128

quartername 1141
quarterstart 1153
second 1165
setdateyear 1170
setdateyearmonth 1172
timezone 1174
today 1175
UTC 1181
week 1181
weekday 1197
weekend 1206
weekname 1218
weekstart 1233
weekyear 1245
year 1255
yearend 1262
yearname 1274
yearstart 1286
yeartodate 1298
8.8 Exponentiële en logaritmische functies 1314
8.9 Veldfuncties 1315
Tellerfuncties 1316
Veld- en selectiefuncties 1316
GetAlternativeCount - diagramfunctie 1317
GetCurrentSelections - diagramfunctie 1323
GetExcludedCount - diagramfunctie 1328
GetFieldSelections - diagramfunctie 1333
GetObjectDimension - diagramfunctie 1337
GetObjectField - diagramfunctie 1341
GetObjectMeasure - diagramfunctie 1345
GetPossibleCount - diagramfunctie 1347
GetSelectedCount - diagramfunctie 1351
GetStateCounts - diagramfunctie 1356
8.10 Bestandsfuncties 1361
Overzicht van bestandsfuncties 1361
Attribute 1363
ConnectString 1371
FileBaseName 1372
FileDir 1372
FileExtension 1373
FileName 1373
FilePath 1373
FileSize 1374
FileTime 1375
GetFolderPath 1375
QvdCreateTime 1376
QvdFieldName 1377
QvdNoOfFields 1378

QvdNoOfRecords 1379
QvdTableName 1380
8.11 Financiële functies 1381
Overzicht van financiële functies 1382
BlackAndSchole 1382
FV 1383
nPer 1384
Pmt 1385
PV 1386
Rate 1387
8.12 Opmaakfuncties 1388
Overzicht van opmaakfuncties 1388
ApplyCodepage 1390
Date 1391
Dual 1395
Interval 1400
Money 1404
Num 1409
Time 1415
Timestamp 1419
8.13 Algemene numerieke functies 1425
Overzicht van algemene numerieke functies 1425
Combinatie- en permutatiefuncties 1426
Modulofuncties 1426
Pariteitsfuncties 1426
Afrondingsfuncties 1427
BitCount 1427
Ceil 1427
Combin 1429
Div 1429
Even 1430
Fabs 1430
Fact 1430
Floor 1431
Fmod 1432
Frac 1433
Mod 1434
Odd 1434
Permut 1435
Round 1435
Sign 1437
8.14 Geospatiale functies 1437
Overzicht van geospatiale functies 1438
GeoAggrGeometry 1439
GeoBoundingBox 1440
GeoCountVertex 1441
GeoGetBoundingBox 1441
GeoGetPolygonCenter 1442

GeoInvProjectGeometry 1443
GeoMakePoint 1443
GeoProject 1444
GeoProjectGeometry 1445
GeoReduceGeometry 1445
8.15 Interpretatiefuncties 1447
Overzicht van interpretatiefuncties 1447
Date# 1448
Interval# 1452
Money# 1455
Num# 1460
Text 1465
Time# 1468
Timestamp# 1472
8.16 Interrecord-functies 1476
Rijfuncties 1476
Kolomfuncties 1477
Veldfuncties 1477
Draaitabelfuncties 1478
Interrecord-functies in het load-script voor gegevens 1479
Above - diagramfunctie 1479
Below - diagramfunctie 1493
Bottom - diagramfunctie 1503
Column - diagramfunctie 1510
Dimensionality - diagramfunctie 1515
Exists 1520
FieldIndex 1524
FieldValue 1525
FieldValueCount 1527
LookUp 1529
NoOfRows 1531
Peek 1533
Previous 1540
Top - diagramfunctie 1542
SecondaryDimensionality - diagramfunctie 1549
After - diagramfunctie 1553
Before - diagramfunctie 1558
First - diagramfunctie 1562
Last - diagramfunctie 1568
ColumnNo - diagramfunctie 1574
NoOfColumns - diagramfunctie 1578
8.17 Logische functies 1581
8.18 Toewijzingsfuncties 1582
Overzicht toewijzingsfuncties 1582
ApplyMap 1583
MapSubstring 1584
8.19 Rekenkundige functies 1586
8.20 NULL-functies 1587

Overzicht NULL-functies ... 1587
EmptyIsNull ... 1587
IsNull ... 1592
NULL ... 1596
8.21 Bereikfuncties ... 1600
Elementaire bereikfuncties ... 1600
Bereikfuncties voor tellers ... 1601
Statistische bereikfuncties ... 1602
Financiële bereikfuncties ... 1602
RangeAvg ... 1603
RangeCorrel ... 1605
RangeCount ... 1608
RangeFractile ... 1610
RangeIRR ... 1612
RangeKurtosis ... 1614
RangeMax ... 1614
RangeMaxString ... 1617
RangeMin ... 1618
RangeMinString ... 1620
RangeMissingCount ... 1622
RangeMode ... 1623
RangeNPV ... 1626
RangeNullCount ... 1627
RangeNumericCount ... 1629
RangeOnly ... 1630
RangeSkew ... 1631
RangeStdev ... 1632
RangeSum ... 1634
RangeTextCount ... 1636
RangeXIRR ... 1638
RangeXNPV ... 1639
8.22 Relationele functies ... 1642
Rangordefuncties ... 1642
Clusterfuncties ... 1642
Ontledingsfuncties voor tijdreeksen ... 1644
Rank - diagramfunctie ... 1644
HRank - diagramfunctie ... 1649
Optimaliseren met k-means: een voorbeeld uit de praktijk ... 1651
KMeans2D - diagramfunctie ... 1660
KMeansND - diagramfunctie ... 1675
KMeansCentroid2D - diagramfunctie ... 1690
KMeansCentroidND - diagramfunctie ... 1691
STL_Trend - diagramfunctie ... 1692
STL_Seasonal - diagramfunctie ... 1694
STL_Residual - diagramfunctie ... 1696
Tutorial: Uitsplitsing van tijdreeksen in Qlik Sense ... 1698
8.23 Statistische verdelingsfuncties ... 1702
Overzicht van statistische verdelingsfuncties ... 1703

BetaDensity 1705
BetaDist 1706
BetaInv 1706
BinomDist 1707
BinomFrequency 1707
BinomInv 1708
ChiDensity 1708
ChiDist 1709
ChiInv 1709
FDensity 1710
FDist 1710
FInv 1711
GammaDensity 1712
GammaDist 1712
GammaInv 1713
NormDist 1713
NormInv 1714
PoissonDist 1715
PoissonFrequency 1715
PoissonInv 1716
TDensity 1716
TDist 1716
TInv 1717
8.24 Tekenreeksfuncties 1718
Overzicht van tekenreeksfuncties 1718
Werken met reguliere expressies in load-scripts en diagramuitdrukkingen 1723
Capitalize 1725
Chr 1728
CountRegEx 1730
Evaluate 1736
ExtractRegEx 1740
ExtractRegExGroup 1746
FindOneOf 1755
Hash128 1759
Hash160 1764
Hash256 1769
Index 1774
IndexRegEx 1777
IndexRegExGroup 1781
IsJson 1786
IsRegEx 1790
JsonArray 1794
JsonGet 1795
JsonObject 1797
JsonSet 1798
JsonSetEx 1800
KeepChar 1801
Left 1805

Len 1809
LevenshteinDist 1813
Lower 1817
LTrim 1820
MatchRegEx 1824
Mid 1828
Ord 1832
PurgeChar 1835
Repeat 1838
Replace 1842
ReplaceRegEx 1845
ReplaceRegExGroup 1851
Right 1857
RTrim 1860
SubField 1864
SubFieldRegEx 1870
SubStringCount 1874
TextBetween 1880
Trim 1884
Upper 1888
8.25 Systeemfuncties 1892
Overzicht van systeemfuncties 1892
CalcDim - diagramfunctie 1895
EngineVersion 1902
Systeemfuncties 1902
GetSysAttr 1905
GetUserAttr 1906
InObject - diagramfunctie 1907
IsPartialReload 1912
ObjectId - diagramfunctie 1912
ProductVersion 1915
StateName - diagramfunctie 1916
8.26 Tabelfuncties 1916
Overzicht van tabelfuncties 1916
FieldName 1918
FieldNumber 1919
NoOfFields 1919
NoOfRows 1920
8.27 Trigonometrische en hyperbolische functies 1920
8.28 Vensterfuncties 1922
Window 1923
WRank 1932
**9 Beperking van toegang tot bestandssysteem 1938**
9.1 Beveiligingsaspecten bij verbinding met op bestanden gebaseerde ODBC- en OLE DB-gegevensverbindingen 1938
9.2 Beperkingen in standaardmodus 1938
Systeemvariabelen 1939

Reguliere scriptopdrachten ... 1940
Scriptbesturingsopdrachten ... 1942
Bestandsfuncties ... 1942
Systeemfuncties ... 1944
9.3 Standaardmodus uitschakelen ... 1945
Qlik Sense ... 1945
Qlik Sense Desktop ... 1945
**10 Scripts op diagramniveau ... 1946**
10.1 Scriptbesturingsopdrachten ... 1946
Overzicht besturingsopdracht diagramaanpassing ... 1946
Call ... 1948
Do..lus ... 1949
End ... 1950
Exit ... 1950
Exit script ... 1950
For..next ... 1951
For each..next ... 1952
If..then..elseif..else..end if ... 1955
Next ... 1956
Sub..end sub ... 1957
Switch..case..default..end switch ... 1958
To ... 1959
10.2 Prefixen ... 1959
Overzicht prefixen diagramaanpassing ... 1959
Add ... 1960
Replace ... 1960
10.3 Reguliere scriptopdrachten ... 1961
Overzicht besturingsopdracht diagramaanpassing ... 1961
Load ... 1962
Let ... 1966
Set ... 1967
Put ... 1967
HCValue ... 1968
**11 QlikView 12-functies en -opdrachten die niet worden ondersteund in Qlik Sense ... 1970**
11.1 Scriptopdrachten die niet worden ondersteund in Qlik Sense ... 1970
11.2 Functies die niet worden ondersteund in Qlik Sense ... 1970
11.3 Prefixes die niet worden ondersteund in Qlik Sense ... 1970
**12 Functies en opdrachten die niet worden aanbevolen in Qlik Sense ... 1971**
12.1 Scriptopdrachten die niet worden aanbevolen in Qlik Sense ... 1971
12.2 Parameters voor scriptopdrachten die niet worden aanbevolen in Qlik Sense ... 1971
12.3 Functies die niet worden aanbevolen in Qlik Sense ... 1973
Kwalificatie ALL ... 1973

# 1 Wat is Qlik Sense?

Qlik Sense is een platform voor gegevensanalyse. Met Qlik Sense kunt u gegevens analyseren en zelf ontdekkingen doen in gegevens. U kunt kennis uitwisselen en gegevens analyseren in groepen en tussen organisaties. Met Qlik Sense kunt u vragen stellen en uw eigen vragen beantwoorden en uw eigen paden naar inzicht volgen. Met Qlik Sense kunnen u en uw collega's samen tot besluiten komen.

## 1.1 Wat kunt u doen in Qlik Sense?

De meeste BI-producten (Business Intelligence) kunnen u helpen vragen te beantwoorden die vooraf duidelijk zijn. Maar hoe zit het met uw vervolgvragen? De vragen die opkomen nadat iemand uw rapport heeft gelezen of uw visualisatie heeft gezien? Met de associatieve ervaring van Qlik Sense kunt u vraag na vraag na vraag beantwoorden en uw eigen pad naar inzicht volgen. Met Qlik Sense kunt u vrijelijk uw gegevens verkennen, met slechts enkele klikken, en bij elke stap meer te weten komen en de volgende stappen bedenken op basis van eerdere bevindingen.

## 1.2 Hoe werkt Qlik Sense?

Qlik Sense genereert weergaven van informatie voor u. Qlik Sense vereist geen vooraf gedefinieerde en statische rapporten en maakt u niet afhankelijk van andere gebruikers - alles wat u doet is simpelweg klikken en leren. Telkens wanneer u klikt, reageert Qlik Sense direct en werkt elke Qlik Sense-visualisatie en weergave in de app bij met de nieuw berekende set gegevens en visualisaties die specifiek zijn voor uw selecties.

### Het app-model

In plaats van enorme bedrijfstoepassingen te implementeren en beheren, kunt u uw eigen Qlik Sense-apps maken die u kunt hergebruiken, wijzigen en delen met anderen. Het app-model helpt u zelf de volgende vraag te stellen en te beantwoorden, zonder terug te hoeven gaan naar een specialist voor een nieuw rapport of een nieuwe visualisatie.

### De associatieve ervaring

Qlik Sense beheert automatisch alle relaties in de gegevens en presenteert informatie met behulp van een **green/white/gray** metafoor. Selecties worden groen gemarkeerd, bijbehorende gegevens worden in wit weergegeven en uitgesloten (gekoppelde) gegevens in grijs. Deze directe feedback stelt u in staat om na te denken over nieuwe vragen en om verder te verkennen en te ontdekken.

### Samenwerking en mobiliteit

Qlik Sense stelt u verder in staat samen te werken met collega's op welk tijdstip dan ook en waar deze zich ook bevinden. Alle Qlik Sense-mogelijkheden, met inbegrip van de associatieve ervaring en samenwerking, zijn beschikbaar op mobiele apparaten. Met Qlik Sense kunt u vragen stellen en uw eigen vragen en vervolgvragen beantwoorden, samen met uw collega's, waar u ook bent.

## 1.3 Hoe implementeert u Qlik Sense?

Er zijn twee versies van Qlik Sense om te implementeren, Qlik Sense Desktop en Qlik Sense Enterprise.

### Qlik Sense Desktop

Dit is een eenvoudig te installeren versie voor één gebruiker die gewoonlijk op een lokale computer wordt geïnstalleerd.

### Qlik Sense Enterprise

Deze versie wordt gebruikt voor het implementeren van Qlik Sense-sites. Een site is een verzameling van één of meer servers die zijn verbonden met een gemeenschappelijk logische bibliotheek of een centraal knooppunt.

## 1.4 Een Qlik Sense-site besturen en beheren

Met de Qlik Beheerconsole kunt u Qlik Sense-sites op een eenvoudige en intuïtieve wijze beheren en bewaken. U kunt licenties, toegangs- en beveiligingsregels beheren, knooppunten en verbindingen met gegevensbronnen configureren en inhoud en gebruikers synchroniseren, naast nog vele andere activiteiten en hulpbronnen.

## 1.5 Qlik Sense uitbreiden en aanpassen voor uw eigen doeleinden

Qlik Sense biedt u flexibele API's en SDK's voor het ontwikkelen van uw eigen uitbreidingen en voor het aanpassen en integreren van Qlik Sense voor verschillende doeleinden, zoals:

### Uitbreidingen en mashups bouwen

Hier kunt u aan webontwikkeling doen met behulp van JavaScript om uitbreidingen te bouwen die aangepaste visualisaties vormen in Qlik Sense-apps of u gebruikt de API's van een mashup om websites te bouwen met Qlik Sense-inhoud.

### Clients bouwen

U kunt clients bouwen in .NET en Qlik Sense-objecten insluiten in uw eigen toepassingen. U kunt ook native clients bouwen in elke programmeertaal die WebSocket-communicatie kan verwerken met behulp van het Qlik Sense-clientprotocol.

### Serverhulpmiddelen bouwen

Met service- en gebruikersdirectory-API's kunt u uw eigen hulpmiddelen bouwen voor het besturen en beheren van Qlik Sense-sites.

### Verbinding maken met andere gegevensbronnen

Maak Qlik Sense-connectoren voor het ophalen van gegevens uit aangepaste gegevensbronnen.

# 2 Overzicht scriptsyntaxis

## 2.1 Inleiding tot scriptsyntaxis

In een script worden de namen gedefinieerd van de desbetreffende gegevensbron, tabellen en velden die bij de logica betrokken zijn. Daarnaast worden de velden in de definitie van de toegangsrechten in het script gedefinieerd. Een script bestaat uit een aantal opdrachten die achter elkaar worden uitgevoerd.

De syntaxis van de Qlik Sense-opdrachtregel en de scriptsyntaxis worden beschreven in een notatie die het Backus-Naur-formalisme of BNF-code wordt genoemd.

Als een nieuw Qlik Sense-bestand wordt gemaakt, worden de eerste regels met code automatisch gegenereerd. De standaardwaarden van deze variabelen voor getalinterpretatie worden afgeleid van de landinstellingen in het besturingssysteem.

Het script bestaat uit een aantal scriptopdrachten en sleutelwoorden die achter elkaar worden uitgevoerd. Alle scriptopdrachten moeten met een puntkomma **";"** worden afgesloten.

U kunt uitdrukkingen en functies gebruiken in de **LOAD**-opdrachten om de geladen gegevens te transformeren.

Voor een tabelbestand met komma's, tabs of puntkomma's als scheidingsteken kan een **LOAD**-opdracht worden gebruikt. Met een **LOAD**-opdracht worden standaard alle velden in het bestand geladen.

Algemene databases kunnen worden geopend via ODBC- of OLE DB-databaseconnectoren. Daar worden standaard SQL-opdrachten voor gebruikt. De juiste SQL-syntaxis kan per ODBC-stuurprogramma verschillen.

Bovendien kunt u andere gegevensbronnen openen via aangepaste connectoren.

## 2.2 Wat is het Backus-Naur-formalisme?

De syntaxis van de Qlik Sense-opdrachtregel en de scriptsyntaxis worden beschreven in een notatie die het Backus-Naur-formalisme of BNF-code wordt genoemd.

De volgende tabel biedt een lijst met symbolen die worden gebruikt in BNF-code, met een beschrijving van hoe zij worden geïnterpreteerd:

Symbolen

| Symbool | Beschrijving |
|---|---|
| \| | Logische OR: het symbool aan beide zijden kan worden gebruikt. |
| ( ) | Haakjes geven prioriteit aan: ze worden gebruikt om de BNF-syntaxis te structureren. |
| [ ] | Vierkante haken: items tussen vierkante haakjes zijn optioneel. |
| { } | Accolades: items tussen accolades kunnen nul of meer keren worden herhaald. |

| Symbool | Beschrijving |
|---|---|
| Symbool | Een niet-eindige syntactische categorie: een categorie die verder kan worden opgesplitst in andere symbolen. Bijvoorbeeld samenstellingen van de bovenstaande, andere niet-terminal symbolen, teksttekenreeksen, enzovoort. |
| ::= | Geeft het begin aan van een blok dat een symbool definieert. |
| **LOAD** | Een eindsymbool dat bestaat uit een teksttekenreeks. Moet precies zoals aangegeven worden geschreven in het script. |

Alle eindsymbolen worden afdrukt in een **bold face** lettertype. Bijvoorbeeld: "(" moet worden geïnterpreteerd als een haakje dat prioriteit aangeeft, terwijl "**(**" moet worden geïnterpreteerd als een teken dat moet worden afgedrukt in het script.

**Voorbeeld:**

De omschrijving van de alias-opdracht is:

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

Dit moet worden geïnterpreteerd als de teksttekenreeks "alias", gevolgd door een willekeurige veldnaam, gevolgd door de teksttekenreeks "as", gevolgd door een willekeurige aliasnaam. Elk aantal aanvullende combinaties van "fieldname as alias" kan worden gegeven, gescheiden door komma's.

De volgende opdrachten zijn correct:

```
alias a as first;

alias a as first, b as second;

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

De volgende opdrachten zijn niet correct:

```
alias a as first b as second;

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

# 3 Scriptinstructies en sleutelwoorden

Het Qlik Sense-script bestaat uit een aantal opdrachten. Een opdracht kan een normale scriptopdracht of een scriptbesturingsopdracht zijn. Bepaalde opdrachten kunnen worden voorafgegaan door prefixen.

Normale opdrachten worden over het algemeen gebruikt om gegevens op een bepaalde manier te manipuleren. Deze opdrachten kunnen worden geschreven op een willekeurig aantal regels in het script en moeten altijd worden afgesloten met een puntkomma (;).

Besturingsopdrachten worden gebruikt om het verloop van de scriptuitvoering te besturen. Elke zin van een besturingsopdracht moet op één scriptregel blijven en kan worden afgesloten met een puntkomma of een regeleinde.

Prefixen kunnen worden toegevoegd aan normale opdrachten, maar nooit aan besturingsopdrachten. De prefixen **when** en **unless** kunnen wel als suffix bij enkele specifieke clausules van besturingsopdrachten worden gebruikt.

In de volgende sectie vindt u een alfabetisch overzicht van alle scriptopdrachten, besturingsopdrachten en prefixen.

Bij het typen van sleutelwoorden in het script kunt u elke combinatie van hoofdletters en kleine letters gebruiken. Gebruikte namen van velden en variabelen in de opdrachten zijn echter hoofdlettergevoelig.

## 3.1 Scriptbesturingsopdrachten

Het Qlik Sense-script bestaat uit een aantal opdrachten. Een opdracht kan een normale scriptopdracht of een scriptbesturingsopdracht zijn.

Besturingsopdrachten worden gebruikt om het verloop van de scriptuitvoering te besturen. Elke zin van een besturingsopdracht moet op één scriptregel blijven en kan worden afgesloten met een puntkomma of een regeleinde.

Prefixen zijn nooit van toepassing op besturingsopdrachten, met uitzondering van de prefixen **when** en **unless**, die bij een aantal specifieke besturingsopdrachten kunnen worden gebruikt.

Bij het typen van sleutelwoorden in het script kunt u elke combinatie van hoofdletters en kleine letters gebruiken.

### Overzicht van scriptbesturingsopdrachten

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

#### Call

De besturingsopdracht **call** roept een subroutine aan die door een vorige **sub**-opdracht is gedefinieerd.

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

### Do..loop

De besturingsopdracht **do..loop** is een herhalingsconstructie voor gebruik in scripts waarmee één of meer opdrachten worden uitgevoerd totdat aan een logische voorwaarde wordt voldaan.

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

### Exit script

Met deze besturingsopdracht wordt de uitvoering van het script stopgezet. Deze opdracht kan overal in het script worden ingevoegd.

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

### For each ..next

De besturingsopdracht **for each..next** is een herhalingsconstructie voor gebruik in scripts waarmee een of meer opdrachten worden uitgevoerd voor elke waarde in een door komma's gescheiden lijst. De opdrachten in de lus tussen **for** en **next** worden uitgevoerd voor elke waarde van de lijst.

```
For each..next var in list

[statements]

[exit for [ ( when | unless ) condition ]

[statements]

next [var]
```

### For..next

De besturingsopdracht **for..next** is een herhalingsconstructie voor gebruik in scripts met een teller. De opdrachten in de lus tussen **for** en **next** worden uitgevoerd voor elke waarde van de tellervariabele tussen de opgegeven onder- en bovengrens.

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

[statements]

[exit for [ ( when | unless ) condition ]

[statements]

Next [counter]
```

### If..then

De besturingsopdracht **if..then** is een constructie voor scriptselectie waarmee de uitvoering van het script, afhankelijk van één of meer logische voorwaarden, op verschillende manieren kan verlopen.

*Omdat de instructie **if..then** een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag geen van de vier mogelijke clausules (**if..then**, **elseif..then**, **else** en **end if**) de regelgrens overschrijden.*

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

  [ statements ]

{ elseif condition then

  [ statements ] }

[ else

  [ statements ] ]

end if
```

### Sub

De besturingsopdracht **sub..end sub** geeft een subroutine aan die vanuit een **call**-opdracht kan worden aangeroepen.

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

### Switch

De besturingsopdracht **switch** is een constructie voor scriptselectie waarmee de uitvoering van het script, afhankelijk van de waarde van een uitdrukking, op verschillende manieren kan verlopen.

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

## Call

De besturingsopdracht **call** roept een subroutine aan die door een vorige **sub**-opdracht is gedefinieerd.

### Syntaxis:

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

### Argumenten:

Argumenten

| Argument | Beschrijving |
|---|---|
| name | De naam van de subroutine. |

| Argument | Beschrijving |
|---|---|
| paramlist | Een door komma's gescheiden lijst met de werkelijke parameters die worden gezonden aan de subroutine. Elk item in de lijst kan een veldnaam, een variabele of een willekeurige uitdrukking zijn. |

De subroutine die wordt opgeroepen door een **call**-opdracht moet op een eerdere positie in het script door een **sub** zijn gedefinieerd.

Parameters worden gekopieerd naar de subroutine en worden, als de parameter in de **call**-instructie een variabele is en geen uitdrukking, bij het beëindigen van de subroutine weer teruggekopieerd.

**Beperkingen:**

- Aangezien de **call**-opdracht een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag de opdracht niet op meerdere regels staan.
- Als u een subroutine definieert met `Sub..end sub` binnen een besturingsopdracht, bijvoorbeeld `if..then`, kunt u de subroutine alleen oproepen binnen dezelfde besturingsopdracht.

**Voorbeeld:**

In dit voorbeeld worden alle aan Qlik gerelateerde bestanden in een map en in de bijbehorende submap opgesomd en wordt bestandsinformatie opgeslagen in een tabel. Er wordt vanuit gegaan dat u een gegevensverbinding hebt gemaakt met de naam Apps in de map.

De subroutine DoDir wordt aangeroepen met de verwijzing naar de map , 'lib://Apps', als parameter. Binnen de subroutine, is er een recursieve aanroep, `Call DoDir (Dir)`, waarmee de functie recursief naar bestanden kijkt in submappen.

```
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')
```

## Do..loop

De besturingsopdracht **do..loop** is een herhalingsconstructie voor gebruik in scripts waarmee één of meer opdrachten worden uitgevoerd totdat aan een logische voorwaarde wordt voldaan.

**Syntaxis:**

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

*Omdat de instructie **do..loop** een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag geen van de drie mogelijke clausules ervan (**do**, **exit do** en **loop**) de regelgrens overschrijden.*

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| condition | Een logische uitdrukking die resulteert in True of False. |
| statements | Een willekeurige groep van een of meer Qlik Sense-scriptopdrachten. |
| while / until | De voorwaardelijke clausule **while** of **until** mag slechts eenmaal voorkomen in elke **do..loop**-opdracht, namelijk na **do** of na **loop**. Elke voorwaarde wordt alleen de eerste keer geïnterpreteerd wanneer deze wordt aangetroffen, maar wordt telkens in de lus geëvalueerd. |
| exit do | Als een clausule **exit do** binnen de lus wordt aangetroffen, gaat de uitvoering van het script over op de eerste opdracht na de clausule **loop** die het einde van de lus aangeeft. De clausule **exit do** kan voorwaardelijk worden gemaakt met het optionele suffix **when** of **unless**. |

**Voorbeeld:**

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

Set a=1;

Do while a<10

LOAD * from file$(a).csv;

Let a=a+1;

Loop
```

## End

Het sleutelwoord **End** voor scripts wordt gebruikt om **If**-, **Sub**- en **Switch**-clausules af te sluiten.

## Exit

Het sleutelwoord **Exit** voor scripts maakt onderdeel uit van de **Exit Script**-opdracht, maar kan ook worden gebruikt voor het afsluiten van **Do**-, **For**- of **Sub**-clausules.

## Exit script

Met deze besturingsopdracht wordt de uitvoering van het script stopgezet. Deze opdracht kan overal in het script worden ingevoegd.

**Syntaxis:**

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

Aangezien de **exit script**-opdracht een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag de opdracht niet op meerdere regels staan.

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| condition | Een logische uitdrukking die resulteert in True of False. |
| when / unless | De opdracht **exit script** kan voorwaardelijk worden gemaakt met de optionele clausule **when** of **unless**. |

**Voorbeelden:**

```
//Exit script
Exit Script;

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

## For..next

De besturingsopdracht **for..next** is een herhalingsconstructie voor gebruik in scripts met een teller. De opdrachten in de lus tussen **for** en **next** worden uitgevoerd voor elke waarde van de tellervariabele tussen de opgegeven onder- en bovengrens.

**Syntaxis:**

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

```
[statements]

[exit for [ ( when | unless ) condition ]

[statements]

Next [counter]
```

De uitdrukkingen *expr1*, *expr2* en *expr3* worden alleen de eerste keer dat de lus wordt ingevoerd geëvalueerd. De waarde van de tellervariabele kan worden gewijzigd door opdrachten binnen de lus. Dit is echter met het oog op de gebruikelijke manier van programmeren niet aan te raden.

Als een clausule **exit for** binnen de lus wordt aangetroffen, gaat de uitvoering van het script over op de eerste opdracht na de clausule **next** die het einde van de lus aangeeft. De clausule **exit for** kan voorwaardelijk worden gemaakt met het optionele suffix **when** of **unless**.

*Omdat de instructie **for..next** een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag geen van de drie mogelijke clausules ervan (**for..to..step**, **exit for** en **next**) de regelgrens overschrijden.*

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| counter | De naam van een variabele. Als *counter* wordt opgegeven na **next** moet de variabele dezelfde naam hebben als de variabele die wordt gevonden na de overeenkomende **for**. |
| expr1 | Een uitdrukking die de eerste waarde van de variabele *counter* bepaalt waarvoor de lus wordt uitgevoerd. |
| expr2 | Een uitdrukking die de laatste waarde van de variabele *counter* bepaalt waarvoor de lus wordt uitgevoerd. |
| expr3 | Een uitdrukking die de waarde bepaalt van de stap waarmee de variabele *counter* toeneemt bij elke uitvoering van de lus. |
| condition | Een logische uitdrukking die resulteert in True of False. |
| statements | Een willekeurige groep van een of meer Qlik Sense-scriptopdrachten. |

**Example 1: Een reeks van bestanden laden**

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

for a=1 to 9

      LOAD * from file$(a).csv;

next
```

#### Example 2: Een willekeurig aantal bestanden laden

In dit voorbeeld gaan we ervan uit dat we de gegevensbestanden *x1.csv, x3.csv, x5.csv, x7.csv* en *x9.csv* hebben. Het laden wordt bij een willekeurig punt gestopt met de voorwaarde `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

De besturingsopdracht **for each..next** is een herhalingsconstructie voor gebruik in scripts waarmee een of meer opdrachten worden uitgevoerd voor elke waarde in een door komma's gescheiden lijst. De opdrachten in de lus tussen **for** en **next** worden uitgevoerd voor elke waarde van de lijst.

#### Syntaxis:

Door een speciale syntaxis kunnen lijsten worden gegenereerd met namen van bestanden en mappen in de huidige map.

```
for each var in list

[statements]

[exit for [ ( when | unless ) condition ]

[statements]

next [var]
```

#### Argumenten:

Argumenten

| Argument | Beschrijving |
|---|---|
| var | De naam van een scriptvariabele die een nieuwe waarde uit de lijst ophaalt, telkens wanneer de lus wordt uitgevoerd. Als **var** wordt opgegeven na **next** moet de variabele dezelfde naam hebben als de variabele die wordt gevonden na de overeenkomende **for each**. |

De waarde van de **var** kan worden gewijzigd door opdrachten binnen de lus. Dit is echter met het oog op de gebruikelijke manier van programmeren niet aan te raden.

Als een clausule **exit for** binnen de lus wordt aangetroffen, gaat de uitvoering van het script over op de eerste opdracht na de clausule **next** die het einde van de lus aangeeft. De clausule **exit for** kan voorwaardelijk worden gemaakt met het optionele suffix **when** of **unless**.

> *Omdat de instructie **for each..next** een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag geen van de drie mogelijke clausules ervan (**for each**, **exit for** en **next**) de regelgrens overschrijden.*

**Syntaxis:**

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

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

Argumenten

| Argument | Beschrijving |
|---|---|
| constant | Elk getal of elke tekenreeks. Een tekenreeks die rechtstreeks in het script wordt opgenomen moet tussen enkele aanhalingstekens worden geplaatst. Een tekenreeks zonder enkele aanhalingstekens wordt geïnterpreteerd als een variabele en vervolgens wordt de waarde van de variabele gebruikt. Getallen hoeven niet tussen enkele aanhalingstekens te worden geplaatst. |
| expression | Een willekeurige uitdrukking. |
| mask | Een masker voor de naam van een bestand of map die geldige tekens voor bestandsnamen plus de standaardjokertekens, ***** en **?**, kan bevatten.<br><br>U kunt absolute bestandspaden of paden lib:// gebruiken. |
| condition | Een logische uitdrukking die resulteert in True of False. |
| statements | Een willekeurige groep van een of meer Qlik Sense-scriptopdrachten. |
| filelist mask | Deze syntaxis produceert een lijst met door komma's gescheiden namen van alle bestanden in de huidige map die overeenkomen met het bestandsnaammasker.<br><br>*Dit argument ondersteunt alleen bibliotheekverbindingen in de standaardmodus.* |

| Argument | Beschrijving |
|---|---|
| dirlist mask | Deze syntaxis produceert een lijst met door komma's gescheiden namen van alle mappen in de huidige map die overeenkomen met het mapnaammasker.<br><br>*Dit argument ondersteunt alleen bibliotheekverbindingen in de standaardmodus.* |
| fieldvaluelist mask | Deze syntaxis itereert door de waarden van een veld die al in Qlik Sense zijn geladen. |

*De Qlik Connectoren van internetopslagprovider en andere DataFiles-verbindingen bieden geen ondersteuning voor filtermaskeringen die jokertekens (* en ?) gebruiken.*

### Example 1: Een lijst van bestanden laden

```
// 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: Een lijst van bestanden maken op schijf

Via dit voorbeeld wordt een lijst met alle aan Qlik Sense gerelateerde bestanden in een map geladen.

```
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: Itereren door de waarden van een veld

Dit voorbeeld itereert door de waarden van geladen waarden van FIELD en genereert een nieuw veld, NEWFIELD. Voor elke waarde van FIELD, worden twee NEWFIELD-records gemaakt.

```
load * inline [
FIELD
one
two
three
];

FOR Each a in FieldValueList('FIELD')
LOAD '$(a)' &'-'&RecNo() as NEWFIELD AutoGenerate 2;
NEXT a
```

De resulterende tabel ziet er als volgt uit:

Example table

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

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

De besturingsopdracht **if..then** is een constructie voor scriptselectie waarmee de uitvoering van het script, afhankelijk van één of meer logische voorwaarden, op verschillende manieren kan verlopen.

Besturingsopdrachten worden gebruikt om het verloop van de scriptuitvoering te besturen. Gebruik in een diagramuitdrukking in plaats daarvan de **if** voorwaardelijke functie.

#### Syntaxis:

```
If  condition then

  [ statements ]

{ elseif condition then

  [ statements ] }

[ else
```

```
  [ statements ] ]

end if
```

Omdat de instructie **if..then** een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag geen van de vier mogelijke clausules (**if..then**, **elseif..then**, **else** en **end if**) de regelgrens overschrijden.

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| condition | Een logische uitdrukking die resulteert in True of False. |
| statements | Een willekeurige groep van een of meer Qlik Sense-scriptopdrachten. |

**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

Het sleutelwoord **Next** voor scripts wordt gebruikt om**For**-lussen af te sluiten.

## Sub..end sub

De besturingsopdracht **sub..end sub** geeft een subroutine aan die vanuit een **call**-opdracht kan worden aangeroepen.

**Syntaxis:**

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

De argumenten worden in de subroutine gekopieerd en, als de corresponderende parameter in de **call**-opdracht een variabelenaam is, weer teruggekopieerd bij het verlaten van de subroutine.

Als een subroutine meer formele parameters heeft dan er aan daadwerkelijke parameters door een **call**-opdracht worden doorgegeven, worden de overtollige parameters op NULL ingesteld en kunnen ze als lokale variabelen binnen de subroutine worden gebruikt.

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| name | De naam van de subroutine. |
| paramlist | Een door komma's gescheiden lijst variabelen voor de formele parameters van de subroutine. Deze kunnen worden gebruikt als willekeurige variabelen binnen de subroutine. |
| statements | Een willekeurige groep van een of meer Qlik Sense-scriptopdrachten. |

**Beperkingen:**

- Omdat de instructie **sub** een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag geen van de twee mogelijke clausules (**sub** en **end sub**) de regelgrens overschrijden.
- Als u een subroutine definieert met `Sub..end sub` binnen een besturingsopdracht, bijvoorbeeld `if..then`, kunt u de subroutine alleen oproepen binnen dezelfde besturingsopdracht.

**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: - parameteroverdracht**

```
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)
```

Het resultaat van het bovenstaande is dat lokaal, binnen de subroutine, A wordt geïnitialiseerd op 1, B op 4 en C op NULL.

Bij het verlaten van de subroutine krijgt de globale variabele A de waarde 2 (teruggekopieerd van de subroutine). De tweede werkelijke parameter “(X+1)*2” wordt niet teruggekopieerd omdat dit geen variabele is. Tot slot verandert de globale variabele C niet door de aanroep van de subroutine.

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

De besturingsopdracht **switch** is een constructie voor scriptselectie waarmee de uitvoering van het script, afhankelijk van de waarde van een uitdrukking, op verschillende manieren kan verlopen.

**Syntaxis:**

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

> *Omdat de instructie **switch** een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag geen van de vier mogelijke clausules (**switch**, **case**, **default** en **end switch**) de regelgrens overschrijden.*

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| expression | Een willekeurige uitdrukking. |
| valuelist | Een door komma's gescheiden lijst waarden waarmee de waarde van de uitdrukking zal worden vergeleken. Uitvoering van het script wordt vervolgd met de opdrachten in de eerste groep waarvoor een waarde in valuelist is gevonden die gelijk is aan de waarde in de uitdrukking. Elke waarde in valuelist kan een willekeurige uitdrukking zijn. Als in geen van de **case**-opdrachten een overeenkomst is gevonden, worden de opdrachten achter **default**, indien gespecificeerd, uitgevoerd. |
| statements | Een willekeurige groep van een of meer Qlik Sense-scriptopdrachten. |

**Voorbeeld:**

```
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

Het sleutelwoord **To** voor scripts wordt gebruikt in verschillende scriptopdrachten.

## 3.2 Scriptprefixen

Prefixen kunnen worden toegevoegd aan normale opdrachten, maar nooit aan besturingsopdrachten. De prefixen **when** en **unless** kunnen wel als suffix bij enkele specifieke clausules van besturingsopdrachten worden gebruikt.

Bij het typen van sleutelwoorden in het script kunt u elke combinatie van hoofdletters en kleine letters gebruiken. Gebruikte namen van velden en variabelen in de opdrachten zijn echter hoofdlettergevoelig.

### Overzicht van scriptprefixen

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

**Add**

Het voorvoegsel **Add** kan aan elke **LOAD**- of **SELECT**-opdracht in het script worden toegevoegd om op te geven dat er records aan een andere tabel moeten worden toegevoegd. Dit geeft ook aan dat deze opdracht met behulp van gedeeltelijk laden moet worden uitgevoerd. Het voorvoegsel **Add** kan ook worden gebruikt in een **Map**-opdracht.

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

```
Add [ Only ] mapstatement
```

### Buffer

QVD-bestanden kunnen automatisch worden gemaakt en beheerd met behulp van het prefix **buffer**. Dit prefix kan worden gebruikt bij de meeste **LOAD**- en **SELECT**-opdrachten in een script. Hiermee wordt aangegeven dat QVD-bestanden worden gebruikt als cache/buffer om het resultaat van de opdracht op te slaan.

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

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

### Concatenate

Als twee tabellen die u wilt aaneenschakelen verschillende sets velden hebben, kunt u aaneenschakeling toch afdwingen met het prefix **Concatenate**.

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

### Crosstable

Het laadvoorvoegsel **crosstable** wordt gebruikt om gestructureerde gegevens in "kruistabel" of "draaitabel" te transponeren. Gegevens die op deze manier zijn gestructureerd, worden vaak aangetroffen bij het werken met spreadsheetbronnen. De uitvoer en het doel van het laadvoorvoegsel **crosstable** is om dergelijke structuren om te zetten in een normaal kolomgeoriënteerd tabelequivalent, aangezien deze structuur over het algemeen beter geschikt is voor analyse in Qlik Sense.

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

### First

Het prefix **First** voor een **LOAD**- of **SELECT (SQL)**-opdracht wordt gebruikt voor het laden van een vast maximumaantal records uit een gegevensbrontabel.

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

### Generic

Het laadvoorvoegsel **Generic** maakt conversie mogelijk van entiteit-attribuut-waarde gemodelleerde gegevens (EAV) naar een traditionele, genormaliseerde relationele tabelstructuur. EAV-modellering wordt ook wel "generieke gegevensmodellering" of "open schema" genoemd.

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

### Hierarchy

Het prefix **hierarchy** wordt gebruikt om een hiërarchietabel met bovenliggende-onderliggende objecten om te zetten in een tabel die bruikbaar is in een Qlik Sense-gegevensmodel. Het prefix kan voor een **LOAD**- of **SELECT**-opdracht worden geplaatst en gebruikt het resultaat van de laadopdracht als invoer voor een tabeltransformatie.

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

### HierarchBelongsTo

Dit prefix wordt gebruikt om een hiërarchietabel met bovenliggende-onderliggende objecten om te zetten in een tabel die bruikbaar is in een Qlik Sense-gegevensmodel. Het prefix kan voor een **LOAD**- of **SELECT**-opdracht worden geplaatst en gebruikt het resultaat van de laadopdracht als invoer voor een tabeltransformatie.

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

### Inner

De prefixen **join** en **keep** kunnen worden voorafgegaan door het prefix **inner**.

Indien gebruikt voor **join**, geeft de opdracht aan dat een inner join moet worden gebruikt. De resulterende tabel bevat dan alleen gecombineerde veldwaarden uit de tabellen met onbewerkte gegevens waarvoor de koppelende veldwaarden in beide tabellen voorkomen. Indien gebruikt voor **keep**, geeft de opdracht aan dat beide tabellen met onbewerkte gegevens worden gereduceerd tot de gemeenschappelijke doorsnede voordat ze worden opgeslagen in Qlik Sense.
zijn.

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

### IntervalMatch

Het prefix **IntervalMatch** wordt gebruikt voor het maken van een tabel waarin discrete numerieke waarden worden gekoppeld aan een of meer numerieke intervallen, en waarin de waarden optioneel worden gekoppeld aan een of meer extra sleutels.

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

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

### Join

Het prefix **join** koppelt de geladen tabel aan een bestaande benoemde tabel of de laatste eerder gemaakte gegevenstabel.

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

### Keep

Het prefix **keep** is vergelijkbaar met het prefix **join**. Net als bij het prefix **join** wordt de geladen tabel vergeleken met een bestaande benoemde tabel of de laatste eerder gemaakte gegevenstabel. Maar de geladen tabel wordt niet samengevoegd met een bestaande tabel, in plaats daarvan wordt een tabel of worden beide tabellen voordat ze worden opgeslagen in Qlik Sense gereduceerd op basis van de doorsnede van de tabelgegevens. De uitgevoerde vergelijking is equivalent aan een natuurlijke join voor alle gemeenschappelijke velden, net zoals bij een overeenkomstige join. De twee tabellen worden echter niet samengevoegd, maar als twee afzonderlijk benoemde tabellen in Qlik Sense bewaard.

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

### Left

De prefixen **Join** en **Keep** kunnen worden voorafgegaan door het prefix **left**.

Gebruikt voor **join** geeft het prefix aan dat er een join links moet plaatsvinden. De resulterende tabel bevat dan alleen gecombineerde veldwaarden uit de tabellen met onbewerkte gegevens waarvoor de koppelende veldwaarden in de eerste tabel voorkomen. Als het prefix wordt gebruikt vóór **keep**, geeft het aan dat de tweede tabel met onbewerkte gegevens wordt gereduceerd tot de gemeenschappelijke doorsnede met de eerste tabel, voordat deze wordt opgeslagen in Qlik Sense.

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

### Mapping

Het prefix **mapping** wordt gebruikt om een toewijzingstabel te maken die bijvoorbeeld kan worden gebruikt voor het vervangen van veldwaarden en veldnamen tijdens de uitvoering van het script.

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

### Merge

Het voorvoegsel **Merge** kan aan elke **LOAD**- of **SELECT**-opdracht in het script worden toegevoegd om op te geven dat de geladen tabel met een andere tabel moet worden samengevoegd. Dit geeft ook aan dat deze opdracht met behulp van gedeeltelijk laden moet worden uitgevoerd.

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

### NoConcatenate

Het prefix **NoConcatenate** zorgt ervoor dat twee geladen tabellen met identieke veldensets worden behandeld als twee aparte interne tabellen, terwijl ze anders automatisch zouden worden aaneengeschakeld.

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

### Outer

Het expliciete prefix **Join** kan worden voorafgegaan door het prefix **Outer** om een outer join op te geven. In een outer join worden alle combinaties tussen de twee tabellen gegenereerd. De resulterende tabel bevat dus gecombineerde veldwaarden uit de tabellen met onbewerkte gegevens waarvoor de koppelende veldwaarden in één of beide tabellen voorkomen. Het trefwoord **Outer** is optioneel en is het standaard join type dat wordt gebruikt als geen join prefix is opgegeven.

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

### Partial reload

Bij een volledige lading worden altijd eerst alle tabellen in het bestaande gegevensmodel verwijderd en vervolgens wordt het load-script uitgevoerd.

Bij een *Gedeeltelijke lading (page 103)* gebeurt dit niet. Hierbij blijven alle tabellen in het gegevensmodel behouden en worden alleen de opdrachten **Load** en **Select** uitgevoerd voorafgegaan door het voorvoegsel **Add**, **Merge** of **Replace**. Andere gegevenstabellen worden niet door de instructie beïnvloed. Het **only**-

argument geeft aan dat de opdracht alleen tijdens gedeeltelijke ladingen moet worden uitgevoerd, en moet worden genegeerd tijdens volledige ladingen. In de volgende tabel vindt u een overzicht van de uitvoering van opdrachten voor gedeeltelijke en volledige ladingen.

### Replace

het voorvoegsel **Replace** kan aan elke **LOAD**- of **SELECT**-opdracht in het script worden toegevoegd om op te geven dat de geladen tabel een andere tabel moet vervangen. Dit geeft ook aan dat deze opdracht met behulp van gedeeltelijk laden moet worden uitgevoerd. Het voorvoegsel **Replace** kan ook worden gebruikt in een **Map**-opdracht.

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

```
Replace [only] mapstatement
```

### Right

De prefixen **Join** en **Keep** kunnen worden voorafgegaan door het prefix **right**.

Als het prefix wordt gebruikt vóór **join**, geeft het aan dat er een join rechts moet plaatsvinden. De resulterende tabel bevat dan alleen gecombineerde veldwaarden uit de tabellen met onbewerkte gegevens waarvoor de koppelende veldwaarden in de tweede tabel voorkomen. Als het prefix wordt gebruikt voor **keep**, geeft het aan dat de eerste tabel met onbewerkte gegevens wordt gereduceerd tot de gemeenschappelijke doorsnede met de tweede tabel, voordat deze wordt opgeslagen in Qlik Sense.

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

### Sample

Als u het prefix **sample** voor een **LOAD**- of **SELECT**-opdracht plaatst, wordt er een willekeurige selectie records uit de gegevensbron geladen.

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

### Semantic

Tabellen met relaties tussen records kunnen worden geladen met het prefix **semantic**. Dit kunnen bijvoorbeeld interne referenties binnen een tabel zijn waarbij een record wijst naar een andere, zoals bovenliggend, behoort tot of voorganger.

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

### Unless

Met het prefix en suffix **unless** maakt u een conditionele clausule waarmee wordt bepaald of een opdracht of exit-clausule wel of niet moet worden geëvalueerd. De opdracht kan worden beschouwd als een compact alternatief voor een volledige **if..end if**-opdracht.

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

### When

Met het prefix en suffix **when** maakt u een conditionele clausule waarmee wordt bepaald of een opdracht of exit-clausule wel of niet moet worden uitgevoerd. De opdracht kan worden beschouwd als een compact alternatief voor een volledige **if..end if**-opdracht.

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

## Add

Het voorvoegsel **Add** kan aan elke **LOAD**- of **SELECT**-opdracht in het script worden toegevoegd om op te geven dat er records aan een andere tabel moeten worden toegevoegd. Dit geeft ook aan dat deze opdracht met behulp van gedeeltelijk laden moet worden uitgevoerd. Het voorvoegsel **Add** kan ook worden gebruikt in een **Map**-opdracht.

> *Om ervoor te zorgen dat een gedeeltelijke lading correct werkt, moet de app met gegevens worden geopend voordat een gedeeltelijke lading wordt geactiveerd.*

Voer een gedeeltelijke lading uit met de knop **Opnieuw laden**. U kunt ook de Qlik Engine JSON API gebruiken.

**Syntaxis:**

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

```
Add [only] mapstatement
```

Tijdens een normale (niet-gedeeltelijke) lading, dient de constructie **Add LOAD** als een normale **LOAD**-instructie. Records worden gegenereerd en opgeslagen in een tabel.

Als het voorvoegsel **Concatenate** is gebruikt of is een tabel aanwezig met dezelfde verzameling van velden, worden de records toegevoegd aan de relevante bestaande tabel. Ander zorgt de constructie **Add LOAD** dat er een nieuwe tabel wordt gemaakt.

Bij een gedeeltelijke lading gebeurt hetzelfde. Het enige verschil is dat de bij de constructie **Add LOAD** nooit een nieuwe tabel wordt gemaakt. Er bestaat altijd een relevante tabel van de uitvoering van het vorige script waarmee de records moeten worden samengevoegd.

Er wordt geen controle op duplicaten uitgevoerd. Daarom bevat een instructie waarbij het prefix **Add** wordt gebruikt gewoonlijk een kwalificatie voor distinct of een where-clausule om duplicaten te voorkomen.

De instructie **Add Map...Using** zorgt ervoor dat de toewijzing ook plaatsvindt als het script gedeeltelijk wordt uitgevoerd.

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| only | Een optionele kwalificatie die aangeeft dat de opdracht alleen moet worden uitgevoerd tijdens het gedeeltelijk laden van gegevens. Het moet worden genegeerd tijdens een normale (niet-gedeeltelijke) lading. |

Voorbeelden en resultaten:

| Voorbeeld | Resultaat |
|---|---|
| `Tab1:`<br><br>`LOAD Name, Number`<br>`FROM Persons.csv;`<br><br>`Add LOAD Name,`<br>`Number FROM`<br>`newPersons.csv;` | Tijdens normaal opnieuw laden worden gegevens geladen uit *Persons.csv* en opgeslagen in de Qlik Sense-tabel Tab1. Gegevens uit *NewPersons.csv* worden vervolgens samengevoegd met dezelfde Qlik Sense-tabel.<br><br>Tijdens het gedeeltelijk opnieuw laden worden gegevens geladen uit *NewPersons.csv* en toegevoegd aan de Qlik Sense-tabel Tab1. Er wordt geen controle op duplicaten uitgevoerd. |
| `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);` | Controle op duplicaten wordt uitgevoerd door te kijken of Name bestaat in de eerder geladen tabelgegevens.<br><br>Tijdens normaal opnieuw laden worden gegevens geladen uit *Persons.csv* en opgeslagen in de Qlik Sense-tabel Tab1. Gegevens uit *NewPersons.csv* worden vervolgens samengevoegd met dezelfde Qlik Sense-tabel.<br><br>Tijdens het gedeeltelijk opnieuw laden worden gegevens geladen uit *NewPersons.csv* die wordt toegevoegd aan de Qlik Sense-tabel Tab1. Controle op duplicaten wordt uitgevoerd door te kijken of Name bestaat in de eerder geladen tabelgegevens. |
| `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);` | Tijdens normaal opnieuw laden worden gegevens geladen uit *Persons.csv* en opgeslagen in de Qlik Sense-tabel Tab1. De opdracht voor het laden van *NewPersons.csv* wordt genegeerd.<br><br>Tijdens het gedeeltelijk opnieuw laden worden gegevens geladen uit *NewPersons.csv* die wordt toegevoegd aan de Qlik Sense-tabel Tab1. Controle op duplicaten wordt uitgevoerd door te kijken of Name bestaat in de eerder geladen tabelgegevens. |

## Buffer

QVD-bestanden kunnen automatisch worden gemaakt en beheerd met behulp van het prefix **buffer**. Dit prefix kan worden gebruikt bij de meeste **LOAD**- en **SELECT**-opdrachten in een script. Hiermee wordt aangegeven dat QVD-bestanden worden gebruikt als cache/buffer om het resultaat van de opdracht op te slaan.

**Syntaxis:**

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

Als deze opties niet zijn gedefinieerd, wordt de QVD-buffer die bij de eerste uitvoering van het script werd gemaakt, voor onbepaalde tijd gebruikt.

Het bufferbestand wordt opgeslagen in de submap *Buffers* normaliter *C:\ProgramData\Qlik\Sense\Engine\Buffers* (serverinstallatie) of *C:\Gebruikers\{gebruiker}\Documenten\Qlik\Sense\Buffers* (Qlik Sense Desktop).

De naam van het QVD-bestand is een berekende naam, een 160-bits hexadecimale hash van de gehele volgende **LOAD** of **SELECT**-opdracht en andere onderscheidende informatie. Dit betekent dat de QVD-buffer ongeldig wordt door een wijziging in de volgende **LOAD**- of **SELECT**-opdracht.

Een QVD-buffer wordt normaal gesproken verwijderd als er tijdens de volledige uitvoering van het script in de app waarvoor de buffer werd gemaakt, niet langer naar wordt verwezen of als de app waarvoor de buffer werd gemaakt, niet meer bestaat.

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| incrementeel | Met de optie incremental wordt aangegeven dat alleen een gedeelte van een onderliggend bestand hoeft te worden gelezen. De vorige grootte van het bestand wordt opgeslagen in de XML-header van het QVD-bestand. Dit is vooral nuttig bij logbestanden. Alle records die eerder zijn geladen, worden gelezen uit het QVD-bestand. Nieuwe bestanden worden gelezen uit de originele bron en vervolgens wordt een bijgewerkt QVD-bestand gemaakt.<br><br>De optie incremental kan alleen worden gebruikt met **LOAD**-opdrachten en tekstbestanden. Incrementeel laden kan niet worden gebruikt als oude gegevens zijn gewijzigd of verwijderd. |
| verouderd [na] hoeveelheid [(dagen \| uren)] | amount is een getal dat de tijdsperiode aangeeft. Het gebruik van decimalen is toegestaan. Als er geen eenheid is aangegeven, wordt verondersteld dat het om dagen gaat.<br>De optie stale after wordt meestal gebruikt voor DB-bronnen waarbij de gegevens niet zijn voorzien van een tijdsstempel. In plaats daarvan geeft men aan hoe oud een gebruikt QVD-snapshot mag zijn. Een stale after-clausule geeft een tijdsperiode aan. Als de QVD-buffer ouder is dan deze periode, wordt de buffer niet meer als geldig beschouwd. Voor deze tijd wordt de QVD-buffer gebruikt als gegevensbron, daarna wordt de oorspronkelijke gegevensbron gebruikt. Het QVD-bufferbestand wordt dan automatisch bijgewerkt en er begint een nieuwe periode. |

**Beperkingen:**

Er zijn talloze beperkingen. De belangrijkste is dat er een **LOAD**- of een **SELECT**-opdracht voor een bestand moet voorkomen in de "basis" van een complexe opdracht.

**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` is een load-prefix van het script waardoor een gegevensverzameling kan worden bijgevoegd bij een bestaande tabel in het geheugen. Het wordt vaak gebruikt om verschillende sets transactiegegevens toe te voegen aan één centrale feitentabel, of veel voorkomende referentiegegevenssets van een specifiek type op te bouwen die voortkomen uit verschillende bronnen. Het is vergelijkbaar wat betreft functionaliteit met de SQL UNION-operator.

De resulterende tabel van een `concatenate`-bewerking omvat vervolgens de oorspronkelijke gegevensset met de nieuwe rijen gegevens die onderaan aan die tabel zijn toegevoegd. In de bron- en doeltabellen kunnen verschillende velden staan. Indien velden anders zijn, wordt de resulterende tabel verbreed om het gecombineerde resultaat van alle aanwezige velden in zowel de bron- als de doeltabel weer te geven.

**Syntaxis:**

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

Argumenten

| Argument | Beschrijving |
|---|---|
| `tablename` | De naam van een bestaande tabel. De genoemde tabel wordt het doel van de `Concatenate`-bewerking en eventuele records van geladen gegevens worden aan die tabel toegevoegd. Als de parameter `tablename` niet wordt gebruikt, wordt de doeltabel de laatst geladen tabel vóór deze opdracht. |
| `loadstatement/selectstatement` | Het `loadstatement/selectstatement`-argument dat op het `tablename`-argument volgt, wordt aaneengeschakeld met de opgegeven tabel. |

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze

landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

Functievoorbeeld

| Voorbeeld | Resultaat |
|---|---|
| `Concatenate (Transactions) Load …. ;` | De gegevens die in de load-instructie onder het `Concatenate`-prefix zijn geladen, worden toegevoegd aan de bestaande tabel in het geheugen, `Transactions`. Er wordt hierbij vanuit gegaan dat de tabel `Transactions` eerder dan dit punt in het load-script al is geladen. |

## Voorbeeld 1 – Meerdere gegevenssets toegevoegd aan een doeltabel met de load-prefix voor aaneenschakelen

Load-script en resultaten

### Overzicht

In dit voorbeeld laadt u twee scripts achter elkaar.

- Het eerste load-script bevat een aanvankelijke gegevensset met datums en bedragen, die naar de tabel `Transactions` wordt gestuurd.
- Het tweede load-script bevat:
  - Een tweede gegevensset die aan de eerste gegevensset wordt toegevoegd met behulp van de `Concatenate`-prefix. Deze gegevensset heeft een extra veld, `type`, dat niet in de eerste gegevensset staat.
  - Het `Concatenate`-prefix.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

### Eerste load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date
- amount

Resultatentabel eerste load-script

| 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 |

De tabel toont de eerste gegevensset.

### Tweede load-script

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe.

```
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
];
```

### Resultaten

Laad de gegevens en ga naar het werkblad. Maak dit veld als een dimensie.

- type

Resultatentabel tweede load-script

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

| id | date | amount | type |
|---|---|---|---|
| 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 | Intern |
| 3759 | 10/03/2018 | 384.00 | Extern |
| 3760 | 10/06/2018 | 25.82 | Intern |
| 3761 | 10/09/2018 | 312.00 | Intern |
| 3762 | 10/15/2018 | 4.56 | Intern |
| 3763 | 10/16/2018 | 90.24 | Intern |
| 3764 | 10/18/2018 | 19.32 | Extern |

Voor de nulwaarden in het `type`-veld voor de eerste zeven geladen records was `type` nog niet was gedefinieerd.

## Voorbeeld 2 – Meerdere gegevenssets toegevoegd aan een doeltabel met impliciete aaneenschakeling

Load-script en resultaten

### Overzicht

Een typische gebruikssituatie voor impliciete toevoeging van gegevens is wanneer u verschillende bestanden met identiek gestructureerde gegevens laadt en u ze allemaal aan een doeltabel wilt toevoegen.

Bijvoorbeeld door `wildcards` in bestandsnamen te gebruiken met een syntax zoals:

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

of in lussen met gebruik van constructies zoals:

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

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

next file
```

> *Impliciete aaneenschakeling vindt plaats tussen twee geladen tabellen met identieke veldnamen, ook als ze niet na elkaar in het script zijn gedefinieerd. Dit kan ertoe leiden dat gegevens onbedoeld aan tabellen worden toegevoegd. Als u niet wilt dat een secundaire tabel met identieke velden op deze manier wordt toegevoegd, gebruikt u het load-prefix `NoConcatenate`. De tabel een andere tabelnaam geven is niet afdoende om impliciete aaneenschakeling te voorkomen. Ga voor meer informatie naar NoConcatenate (page 93).*

In dit voorbeeld laadt u twee scripts achter elkaar.

- Het eerste load-script bevat een aanvankelijke gegevensset met vier velden, die naar de tabel `Transactions` wordt gestuurd.
- Het tweede load-script bevat een gegevensset met dezelfde velden als de eerste gegevensset.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

### Eerste load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

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

Resultatentabel eerste load-script

| id | date | type | amount |
|---|---|---|---|
| 3758 | 10/01/2018 | Intern | 164.27 |
| 3759 | 10/03/2018 | Extern | 384.00 |
| 3760 | 10/06/2018 | Intern | 25.82 |
| 3761 | 10/09/2018 | Intern | 312.00 |
| 3762 | 10/15/2018 | Intern | 4.56 |

| id | date | type | amount |
|---|---|---|---|
| 3763 | 10/16/2018 | Intern | 90.24 |
| 3764 | 10/18/2018 | Extern | 19.32 |

De tabel toont de eerste gegevensset.

#### Tweede load-script

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe.

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

#### Resultaten

Laad de gegevens en ga naar het werkblad.

Resultatentabel tweede load-script

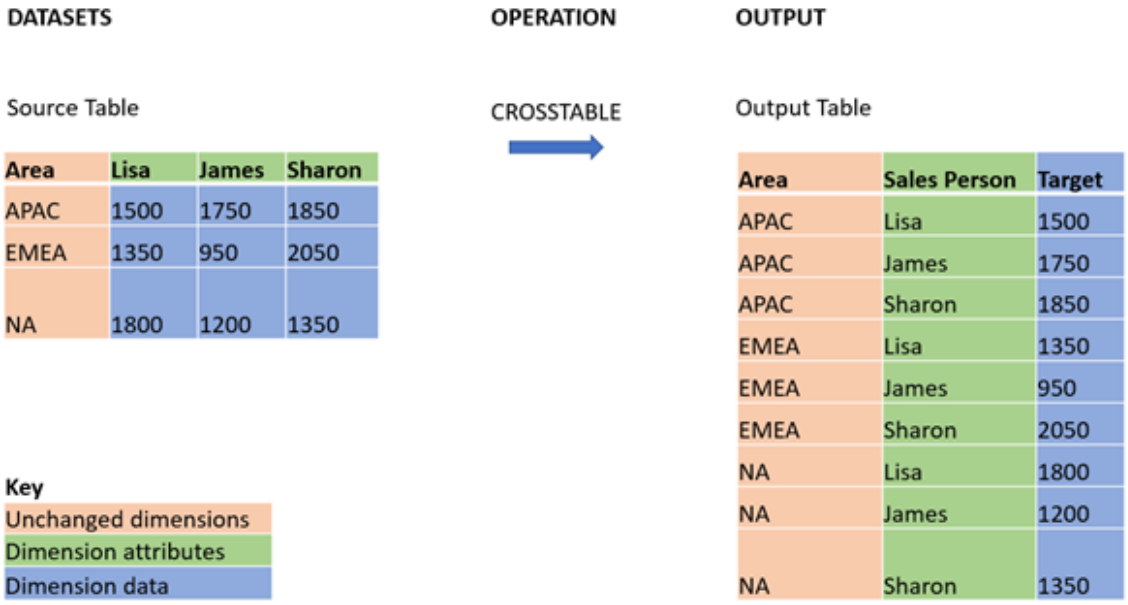
| id | date | type | amount |
|---|---|---|---|
| 3758 | 10/01/2018 | Intern | 164.27 |
| 3759 | 10/03/2018 | Extern | 384.00 |
| 3760 | 10/06/2018 | Intern | 25.82 |
| 3761 | 10/09/2018 | Intern | 312.00 |
| 3762 | 10/15/2018 | Intern | 4.56 |
| 3763 | 10/16/2018 | Intern | 90.24 |
| 3764 | 10/18/2018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

De tweede gegevensset is impliciet aaneengeschakeld met de eerste gegevensset omdat de sets identieke velden bevatten.

## Crosstable

Het laadvoorvoegsel **crosstable** wordt gebruikt om gestructureerde gegevens in "kruistabel" of "draaitabel" te transponeren. Gegevens die op deze manier zijn gestructureerd, worden vaak aangetroffen bij het werken met spreadsheetbronnen. De uitvoer en het doel van het laadvoorvoegsel **crosstable** is om dergelijke structuren om te zetten in een normaal kolomgeoriënteerd tabelequivalent, aangezien deze structuur over het algemeen beter geschikt is voor analyse in Qlik Sense.

*Voorbeeld van gegevens die zijn gestructureerd als een kruistabel en de equivalente structuur ervan na een kruistabeltransformatie*




**Syntaxis:**

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

Argumenten

| Argument | Beschrijving |
|---|---|
| attribute field name | De gewenste uitvoerveldnaam die de horizontaal georiënteerde dimensie beschrijft die moet worden getransponeerd (de koprij). |
| data field name | De gewenste uitvoerveldnaam die de horizontaal georiënteerde gegevens beschrijft van de dimensie die moet worden getransponeerd (de matrix van gegevenswaarden onder de koprij). |
| n | Het aantal kwalificatievelden, of ongewijzigde dimensies, dat voor de tabel staat, die in generieke vorm moet worden veranderd. De standaardwaarde is 1. |

Deze scriptfunctie is gerelateerd aan de volgende functies:

Gerelateerde functies

| Functie | Interactie |
|---|---|
| Generic (page 59) | Een voorvoegsel voor het laden van de transformatie dat een gestructureerde gegevensset van entiteit-attribuut-waarde neemt en deze omzet in een normale relationele tabelstructuur, waarbij elk gevonden attribuut wordt gescheiden in een nieuw veld of nieuwe kolom met gegevens. |

## Voorbeeld 1 – Omzetten van gedraaide verkoopgegevens (eenvoudig)

Load-scripts en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het eerste onderstaande load-script toe aan een nieuw tabblad.

Het eerste load-script bevat een gegevensverzameling waarop het scriptvoorvoegsel `crosstable` later zal worden toegepast, waarbij de sectie die van toepassing is op `crosstable` wordt weggelaten. Dit betekent dat de syntaxis van opmerkingen is gebruikt om deze sectie in het load-script uit te schakelen.

Het tweede load-script is hetzelfde als het eerste, maar met de toepassing van `crosstable` zonder commentaar (ingeschakeld door de commentaarsyntaxis te verwijderen). De scripts worden op deze manier weergegeven om de waarde van deze scriptfunctie bij het transformeren van gegevens te benadrukken.

### Eerste load-script (functie niet toegepast)

```
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;
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

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

Resultatentabel

| Product | Jan 2021 | Feb 2021 | Mrt 2021 | Apr 2021 | Mei 2021 | Jun 2021 |
|---|---|---|---|---|---|---|
| A | 100 | 98 | 103 | 63 | 108 | 82 |
| B | 284 | 279 | 297 | 305 | 294 | 292 |
| C | 50 | 53 | 50 | 54 | 49 | 51 |

Met dit script kan een kruistabel met een kolom voor elke maand en één rij per product worden gemaakt. In de huidige indeling kunnen deze gegevens niet gemakkelijk worden geanalyseerd. Het is veel beter als alle getallen zich in één veld bevinden en alle maanden in een ander veld, in een tabel met drie kolommen. In de volgende sectie wordt uitgelegd hoe u deze transformatie naar de kruistabel kunt uitvoeren.

### Tweede load-script (functie toegepast)

Verwijder het commentaar van het script door de `//` te verwijderen. Het load-script zou er als volgt moeten uitzien:

```
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;
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `Product`
- `Month`
- `Sales`

Resultatentabel

| Product | Maand | Sales. |
|---|---|---|
| A | Jan 2021 | 100 |

| Product | Maand | Sales. |
|---|---|---|
| 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 |

Wanneer het scriptvoorvoegsel is toegepast, wordt de kruistabel getransformeerd in een rechte tabel met één kolom voor `Month` en nog een voor `Sales`. Dit verbetert de leesbaarheid van de gegevens.

## Voorbeeld 2 - Omzetten van gedraaide verkoopdoelgegevens in een verticale tabelstructuur (tussenliggend)

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset die wordt geladen in een tabel met de naam Doelen.
- Het laadvoorvoegsel `crosstable`, dat de gedraaide verkopersnamen omzet in een eigen veld, met het label `Sales Person`.
- De bijbehorende verkoopdoelgegevens, die zijn gestructureerd in een veld met de naam `Target`.

**Load-script**

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

**Resultaten**

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `Area`
- `Sales Person`

Voeg deze meting toe:

```
=Sum(Target)
```

Resultatentabel

| Gebied | Verkoop-medewerker | =Sum(Target) |
|---|---|---|
| APAC | James | 1750 |
| APAC | Lisa | 1500 |
| APAC | Sharon | 1850 |
| EMEA | James | 950 |
| EMEA | Lisa | 1350 |
| EMEA | Sharon | 2050 |
| n.v.t. | James | 1200 |
| n.v.t. | Lisa | 1800 |
| n.v.t. | Sharon | 1350 |

Als u de weergave van gegevens wilt repliceren als de gedraaide invoertabel, kunt u een equivalente draaitabel in een werkblad maken.

**Doe het volgende:**

1. Kopieer en plak de tabel die u zojuist hebt gemaakt in het werkblad.
2. Sleep het diagramobject **Draaitabel** bovenop de nieuw gemaakte tabelkopie. Selecteer **Converteren**.

3. Klik op ✔ **Klaar met bewerken**.
4. Sleep het veld `Sales Person` van de verticale kolomplank naar de horizontale kolomplank.

De volgende tabel toont de gegevens in de oorspronkelijke tabelvorm, zoals weergegeven in Qlik Sense:

Originele resultatentabel, zoals weergegeven in Qlik Sense

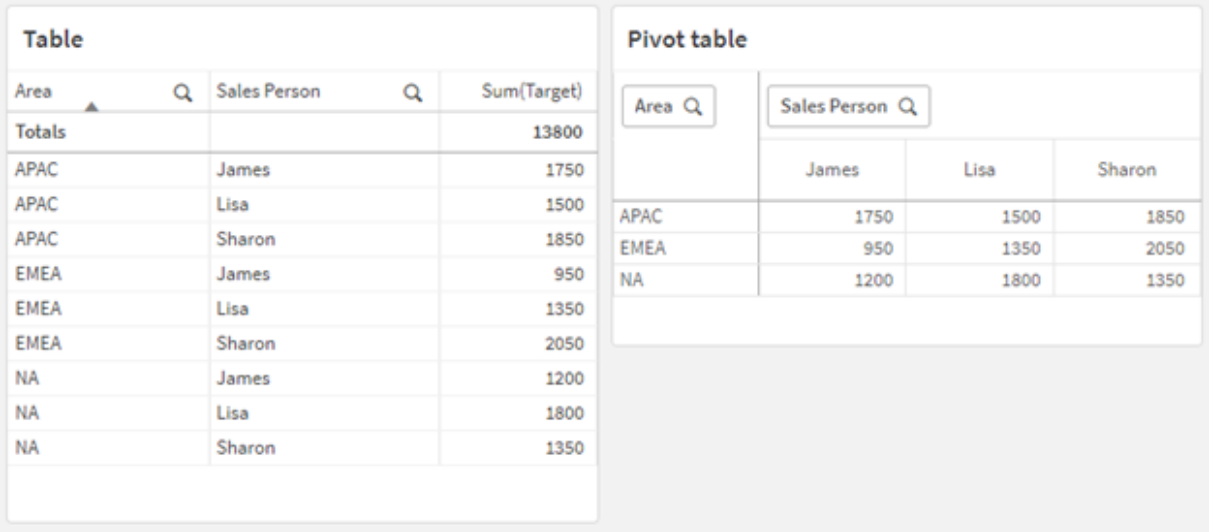
| Gebied | Verkoop-medewerker | =Sum(Target) |
|---|---|---|
| Totalen | - | 13800 |
| APAC | James | 1750 |
| APAC | Lisa | 1500 |
| APAC | Sharon | 1850 |
| EMEA | James | 950 |
| EMEA | Lisa | 1350 |
| EMEA | Sharon | 2050 |
| n.v.t. | James | 1200 |
| n.v.t. | Lisa | 1800 |
| n.v.t. | Sharon | 1350 |

De equivalente draaitabel ziet er als volgt uit, met de kolom voor de naam van elke verkoper in de grotere rij voor `Sales Person`:

Equivalente draaitabel met het veld `Sales Person` horizontaal gedraaid

| Gebied | James | Lisa | Sharon |
|---|---|---|---|
| APAC | 1750 | 1500 | 1850 |
| EMEA | 950 | 1350 | 2050 |
| n.v.t. | 1350 | 1350 | 1350 |

*Voorbeeld van gegevens weergegeven als tabel en een equivalente draaitabel met het veld `Sales Person` horizontaal gedraaid*



**Table**

| Area | Sales Person | Sum(Target) |
|---|---|---|
| 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 |

**Pivot table**

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

## Voorbeeld 3 - Omzetten van gedraaide verkoop- en doelgegevens in een verticale tabelstructuur (geavanceerd)

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met verkoop- en doelgegevens, geordend op gebied en maand van het jaar. Dit wordt geladen in een tabel genaamd `SalesAndTargets`.
- Het laadvoorvoegsel `crosstable`. Dit wordt gebruikt om het draaien van de dimensie `Month Year` ongedaan te maken in een speciaal veld en om de matrix van verkoop- en doelbedragen om te zetten in een speciaal veld met de naam `Amount`.
- Een conversie van het veld `Month Year` van tekst naar een goede datum, met behulp van de tekst-naar-datum-conversiefunctie `date#`. Dit naar datum geconverteerde veld `Month Year` wordt teruggekoppeld naar de tabel `SalesAndTarget` via een laadvoorvoegsel `Join`.

### Load-script

```
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
```

```
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
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
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;
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `Area`
- `Month Year`

Maak de volgende meting, met het label `Actual`:

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

Maak ook deze meting, met het label `Target`:

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

Resultatentabel (bijgesneden)

| Gebied | Maand Jaar | Werkelijk | Doel |
|---|---|---|---|
| APAC | Jan-22 | 435 | 425 |
| APAC | Feb-22 | 434 | 425 |
| APAC | Mar-22 | 397 | 425 |
| APAC | Apr-22 | 404 | 425 |
| APAC | Mei-22 | 458 | 425 |
| APAC | Jun-22 | 447 | 425 |
| APAC | Jul-22 | 413 | 425 |
| APAC | Aug-22 | 458 | 425 |
| APAC | Sep-22 | 385 | 425 |
| APAC | Oct-22 | 421 | 425 |
| APAC | Nov-22 | 448 | 425 |

| Gebied | Maand Jaar | Werkelijk | Doel |
|---|---|---|---|
| APAC | Dec-22 | 397 | 425 |
| EMEA | Jan-22 | 363.5 | 362.5 |
| EMEA | Feb-22 | 359.5 | 362.5 |

Als u de weergave van gegevens wilt repliceren als de gedraaide invoertabel, kunt u een equivalente draaitabel in een werkblad maken.

**Doe het volgende:**

1. Kopieer en plak de tabel die u zojuist hebt gemaakt in het werkblad.
2. Sleep het diagramobject **Draaitabel** bovenop de nieuw gemaakte tabelkopie. Selecteer **Converteren**.
3. Klik op ✔ **Klaar met bewerken**.
4. Sleep het veld `Month Year` van de verticale kolomplank naar de horizontale kolomplank.
5. Sleep het item `Values` van de horizontale kolomplank naar de verticale kolomplank.

De volgende tabel toont de gegevens in de oorspronkelijke tabelvorm, zoals weergegeven in Qlik Sense:

Originele resultatentabel (bijgesneden), zoals weergegeven in Qlik Sense

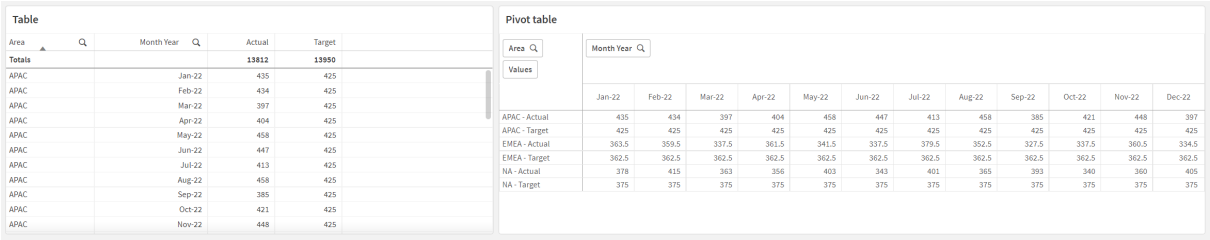
| Gebied | Maand Jaar | Werkelijk | Doel |
|---|---|---|---|
| Totalen | - | 13812 | 13950 |
| APAC | Jan-22 | 435 | 425 |
| APAC | Feb-22 | 434 | 425 |
| APAC | Mar-22 | 397 | 425 |
| APAC | Apr-22 | 404 | 425 |
| APAC | Mei-22 | 458 | 425 |
| APAC | Jun-22 | 447 | 425 |
| APAC | Jul-22 | 413 | 425 |
| APAC | Aug-22 | 458 | 425 |
| APAC | Sep-22 | 385 | 425 |
| APAC | Oct-22 | 421 | 425 |
| APAC | Nov-22 | 448 | 425 |
| APAC | Dec-22 | 397 | 425 |
| EMEA | Jan-22 | 363.5 | 362.5 |
| EMEA | Feb-22 | 359.5 | 362.5 |

De equivalente draaitabel ziet er als volgt uit, met de kolom voor elke afzonderlijke maand van het jaar in de grotere rij voor `Month Year`:

Equivalente draaitabel (bijgesneden) met het veld `Month Year` horizontaal gedraaid

| Gebied (Waarden) | Jan-22 | Feb-22 | Mar-22 | Apr-22 | Mei-22 | Jun-22 | Jul-22 | Aug-22 | Sep-22 | Oct-22 | Nov-22 | Dec-22 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| APAC - Actueel | 435 | 434 | 397 | 404 | 458 | 447 | 413 | 458 | 385 | 421 | 448 | 397 |
| APAC - Doel | 425 | 425 | 425 | 425 | 425 | 425 | 425 | 425 | 425 | 425 | 425 | 425 |
| EMEA - Actueel | 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 - Doel | 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.v.t. - Actueel | 378 | 415 | 363 | 356 | 403 | 343 | 401 | 365 | 393 | 340 | 360 | 405 |
| N.v.t. - Doel | 375 | 375 | 375 | 375 | 375 | 375 | 375 | 375 | 375 | 375 | 375 | 375 |

*Voorbeeld van gegevens weergegeven als tabel en een equivalente draaitabel met het veld `Month Year` horizontaal gedraaid*




## First

Het voorvoegsel `First` voor een `LOAD`- of `SELECT`-opdracht (SQL) wordt gebruikt voor het laden van een vast maximumaantal records uit een gegevensbrontabel. Een typisch gebruiksscenario voor het gebruik van het voorvoegsel `First` is wanneer u een kleine subset records wilt ophalen uit een grote en/or langzame stap voor het laden van gegevens. Zodra het gedefinieerde '`n`' aantal records is geladen, wordt de laadstap voortijdig beëindigd en gaat de rest van de uitvoering van het script gewoon door.

**Syntaxis:**

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

Argumenten

| Argument | Beschrijving |
|---|---|
| `n` | Een willekeurige uitdrukking die resulteert in een geheel getal waarmee het maximale aantal te lezen records wordt aangegeven. `n` kan ook worden omsloten door haken: `(n)`. |
| `loadstatement \| selectstatement` | Het `load statement/select statement` dat volgt op het n-argument definieert de gespecificeerde tabel die moet worden geladen met het ingestelde maximale aantal records. |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `FIRST 10 LOAD * from abc.csv;` | Dit voorbeeld haalt de eerste tien regels op uit een Excel-bestand. |
| `FIRST (1) SQL SELECT * from Orders;` | In dit voorbeeld wordt de eerste geselecteerde regel opgehaald uit de gegevensverzameling `Orders`. |

## Voorbeeld – De eerste vijf rijen laden

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met datums uit de eerste twee weken van 2020.
- De variabele `First` die de toepassing de opdracht geeft om alleen de eerste vijf records te laden.

**Load-script**

```
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
];
```

**Resultaten**

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg `Date` toe als een veld en `sum(sales)` als een meting:

Resultatentabel

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

Het script laadt alleen de eerste vijf records van de tabel `Sales`.

## Generic

Het laadvoorvoegsel **Generic** maakt conversie mogelijk van entiteit-attribuut-waarde gemodelleerde gegevens (EAV) naar een traditionele, genormaliseerde relationele tabelstructuur. EAV-modellering wordt ook wel "generieke gegevensmodellering" of "open schema" genoemd.

*Voorbeeld van EAV-gemodelleerde gegevens en een equivalente gedenormaliseerde relationele tabel*

| 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 |

*Voorbeeld van EAV-gemodelleerde gegevens en een equivalente set van genormaliseerde relationele tabellen*

| 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 |

Hoewel het technisch mogelijk is om EAV-gemodelleerde gegevens in Qlik te laden en te analyseren, is het vaak gemakkelijker om met een equivalente traditionele relationele gegevensstructuur te werken.

**Syntaxis:**

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

Deze onderwerpen kunnen u helpen bij het werken met deze functie:

Gerelateerde onderwerpen

| Onderwerp | Beschrijving |
|---|---|
| *Crosstable (page 47)* | Het laadvoorvoegsel `Crosstable` transformeert gegevens die horizontaal zijn georiënteerd in verticaal georiënteerde gegevens. Vanuit een puur functioneel perspectief voert het de tegenovergestelde transformatie uit naar het laadvoorvoegsel `Generic`, hoewel de voorvoegsels doorgaans geheel andere gebruiksscenario's dienen. |
| **Generieke databases** in *Gegevens beheren* | EAV-gestructureerde gegevensmodellen worden hier verder beschreven. |

## Voorbeeld 1 – Gestructureerde EAV-gegevens transformeren met het algemene laadvoorvoegsel

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat een gegevensverzameling die wordt geladen in een tabel met de naam `Transactions`. De gegevensverzameling bevat een datumveld. De standaard `MonthNames`-definitie wordt gebruikt.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `Color`.

Voeg deze meting toe:

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

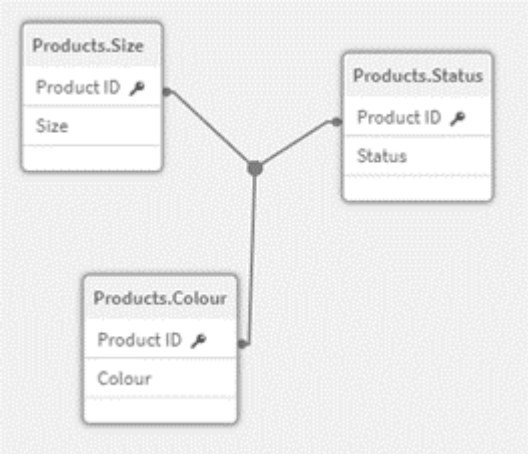
Nu kunt u het aantal producten op kleur inspecteren.

Resultatentabel

| Kleur | =Count([Product ID]) |
|---|---|
| Bruin | 4 |
| Wit | 2 |

Let op de vorm van het gegevensmodel, waarbij elk attribuut is uitgesplitst in een aparte tabel die wordt genoemd volgens de oorspronkelijke tag van de doeltabel `Product`. Elke tabel heeft het attribuut als achtervoegsel. Een voorbeeld hiervan is `Product.Color`. De resulterende uitvoerrecords Productattribuut worden gekoppeld door de `Product ID`.

*Weergave van de gegevensmodelviewer van de resultaten*




Resulterende tabel met records: Products.Status

| Product-id | Status |
|---|---|
| 13 | Stopgezet |
| 2 | Stopgezet |

Resulterende tabel met records: Products.Size

| Product-id | Grootte |
|---|---|
| 13 | 13-15 |
| 20 | 16-18 |
| 45 | 16-18 |

Resulterende tabel met records: Products.Color

| Product-id | Kleur |
|---|---|
| 13 | Bruin |
| 5 | Bruin |
| 44 | Bruin |
| 45 | Bruin |

| Product-id | Kleur |
|---|---|
| 20 | Wit |
| 2 | Wit |

## Voorbeeld 2 – Gestructureerde EAV-gegevens analyseren zonder het algemene laadvoorvoegsel

Load-script en diagramuitdrukking

### Overzicht

Dit voorbeeld laat zien hoe u gestructureerde EAV-gegevens in hun oorspronkelijke vorm kunt analyseren.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat een gegevensset die wordt geladen in een tabel met de naam `Products` in een EAV-structuur.

In dit voorbeeld tellen we nog steeds producten op kleurattribuut. Om op deze manier gestructureerde gegevens te analyseren, moet u filtering op uitdrukkingsniveau toepassen op producten met de attribuutwaarde `Color`.

Bovendien zijn individuele attributen niet beschikbaar om te selecteren als dimensies of velden, waardoor het moeilijker wordt om te bepalen hoe effectieve visualisaties kunnen worden gemaakt.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `Value`.

Maak de volgende meting:

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

Nu kunt u het aantal producten op kleur inspecteren.

Resulterende tabel met records: Products.Status

| Waarde | =Count({<Attribute={'Color'}>} [Product ID]) |
|---|---|
| Bruin | 4 |
| Wit | 2 |

## Voorbeeld 3 - Denormaliseren van de resulterende uitvoertabellen van een generieke lading (geavanceerd)

Load-script en diagramuitdrukking

### Overzicht

In dit voorbeeld laten we zien hoe de genormaliseerde gegevensstructuur die wordt geproduceerd door het laadvoorvoegsel `Generic`, weer kan worden gedenormaliseerd naar een geconsolideerde `Product`-dimensietabel. Dit is een geavanceerde modelleringstechniek die kan worden gebruikt als onderdeel van het afstemmen van de prestaties van gegevensmodellen.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

### Load-script

```
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;

OUTER JOIN (Products)
```

```
LOAD * RESIDENT Products.Status;
DROP TABLES Products.Size,Products.Status;
```

### Resultaten

Open de gegevensmodelviewer en let op de vorm van het resulterende gegevensmodel. Er is slechts één gedenormaliseerde tabel aanwezig. Het is een combinatie van de drie tussenliggende uitvoertabellen: `Products.Size`, `Products.Status` en `Products.Color`.

Resulterend intern gegevensmodel

| Producten |
|---|
| Product-id |
| Status |
| Kleur |
| Grootte |

Resulterende tabel met records: Products

| Product-id | Status | Kleur | Grootte |
|---|---|---|---|
| 13 | Stopgezet | Bruin | 13-15 |
| 20 | - | Wit | 16-18 |
| 2 | Stopgezet | Wit | - |
| 5 | - | Bruin | - |
| 44 | - | Bruin | - |
| 45 | - | Bruin | 16-18 |

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `Color`.

Voeg deze meting toe:

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

Resultatentabel

| Kleur | =Count([Product ID]) |
|---|---|
| Bruin | 4 |
| Wit | 2 |

## Hierarchy

Het prefix **hierarchy** wordt gebruikt om een hiërarchietabel met bovenliggende-onderliggende objecten om te zetten in een tabel die bruikbaar is in een Qlik Sense-gegevensmodel. Het prefix kan voor een **LOAD**- of **SELECT**-opdracht worden geplaatst en gebruikt het resultaat van de laadopdracht als invoer voor een tabeltransformatie.

De prefix maakt een uitgebreide knooppuntentabel, die normaliter hetzelfde aantal records bevat als de invoertabel, maar daarnaast wordt elk niveau in de hiërarchie ook nog een opgeslagen in een apart veld. Het padveld kan worden gebruikt in een boomstructuur.

**Syntaxis:**

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

De invoertabel moet een aangrenzende knooppuntentabel zijn. Aangrenzende knooppuntentabellen zijn tabellen waarin elk record correspondeert met een knooppunt en een veld heeft dat de referentie naar het bovenliggende knooppunt bevat. In een dergelijke tabel wordt het knooppunt opgeslagen in slechts één record, maar kan het knooppunt wel meerdere onderliggende knooppunten hebben. De tabel kan natuurlijk aanvullende velden bevatten waarin kenmerken van de knooppunten worden beschreven.

De prefix maakt een uitgebreide knooppuntentabel, die normaliter hetzelfde aantal records bevat als de invoertabel, maar daarnaast wordt elk niveau in de hiërarchie ook nog een opgeslagen in een apart veld. Het padveld kan worden gebruikt in een boomstructuur.

Gewoonlijk heeft de interne tabel precies één record per knooppunt. In dat geval heeft de uitvoertabel hetzelfde aantal records. Soms zijn er echter knooppunten met meerdere bovenliggende knooppunten. Een knooppunt wordt dan gerepresenteerd door meerdere records in de invoertabel. In dat geval kan de uitvoertabel meer records hebben dan de invoertabel.

Alle knooppunten met een bovenliggende ID die niet is gevonden in de kolom node id (met inbegrip van knooppunten met ontbrekende bovenliggende ID) beschouwd worden als roots. Er worden dus alleen knooppunten geladen met een directe of indirecte verbinding met een root-knooppunt. Op deze wijze worden cirkelreferenties voorkomen.

U kunt aanvullende velden met de naam van het bovenliggende knooppunt en het pad of de diepte van het knooppunt maken.

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| NodeID | De naam van het veld dat de knooppunt-ID bevat. Dit veld moet bestaan in de invoertabel. |

| Argument | Beschrijving |
|---|---|
| ParentID | De naam van het veld dat de knooppunt-ID van het bovenliggende knooppunt bevat. Dit veld moet bestaan in de invoertabel. |
| NodeName | De naam van het veld dat de naam van het knooppunt bevat. Dit veld moet bestaan in de invoertabel. |
| ParentName | Een tekenreeks die wordt gebruikt als naam voor een nieuw veld **ParentName**. Als deze parameter wordt weggelaten, wordt dit veld niet gemaakt. |
| ParentSource | De naam van het veld dat de naam van het knooppunt bevat dat wordt gebruikt om het knooppuntpad te bouwen. Optionele parameter. Als deze parameter wordt weggelaten, wordt **NodeName** gebruikt. |
| PathName | Een tekenreeks die wordt gebruikt als naam voor het nieuwe veld **Path**. Hierin staat het pad van de root naar het knooppunt. Optionele parameter. Als deze parameter wordt weggelaten, wordt dit veld niet gemaakt. |
| PathDelimiter | Een tekenreeks die wordt gebruikt als scheidingsteken in het nieuwe veld **Path**. Optionele parameter. Als deze parameter wordt weggelaten, wordt '/' gebruikt. |
| Depth | Een tekenreeks die wordt gebruikt als naam voor het nieuwe veld **Depth**. Hiermee wordt aangegeven wat de diepte van het knooppunt in de hiërarchie is. Optionele parameter. Als deze parameter wordt weggelaten, wordt dit veld niet gemaakt. |

**Voorbeeld:**

```
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

];
```

| NodeID | 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 | Germany | Europe | Germany | - | Europe | Europe\Germany | 2 |
| 4 | 5 | UK | Europe | UK | - | Europe | Europe\UK | 2 |
| 5 | | Europe | Europe | - | - | - | Europe | 1 |

## HierarchyBelongsTo

Dit prefix wordt gebruikt om een hiërarchietabel met bovenliggende-onderliggende objecten om te zetten in een tabel die bruikbaar is in een Qlik Sense-gegevensmodel. Het prefix kan voor een **LOAD**- of **SELECT**-opdracht worden geplaatst en gebruikt het resultaat van de laadopdracht als invoer voor een tabeltransformatie.

Met het prefix maakt u een tabel waarin alle relaties tussen voorgangers en onderliggende elementen van de hiërarchie staan. Met de voorgangervelden kunt u vervolgens hele bomen in de hiërarchie selecteren. De uitvoertabel bevat in de meeste gevallen meerdere records per knooppunt.

**Syntaxis:**

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

De invoertabel moet een aangrenzende knooppuntentabel zijn. Aangrenzende knooppuntentabellen zijn tabellen waarin elk record correspondeert met een knooppunt en een veld heeft dat de referentie naar het bovenliggende knooppunt bevat. In een dergelijke tabel wordt het knooppunt opgeslagen in slechts één record, maar kan het knooppunt wel meerdere onderliggende knooppunten hebben. De tabel kan natuurlijk aanvullende velden bevatten waarin kenmerken van de knooppunten worden beschreven.

Met het prefix maakt u een tabel waarin alle relaties tussen voorgangers en onderliggende elementen van de hiërarchie staan. Met de voorgangervelden kunt u vervolgens hele bomen in de hiërarchie selecteren. De uitvoertabel bevat in de meeste gevallen meerdere records per knooppunt.

U kunt een extra veld maken waarin het diepteverschil van de knooppunten wordt aangegeven.

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| NodeID | De naam van het veld dat de knooppunt-ID bevat. Dit veld moet bestaan in de invoertabel. |
| ParentID | De naam van het veld dat de knooppunt-ID van het bovenliggende knooppunt bevat. Dit veld moet bestaan in de invoertabel. |

| Argument | Beschrijving |
|---|---|
| NodeName | De naam van het veld dat de naam van het knooppunt bevat. Dit veld moet bestaan in de invoertabel. |
| AncestorID | Een tekenreeks die wordt gebruikt voor het geven van een naam aan het nieuwe veld ancestor-ID. Hierin staat de ID van het voorgangerknooppunt. |
| AncestorName | Een tekenreeks die wordt gebruikt voor het geven van een naam aan het nieuwe veld ancestor-ID. Hierin staat de naam van het voorgangerknooppunt. |
| DepthDiff | Een tekenreeks die wordt gebruikt als naam voor het nieuwe veld **DepthDiff**. Hiermee wordt aangegeven wat de diepte van het knooppunt in de hiërarchie in relatie tot het voorgangerknooppunt. Optionele parameter. Als deze parameter wordt weggelaten, wordt dit veld niet gemaakt. |

**Voorbeeld:**

```
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 |

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

## Inner

De prefixen **join** en **keep** kunnen worden voorafgegaan door het prefix **inner**. Indien gebruikt voor **join**, geeft de opdracht aan dat een inner join moet worden gebruikt. De resulterende tabel bevat dan alleen gecombineerde veldwaarden uit de tabellen met onbewerkte gegevens waarvoor de koppelende veldwaarden in beide tabellen voorkomen. Indien gebruikt voor **keep**, geeft de opdracht aan dat beide tabellen met onbewerkte gegevens worden gereduceerd tot de gemeenschappelijke doorsnede voordat ze worden opgeslagen in Qlik Sense.

**Syntaxis:**

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

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| tablename | De benoemde tabel die moet worden vergeleken met de geladen tabel. |
| loadstatementof selectstatement | De **LOAD** of **SELECT**-opdracht voor de geladen tabel. |

Voorbeeld

### Load-script

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
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 ];
```

### Resultaat

Resultaattabel

| Kolom1 | Kolom2 | Kolom3 |
|---|---|---|
| A | B | C |
| 1 | aa | xx |

### Uitleg

Dit voorbeeld toont de uitvoer voor inner join waarbij alleen waarden die aanwezig zijn in de eerste (links) en de tweede (rechts) tabellen zijn samengevoegd.

## IntervalMatch

Het prefix **IntervalMatch** wordt gebruikt voor het maken van een tabel waarin discrete numerieke waarden worden gekoppeld aan een of meer numerieke intervallen, en waarin de waarden optioneel worden gekoppeld aan een of meer extra sleutels.

**Syntaxis:**

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

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

Het prefix **IntervalMatch** moet voor een **LOAD**- of **SELECT**-opdracht worden geplaatst waarmee de intervallen worden geladen. Het veld met de discrete gegevenspunten (Tijd in het voorbeeld verderop) en extra sleutels moet al in Qlik Sense geladen zijn vóór de opdracht met het prefix **IntervalMatch**. Het prefix leest dit veld niet zelf uit de databasetabel. Het prefix zet de geladen tabel met intervallen en sleutels om in een tabel die een extra kolom bevat: de discrete numerieke gegevenspunten. Tevens wordt het aantal records zodanig uitgebreid dat de nieuwe tabel één record per mogelijke combinatie van discreet gegevenspunt, interval en waarde van het sleutelveld of de sleutelvelden bevat.

De intervallen kunnen elkaar overlappen en de discrete waarden worden aan alle relevante intervallen gekoppeld.

Als het prefix IntervalMatch  wordt uitgebreid met sleutelvelden, wordt het gebruikt voor het maken van een tabelkoppeling van discrete numerieke waarden met een of meer numerieke intervallen, terwijl tegelijkertijd de waarden van een of meer extra sleutels worden gekoppeld.

Om te voorkomen dat ongedefinieerde intervallimieten worden genegeerd, moet u mogelijk NULL-waarden toewijzen aan andere velden met de onderste en bovenste limieten voor het interval. Dit kan

worden afgehandeld met de opdracht **NullAsValue** of door een expliciete test waarbij NULL-waarden worden vervangen door een numerieke waarde ruim voor of na een van de discrete numerieke gegevenspunten.

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| matchfield | Het veld met de discrete numerieke waarden voor koppeling aan intervallen. |
| keyfield | Velden met de extra kenmerken die bij de transformatie moeten worden gekoppeld. |
| loadstatementor selectstatement | Moet resulteren in een tabel waarvan het eerste veld de laagste limiet van elk interval bevat en het twee veld de hoogste limiet. Als sleutelkoppeling wordt gebruikt, moeten het derde veld en de volgende velden de sleutelvelden bevatten van de opdracht **IntervalMatch**. De intervallen zijn altijd gesloten, dat wil zeggen dat de eindpunten in het interval zijn opgenomen. Niet-numerieke limieten zorgen ervoor dat het interval buiten beschouwing wordt geladen (ongedefinieerd). |

**Example 1:**

In de twee onderstaande tabellen wordt in de eerste een aantal discrete gebeurtenissen weergegeven en worden in de tweede de begin- en eindtijden voor de productie van verschillende bestellingen gedefinieerd. Met behulp van het prefix **IntervalMatch** kan een logische verbinding tot stand worden gebracht tussen de twee tabellen om bijvoorbeeld uit te zoeken welke bestellingen last hadden van storingen en welke bestellingen door welke ploegen zijn verwerkt.

```
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;
```

De tabel **OrderLog** bevat nu een extra kolom: *Time*. Het aantal records is eveneens uitgebreid.

Table with additional column

| Time | Start | End | Order |
|---|---|---|---|
| 00:00 | - | - | - |
| 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: (met keyfield)

Hetzelfde voorbeeld als boven, waarbij *ProductionLine* is toegevoegd als sleutelveld.

```
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

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;
```

Het volgende tabelvak kan nu worden gemaakt:

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

Het prefix **join** koppelt de geladen tabel aan een bestaande benoemde tabel of de laatste eerder gemaakte gegevenstabel.

Het effect van het samenvoegen van gegevens is dat de doeltabel wordt uitgebreid met een extra set velden of attributen, namelijk die welke nog niet in de doeltabel aanwezig zijn. Alle gemeenschappelijke veldnamen tussen de brongegevensset en de doeltabel worden gebruikt om uit te zoeken hoe de nieuwe binnenkomende records moeten worden gekoppeld. Dit wordt gewoonlijk een 'natuurlijke join' genoemd. Een Qlik-samenvoegbewerking kan ertoe leiden dat de resulterende doeltabel meer of minder records heeft dan waarmee hij begon, afhankelijk van de uniekheid van de join-koppeling en het type join dat wordt gebruikt.

Er zijn vier soorten joins:

#### Left join

Left joins zijn het meest voorkomende join-type. Als u bijvoorbeeld een transactiegegevensset hebt en deze wilt combineren met een referentiegegevensset, gebruikt u doorgaans een `Left Join`. U zou eerst de transactietabel laden en vervolgens de referentiegegevensset laden terwijl u deze via een `Left Join`-voorvoegsel aan de reeds geladen transactietabel toevoegt. Een `Left Join` zou alle transacties ongewijzigd laten en de aanvullende referentiegegevensvelden toevoegen waar een overeenkomst wordt gevonden.

#### Inner join

Als u twee datasets hebt waarbij u alleen om resultaten geeft als er een overeenkomende koppeling is, overweeg dan om een `Inner Join` te gebruiken. Hierdoor worden alle records verwijderd uit zowel de geladen brongegevens als de doeltabel als er geen overeenkomst wordt gevonden. Als gevolg hiervan kan uw doeltabel minder records bevatten dan voordat de samenvoegbewerking plaatsvond.

#### Outer join

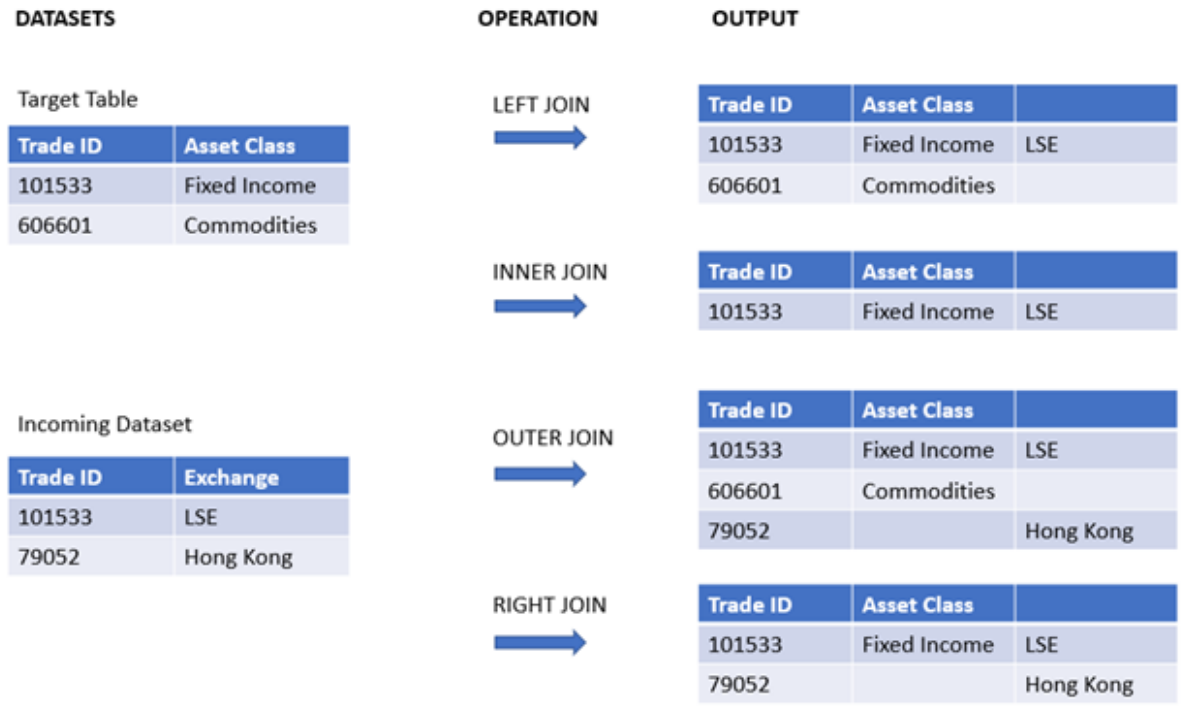
Als u zowel de doelrecords als alle inkomende records moet bewaren, gebruikt u een `Outer Join`. Als er geen overeenkomst wordt gevonden, wordt elke set records nog steeds bewaard, terwijl de velden aan de andere kant van de join niet ingevuld blijven (null).

Als het type trefwoord wordt weggelaten, dan is het standaard join-type een outer join.

#### Right join

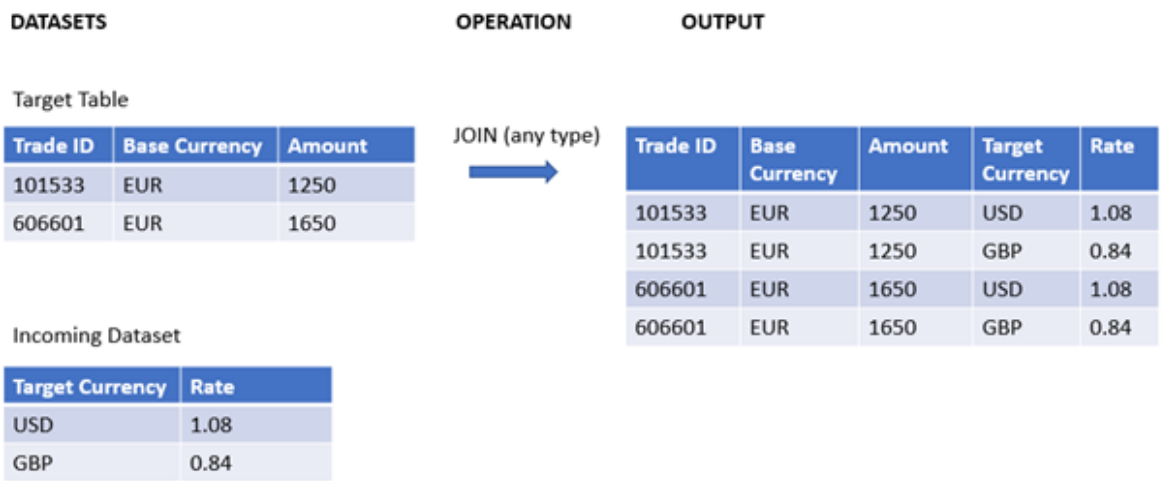
Dit type join houdt alle records die op het punt staan te worden geladen, terwijl de records in de tabel waarop de join is gericht, worden beperkt tot alleen die records waarbij er een koppelingsovereenkomst is in de inkomende records. Dit is een niche-jointype dat soms wordt gebruikt om een reeds vooraf geladen tabel met records in te korten tot een vereiste subset.

*Voorbeelden van resultaatsets van verschillende typen samenvoegbewerkingen*




*Als er geen gemeenschappelijke veldnamen zijn tussen de bron en het doel van een samenvoegbewerking, resulteert de join in een cartesiaans product van alle rijen - dit wordt een 'cross join' genoemd.*

*Voorbeeld resultaatset van een 'cross join'-bewerking*




**Syntaxis:**

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

Argumenten

| Argument | Beschrijving |
|---|---|
| tablename | De benoemde tabel die moet worden vergeleken met de geladen tabel. |
| loadstatementof selectstatement | De **LOAD** of **SELECT**-opdracht voor de geladen tabel. |

Deze onderwerpen kunnen u helpen bij het werken met deze functie:

Gerelateerde onderwerpen

| Onderwerp | Beschrijving |
|---|---|
| **Tabellen samenvoegen met Join en Keep** in *Gegevens beheren* | In dit onderwerp vindt u meer uitleg over de concepten 'samenvoegen' en 'bewaren' van gegevenssets. |
| *Keep (page 85)* | Het laadvoorvoegsel `Keep` is vergelijkbaar met het voorvoegsel `Join`, maar combineert de bron- en doelgegevenssets niet. In plaats daarvan trimt het elke gegevensverzameling op basis van het type bewerking dat is aangenomen (binnen, buiten, links of rechts). |

## Voorbeeld 1 - Left join: een doeltabel verrijken met een referentiegegevensverzameling

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset die wijzigingsrecords vertegenwoordigt, die in een tabel met de naam `Changes` wordt geladen. Bevat een sleutelveld Status-id.
- Een tweede gegevensset die wijzigingsstatussen vertegenwoordigt, die wordt geladen en gecombineerd met de originele wijzigingsrecords door deze te combineren met een linker `Join`-laadvoorvoegsel.

Deze left join zorgt ervoor dat de wijzigingsrecords intact blijven terwijl er statusattributen worden toegevoegd waar een match in de inkomende statusrecords wordt gevonden op basis van een gemeenschappelijk Status-id.

### Load-script

```
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');

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');
```

### Resultaten

Open de gegevensmodelviewer en let op de vorm van het gegevensmodel. Er is slechts één gedenormaliseerde tabel aanwezig. Het is een combinatie van alle originele wijzigingsrecords, met de overeenkomende statusattributen samengevoegd met elk wijzigingsrecord.

Resulterend intern gegevensmodel

| **Wijzigingen** |
| --- |
| Wijzigings-id |
| Status-id |
| Geplande begindatum |
| Geplande einddatum |
| Bedrijfsimpact |
| Status |

| Wijzigingen |
| --- |
| Substatus |

Als u het voorbeeldvenster in de gegevensmodelviewer uitvouwt, ziet u een deel van deze volledige resultatenset, georganiseerd in een tabel:

Voorbeeld van de tabel Wijzigingen in de gegevensmodelviewer

| Wijzigings-id | Status-id | Geplande begindatum | Geplande einddatum | Bedrijfsimpact | Status | Substatus |
| --- | --- | --- | --- | --- | --- | --- |
| 10030 | 4 | 19/01/2022 | 23/02/2022 | Geen | Gesloten | Geannuleerd |
| 10031 | 3 | 20/01/2022 | 25/03/2022 | Laag | Gesloten | Voltooid |
| 10015 | 3 | 04/01/2022 | 15/02/2022 | Laag | Gesloten | Voltooid |
| 10103 | 1 | 02/04/2022 | 29/05/2022 | Gemiddeld | Open | Niet gestart |
| 10116 | 1 | 15/04/2022 | 24/04/2022 | Geen | Open | Niet gestart |
| 10134 | 1 | 03/05/2022 | 08/07/2022 | Laag | Open | Niet gestart |
| 10264 | 1 | 10/09/2022 | 17/10/2022 | Gemiddeld | Open | Niet gestart |
| 10040 | 1 | 29/01/2022 | 22/04/2022 | Geen | Open | Niet gestart |
| 10323 | 1 | 08/11/2022 | 26/11/2022 | Hoog | Open | Niet gestart |
| 10187 | 2 | 25/06/2022 | 24/08/2022 | Laag | Open | Gestart |
| 10185 | 2 | 23/06/2022 | 08/09/2022 | Geen | Open | Gestart |
| 10220 | 2 | 28/07/2022 | 06/09/2022 | Geen | Open | Gestart |
| 10326 | 2 | 11/11/2022 | 05/12/2022 | Geen | Open | Gestart |
| 10138 | 2 | 07/05/2022 | 03/08/2022 | Geen | Open | Gestart |
| 10334 | 2 | 19/11/2022 | 06/02/2023 | Laag | Open | Gestart |

Omdat de vijfde rij in de tabel Status (Status-id: '5', Status: 'Closed', substatus: 'Obsolete') niet overeenkomt met records in de tabel Changes, verschijnen de gegevens in deze rij niet in de bovenstaande resultaten.

Ga terug naar de editor voor laden van gegevens. Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `Status`.

Voeg deze meting toe:

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

Nu kunt u het aantal wijzigingen per status bekijken.

Resultatentabel

| Status | =Aantal([Wijzigings-ID]) |
|---|---|
| Open | 12 |
| Gesloten | 3 |

## Voorbeeld 2 – Inner join: alleen overeenkomende records combineren

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset die wijzigingsrecords vertegenwoordigt, die in een tabel met de naam `Changes` wordt geladen.
- Een tweede gegevensset die wijzigingsrecords uit het bronsysteem `JIRA` weergeeft. Deze wordt geladen en gecombineerd met de oorspronkelijke records door deze samen te voegen met een `Inner Join`-laadvoorvoegsel.

Dit `Inner Join` zorgt ervoor dat alleen de vijf wijzigingsrecords die in beide gegevenssets voorkomen, worden bewaard.

### Load-script

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

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `Source System`
- `Change ID`
- `Business Impact`

Nu kunt u de vijf resulterende records inspecteren. De resulterende tabel van een `Inner Join` bevat alleen records met overeenkomende gegevens in beide gegevensverzamelingen.

Resultatentabel

| Bronsysteem | Wijzigings-id | Bedrijfsimpact |
|---|---|---|
| JIRA | 10030 | Geen |
| JIRA | 10134 | Laag |
| JIRA | 10220 | Geen |
| JIRA | 10323 | Hoog |
| JIRA | 10334 | Laag |

## Voorbeeld 3 – Outer join: overlappende recordsets combineren

#### Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset die wijzigingsrecords vertegenwoordigt, die in een tabel met de naam `Changes` wordt geladen.

- Een tweede gegevensset die wijzigingsrecords uit het bronsysteem `JIRA` weergeeft, die wordt geladen en gecombineerd met de oorspronkelijke records door deze samen te voegen met een `Outer Join`-laadvoorvoegsel.

Dit zorgt ervoor dat alle overlappende wijzigingsrecords van beide gegevenssets behouden blijven.

### Load-script

```
// 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');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `Source System`
- `Change ID`
- `Business Impact`

Nu kunt u de 10 resulterende records inspecteren.

Resultatentabel

| Bronsysteem | Wijzigings-id | Bedrijfsimpact |
|---|---|---|
| JIRA | 10030 | Geen |
| JIRA | 10134 | Laag |
| JIRA | 10220 | Geen |
| JIRA | 10323 | - |
| JIRA | 10334 | Laag |
| JIRA | 10597 | - |
| - | 10015 | Laag |
| - | 10031 | Laag |
| - | 10040 | Geen |
| - | 10138 | Geen |

## Voorbeeld 4 – Right join: een doeltabel inkorten door een secundaire mastergegevensset

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset die wijzigingsrecords vertegenwoordigt, die in een tabel met de naam `Changes` wordt geladen.
- Een tweede gegevensverzameling met alle wijzigingsrecords van het bronsysteem `Teamwork`. Dit wordt geladen en met de originele records gecombineerd met behulp van een `Right Join`-laadprefix.

Dit zorgt ervoor dat alleen `Teamwork`-wijzigingsrecords worden bewaard, terwijl er geen `Teamwork`-records verloren gaan als de doeltabel geen overeenkomende `Change ID` heeft.

### Load-script

```
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');

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');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `Source System`
- `Change ID`
- `Business Impact`

Nu kunt u de vijf resulterende records inspecteren.

Resultatentabel

| Bronsysteem | Wijzigings-id | Bedrijfsimpact |
|---|---|---|
| Teamwerk | 10015 | Laag |
| Teamwerk | 10031 | Laag |
| Teamwerk | 10040 | Geen |
| Teamwerk | 10103 | Gemiddeld |
| Teamwerk | 50231 | - |

## Keep

Het prefix **keep** is vergelijkbaar met het prefix **join**. Net als bij het prefix **join** wordt de geladen tabel vergeleken met een bestaande benoemde tabel of de laatste eerder gemaakte gegevenstabel. Maar de geladen tabel wordt niet samengevoegd met een bestaande tabel, in plaats daarvan wordt een tabel of worden beide tabellen voordat ze worden opgeslagen in Qlik Sense gereduceerd op basis van de doorsnede van de tabelgegevens. De uitgevoerde vergelijking is equivalent aan een natuurlijke join voor alle gemeenschappelijke velden, net zoals bij een overeenkomstige join. De twee tabellen worden echter niet samengevoegd, maar als twee afzonderlijk benoemde tabellen in Qlik Sense bewaard.

**Syntaxis:**

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

Het prefix **keep** moet worden voorafgegaan door een van de volgende prefixen **inner**, **left** of **right**.

Het expliciete prefix **join** in de scripttaal van Qlik Sense zorgt voor een volledige koppeling van de twee tabellen. Het resultaat is één tabel. Dergelijke samenvoegingen resulteren vaak in zeer grote tabellen. Een van belangrijkste functies in Qlik Sense is de mogelijkheid om meerdere tabellen met elkaar te associëren in plaats van ze samen te voegen. Dit vereist veel minder geheugen, versnelt de verwerking en biedt een enorme flexibiliteit. Expliciete joins kunnen dan ook beter worden vermeden in Qlik Sense-scripts. De functionaliteit van keep is ontworpen om het aantal gevallen waar expliciete joins moeten worden gebruikt te verkleinen.

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| tablename | De benoemde tabel die moet worden vergeleken met de geladen tabel. |
| loadstatementof selectstatement | De **LOAD** of **SELECT**-opdracht voor de geladen tabel. |

**Voorbeeld:**

```
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

De prefixen **Join** en **Keep** kunnen worden voorafgegaan door het prefix **left**.

Gebruikt voor **join** geeft het prefix aan dat er een join links moet plaatsvinden. De resulterende tabel bevat dan alleen gecombineerde veldwaarden uit de tabellen met onbewerkte gegevens waarvoor de koppelende veldwaarden in de eerste tabel voorkomen. Als het prefix wordt gebruikt vóór **keep**, geeft het aan dat de tweede tabel met onbewerkte gegevens wordt gereduceerd tot de gemeenschappelijke doorsnede met de eerste tabel, voordat deze wordt opgeslagen in Qlik Sense.

*Was u op zoek naar de tekenreeksfunctie met dezelfde naam? Zie: Left (page 1805)*

**Syntaxis:**

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

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| tablename | De benoemde tabel die moet worden vergeleken met de geladen tabel. |
| loadstatementof selectstatement | De **LOAD** of **SELECT**-opdracht voor de geladen tabel. |

Voorbeeld

### Load-script

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
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 ];
```

### Resultaat

Resultaattabel

| Kolom1 | Kolom2 | Kolom3 |
|---|---|---|
| A | B | C |
| 1 | aa | xx |
| 2 | cc | - |
| 3 | ee | - |

### Uitleg

Dit voorbeeld toont de uitvoer voor left join waarbij alleen waarden die aanwezig zijn in de eerste (links) tabel zijn samengevoegd.

## Mapping

Het prefix **mapping** wordt gebruikt om een toewijzingstabel te maken die bijvoorbeeld kan worden gebruikt voor het vervangen van veldwaarden en veldnamen tijdens de uitvoering van het script.

**Syntaxis:**

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

Het prefix **mapping** kan voor een **LOAD**- of **SELECT**-opdracht worden gezet en gebruikt het resultaat van de laadopdracht als toewijzingstabel. Toewijzing vormt een efficiënt manier om veldwaarden te vervangen tijdens de uitvoering van het script, zoals het vervangen van US, U.S. of Amerika door USA. Een toewijzingstabel bestaat uit twee kolommen, de eerste met vergelijkingswaarden en de tweede met de gewenste toewijzingswaarden. Toewijzingstabellen worden tijdelijk in het geheugen opgeslagen en automatisch gewist na uitvoering van het script.

De inhoud van de toewijzingstabel is toegankelijk via onder andere de opdracht **Map ... Using**, de opdracht **Rename Field**, de functie **Applymap()** of de functie **Mapsubstring()**.

**Voorbeeld:**

In dit voorbeeld laden wij een lijst met verkopers waarbij een landcode het land aangeeft waar zijn wonen. Wij gebruiken een tabel waarin een landcode wordt toegewezen aan een land ter vervanging van de landcode door de landnaam. Er zijn slechts drie landen gedefinieerd in de toewijzingstabel, andere landcode worden toegewezen aan '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';
```

De resulterende tabel ziet er als volgt uit:

Mapping table

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

## Merge

Het voorvoegsel **Merge** kan aan elke **LOAD**- of **SELECT**-opdracht in het script worden toegevoegd om op te geven dat de geladen tabel met een andere tabel moet worden samengevoegd. Dit geeft ook aan dat deze opdracht met behulp van gedeeltelijk laden moet worden uitgevoerd.

Deze instructie wordt meestal gebruikt als u een wijzigingslogboek laadt en inserts, updates en deletes wilt toepassen op een bestaande tabel.

> *Om ervoor te zorgen dat een gedeeltelijke lading correct werkt, moet de app met gegevens worden geopend voordat een gedeeltelijke lading wordt geactiveerd.*

Voer een gedeeltelijke lading uit met de knop **Opnieuw laden**. U kunt ook de Qlik Engine JSON API gebruiken.

**Syntaxis:**

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

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| only | Een optionele kwalificatie die aangeeft dat de opdracht alleen moet worden uitgevoerd tijdens het gedeeltelijk laden van gegevens. De opdracht wordt genegeerd tijdens het normaal (niet-gedeeltelijk) laden van gegevens. |
| SequenceNoField | De naam van het veld dat een tijdstempel bevat of een volgnummer dat de volgorde van de bewerkingen definieert. |
| SequenceNoVar | De naam van de variabele die wordt toegewezen aan de maximumwaarde voor SequenceNoField van de tabel die wordt samengevoegd. |
| ListOfKeys | Een door komma's gescheiden lijst met veldnamen waarin de primaire sleutel is opgegeven. |
| Operation | Het eerste veld van de lOAD-instructie moet de bewerking als tekstreeks bevatten: 'Insert', 'Update' of 'Delete'. ‘i’, ‘u’ en ‘d’ worden ook geaccepteerd. |

## Algemene functionaliteit

Tijdens een normale (niet-gedeeltelijke) lading zal de constructie **Merge LOAD** worden uitgevoerd als een normale **LOAD**-instructie, maar zorgt dit ook voor de verwijdering van verouderde records en records die zijn gemarkeerd voor verwijdering. Het eerste veld van de **LOAD**-instructie moet informatie over de bewerking bevatten: Insert, Update of Delete.

Voor elke geladen record, wordt de identificatie van een record vergelijken met eerder geladen records en alleen het laatste record (op basis van het volgnummer) wordt behouden. Als het laatste record is gemarkeerd met Delete, worden er geen record behouden.

### Doeltabel

Welke tabel moet worden gewijzigd, wordt bepaald door de reeks velden. Als een tabel met dezelfde reeks velden (met uitzondering van het eerste veld; de bewerking) al bestaat, wordt dit de tabel die u moet wijzigen. U kunt ook het prefix **Concatenate** gebruiken om de tabel op te geven. Als de doeltabel niet is bepaald, wordt het resultaat van de **Merge LOAD**-constructie opgeslagen in een nieuwe tabel.

Als het prefix Concatenate wordt gebruikt, bevat de resulterende tabel een reeks velden die overeenkomt met de verbintenis van de bestaande tabel en de invoer voor de samenvoeging De doeltabel kan daardoor meer tabellen krijgen dan het wijzigingslogboek dat is gebruik als invoer voor de samenvoeging.

Bij een gedeeltelijke lading gebeurt hetzelfde als bij een volledige lading. Het enige verschil is dat bij een gedeeltelijke lading vrijwel nooit een nieuwe tabel wordt gemaakt. Tenzij u de **Only**-clausule hebt gebruikt, bestaat er altijd een doeltabel met dezelfde reek velden van de vorige uitvoering van het script.

### Volgnummer

Als het geladen wijzigingslogboek een samengevoegd logboek is, dan bevat deze wijzigingen die al zijn geladen, de parameter SequenceNoVar kan worden gebruikt in een **Where**-clausule om een limiet in te stellen voor de invoergegevens. De constructie **Merge LOAD** kan dan alleen worden gebruikt voor laadrecords waarbij het veld SequenceNoField groter is dan SequenceNoVar. Na voltooiing wijst de constructie **Merge LOAD** een nieuwe waarde toe aan de SequenceNoVar met de maximumwaarde die te zien is in het veld SequenceNoField.

### Bewerkingen

De **Merge LOAD** kan minder velden bevatten dan de doeltabel. De verschillende bewerkingen gaan anders met missende velden om:

**Invoegen**: ontbrekende velden in de **Merge LOAD**, die wel bestaan in de doeltabel, krijgen een NULL-waarde in de doeltabel.

**Verwijderen**: ontbrekende velden hebben geen invloed op het resultaat. De relevante records worden toch verwijderd.

**Bijwerken**: velden die worden weergegeven in de **Merge LOAD** worden bijgewerkt in de doeltabel. Ontbrekende velden worden niet gewijzigd. Dit betekent dat de twee volgende opdrachten niet identiek zijn:

- 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 ...;

Met de eerste opdracht worden de weergegeven records bijgewerkt en F2 gewijzigd in NULL. Bij de tweede opdracht wordt F2 niet gewijzigd, maar blijven de waarden van de doeltabel behouden.

#### Voorbeelden

## Voorbeeld 1: eenvoudige samenvoeging met opgegeven tabel

In dit voorbeeld wordt een uitgelijnde tabel met vier rijen met de naam Persons geladen. Met **Merge** wordt de tabel als volgt gewijzigd:

- De rij *Mary, 4* wordt toegevoegd.
- De rij *Steven, 3* wordt verwijderd.
- Het getal *5* wordt toegewezen aan *Jake*.

De variabele *LastChangeDate* wordt ingesteld op de maximumwaarde in de kolom *ChangeDate* nadat **Merge** is uitgevoerd.

### Load-script

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

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

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
];
```

## Resultaat

Voorafgaand aan de lading **Merge Load** verschijnt de volgende tabel:

Resulting table

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

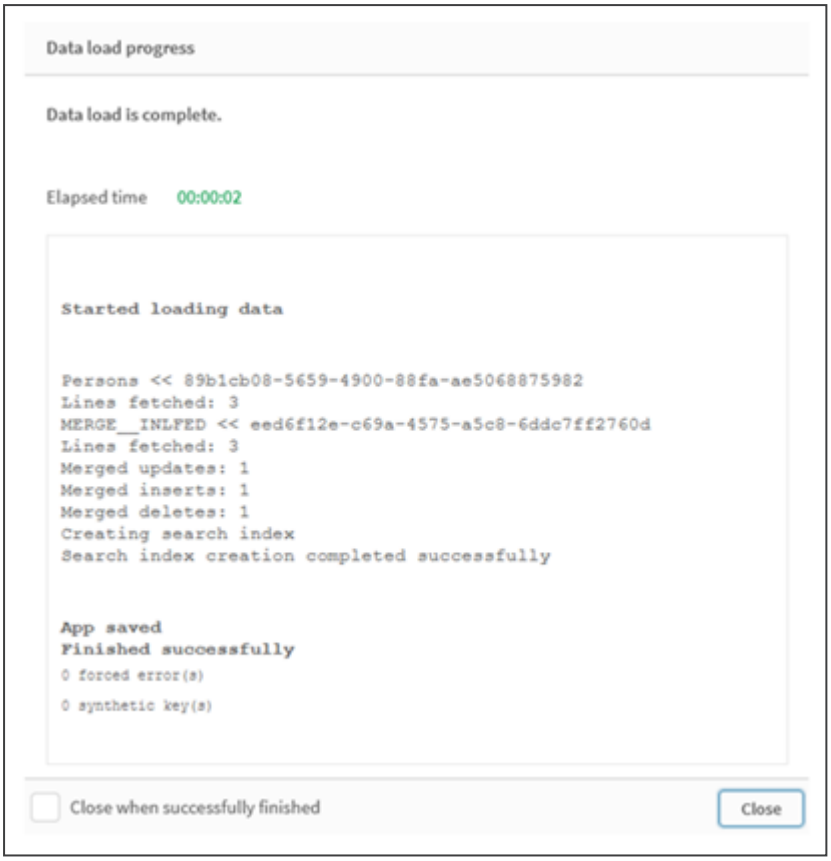
Na de **Merge Load** verschijnt de volgende tabel:

Resulting table

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

Als de gegevens zijn geladen toont het dialoogvenster **Voortgang laden van gegevens** de bewerkingen die worden uitgevoerd:

*Dialoogvenster Voortgang laden van gegevens*




## Voorbeeld 2: load-script voor gegevens met ontbrekende velden

In dit voorbeeld worden dezelfde gegevens als hierboven geladen, maar nu met een id voor elke persoon.

Met **Merge** wordt de tabel als volgt gewijzigd:

- De rij *Mary, 4* wordt toegevoegd.
- De rij *Steven, 3* wordt verwijderd.
- Het getal *5* wordt toegewezen aan *Jake*.
- Het getal *6* wordt toegewezen aan *Jill*.

### Load-script

Hier gebruiken we twee **Merge LOAD**-opdrachten, een voor 'Invoegen' en 'Verwijderen' en een voor het 'Bijwerken'.

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

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

```
];

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
];
```

### Resultaat

Na de **Merge LOAD**-opdrachten verschijnt de volgende tabel:

Resulting table

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

De tweede **Merge**-opdracht bevat niet het veld **Naam** en daardoor zijn de namen mogelijk niet gewijzigd.

## Voorbeeld 3: load-script voor gegevens - gedeeltelijke lading met een Where-clausule met ChangeDate

In het volgende voorbeeld, geeft het argument **Only** op dat de instructie **Merge** alleen wordt uitgevoerd tijdens een gedeeltelijke lading. Updates worden gefilterd op basis van de eerder vastgelegde LastChangeDate. Nadat **Merge** is voltooid, wordt de variabel LastChangeDate toegewezen aan de maximumwaarde van de kolom ChangeDate die tijdens het samenvoegen is verwerkt

### Load-script

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

# NoConcatenate

Het prefix **NoConcatenate** zorgt ervoor dat twee geladen tabellen met identieke veldensets worden behandeld als twee aparte interne tabellen, terwijl ze anders automatisch zouden worden aaneengeschakeld.

**Syntaxis:**

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

Als een tabel met een identiek aantal velden en overeenkomende veldnamen eerder in het script in een tabel is geladen, zal Qlik Sense standaard deze twee tabellen automatisch aaneenschakelen. Dit gebeurt ook als de tweede tabel een andere naam heeft.

Maar als het script-prefix `NoConcatenate` is ingevoegd vóór de load-opdracht of de selecteeropdracht van de tweede tabel, dan worden deze twee tabellen afzonderlijk geladen.

Een typische gebruikssituatie voor `NoConcatenate` is wanneer u mogelijk een tijdelijke kopie van een tabel moet maken om een aantal tijdelijke transformaties in die kopie uit te voeren, terwijl u een kopie van de oorspronkelijke gegevens bewaart. `NoConcatenate` zorgt ervoor dat u die kopie kunt maken zonder deze impliciet weer toe te voegen aan de brontabel.

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

Functievoorbeeld

| Voorbeeld | Resultaat |
|---|---|
| `Source: LOAD A,B from file1.csv; CopyOfSource: NoConcatenate LOAD A,B resident Source;` | Er wordt een tabel met `A` en `B` als metingen geladen. Een tweede tabel met dezelfde velden wordt afzonderlijk geladen door gebruik van de `NoConcatenate`-variabele. |

## Voorbeeld 1 – Impliciete aaneenschakeling

Load-script en resultaten

### Overzicht

In dit voorbeeld voegt u twee load-scripts achter elkaar toe.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een eerste gegevensverzameling met datums en bedragen die naar de tabel `Transactions` wordt gestuurd.

### Eerste load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

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

Eerste resultatentabel

| 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 |

### Tweede load-script

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een tweede gegevensset met identieke velden wordt naar de tabel `Sales` gestuurd.

```
Sales:
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
];
```

### Resultaten

Laad de gegevens en ga naar de tabel.

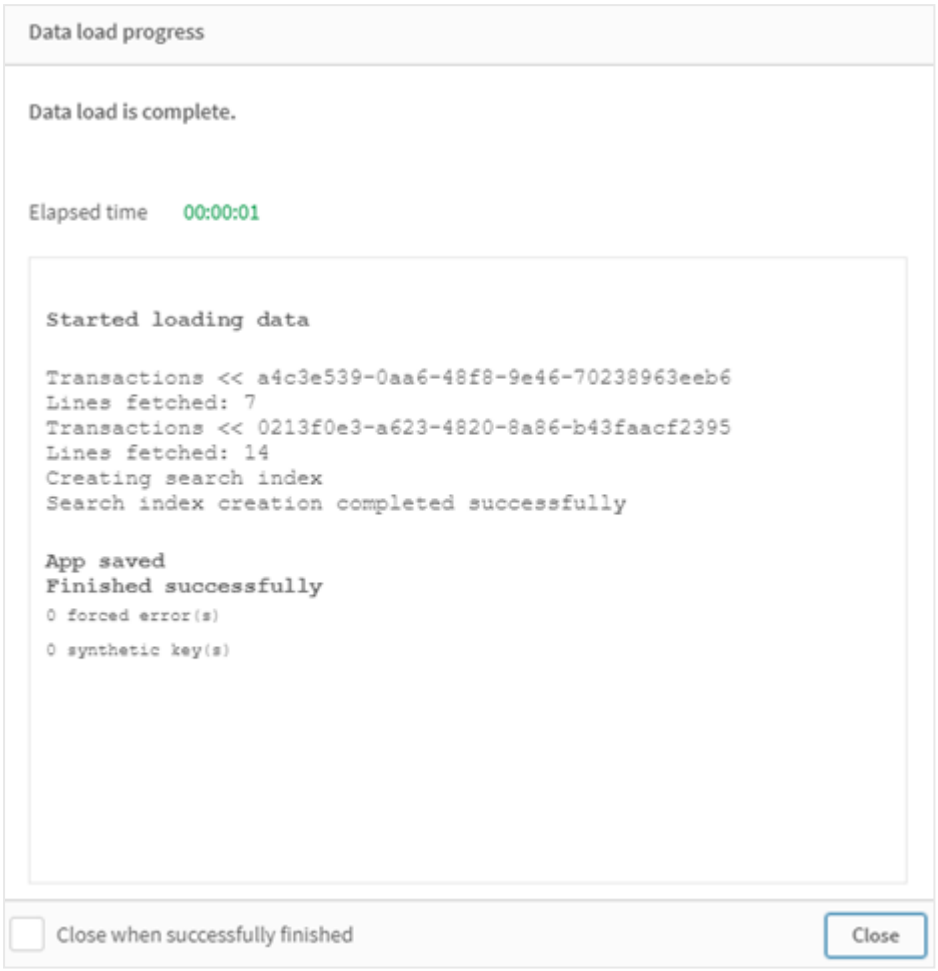
Tweede resultatentabel

| 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 |
| 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 |

Wanneer het script wordt uitgevoerd, wordt de tabel `Sales` impliciet aaneengeschakeld met de bestaande `Transactions`-tabel omdat de twee gegevensset een identiek aantal velden delen met identieke veldnamen. Dit gebeurt ondanks het feit dat het tweede tabelnaamlabel de resultatenset `'Sales'` probeert te noemen.

Als u naar het logbestand **Voortgang laden van gegevens** kijkt, kunt u zien dat de verkoopgegevensset impliciet aaneengeschakeld wordt.

*Het logbestand Voortgang laden van gegevens toont dat transactiegegevens impliciet aaneengeschakeld worden.*




## Voorbeeld 2 – Scenario voor gebruikssituatie

Load-script en resultaten

### Overzicht

In deze gebruikssituatie hebt u:

- Een transactiegegevensset met:
  - id
  - date
  - bedrag (in GBP)
- Een valutatabel met:
  - Wisselkoers voor USD naar GBP
- Een tweede transactiegegevensset met:
  - id

- date
- bedrag (in USD)

U laadt vijf scripts achter elkaar.

- Het eerste load-script bevat een aanvankelijke gegevensset met datums en bedragen in GBP, die naar de tabel `Transactions` wordt gestuurd.
- Het tweede load-script bevat:
  - Een tweede gegevensset met datums en bedragen in USD, die naar de tabel `Transactions_in_USD` wordt gestuurd.
  - Het `noconcatenate`-prefix dat vóór de load-instructie van de `Transactions_in_USD`-gegevensset is geplaatst om impliciete aaneenschakeling te voorkomen.
- Het derde load-script bevat het `join`-prefix dat wordt gebruikt om een wisselkoers op te zetten tussen GBP en USD in de `Transactions_in_USD`-tabel.
- Het vierde load-script bevat het `concatenate`-prefix dat de `Transactions_in_USD` toevoegt aan de aanvankelijke `Transactions`-tabel.
- Het vijfde load-script bevat de `drop table`-opdracht die de `Transactions_in_USD`-tabel verwijdert waaruit gegevens zijn aaneengeschakeld met de `Transactions`-tabel.

### Eerste load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

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

Resultaten eerste load-script

| id | date | amount |
|---|---|---|
| 1 | 12/30/2018 | 23.56 |
| 2 | 12/07/2018 | 556.31 |

| id | date | amount |
|---|---|---|
| 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 |

De tabel toont de eerste gegevensset met bedragen in GBP.

### Tweede load-script

```
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
];
```

### Resultaten

Laad de gegevens en ga naar de tabel.

Resultaten tweede load-script

| 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 |
| 8 | 01/01/2019 | 164.27 |
| 9 | 01/03/2019 | 384.00 |
| 10 | 01/06/2019 | 25.82 |
| 11 | 01/09/2019 | 312.00 |

| id | date | amount |
|---|---|---|
| 12 | 01/15/2019 | 4.56 |
| 13 | 01/16/2019 | 90.24 |
| 14 | 01/18/2019 | 19.32 |

U ziet dat de tweede gegevensset uit de `Transactions_in_USD`-tabel is toegevoegd.

### Derde load-script

Dit load-script koppelt een wisselkoers van USD naar GBP aan de `Transactions_in_USD`-tabel.

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

### Resultaten

Laad de gegevens en ga naar de gegevensmodelviewer. Selecteer de `Transactions_in_USD`-tabel en u zult zien dat elk bestaand record een koersveldwaarde heeft van 0,7.

### Vierde load-script

Met gebruik van een resident-load zal dit load-script de `Transactions_in_USD`-tabel aaneenschakelen met de `Transactions`-tabel nadat de bedragen zijn omgezet in USD.

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

### Resultaten

Laad de gegevens en ga naar de tabel. U ziet nieuwe invoeren met bedragen in GBP van regels acht t/m veertien.

Resultaten vierde load-script

| 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 |

| id | date | amount |
|---|---|---|
| 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 |

### Vijfde load-script

Dit load-script verwijdert de dubbele invoeren uit de resultatentabel van het vierde load-script en laat alleen invoeren met bedragen in GBP staan.

```
drop tables Transactions_in_USD;
```

### Resultaten

Laad de gegevens en ga naar de tabel.

Resultaten vijfde load-script

| 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 |

| id | date | amount |
|---|---|---|
| 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 |

Nadat het vijfde load-script is geladen, toont de resultatentabel alle veertien transacties die in beide transactiegegevenssets stonden. Maar voor transacties 8-14 zijn de bedragen omgezet in GBP.

Als we het `NoConcatenate`-prefix verwijderen dat werd gebruikt vóór de `Transactions_in_USD` in het tweede load-script, geeft het script de volgende foutmelding weer: "Tabel `Transactions_in_USD` niet gevonden". Dit komt doordat de `Transactions_in_USD`-tabel automatisch had zullen worden aaneengeschakeld met de oorspronkelijke `Transactions`-tabel.

## Only

Het sleutelwoord **Only** in scripts wordt gebruikt als aggregatiefunctie of als onderdeel van de syntaxis in de prefixen **Add**, **Replace** en **Merge** voor gedeeltelijk opnieuw laden.

## Outer

Het expliciete prefix **Join** kan worden voorafgegaan door het prefix **Outer** om een outer join op te geven. In een outer join worden alle combinaties tussen de twee tabellen gegenereerd. De resulterende tabel bevat dus gecombineerde veldwaarden uit de tabellen met onbewerkte gegevens waarvoor de koppelende veldwaarden in één of beide tabellen voorkomen. Het trefwoord **Outer** is optioneel en is het standaard join type dat wordt gebruikt als geen join prefix is opgegeven.

**Syntaxis:**

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

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| tablename | De benoemde tabel die moet worden vergeleken met de geladen tabel. |
| loadstatementof selectstatement | De **LOAD** of **SELECT**-opdracht voor de geladen tabel. |

Voorbeeld

### Load-script

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
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 ];
```

Resultaattabel

| Kolom1 | Kolom2 | Kolom3 |
|---|---|---|
| A | B | C |
| 1 | aa | xx |
| 2 | cc | - |
| 3 | ee | - |
| 4 | - | yy |

### Uitleg

In dit voorbeeld worden de twee tabellen, Table1 en Table2 samengevoegd in één tabel met de naam Table1. In dit soort gevallen wordt vaak het prefix **outer** gebruikt om verschillende tabellen samen te voegen tot één tabel om aggregaties uit te voeren voor de waarden van één tabel.

## Gedeeltelijke lading

Bij een volledige lading worden altijd eerst alle tabellen in het bestaande gegevensmodel verwijderd en vervolgens wordt het load-script uitgevoerd.

Bij een gedeeltelijke lading gebeurt dit niet. Hierbij blijven alle tabellen in het gegevensmodel behouden en worden alleen de instructies **Load** en **Select** uitgevoerd voorafgegaan door het voorvoegsel **Add**, **Merge** of **Replace**. Andere gegevenstabellen worden niet door de instructie beïnvloed. Het **only**-argument geeft aan dat de opdracht alleen tijdens gedeeltelijke ladingen moet worden uitgevoerd, en moet worden genegeerd tijdens volledige ladingen. In de volgende tabel vindt u een overzicht van de uitvoering van opdrachten voor gedeeltelijke en volledige ladingen.

| Opdracht | Volledige lading | Gedeeltelijke lading |
|---|---|---|
| Load ... | Opdracht wordt uitgevoerd | Opdracht wordt niet uitgevoerd |
| Lading toevoegen/vervangen/samenvoegen ... | Opdracht wordt uitgevoerd | Opdracht wordt uitgevoerd |
| Alleen lading toevoegen/vervangen/samenvoegen ... | Opdracht wordt niet uitgevoerd | Opdracht wordt uitgevoerd |

Gedeeltelijk opnieuw laden heeft verschillende voordelen ten opzichte van volledig opnieuw laden:

- Sneller, omdat alleen recent gewijzigde gegevens hoeven te worden geladen. Bij grote gegevensverzamelingen is het verschil aanzienlijk.
- Er wordt minder geheugen verbruikt, omdat er minder gegevens worden geladen.
- Betrouwbaarder, omdat query's naar brongegevens sneller verlopen, waardoor het risico op netwerkproblemen wordt verminderd.

*Om ervoor te zorgen dat een gedeeltelijke lading correct werkt, moet de app met gegevens worden geopend voordat een gedeeltelijke lading wordt geactiveerd.*

Voer een gedeeltelijke lading uit met de knop **Opnieuw laden**. U kunt ook de Qlik Engine JSON API gebruiken.

## Beperkingen

Gedeeltelijk laden zal mislukken als er opdrachten zijn met verwijzingen naar tabellen die tijdens de volledige laden al bestonden, maar niet tijdens de gedeeltelijke lading.

Voorbeeld

### Voorbeeldopdrachten

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

Waar <Table_removed_after_full_reload> een tabel is die in de volledige lading bestond maar niet in de gedeeltelijke lading.

### Tijdelijke oplossing

Als tijdelijke oplossing kunt u de opdracht integreren in de volgende if-statement:

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

Bij een gedeeltelijke lading kunnen waarden van de gegevens worden verwijderd. Dit wordt echter niet weergegeven in de lijst met bijzondere waarden. Dit is een lijst die intern wordt bijgehouden. Dat betekent dat de lijst, na gedeeltelijk opnieuw laden, alle bijzondere waarden zal bevatten die in het veld hebben bestaan sinds de laatste volledige lading. Dit kan meer zijn dan wat er op dit moment bestaat na de

gedeeltelijke lading. Dit treft de uitvoer van de functies FieldValueCount() en FieldValue(). De functie FieldValueCount() zou mogelijk een aantal kunnen retourneren dat groter is dan het huidige aantal veldwaarden.

Voorbeeld

## Voorbeeld 1

### Load-script

Voeg het voorbeeldscript toe aan uw app en voer een gedeeltelijke lading uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
T1:
Add only Load distinct recno()+10 as Num autogenerate 10;
```

### Resultaat

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 |

### Uitleg

De opdracht wordt alleen tijdens een gedeeltelijke lading uitgevoerd. Als het prefix 'distinct' wordt weggelaten, zal het aantal van het veld **Num** na elke opeenvolgende gedeeltelijke lading worden verhoogd.

## Voorbeeld 2

### Load-script

Voeg het voorbeeldscript toe aan uw app. Voer een volledige lading uit en bekijk het resultaat. Voer vervolgens een gedeeltelijke lading uit en bekijk het resultaat. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om de resultaten te bekijken.

```
T1:
Load recno() as ID, recno() as Value autogenerate 10;
```

```
T1:
Replace only Load recno() as ID, repeat(recno(),3) as Value autogenerate 10;
```

### Resultaat

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 |
| 8 | 888 |
| 9 | 999 |
| 10 | 101010 |

### Uitleg

De eerste tabel is geladen tijdens een volledige lading en de tweede tabel vervangt de eerste tabel tijdens een gedeeltelijke lading.

# Replace

Het sleutelwoord **Replace** voor scripts wordt gebruikt als tekenreeksfunctie of als prefix bij gedeeltelijk opnieuw laden.

## Replace

het voorvoegsel **Replace** kan aan elke **LOAD**- of **SELECT**-opdracht in het script worden toegevoegd om op te geven dat de geladen tabel een andere tabel moet vervangen. Dit geeft ook aan dat deze opdracht met behulp van gedeeltelijk laden moet worden uitgevoerd. Het voorvoegsel **Replace** kan ook worden gebruikt in een **Map**-opdracht.

> *Om ervoor te zorgen dat een gedeeltelijke lading correct werkt, moet de app met gegevens worden geopend voordat een gedeeltelijke lading wordt geactiveerd.*

Voer een gedeeltelijke lading uit met de knop **Opnieuw laden**. U kunt ook de Qlik Engine JSON API gebruiken.

**Syntaxis:**

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

```
Replace [only] mapstatement
```

Tijdens een normale (niet-gedeeltelijke) lading, dient de constructie **Replace LOAD** als een normale **LOAD**-instructie, maar wordt voorafgegaan door **Drop Table**. Eerst wordt de oude tabel verwijderd vervolgens worden de records gegenereerd en opgeslagen als een nieuwe tabel.

Als het voorvoegsel **Concatenate** is gebruikt of er is een tabel aanwezig met dezelfde verzameling van velden, is dit de relevante tabel die wordt verwijderd. Anders is er geen tabel om te negeren en is de constructie **Replace LOAD** identiek aan een normale **LOAD**.

Bij een gedeeltelijke lading gebeurt hetzelfde. Het enige verschil is dat er altijd een tabel uit het vorige uitgevoerde script is om te verwijderen. Bij de constructie **Replace LOAD** wordt altijd eerste de oude tabel verwijderd en wordt er vervolgens een nieuwe tabel gemaakt.

De instructie **Replace Map...Using** zorgt ervoor dat de toewijzing ook plaatsvindt als het script gedeeltelijk wordt uitgevoerd.

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| only | Een optionele kwalificatie die aangeeft dat de opdracht alleen moet worden uitgevoerd tijdens het gedeeltelijk laden van gegevens. Het moet worden genegeerd tijdens een normale (niet-gedeeltelijke) lading. |

Voorbeelden en resultaten:

| Voorbeeld | Resultaat |
|---|---|
| `Tab1: Replace LOAD * from File1.csv;` | Tijdens zowel volledig als gedeeltelijk opnieuw laden wordt de Qlik Sense-tabel Tab1 aanvankelijk verwijderd. Daarna worden er nieuwe gegevens uit File1.csv geladen en opgeslagen in Tab1. |
| `Tab1: Replace only LOAD * from File1.csv;` | Tijdens normaal opnieuw laden wordt deze opdracht genegeerd.<br><br>Tijdens gedeeltelijk opnieuw laden wordt elke Qlik Sense-tabel die eerst Tab1 heette, aanvankelijk verwijderd. Daarna worden er nieuwe gegevens uit File1.csv geladen en opgeslagen in Tab1. |
| `Tab1: LOAD a,b,c from File1.csv; Replace LOAD a,b,c from File2.csv;` | Tijdens normaal opnieuw laden wordt File1.csv eerst in de Qlik Sense-tabel Tab1 gelezen, maar meteen daarna verwijderd en vervangen door nieuwe gegevens, geladen uit File2.csv. Alle gegevens uit File1.csv zijn verloren gegaan.<br><br>Tijdens gedeeltelijk opnieuw laden wordt de gehele Qlik Sense-tabel Tab1 aanvankelijk verwijderd. Daarna wordt deze vervangen door nieuwe gegevens, geladen uit File2.csv. |
| `Tab1: LOAD a,b,c from File1.csv; Replace only LOAD a,b,c from File2.csv;` | Tijdens normaal opnieuw laden worden gegevens geladen uit File1.csv en opgeslagen in de Qlik Sense-tabel Tab1. File2.csv wordt genegeerd.<br><br>Tijdens gedeeltelijk opnieuw laden wordt de gehele Qlik Sense-tabel Tab1 aanvankelijk verwijderd. Daarna wordt deze vervangen door nieuwe gegevens, geladen uit File2.csv. Alle gegevens uit File1.csv zijn verloren gegaan. |

## Right

De prefixen **Join** en **Keep** kunnen worden voorafgegaan door het prefix **right**.

Als het prefix wordt gebruikt vóór **join**, geeft het aan dat er een join rechts moet plaatsvinden. De resulterende tabel bevat dan alleen gecombineerde veldwaarden uit de tabellen met onbewerkte gegevens waarvoor de koppelende veldwaarden in de tweede tabel voorkomen. Als het prefix wordt gebruikt voor **keep**, geeft het aan dat de eerste tabel met onbewerkte gegevens wordt gereduceerd tot de gemeenschappelijke doorsnede met de tweede tabel, voordat deze wordt opgeslagen in Qlik Sense.

*Was u op zoek naar de tekenreeksfunctie met dezelfde naam? Zie: Right (page 1857)*

**Syntaxis:**

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

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| tablename | De benoemde tabel die moet worden vergeleken met de geladen tabel. |
| loadstatementof selectstatement | De **LOAD** of **SELECT**-opdracht voor de geladen tabel. |

Voorbeeld

### Load-script

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
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 ];
```

### Resultaat

Resultaattabel

| Kolom1 | Kolom2 | Kolom3 |
|---|---|---|
| A | B | C |
| 1 | aa | xx |
| 4 | - | yy |

### Uitleg

Dit voorbeeld toont de uitvoer voor right join waarbij alleen waarden die aanwezig zijn in de tweede (rechts) tabel zijn samengevoegd.

## Sample

Als u het prefix **sample** voor een **LOAD**- of **SELECT**-opdracht plaatst, wordt er een willekeurige selectie records uit de gegevensbron geladen.

**Syntaxis:**

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

De uitdrukking die wordt beoordeeld, definieert niet het percentage records uit de gegevensverzameling dat wordt geladen in de Qlik Sense-applicatie, maar de waarschijnlijkheid dat elk record dat wordt gelezen in de applicatie zal worden geladen. Met andere woorden: door een waarde van `p = 0.5` te specificeren, wil dat niet zeggen dat 50% van het totale aantal records wordt geladen, maar dat voor elk bestaand record er een kans van 50% is dat het in de Qlik Sense-applicatie zal worden geladen.

Argumenten

| Argument | Beschrijving |
|---|---|
| p | Een willekeurige uitdrukking die wordt geëvalueerd als een getal dat groter dan 0 en kleiner·dan of gelijk is aan 1. Het getal geeft de waarschijnlijkheid aan dat een bepaalde record wordt gelezen.<br><br>Alle records worden gelezen maar slechts enkele daarvan worden in Qlik Sense geladen. |

## Wanneer gebruiken

Een voorbeeld is handig wanneer u wilt dat er uit een grote tabel voorbeeldgegevens worden gehaald om de aard van gegevens, verdeling of veldinhoud te begrijpen. Omdat dit een subset van gegevens ophaalt, worden gegevens sneller geladen zodat scripts ook sneller kunnen worden getest. Anders dan de functie `First`, haalt de functie `Sample` gegevens uit de hele tabel in plaats van beperkt te zijn tot de eerste paar rijen. Dit kan in bepaalde gevallen een nauwkeurigere afspiegeling van gegevens opleveren.

De volgende voorbeelden geven twee mogelijke toepassingen weer van het `Sample`-scriptprefix:

```
Sample 0.15 SQL SELECT * from Longtable;

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

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – Voorbeeld van een inline-tabel

Load-script en resultaten

### Overzicht

In dit voorbeeld laadt het script vanuit een inline-tabel een voorbeeldgegevensset uit een gegevensset met zeven records in de tabel `Transactions`.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `amount`

Voeg de volgende meting toe:

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

Resultatentabel

| id | date | =Sum(amount) |
|---|---|---|
| 2 | 09/07/2018 | 556.31 |
| 4 | 09/22/2018 | 125 |
| 1 | 08/30/2018 | 23.56 |
| 3 | 09/16/2018 | 5.75 |

In de iteratie van de load die in dit voorbeeld wordt gebruikt, zijn alle zeven records gelezen, maar zijn er slechts vier in de gegevenstabel geladen. Elke keer dat de load opnieuw wordt uitgevoerd, kan dit resulteren in een ander aantal en in dat er een andere set records in de applicatie wordt geladen.

## Voorbeeld 2 – Voorbeeld uit een automatisch gegenereerde tabel

Load-script en resultaten

### Overzicht

In dit voorbeeld met `Autogenerate`, wordt een gegevensset van 100 records gemaakt met de velden `date`, `id` en `amount`. Maar het prefix `Sample` wordt gebruikt met een waarde van 0,1.

### Load-script

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

Autogenerate(100);
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `amount`

Voeg de volgende meting toe:

Resultatentabel

| 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 |

In de iteratie van de load die in dit voorbeeld wordt gebruikt, zijn zeven records geladen uit de aangemaakte gegevensset. Wederom geldt dat elke keer dat de load opnieuw wordt uitgevoerd, dit kan resulteren in een ander aantal en in dat er een andere set records in de applicatie wordt geladen.

## Semantic

Het laadvoorvoegsel `semantic` maakt een speciaal type veld dat kan worden gebruikt in Qlik Sense om relationele gegevens te verbinden en te beheren, zoals boomstructuren, zelfverwijzende bovenliggende/onderliggende gestructureerde gegevens en/of gegevens die kunnen worden beschreven als een diagram.

Merk op dat de lading `semantic` op dezelfde manier kan werken als de voorvoegsels *Hierarchy (page 66)* en *HierarchyBelongsTo (page 68)*. Alle drie de voorvoegsels kunnen worden gebruikt als bouwstenen in effectieve front-end-oplossingen voor het doorkruisen van relationele gegevens.

**Syntaxis:**

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

Een semantische lading verwacht een invoer die precies drie of vier velden breed is met een strikte definitie van wat elk geordend veld vertegenwoordigt, zoals weergegeven in de onderstaande tabel:

Semantische ladingsvelden

| Veldnaam | Veldbeschrijving |
|---|---|
| 1e veld: | Deze tag is een weergave van het eerste van twee objecten waartussen een relatie bestaat. |
| 2e veld: | Deze tag wordt gebruikt om de 'voorwaartse' relatie tussen het eerste en tweede object te beschrijven. Als het eerste object een onderliggend object is en het tweede object een bovenliggend object, kunt u een relatietabblad maken met de vermelding 'bovenliggend' of 'bovenliggend van' alsof u de relatie van onderliggend tot bovenliggend volgt. |
| 3e veld: | Deze tag is een weergave van het tweede van twee objecten waartussen een relatie bestaat. |
| 4e veld: | Dit veld is optioneel. Deze tag beschrijft de 'achterwaartse' of 'omgekeerde' relatie tussen het eerste en tweede object. Als het eerste object een onderliggend object is en het tweede object een bovenliggend object, kan een relatietabblad 'onderliggend' of 'onderliggend van' vermelden, alsof u de relatie van bovenliggend tot onderliggend volgt. Als u geen vierde veld toevoegt, wordt de tweede veldtag gebruikt om de relatie in beide richtingen te beschrijven. In dat geval wordt automatisch een pijlsymbool toegevoegd als onderdeel van de tag. |

De volgende code is een voorbeeld van het voorvoegsel `semantic`.

```
Semantic
Load
Object,
‘Parent’ AS Relationship,
NeighbouringObject AS Object,
‘Child’ AS Relationship
from graphdata.csv;
```

*Het is toegestaan en gebruikelijk om het derde veld hetzelfde te labelen als het eerste veld. Dit creëert een zelfverwijzende opzoekactie, zodat u object(en) stap voor stap naar het/de verwante object(en) kunt volgen. Als het 3e veld niet dezelfde naam heeft, dan is het eindresultaat een eenvoudige opzoekactie van (een) object(en) naar zijn directe relationele buur/buren slechts één stap verwijderd, wat een uitvoer is van weinig praktisch nut.*

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

Gerelateerde functies

| Functies | Interactie |
|---|---|
| *Hierarchy (page 66)* | Het laadvoorvoegsel Hiërarchie wordt gebruikt om knooppunten in bovenliggende/onderliggende gegevensstructuren en andere diagramachtige gegevensstructuren te verdelen en te ordenen en deze om te zetten in tabellen. |
| *HierarchyBelongsTo (page 68)* | Het laadvoorvoegsel HierarchyBelongsTo wordt gebruikt om de voorlopers van bovenliggende/onderliggende gegevensstructuren en andere diagramachtige gegevensstructuren te vinden en te ordenen en deze om te zetten in tabellen. |

## Voorbeeld - Een speciaal veld maken voor het verbinden van relaties met behulp van het semantische voorvoegsel

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die geografierelatierecords vertegenwoordigt, die in een tabel met de naam `GeographyTree` wordt geladen.
    - Elk item heeft een id aan het begin van de regel en een ParentID aan het einde van de regel.
- Het voorvoegsel `semantic` dat een speciaal gedragsveld met het label `Relation` zal toevoegen.

### Load-script

```
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;
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies.

- `Id`
- `Geography`

Maak vervolgens een filtervak met `Relation` als dimensie. Klik op **Klaar met bewerken**.

Resultatentabel

| Id | Geografie |
|---|---|
| 1 | Wereld |

| Id | Geografie |
|---|---|
| 2 | Europa |
| 3 | Azië |
| 4 | Noord-Amerika |
| 5 | Zuid-Amerika |
| 6 | VK |
| 7 | Germany |
| 8 | Sweden |
| 9 | Zuid-Korea |
| 10 | Noord-Korea |
| 11 | China |
| 12 | Londen |
| 13 | Birmingham |

Filtervak

| Relatie |
|---|
| Onderliggend |
| Bovenliggend |

Klik op **Europa** vanuit de dimensie `Geography` in de tabel en klik op **Onderliggend** vanuit de dimensie `Relation` in het filtervak. Noteer het verwachte resultaat in de tabel:

Resultatentabel met 'onderliggende objecten' van Europa

| Id | Geografie |
|---|---|
| 6 | VK |
| 7 | Germany |
| 8 | Sweden |

Als u nogmaals op **Onderliggend** klikt, worden plaatsen weergegeven die 'onderliggende objecten' van het VK zijn, een stap verder.

Resultatentabel met 'onderliggende objecten' van het VK

| Id | Geografie |
|---|---|
| 12 | Londen |
| 13 | Birmingham |

## Unless

Met het prefix en suffix **unless** maakt u een conditionele clausule waarmee wordt bepaald of een opdracht of exit-clausule wel of niet moet worden geëvalueerd. De opdracht kan worden beschouwd als een compact alternatief voor een volledige **if..end if**-opdracht.

**Syntaxis:**

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

De **statement** of de **exitstatement** wordt alleen uitgevoerd als de **condition** resulteert in False.

Het prefix **unless** kan worden gebruikt bij opdrachten die al een of meer andere opdrachten bevatten, inclusief aanvullende prefixen **when** of **unless**.

Argumenten

| Argument | Beschrijving |
|---|---|
| condition | Een logische uitdrukking die resulteert in True of False. |
| statement | Elke Qlik Sense-scriptopdracht behalve besturingsopdrachten. |
| exitstatement | Een **exit for**-, **exit do**- of **exit sub**-clausule of een **exit script**-opdracht. |

### Wanneer gebruiken

De `Unless`-opdracht retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als een voorwaarde wanneer de gebruiker delen van het script voorwaardelijk wil laden of uitsluiten.

De volgende regels tonen drie voorbeelden van hoe de `Unless`-functie kan worden gebruikt:

```
exit script unless A=1;

unless A=1 LOAD * from myfile.csv;

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

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – Unless-prefix

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Het maken van variabele A, met een waarde van 1.
- Een gegevensverzameling die in de tabel Transacties wordt geladen, tenzij de variabele A = 2.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date
- amount

Resultatentabel

| 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 |

Omdat variabele A is toegewezen aan de waarde 1 aan het begin van het script, wordt de voorwaarde na het `Unless`-prefix beoordeeld en wordt het resultaat `FALSE` geretourneerd. Het resultaat is dat het script de `Load`-opdracht blijft uitvoeren. In de resultatentabel worden alle records uit de `Transactions`-tabel weergegeven.

Als deze variabelewaarde is ingesteld op gelijk aan 2, dan worden er geen gegevens in het gegevensmodel geladen.

## Voorbeeld 2 – Unless-suffix

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script begint met het laden van een eerste gegevensset in de tabel `Transactions`. Het script wordt vervolgens beëindigd tenzij er minder dan 10 records in de `Transactions`-tabel staan.

Als deze voorwaarde niet resulteert in de beëindiging van het script, wordt er nog een set met transacties aaneengeschakeld in de `Transactions`-tabel en wordt dit proces herhaald.

### Load-script

```
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 [
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 ;
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date
- amount

Resultatentabel

| id | date | amount |
|---|---|---|
| 1 | 08/30/2018 | 23.56 |
| 2 | 09/07/2018 | 556.31 |

| id | date | amount |
|---|---|---|
| 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 |

Er zijn zeven records in elk van de drie gegevenssets van het load-script.

De eerste gegevensset (met transactie `id` 1 t/m 7) wordt in de applicatie geladen. De voorwaarde `Unless` controleert of er minder dan 10 rijen in de tabel `Transactions` staan. Dit resulteert in `TRUE` en daarom wordt de tweede gegevensset (met transactie `id` 8 t/m 14) in de applicatie geladen. De tweede voorwaarde `Unless` controleert of er minder dan 10 records in de tabel `Transactions` staan. Dit resulteert in `FALSE` en dus wordt het script beëindigd.

## Voorbeeld 3 – Meerdere Unless-prefixes

### Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

In dit voorbeeld wordt een gegevensset met één transactie gemaakt als de tabel `Transactions`. Er wordt een for-lus geactiveerd waarin twee geneste unless-opdrachten het volgende controleren:

1. Tenzij (unless) er meer dan 100 records in de `Transactions`-tabel staan
2. Tenzij (unless) het aantal records in de `Transactions`-tabel een veelvoud is van 6

Als deze voorwaarden `FALSE` zijn, worden er nog eens zeven records gegenereerd en aaneengeschakeld met de bestaande `Transactions`-tabel. Dit proces wordt herhaald tot een van de twee transacties de waarde `TRUE` retourneert.

### Load-script

```
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
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:`id`.

Resultatentabel

| id |
| --- |
| 0 |
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |
| Nog 30 rijen |

De geneste unless-opdrachten die optreden in de for-lus controleren het volgende:

1. Staan er meer dan 100 rijen in de `Transactions`-tabel?
2. Is het totale aantal records in de `Transactions`-tabel een veelvoud van 6?

Wanneer beide unless-opdrachten een waarde van `FALSE` retourneren, worden er nog eens zeven records gegenereerd en aaneengeschakeld met de bestaande `Transactions`-tabel.

Deze opdrachten retourneren vijf keer een waarde van `FALSE` waarna er in totaal 36 rijen gegevens in de `Transactions`-tabel staan.

Hierna retourneert de tweede `Unless`-opdracht een waarde van `TRUE`. Daarom wordt de load-instructie hierna niet meer uitgevoerd.

## When

Met het prefix en suffix **when** maakt u een conditionele clausule waarmee wordt bepaald of een opdracht of exit-clausule wel of niet moet worden uitgevoerd. De opdracht kan worden beschouwd als een compact alternatief voor een volledige **if..end if**-opdracht.

**Syntaxis:**

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

**Retourgegevenstypen:** Booleaanse waarde

In Qlik Sense wordt de booleaanse waarde Waar vertegenwoordigd door -1 en de waarde Onwaar door 0.

De **statement** of de **exitstatement** wordt alleen uitgevoerd als de voorwaarde resulteert in True.

Het prefix `When` kan worden gebruikt bij opdrachten die al een of meer andere opdrachten bevatten, inclusief aanvullende prefixen `When` of `Unless`.

**Wanneer gebruiken**

De `When`-opdracht retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als een voorwaarde wanneer de gebruiker delen van een script wil laden of uitsluiten.

Argumenten

| Argument | Beschrijving |
|---|---|
| condition | Een logische uitdrukking die resulteert in TRUE of FALSE |
| statement | Elke Qlik Sense-scriptopdracht behalve besturingsopdrachten. |
| exitstatement | Een **exit for**-, **exit do**- of **exit sub**-clausule of een **exit script**-opdracht. |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `exit script when A=1;` | Wanneer de uitdrukking `A=1` resulteert in TRUE, wordt het script beëindigd. |
| `when A=1 LOAD * from myfile.csv;` | Wanneer de uitdrukking `A=1` resulteert in TRUE, wordt het bestand `myfile.csv` geladen. |
| `when A=1 unless B=2 drop table Tab1;` | Wanneer de opdracht `A=1` resulteert in TRUE en als `B=2` resulteert in FALSE, dan wordt de `Tab1`-tabel verwijderd. |

## Voorbeeld 1 – When-prefix

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met datums en bedragen die naar de tabel `Transactions` wordt gestuurd.
- De `Let`-opdracht die aangeeft dat de variabele `A` is gemaakt en de waarde `1` heeft.
- De `When`-voorwaarde levert de voorwaarde dat als `A` gelijk is aan `1`, het script zal blijven laden.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date
- amount

Resultatentabel

| 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 |

Omdat variabele A is toegewezen aan de waarde 1 aan het begin van het script, wordt de voorwaarde na het When-prefix beoordeeld en wordt het resultaat TRUE geretourneerd. Vanwege het resultaat TRUE, blijft het script de load-instructie uitvoeren. Alle records uit de resultatentabel worden weergegeven.

Als deze variabelewaarde was ingesteld op niet-gelijk aan 1, dan zouden er geen gegevens in het gegevensmodel worden geladen.

## Voorbeeld 2 – When-suffix

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Drie gegevenssets met datums en bedragen die naar de tabel Transactions worden gestuurd.
  - De eerste gegevensset bevat transacties 1-7.
  - De tweede gegevensset bevat transacties 8-14.
  - De derde gegevensset bevat transacties 15-21.
- Een When-voorwaarde die bepaalt of de tabel Transactions meer dan tien rijen bevat. Als een van de When-opdrachten resulteren in TRUE, wordt het load-script beëindigd. Deze voorwaarde wordt aan het eind van elk van de drie gegevenssets geplaatst.

### Load-script

```
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 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 ;
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date
- amount

Resultatentabel

| 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 |
| 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 |

Elk van de drie gegevenssets bevat zeven transacties. De eerste gegevensset bevat transacties 1 t/m 7 en wordt in de applicatie geladen. De `When`-voorwaarde na deze load-instructie resulteert in FALSE omdat er minder dan tien rijen in de `Transactions`-tabel staan. Het load-script gaat verder naar de volgende gegevensset.

De tweede gegevensset bevat transacties 8 t/m 14 en wordt in de applicatie geladen. De tweede `When`-voorwaarde resulteert in TRUE omdat er meer dan tien rijen in de `Transactions`-tabel staan. Daarom wordt het script beëindigd.

## Voorbeeld 3 – Meerdere When-prefixes

### Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met één transactie wordt gemaakt als de tabel `Transactions`.
- Er wordt een `For`-lus geactiveerd die twee geneste `When`-voorwaarden bevat die resulteren in een van de volgende opties:

1. Er staan minder dan `100` records in de `Transactions`-tabel.
2. Het aantal records in de `Transactions`-tabel is geen veelvoud van `6`.

### Load-script

```
RowsCheck = NoOfRows('Transactions') < 100 or mod(NoOfRows('Transactions'),6) <> 0;
Transactions:
Load
	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
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `id`

De resultatentabel toont alleen de eerste vijf transactie-id's, maar het load-script maakt 36 rijen en wordt vervolgens beëindigd zodra aan de `When`-voorwaarde is voldaan.

Resultatentabel

| id |
|---|
| 0 |
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |
| Nog 30 rijen |

De geneste `When`-voorwaarden in de `For`-lus resulteren in de volgende vragen:

- Staan er minder dan 100 rijen in de `Transactions`-tabel?
- Is het totale aantal records in de `Transactions`-tabel geen veelvoud van 6?

Wanneer beide `When`-voorwaarden een waarde van TRUE retourneren, worden er nog eens zeven records gegenereerd en aaneengeschakeld met de bestaande `Transactions`-tabel.

De `When`-voorwaarden retourneren vijf keer de waarde TRUE. Als dat gebeurt, zijn er in totaal 36 rijen gegevens in de `Transactions`-tabel.

Wanneer 36 rijen gegevens worden gemaakt in de `Transactions`-tabel, retourneert de tweede `When`-opdracht de waarde FALSE en wordt de load-instructie hierna niet meer uitgevoerd.

## 3.3 Reguliere scriptopdrachten

Normale opdrachten worden over het algemeen gebruikt om gegevens op een bepaalde manier te manipuleren. Deze opdrachten kunnen worden geschreven op een willekeurig aantal regels in het script en moeten altijd worden afgesloten met een puntkomma (;).

Bij het typen van sleutelwoorden in het script kunt u elke combinatie van hoofdletters en kleine letters gebruiken. Gebruikte namen van velden en variabelen in de opdrachten zijn echter hoofdlettergevoelig.

### Overzicht van normale scriptopdrachten

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

#### Alias

De opdracht **alias** wordt gebruikt om een alias in te stellen. Dit betekent dat een veld een andere naam krijgt als het voorkomt in het script dat volgt.

```
Alias fieldname as aliasname {,fieldname as aliasname}
```

#### Autonumber

Deze instructie genereert een uniek geheel getal voor elke afzonderlijk geëvalueerde waarde in een veld die tijdens de uitvoering van een script wordt aangetroffen.

```
AutoNumber fields [Using namespace] ]
```

#### Binary

De opdracht **binary** wordt gebruikt voor het laden van de gegevens vanuit een ander QlikView 12-document, met inbegrip van sectietoegangsgegevens.

```
Binary [path] filename
```

#### comment

Hiermee kunt u de opmerkingen voor velden (metagegevens) uit databases en spreadsheets weergeven. Veldnamen die niet in de app aanwezig zijn, worden genegeerd. Als meerdere exemplaren van een veldnaam worden aangetroffen, wordt de laatste waarde gebruikt.

```
Comment field *fieldlist using mapname
```

```
Comment field fieldname with comment
```

#### comment table

Hiermee kunt u de opmerkingen voor tabellen (metagegevens) uit databases of spreadsheets weergeven.

```
Comment table tablelist using mapname
```

```
Comment table tablename with comment
```

### Connect

*Deze functionaliteit is niet beschikbaar in Qlik Sense SaaS.*

Met de opdracht **CONNECT** wordt de toegang van Qlik Sense tot een algemene database via de OLE DB/ODBC-interface vastgelegd. Voor ODBC moet de gegevensbron eerst worden opgegeven met behulp van ODBC-beheer.

```
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

De opdracht **Declare** wordt gebruikt voor het maken van velddefinities, waarbij u relaties kunt definiëren tussen velden of functies. Een reeks van velddefinities kan worden gebruikt om automatisch afgeleide velden te genereren, die kunnen worden gebruikt als dimensies. Zo kunt u bijvoorbeeld een agendadefinitie maken en die gebruiken om gerelateerde dimensies te genereren, zoals jaar, maand, week en dag, vanuit een datumveld.

```
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

De opdracht **Derive** wordt gebruikt voor het genereren van afgeleide velden op basis van een velddefinitie die is gemaakt met een opdracht **Declare**. U kunt opgeven voor welke gegevensvelden velden moeten worden afgeleid of u kunt deze expliciet of impliciet afleiden op basis van veldtags.

```
Derive [Field[s]] 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
```

### Direct Query

De instructie **DIRECT QUERY** zorgt ervoor dat u tabellen kunt openen met een ODBC- of OLE DB-verbinding met gebruik van de Direct Discovery-functie.

```
Direct Query [path]
```

### Directory

De opdracht **Directory** legt vast in welke directory moet worden gezocht naar gegevensbestanden in volgende **LOAD**-opdrachten, totdat een nieuwe **Directory**opdracht wordt gegeven.

```
Directory [path]
```

### Disconnect

De opdracht **Disconnect** verbreekt de huidige ODBC-/OLE DB-/aangepaste verbinding. Deze opdracht is optioneel.

```
Disconnect
```

### drop field

Een of meer Qlik Sense-velden kunnen op elk moment tijdens de uitvoering van een script worden verwijderd uit het gegevensmodel, en dus uit het geheugen, met behulp van de opdracht **drop field**. De eigenschap "distinct" van een tabel wordt verwijderd na een **drop field**-opdracht.

> *Zowel **drop field** als **drop fields** zijn toegestane vormen en hebben hetzelfde effect. Als geen tabel wordt opgegeven, wordt het veld uit alle tabellen waarin het voorkomt verwijderd.*

```
Drop field fieldname [ , fieldname2 ...] [from tablename1 [ , tablename2
...]]
```

```
drop fields fieldname [ , fieldname2 ...] [from tablename1 [ , tablename2
...]]
```

### drop table

Een of meer interne tabellen in Qlik Sense kunnen op elk moment tijdens de uitvoering van een script worden verwijderd uit het gegevensmodel, en dus uit het geheugen, met behulp van de opdracht **drop table**.

*De vormen **drop table** en **drop tables** worden beide geaccepteerd.*

```
Drop table  tablename [, tablename2 ...]
```

```
drop tables[ tablename [, tablename2 ...]
```

### Execute

De **Execute**-opdracht wordt gebruikt voor het uitvoeren van andere programma's terwijl gegevens worden geladen in Qlik Sense. Bijvoorbeeld voor het uitvoeren van noodzakelijke conversies.

```
Execute commandline
```

### FlushLog

Met de opdracht **FlushLog** dwingt u af dat Qlik Sense de inhoud van de scriptbuffer naar het scriptlogbestand schrijft.

```
FlushLog
```

### Force

Met de opdracht **force** kunt u afdwingen dat Qlik Sense veldnamen en veldwaarden van de daaropvolgende **LOAD**- en **SELECT**-opdrachten interpreteert als geschreven met alleen hoofdletters, met alleen kleine letters, met beginhoofdletters of zoals ze worden weergegeven (gemengd). Met deze opdracht kunt u veldwaarden van tabellen interpreteren volgens verschillende conventies.

```
Force ( capitalization | case upper | case lower | case mixed )
```

### LOAD

Met de opdracht **LOAD** worden velden geladen uit een bestand, uit gegevens die in het script zijn gedefinieerd, uit een eerder geladen tabel, van een webpagina, uit het resultaat van een daaropvolgende **SELECT**-opdracht of door gegevens automatisch te genereren. Het is ook mogelijk om gegevens via analytische verbindingen op te halen.

```
 Load [ distinct ] *fieldlist
```

```
[( from file [ format-spec ] |
```

```
from_field fieldassource [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

De opdracht **let** is een aanvulling op de opdracht **set** voor het definiëren van scriptvariabelen. In tegenstelling tot de opdracht **set** evalueert de opdracht **let** de uitdrukking rechts van '=' tijdens runtime van het script voordat deze wordt toegewezen aan de variabele.

```
Let variablename=expression
```

### Loosen Table

Een of meer interne gegevenstabellen van Qlik Sense kunt u tijdens de uitvoering van het script expliciet declareren als losjes gekoppeld met behulp van de opdracht **Loosen Table**. Als een tabel losjes is gekoppeld, zijn alle associaties tussen veldwaarden in de tabel verwijderd. Een vergelijkbaar effect kan worden verkregen door elk veld van de losjes gekoppelde tabel als onafhankelijke, niet-verbonden tabel te laden. Losjes gekoppeld kan handig zijn tijdens het testen om tijdelijk verschillende onderdelen van de gegevensstructuur te isoleren. Een losjes gekoppelde tabel kan worden geïdentificeerd in de tabelweergave aan de hand van de stippellijnen. Het gebruik van een of meer **Loosen Table**-opdrachten in het script zorgt dat Qlik Sense eventuele instellingen negeert voor losjes gekoppelde tabellen voordat het script wordt uitgevoerd.

```
tablename [ , tablename2 ...]
```

```
Loosen Tables tablename [ , tablename2 ...]
```

### Map ... using

De opdracht **map ... using** wordt gebruikt voor de toewijzing van een bepaalde veldwaarde of uitdrukking aan de waarden van een specifieke toewijzingstabel. De toewijzingstabel wordt gemaakt met behulp van de opdracht **Mapping**.

```
Map  *fieldlist Using  mapname
```

### NullAsNull

De opdracht **NullAsNull** schakelt de conversie van NULL-waarden naar tekenreekswaarden uit die eerder is ingesteld via een **NullAsValue**-opdracht.

```
NullAsNull *fieldlist
```

### NullAsValue

De opdracht **NullAsValue** legt vast voor welke velden NULL moet worden omgezet in een waarde.

```
NullAsValue *fieldlist
```

### Qualify

De opdracht **Qualify** wordt gebruikt om de kwalificatie van veldnamen in te schakelen. Dit houdt in dat veldnamen de tabelnaam als prefix krijgen.

```
Qualify *fieldlist
```

### Rem

Met de opdracht **rem** kunt u opmerkingen of commentaar aan het script toevoegen, of scriptopdrachten tijdelijk deactiveren zonder ze te verwijderen.

```
Rem string
```

### Rename Field

Met deze scriptfunctie wordt de naam van een of meer bestaande Qlik Sense-velden gewijzigd nadat deze zijn geladen.

```
Rename field (using mapname|oldname to newname{ , oldname to newname })
```

```
Rename Fields (using mapname | oldname to newname{ , oldname to newname })
```

### Rename Table

Met deze scriptfunctie wordt de naam van een of meer bestaande interne Qlik Sense-tabellen gewijzigd nadat deze zijn geladen.

```
Rename table (using mapname | oldname to newname{ , oldname to newname })
```

```
Rename Tables (using mapname | oldname to newname{ , oldname to newname })
```

### Section

Met de opdracht **section** kan worden aangegeven of de daarop volgende **LOAD**- en **SELECT**-opdrachten moeten worden beschouwd als gegevens of als een definitie van de toegangsrechten.

```
Section (access | application)
```

### Select

U selecteert velden uit een ODBC-gegevensbron of van een OLE DB-provider met de standaard **SELECT**-opdrachten van SQL. Of de **SELECT**-opdrachten worden geaccepteerd, hangt af van het gebruikte ODBC-stuurprogramma of de gebruikte OLE DB-provider.

```
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

Met de opdracht **set** legt u scriptvariabelen vast. Deze kunnen worden gebruikt om tekenreeksen, paden, stations en dergelijke te vervangen.

```
Set variablename=string
```

### Sleep

Met de opdracht **sleep** onderbreekt u de uitvoering van een script voor een bepaalde tijd.

```
Sleep n
```

### SQL

Met de opdracht **SQL** kunt u via een SQL- of ODBC-verbinding een willekeurige OLE DB-opdracht verzenden.

```
SQL sql_command
```

### SQLColumns

Met de opdracht **sqlcolumns** wordt een set velden geretourneerd waarin de kolommen worden beschreven van een ODBC- of OLE DB-gegevensbron waarmee via **connect** verbinding is gemaakt.

```
SQLColumns
```

### SQLTables

Met de opdracht **sqltables** wordt een set velden geretourneerd waarin de tabellen worden beschreven van een ODBC- of OLE DB-gegevensbron waarmee via **connect** verbinding is gemaakt.

```
SQLTables
```

### SQLTypes

Met de opdracht **sqltypes** wordt een set velden geretourneerd waarin de typen worden beschreven van een ODBC- of  OLE DB-gegevensbron waarmee via **connect** verbinding is gemaakt.

```
SQLTypes
```

### Star

Met de opdracht **star** kunt u een tekenreeks opgeven die de set van alle waarden van een veld in een database vertegenwoordigt. De opdracht beïnvloedt de **LOAD**- en **SELECT**-opdrachten die volgen.

```
Star is [ string ]
```

### Store

De **Store**-instructie genereert een QVD-, Parquet-, CSV- of TXT-bestand.

```
Store [ *fieldlist from] table into filename [ format-spec ];
```

### Tag

Met de scriptopdracht kunnen tags aan één of meer velden of tabellen worden toegewezen. Als wordt geprobeert om een tag toe te wijzen aan een veld of tabel die niet bestaat in de app, wordt de tag genegeerd. Als conflicterende exemplaren van een veld- of tagnaam zijn aangetroffen, wordt de laatste waarde gebruikt.

```
Tag[field|fields] fieldlist with tagname
```

```
Tag [field|fields] fieldlist using mapname
```

```
Tag table tablelist with tagname
```

### Trace

Met de opdracht **trace** wordt eventueel een tekenreeks naar het venster **Voortgang scriptuitvoering** en in het scriptlogbestand geschreven. Dit is zeer handig bij het opsporen van fouten. Met behulp van $-uitbreidingen van variabelen die zijn berekend vóór de **trace**-opdracht, kunt u het bericht aanpassen.

```
Trace string
```

### Unmap

Met de opdracht **Unmap** wordt de toewijzing van veldwaarden die is vastgelegd door een eerdere opdracht **Map … Using**, uitgeschakeld voor velden die daarna worden geladen.

```
Unmap *fieldlist
```

### Unqualify

Met de opdracht **Unqualify** wordt de kwalificatie van veldnamen die eerder was ingeschakeld met de opdracht **Qualify**, uitgeschakeld.

```
Unqualify *fieldlist
```

### Untag

Met de scriptopdracht kunnen tags van één of meer velden of tabellen worden verwijderd. Als wordt geprobeert om een tag te verwijderen van een veld of tabel die niet bestaat in de app, wordt de tag niet verwijderd.

```
Untag[field|fields] fieldlist with tagname
```

```
Tag [field|fields] fieldlist using mapname
```

```
Tag table tablelist with tagname
```

## Alias

De opdracht **alias** wordt gebruikt om een alias in te stellen. Dit betekent dat een veld een andere naam krijgt als het voorkomt in het script dat volgt.

### Syntaxis:

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

### Argumenten:

Argumenten

| Argument | Beschrijving |
|---|---|
| fieldname | De naam van het veld in de brongegevens |
| aliasname | Een aliasnaam die u ervoor in de plaats wilt gebruiken |

## Voorbeelden met resultaten:

Voor de voorbeelden in de tabel hieronder zullen de gedefinieerde veldnaamwijzigingen worden toegepast op verwijzingen naar de oorspronkelijke veldnaam in alle volgende **SELECT**- en **LOAD**-opdrachten. Een nieuwe alias voor een veldnaam kan worden gedefinieerd met een nieuwe **alias**-opdracht verderop in het script.

Naar het veld wordt ook verwezen met de aliasnaam in het gegevensmodel en de app-inhoud.

Voorbeelden en resultaten

| Voorbeeld | Resultaat |
|---|---|
| `Alias ID_N as NameID;` | Veld *ID_N* is hernoemd naar *NameID*. |
| `Alias A as Name, B as Number, C as Date;` | De velden *A*, *B*, en *C* zijn hernoemd naar respectievelijk *Name*, *Number*, en *Date*. |

### Voorbeeld in lange vorm

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

#### Load-script

```
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;
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- AAA

Resultatentabel

| AAA |
|---|
| string 1 |
| string 3 |

# AutoNumber

Deze instructie genereert een uniek geheel getal voor elke afzonderlijk geëvalueerde waarde in een veld die tijdens de uitvoering van een script wordt aangetroffen.

Het is ook mogelijk om de functie *autonumber (page 697)* te gebruiken in een **LOAD**-instructie, maar dit heeft enige beperkingen als u een geoptimaliseerde lading wilt gebruiken. U kunt een geoptimaliseerde lading maken door de gegevens eerst op te halen vanuit een **QVD**-bestand. Vervolgens kunt u de instructie **AutoNumber** gebruiken om de waarden naar symboolsleutels te converteren.

**Syntaxis:**

```
AutoNumber *fieldlist [Using namespace] ]
```

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| *fieldlist | Een door komma's gescheiden lijst met de velden waarin de waarden moeten worden vervangen door een uniek geheel getal.<br><br>U kunt de jokertekens ? en * in de veldnamen gebruiken om alle velden met een overeenkomstige naam toe te voegen U kunt ook * gebruiken om alle velden toe te voegen. U moet de veldnamen tussen aanhalingstekens plaatsen als u jokertekens gebruikt. |
| namespace | **Using** `namespace` is optioneel. U kunt deze optie gebruiken als u een naamruimte wilt maken waarin identieke waarden in verschillende velden dezelfde sleutel delen.<br><br>Als u deze optie niet gebruikt, wordt voor alle velden een afzonderlijke sleutelindex gebruikt. |

**Beperkingen:**

Als u verschillende **LOAD**-instructies gebruikt in het script, moet de **AutoNumber**-instructie achter de laatste **LOAD**-instructie worden geplaatst.

Voorbeeld - script met AutoNumber

## Scriptvoorbeeld

In dit voorbeeld worden de gegevens eerst geladen zonder de **AutoNumber**-instructie. Vervolgens wordt de **AutoNumber**-instructie toegevoegd om het effect ervan te laten zien.

### Gebruikte gegevens in voorbeeld

Laad de volgende gegevens als een inline-load in de editor voor laden van gegevens om het onderstaande scriptvoorbeeld te maken. Laat de **AutoNumber**-instructie eerst uitgeschakeld in de vorm van een commentaarregel.

```
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
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;
```

### Visualisaties maken

Maak twee tabelvisualisaties in een Qlik Sense-werkblad. Voeg **KeyToOtherTable**, **Region**, **Year**, **Month** en **Sales** toe als dimensies aan de eerste tabel. Voeg **KeyToOtherTable**, **Region**, **Year**, **Month** en **Budget** toe als dimensies aan de tweede tabel.

### Resultaat

Tabel 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 |

Tabel 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 |

| KeyToOtherTable | Region | Year | Month | Budget |
|---|---|---|---|---|
| South\|2013\|May | South | 2013 | May | 200 |
| South\|2013\|May | South | 2013 | May | 300 |
| South\|2014\|June | South | 2014 | June | 500 |

### Uitleg

In het voorbeeld ziet u een composietveld **KeyToOtherTable** waarmee de twee tabellen worden gekoppeld. **AutoNumber** wordt niet gebruikt. Kijk naar de lengte van de **KeyToOtherTable**-waarden.

### AutoNumber-instructie toevoegen

Verwijder de commentaaraanduiding van de **AutoNumber**-instructie in het load-script.

```
AutoNumber KeyToOtherTable;
```

### Resultaat

Tabel 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 |

Tabel 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 |

### Uitleg

De **KeyToOtherTable**-veldwaarden zijn vervangen door unieke gehele getalswaarden en daardoor is de lengte van de veldwaarden gereduceerd, wat geheugen bespaart. **AutoNumber** heeft een effect op de sleutelvelden in beide tabellen en de tabellen blijven gekoppeld. Het voorbeeld is kort vanwege

demonstratiedoeleinden, maar zou relevant zijn bij een tabel met een groot aantal rijen.

## Binary

De opdracht **binary** wordt gebruikt voor het laden van de gegevens vanuit een andere Qlik Sense-app of een QlikView 12-document, met inbegrip van sectietoegangsgegevens. Er zijn geen andere elementen van de app toegevoegd, zoals werkbladen, presentaties, visualisaties, masteritems of variabelen.

Slechts één **binary** opdracht is toegestaan in het script. De **binary** opdracht moet de eerste opdracht van het script zijn, zelfs nog vóór de SET-opdrachten die zich gewoonlijk aan het begin van het script bevinden.

**Syntaxis:**

```
binary [path] filename
```

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| path | Het pad naar het bestand die een verwijzing zou moeten zijn naar een mapgegevensverbinding. Dit is een vereiste als het bestand zich niet in de werkdirectory van Qlik Sense bevindt.<br><br>**Voorbeeld:** *'lib://Table Files/'*<br><br>In de bestaande scriptmodus, worden tevens de volgende padindelingen ondersteund:<br>• absoluut<br>**Voorbeeld:** *c:\data*<br>• relatief ten opzicht van de app die deze scriptregel bevat.<br>**Voorbeeld:** *data* |
| filename | De naam van het bestand, inclusief de bestandsextensie .qvw of .qvf. |

**Beperkingen:**

U kunt **binary** niet gebruiken om gegevens te laden vanuit een app op dezelfde Qlik Sense Enterprise-implementatie door naar de app-ID te verwijzen. U kunt alleen laden vanuit een *.qvf*-bestand.

Voorbeelden

| Tekenreeks | Beschrijving |
|---|---|
| `Binary lib://DataFolder/customer.qvw;` | In dit voorbeeld moet het bestand zich in de **map** gegevensverbinding bevinden. Dit kan bijvoorbeeld een map zijn die uw beheerder op de Qlik Sense-server maakt. Klik op **Nieuwe verbinding maken** in de editor voor laden van gegevens en selecteer vervolgens **map** onder **Bestandslocaties**. |
| `Binary customer.qvf;` | In dit voorbeeld moet het bestand zich in de werkdirectory van Qlik Sense bevinden. |
| `Binary c:\qv\customer.qvw;` | Dit voorbeeld waarbij een absoluut bestandspad wordt gebruikt, werkt alleen in de bestaande scriptmodus. |

## Comment field

Hiermee kunt u de opmerkingen voor velden (metagegevens) uit databases en spreadsheets weergeven. Veldnamen die niet in de app aanwezig zijn, worden genegeerd. Als meerdere exemplaren van een veldnaam worden aangetroffen, wordt de laatste waarde gebruikt.

**Syntaxis:**

```
comment [fields] *fieldlist using mapname
```

```
comment [field] fieldname with comment
```

De toewijzingstabel moet uit twee kolommen bestaan, de eerste met veldnamen en de tweede met de opmerkingen.

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| **fieldlist* | Een met door komma's gescheiden lijst velden die van commentaar moeten worden voorzien. Met een * als veldlijst worden alle velden aangeduid. De jokertekens * en ? zijn toegestaan in de veldnamen. Het gebruik van aanhalingstekens rond veldnamen kan nodig zijn bij het gebruik van jokertekens. |
| *mapname* | De naam van een toewijzingstabel die eerder is gelezen in een opdracht mapping **LOAD** of mapping **SELECT**. |
| *fieldname* | De naam van het veld waaraan de opmerking moet worden toegevoegd. |
| *comment* | De opmerking die aan het veld moet worden toegevoegd. |

**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

Hiermee kunt u de opmerkingen voor tabellen (metagegevens) uit databases of spreadsheets weergeven.

De tabelnamen die niet in de app aanwezig zijn, worden genegeerd. Als meerdere exemplaren van een tabelnaam zijn aangetroffen, wordt de laatste waarde gebruikt. Met het trefwoord kunnen opmerkingen uit een gegevensbron worden gelezen.

**Syntaxis:**

```
comment [tables] tablelist using mapname
comment [table] tablename with comment
```

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| *tablelist* | (table{,table}) |
| *mapname* | De naam van een toewijzingstabel die eerder is gelezen in een opdracht mapping **LOAD** of mapping **SELECT**. |
| *tablename* | De naam van de tabel waaraan de opmerking moet worden toegevoegd. |
| *comment* | De opmerking die aan de tabel moet worden toegevoegd. |

**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

Met de opdracht **CONNECT** wordt de toegang van Qlik Sense tot een algemene database via de OLE DB/ODBC-interface vastgelegd. Voor ODBC moet de gegevensbron eerst worden opgegeven met behulp van ODBC-beheer.

*Deze functionaliteit is niet beschikbaar in Qlik Sense SaaS.*

*Deze opdracht ondersteunt alleen mapgegevensverbindingen in de standaardmodus.*

**Syntaxis:**

```
ODBC CONNECT TO connect-string

OLEDB CONNECT TO connect-string

CUSTOM CONNECT TO connect-string

LIB CONNECT TO connection
```

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| connect-string | `connect-string ::= datasourcename { ; conn-spec-item }`<br>De verbindingstekenreeks bestaat uit de naam van de gegevensbron en een optionele lijst met een of meer specificaties voor de verbinding. Als er spaties voorkomen in de naam van de gegevensbron of als specificaties voor de verbinding worden opgegeven, moet de verbindingstekenreeks tussen aanhalingstekens staan.<br>**datasourcename** moet een gedefinieerde ODBC-gegevensbron zijn of een tekenreeks waarmee een OLE DB-provider wordt gedefinieerd.<br>`conn-spec-item ::=`**`DBQ=`**`database_specifier |`**`DriverID=`**`driver_specifier |`**`UID=`**`userid |`**`PWD=`**`password`<br>Welke specificaties voor de verbinding mogelijk zijn, is afhankelijk van de databases. Voor sommige databases zijn andere specificaties mogelijk dan de hierboven genoemde. Voor OLE DB zijn sommige van de verbindingsspecifieke items verplicht en niet optioneel. |
| connection | De naam van een gegevensverbinding die is opgeslagen in de editor voor het laden van gegevens. |

Als de **ODBC** wordt geplaatst voor **CONNECT**, wordt de ODBC-interface gebruikt, anders wordt OLE DB gebruikt.

Met **LIB CONNECT TO** wordt verbinding gemaakt met een database waarin een gegevensverbinding is opgeslagen die in de editor voor het laden van gegevens werd gemaakt.

**Example 1:**

```
ODBC CONNECT TO 'Sales

DBQ=C:\Program Files\Access\Samples\Sales.mdb';
```

De door deze opdracht gedefinieerde gegevensbron wordt gebruikt door daaropvolgende **Select (SQL)**-opdrachten, tot een nieuwe **CONNECT**-opdracht wordt gemaakt.

**Example 2:**

```
LIB CONNECT TO 'DataConnection';
```

### Connect32

Deze opdracht wordt op dezelfde wijze gebruikt als de **CONNECT**-opdracht, alleen wordt een 64-bits systeem gedwongen om een 32-bits ODBC/OLE DB-provider te gebruiken. Niet van toepassing op custom connect.

### Connect64

Deze opdracht wordt op dezelfde wijze gebruikt als de **CONNECT**-opdracht, alleen wordt het gebruik van een 64-bits provider afgedwongen. Niet van toepassing op custom connect.

## Declare

De opdracht **Declare** wordt gebruikt voor het maken van velddefinities, waarbij u relaties kunt definiëren tussen velden of functies. Een reeks van velddefinities kan worden gebruikt om automatisch afgeleide velden te genereren, die kunnen worden gebruikt als dimensies. Zo kunt u bijvoorbeeld een agendadefinitie maken en die gebruiken om gerelateerde dimensies te genereren, zoals jaar, maand, week en dag, vanuit een datumveld.

U kunt **Declare** gebruiken om een nieuwe velddefinitie op te zetten of om een velddefinitie te maken op basis van een reeds bestaande definitie.

### Een nieuwe velddefinitie opzetten

**Syntaxis:**

```
definition_name:

Declare [Field[s]] Definition [Tagged tag_list ]

[Parameters parameter_list ]

Fields field_list
```

**Argumenten:**

| Argument | Beschrijving |
|---|---|
| definition_name | Naam van de velddefinitie, afgesloten met een dubbele punt.<br><br>*Maak geen gebruik van autoCalendar als naam voor velddefinities, aangezien deze naam is gereserveerd voor automatisch gegenereerde agendasjablonen.*<br><br>**Voorbeeld:**<br><br>`Calendar:` |

| Argument | Beschrijving |
|---|---|
| tag_list | Een door komma's gescheiden lijst met tags om toe te passen op velden die zijn afgeleid van de velddefinitie. Het toepassen van tags is optioneel, maar als u geen tags toepast die worden gebruikt voor het specificeren van de sorteervolgorde, zoals $date, $numeric of $text, wordt het afgeleide veld standaard gesorteerd op laadvolgorde.<br>**Voorbeeld:**<br>`'$date'Thank you for bringing this to our attention, and apologies for the inconvenience.` |
| parameter_list | Een door komma's gescheiden lijst met parameters. Een parameter wordt gedefinieerd in de vorm `name=value` en krijgt een beginwaarde toegewezen, die kan worden vervangen als een velddefinitie opnieuw wordt gebruikt. Optioneel.<br>**Voorbeeld:**<br>`first_month_of_year = 1` |
| field_list | Een door komma's gescheiden lijst met velden die worden gegenereerd als de velddefinitie wordt gebruikt. Een veld wordt gedefinieerd in de vorm `<expression> As field_name tagged tag`. Gebruik `$1` om te verwijzen naar het gegevensveld op basis waarvan de afgeleide velden moeten worden gegenereerd.<br>**Voorbeeld:**<br>`Year($1) As Year tagged ('$numeric')` |

**Voorbeeld:**

```
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')
;
```

De agenda is nu gedefinieerd en u kunt deze toepassen op de datumvelden die zijn geladen, in dit geval OrderDate en ShippingDate, met behulp van een clausule **Derive**.

## Een bestaande velddefinitie opnieuw gebruiken

**Syntaxis:**

```
<definition name>:

Declare [Field][s] Definition

Using <existing_definition>

[With <parameter_assignment> ]
```

**Argumenten:**

| Argument | Beschrijving |
|---|---|
| definition_name | Naam van de velddefinitie, afgesloten met een dubbele punt.<br>**Voorbeeld:**<br>`MyCalendar:` |
| existing_definition | De velddefinitie die opnieuw moet worden gebruikt bij het maken van de nieuwe velddefinitie. De nieuwe velddefinitie functioneert op dezelfde wijze als de definitie waarop deze is gebaseerd, behalve als u parameter_assignment gebruikt om een waarde te wijzigen die wordt gebruikt in de velduitdrukkingen.<br>**Voorbeeld:**<br>`Using Calendar` |
| parameter_assignment | Een door komma's gescheiden lijst met parametertoewijzingen. Een parametertoewijzing wordt gedefinieerd in de vorm `name=value` en vervangt de parameterwaarde die is ingesteld in de basisvelddefinitie. Optioneel.<br>**Voorbeeld:**<br>`first_month_of_year = 4` |

**Voorbeeld:**

In dit voorbeeld gebruiken we de agendadefinitie opnieuw die we in het voorafgaande voorbeeld hebben gemaakt. In dit geval willen we een fiscaal jaar gebruiken dat begint in april. Dit wordt gedaan door de waarde 4 toe te wijzen aan de parameter first_month_of_year, hetgeen van invloed is op het veld DayNumberOfYear dat is wordt gedefinieerd.

In het voorbeeld wordt ervan uitgegaan dat u de voorbeeldgegevens en velddefinitie uit het voorafgaande voorbeeld gebruikt.

```
MyCalendar:
DECLARE FIELD DEFINITION USING Calendar WITH first_month_of_year=4;
```

```
DERIVE FIELDS FROM FIELDS OrderDate,ShippingDate USING MyCalendar;
```

Als u het gegevensscript opnieuw hebt geladen, zijn de gegenereerde velden beschikbaar in de werkbladeditor, met de namen OrderDate.MyCalendar.* en ShippingDate.MyCalendar.*.

## Derive

De opdracht **Derive** wordt gebruikt voor het genereren van afgeleide velden op basis van een velddefinitie die is gemaakt met een opdracht **Declare**. U kunt opgeven voor welke gegevensvelden velden moeten worden afgeleid of u kunt deze expliciet of impliciet afleiden op basis van veldtags.

**Syntaxis:**

```
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
```

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| definition | Naam van de velddefinitie die moet worden gebruikt bij het afleiden van velden.<br><br>**Voorbeeld:** `Calendar` |
| field_list | Een door komma's gescheiden lijst met gegevensvelden waaruit de afgeleide velden moeten worden gegenereerd, op basis van de velddefinitie. De gegevensvelden moeten velden zijn die u al hebt geladen in het script.<br><br>**Voorbeeld:** `OrderDate, ShippingDate` |
| tag_list | Een door komma's gescheiden lijst met tags. Afgeleide velden worden gegenereerd voor alle gegevensvelden met enige van de tags in de lijst. De lijst met tags moet tussen ronde haakjes worden geplaatst.<br><br>**Voorbeeld:** `('$date','$timestamp')` |

**Voorbeelden:**

- Velden afleiden voor specifieke gegevensvelden.
  In dit geval geven we de velden OrderDate en ShippingDate op.
  `DERIVE FIELDS FROM FIELDS OrderDate,ShippingDate USING Calendar;`
- Velden afleiden voor alle velden met een specifieke tag.
  In dit geval leiden we velden af die zijn gebaseerd op Calendar voor alle velden met een tag `$date`.
  `DERIVE FIELDS FROM EXPLICIT TAGS ('$date') USING Calendar;`

- Velden afleiden voor alle velden met de velddefinitietag.
  In dit geval leiden we velden af voor alle gegevensvelden met dezelfde tag als de velddefinitie Calendar. In dit geval is dit `$date`.
  ```
  DERIVE FIELDS FROM IMPLICIT TAG USING Calendar;
  ```

## Direct Query

De instructie **DIRECT QUERY** zorgt ervoor dat u tabellen kunt openen met een ODBC- of OLE DB-verbinding met gebruik van de Direct Discovery-functie.

**Syntaxis:**

```
DIRECT QUERY DIMENSION fieldlist [MEASURE fieldlist] [DETAIL fieldlist] FROM
tablelist
[WHERE where_clause]
```

De trefwoorden **DIMENSION**, **MEASURE** en **DETAIL** kunnen in willekeurige volgorde worden gebruikt.

De trefwoordclausules **DIMENSION** en **FROM** zijn vereist in alle **DIRECT QUERY**-opdrachten. Het trefwoord **FROM** moet na het trefwoord **DIMENSION** komen.

De velden die zijn opgegeven direct na het trefwoord **DIMENSION** worden in het geheugen geladen om associaties tot stand te brengen tussen gegevens in het geheugen en Direct Discovery-gegevens.

*De **DIRECT QUERY**-opdracht kan geen **DISTINCT**- of **GROUP BY**-clausules bevatten.*

Met het trefwoord **MEASURE** kunt u velden definiëren waarvan Qlik Sense zich bewust is op “metaniveau”. De feitelijke gegevens van een metingsveld bevinden zich alleen tijdens het laden van de gegevens in de database en worden op ad hoc basis opgehaald via de diagramuitdrukkingen die worden gebruikt in een visualisatie.

Gewoonlijk kunnen velden met discrete waarden die zullen worden gebruikt als dimensies het beste worden geladen met het trefwoord **DIMENSION**, terwijl getallen die zullen worden gebruikt in aggregaties uitsluitend zouden meten worden geselecteerd met het trefwoord **MEASURE**.

Velden van het type **DETAIL** bieden informatie of details, zoals commentaarvelden, die een gebruiker wellicht wil weergeven in een tabelvak waarbij wordt geanalyseerd op detailniveau. Velden van het type **DETAIL** kunnen niet worden gebruikt in diagramuitdrukkingen.

De DIRECT QUERY-opdracht is zodanig ontworpen dat deze gegevensbronneutraal is voor gegevensbronnen die SQL ondersteunen.SQL Om die reden kan dezelfde DIRECT QUERY-opdracht ongewijzigd worden gebruikt voor verschillende SQL-databases.SQL Direct Discovery genereert zo nodig query's voor de desbetreffende database.

Er kan gebruik worden gemaakt van speciale gegevenbronsyntaxis als de gebruiker weet op welke database query's moeten worden uitgevoerd en gebruik wil maken van databasespecifieke uitbreidingen van SQL. Er wordt speciale gegevensbronsyntaxis ondersteund:

- Als velduitdrukkingen in **DIMENSION**- en **MEASURE**-clausules
- Als de inhoud van de **WHERE**-clausule

Voorbeelden:

```
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"')
```

*De volgende termen worden gebruikt als trefwoorden en kunnen niet zonder aanhalingstekens worden gebruikt als kolom- of veldnamen: and, as, detach, detail, dimension, distinct, from, in, is, like, measure, native, not, or, where*

**Argumenten:**

| Argument | Beschrijving |
|---|---|
| fieldlist | Een door komma's gescheiden lijst met veldspecificaties, . *fieldname {, fieldname}*. Een veldspecificatie kan een veldnaam zijn. In dat geval wordt dezelfde naam gebruikt voor de databasekolomnaam en de Qlik Sense-veldnaam. Een veldspecificatie kan ook een "veldalias" zijn. In dat geval wordt een Qlik Sense-veldnaam gegeven aan een database-uitdrukking of kolomnaam. |
| tablelist | Een lijst met de namen van tabellen of weergaven in de database waaruit gegevens worden geladen. Gewoonlijk zijn dit weergaven die een JOIN bevatten die worden uitgevoerd op de database. |
| where_ clause | De volledige syntaxis van **WHERE**-clausules voor databases wordt hier niet gedefinieerd, maar de meesteSQL-uitdrukkingen zijn toegestaan, met inbegrip van het gebruik van functieaanroepen, de **LIKE**-operator voor tekenreeksen, **IS NULL** en **IS NOT NULL**, en **IN. BETWEEN** is niet inbegrepen.<br><br>**NOT** is een unaire operator, in tegenstelling tot een modificatie voor bepaalde trefwoorden.<br><br>Voorbeelden:<br><br>`WHERE x > 100 AND "Region Code" IN ('south', 'west')`<br>`WHERE Code IS NOT NULL and Code LIKE '%prospect'`<br>`WHERE NOT X in (1,2,3)`<br>Het laatste voorbeeld kan niet worden geschreven als:<br><br>`WHERE X NOT in (1,2,3)` |

**Voorbeeld:**

In dit voorbeeld wordt een databasetabel genaamd TableName gebruikt, met de velden Dim1, Dim2, Num1, Num2 en Num3.Dim1 en Dim2 worden geladen in de Qlik Sense gegevensverzameling.

```
DIRECT QUERY DIMENSTION Dim1, Dim2 MEASURE Num1, Num2, Num3 FROM TableName ;
```

Dim1 en Dim2 zijn beschikbaar voor gebruik als dimensies. Num1, Num2 en Num3 zijn beschikbaar voor aggregaties. Dim1 en Dim2 zijn eveneens beschikbaar voor aggregaties. Het type aggregaties waarvoor Dim1 en Dim2 kunnen worden gebruikt is afhankelijk van het gegevenstype. In veel gevallen kunnen bijvoorbeeld velden van het type tekenreeksgegevens bevatten zoals namen of rekeningnummers. Deze velden kunnen niet worden opgeteld maar wel worden geteld: .count(Dim1)

*__DIRECT QUERY__-opdrachten worden rechtstreeks in de scripteditor geschreven. U kunt het samenstellen van __DIRECT QUERY__-opdrachten vergemakkelijken door een __SELECT__-opdracht te maken op basis van een gegevensverbinding en vervolgens het gegenereerde script bewerken om deze in een __DIRECT QUERY__-opdracht te veranderen.*
*Bijvoorbeeld de __SELECT__-opdracht:*

```
SQL SELECT
 SalesOrderID,
 RevisionNumber,
 OrderDate,
 SubTotal,
 TaxAmt
FROM MyDB.Sales.SalesOrderHeader;
```

*kan worden gewijzigd in de volgende __DIRECT QUERY__-opdracht:*

```
DIRECT QUERY
 DIMENSION
 SalesOrderID,
 RevisionNumber

MEASURE
 SubTotal,
 TaxAmt

DETAIL
OrderDate

FROM MyDB.Sales.SalesOrderHeader;
```

## Direct Discovery-veldenlijsten

Een veldenlijst is een door komma's gescheiden lijst met veldspecificaties, *fieldname {, fieldname}*. Een veldspecificatie kan een veldnaam zijn. In dat geval wordt dezelfde naam gebruikt voor de databasekolomnaam en de veldnaam. Een veldspecificatie kan ook een "veldalias" zijn. In dat geval wordt een Qlik Sense-veldnaam gegeven aan een database-uitdrukking of kolomnaam.

Veldnamen kunnen simpele namen of namen tussen aanhalingstekens zijn. Een simpele naam begint met een alfabetisch Unicode-teken en wordt gevolgd door een combinatie van alfabetische of numerieke tekens of onderstrepingstekens. Namen tussen aanhalingstekens beginnen met een dubbel aanhalingsteken en bevatten een willekeurige reeks van tekens. Als een naam tussen aanhalingstekens dubbele aanhalingstekens bevat, worden deze aanhalingstekens aangegeven door middel van twee dubbele aanhalingstekens naast elkaar.

Qlik Sense-veldnamen zijn hoofdlettergevoelig. Databaseveldnamen kunnen al dan niet hoofdlettergevoelig zijn, afhankelijk van de database. Een Direct Discovery-query behoudt de weergave in hoofdletters of kleine letters van alle veldidentificaties en aliassen. In het volgende voorbeeld wordt de alias "MyState" intern gebruikt voor het opslaan van de gegevens uit de databasekolom "STATEID".

```
DIRECT QUERY Dimension STATEID as MyState Measure AMOUNT from SALES_TABLE;
```

Dit wijkt af van het resultaat van een **SQL Select**-opdracht met een alias. Als de alias niet expliciet tussen aanhalingstekens staat, bevat het resultaat de standaardcase van de kolom die is geretourneerd door de doeldatabase. In het volgende voorbeeld, maakt de **SQL Select**-opdracht aan een an Oracle-database "MYSTATE," met allemaal hoofdletters, als de interne Qlik Sense-alias zelfs al is de alias opgegeven als een combinatie van hoofdletters en kleine letters. De **SQL Select**-opdracht maakt gebruik van de kolomnaam die wordt geretourneerd door de database, die in het geval van Oracle volledig in hoofdletters is.

```
SQL Select STATEID as MyState, STATENAME from STATE_TABLE;
```

U kunt dit gedrag vermijden door de LOAD-opdracht te gebruiken om de alias op te geven.

```
Load STATEID as MyState, STATENAME;
SQL Select STATEID, STATEMENT from STATE_TABLE;
```

In dit voorbeeld wordt de kolom "STATEID" intern opgeslagen door Qlik Sense als "MyState".

De meeste scalaire uitdrukkingen in databases zijn toegestaan als veldspecificaties. Functieaanroepen kunnen eveneens worden gebruikt in veldspecificaties. Uitdrukkingen kunnen constanten bevatten die booleaanse, numerieke of tekenreekswaarden zijn tussen enkele aanhalingstekens (ingesloten enkele aanhalingstekens worden aangeduid met twee enkele aanhalingstekens naast elkaar).

**Voorbeelden:**

```
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"

      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 biedt geen ondersteuning voor het gebruik van aggregaties in **LOAD**-opdrachten. Als aggregaties worden gebruikt, zijn de resultaten onvoorspelbaar. Een **LOAD**-opdracht zoals de volgende mag niet worden gebruikt:

```
DIRECT QUERY DIMENSION stateid, SUM(amount*7) AS MultiFirst MEASURE amount FROM sales_table;
```

De **SUM** mag niet in de **LOAD**-opdracht worden opgenomen.

Direct Discovery biedt tevens geen ondersteuning voor Qlik Sense-functies in **Direct Query**-opdrachten. Zo resulteert bijvoorbeeld de volgende specificatie voor een **DIMENSION**-veld in een fout als het "Mth"-veld wordt gebruikt als een dimensie in een visualisatie:

```
month(ModifiedDate) as Mth
```

## Directory

De opdracht **Directory** legt vast in welke directory moet worden gezocht naar gegevensbestanden in volgende **LOAD**-opdrachten, totdat een nieuwe **Directory**opdracht wordt gegeven.

**Syntaxis:**

```
Directory[path]
```

Als de **Directory**-opdracht wordt opgegeven zonder een **path** of als het pad wordt weggelaten, zoekt Qlik Sense in de werkdirectory van Qlik Sense.

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| **path** | Een tekst die kan worden geïnterpreteerd als het pad naar het data-bestand.<br>Het pad is het pad naar het bestand. Het is:<br>• absoluut<br>**Voorbeeld:** *c:\data*<br>• relatief ten opzichte van de werkmap van de Qlik Sense-app.<br>**Voorbeeld:** *data*<br>• URL-adres (HTTP of FTP) dat verwijst naar een locatie op internet of intranet.<br>**Voorbeeld:** *http://www.qlik.com* |

**Voorbeelden:**

```
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

De opdracht **Disconnect** verbreekt de huidige ODBC-/OLE DB-/aangepaste verbinding. Deze opdracht is optioneel.

**Syntaxis:**

```
Disconnect
```

De verbinding wordt automatisch afgebroken wanneer een nieuwe **connect**-opdracht wordt uitgevoerd of wanneer het script volledig is uitgevoerd.

**Voorbeeld:**

```
Disconnect;
```

## Drop

Het sleutelwoord **Drop** voor scripts kan worden gebruikt om tabellen of velden uit de database te verwijderen.

### Drop field

Een of meer Qlik Sense-velden kunnen op elk moment tijdens de uitvoering van een script worden verwijderd uit het gegevensmodel, en dus uit het geheugen, met behulp van de opdracht **drop field**. De eigenschap "distinct" van een tabel wordt verwijderd na een **drop field**-opdracht.

> *Zowel **drop field** als **drop fields** zijn toegestane vormen en hebben hetzelfde effect. Als geen tabel wordt opgegeven, wordt het veld uit alle tabellen waarin het voorkomt verwijderd.*

**Syntaxis:**

```
Drop field fieldname { , fieldname2 ...} [from tablename1 { , tablename2
...}]
Drop fields fieldname { , fieldname2 ...} [from tablename1 { , tablename2
...}]
```

**Voorbeelden:**

```
Drop field A;
Drop fields A,B;
Drop field A from X;
Drop fields A,B from X,Y;
```

### Drop table

Een of meer interne tabellen in Qlik Sense kunnen op elk moment tijdens de uitvoering van een script worden verwijderd uit het gegevensmodel, en dus uit het geheugen, met behulp van de opdracht **drop table**.

**Syntaxis:**

```
drop table  tablename {, tablename2 ...}

drop tables tablename {, tablename2 ...}
```

*De vormen **drop table** en **drop tables** worden beide geaccepteerd.*

De volgende items gaan als gevolg hiervan verloren:

- De feitelijke tabel(len).
- Alle velden die geen onderdeel vormen van de resterende tabellen.
- Veldwaarden in resterende velden die uitsluitend uit de te wissen tabel(len) afkomstig zijn.

Voorbeelden en resultaten:

| Voorbeeld | Resultaat |
|---|---|
| `drop table Orders, Salesmen, T456a;` | Deze regel verwijdert drie tabellen uit het geheugen. |
| `Tab1:`<br><br>`Load * Inline [`<br><br>`Customer, Items, UnitPrice`<br><br>`Bob, 5, 1.50`<br><br>`];`<br><br><br>`Tab2:`<br><br>`LOAD Customer, Sum( Items * UnitPrice ) as Sales`<br><br>`resident Tab1`<br><br>`group by Customer;`<br><br><br>`drop table Tab1;` | Nadat de tabel *Tab2* is gemaakt, wordt de tabel *Tab1* verwijderd. |

## Drop table

Een of meer interne tabellen in Qlik Sense kunnen op elk moment tijdens de uitvoering van een script worden verwijderd uit het gegevensmodel, en dus uit het geheugen, met behulp van de opdracht **drop table**.

**Syntaxis:**

```
drop table  tablename {, tablename2 ...}
```

```
drop tables tablename {, tablename2 ...}
```

*De vormen **drop table** en **drop tables** worden beide geaccepteerd.*

De volgende items gaan als gevolg hiervan verloren:

- De feitelijke tabel(len).
- Alle velden die geen onderdeel vormen van de resterende tabellen.
- Veldwaarden in resterende velden die uitsluitend uit de te wissen tabel(len) afkomstig zijn.

Voorbeelden en resultaten:

| Voorbeeld | Resultaat |
|---|---|
| `drop table Orders, Salesmen, T456a;` | Deze regel verwijdert drie tabellen uit het geheugen. |
| `Tab1:`<br><br>`Load * Inline [`<br><br>`Customer, Items, UnitPrice`<br><br>`Bob, 5, 1.50`<br><br>`];`<br><br><br>`Tab2:`<br><br>`LOAD Customer, Sum( Items * UnitPrice ) as Sales`<br><br>`resident Tab1`<br><br>`group by Customer;`<br><br><br>`drop table Tab1;` | Nadat de tabel *Tab2* is gemaakt, wordt de tabel *Tab1* verwijderd. |

## Execute

De **Execute**-opdracht wordt gebruikt voor het uitvoeren van andere programma's terwijl gegevens worden geladen in Qlik Sense. Bijvoorbeeld voor het uitvoeren van noodzakelijke conversies.

*Deze functionaliteit is niet beschikbaar in Qlik Sense SaaS.*

*Deze opdracht wordt niet ondersteund in de standaardmodus.*

**Syntaxis:**

```
execute commandline
```

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| *commandline* | Een tekst die door het besturingssysteem als een opdrachtregel kan worden geïnterpreteerd. U kunt verwijzen naar een absoluut bestandspad of naar een mappad lib://. |

Als u **Execute** wilt gebruiken, moet aan de volgende voorwaarden worden voldaan:

- U moet werken in de bestaande modus (van toepassing voor Qlik Sense en Qlik Sense Desktop).
- U moet OverrideScriptSecurity instellen op 1 in *Settings.ini* (van toepassing voor Qlik Sense). *Settings.ini* bevindt zich in *C:\ProgramData\Qlik\Sense\Engine* en is gewoonlijk een leeg bestand.

*Als u OverrideScriptSecurity zo instelt, dat **Execute** wordt ingeschakeld, kan elke gebruiker bestanden uitvoeren op de server. Een gebruiker kan bijvoorbeeld een uitvoerbaar bestand aan een app toevoegen en vervolgens het bestand uitvoeren in het load-script voor gegevens.*

**Doe het volgende:**

1. Maak een kopie van *Settings.ini* en open het in een teksteditor.
2. Controleer of het bestand *[Settings 7]* bevat op de eerste regel.
3. Voeg een nieuwe regel in en typ *OverrideScriptSecurity=1*.
4. Voeg een lege regel in aan het einde van het bestand.
5. Sla het bestand op.
6. Vervang *Settings.ini* door uw bewerkte bestand.
7. Start Qlik Sense Engine Service (QES) opnieuw op.

*Als Qlik Sense wordt uitgevoerd als een service, werken sommige opdrachten mogelijk niet zoals verwacht.*

**Voorbeeld:**

```
Execute C:\Program Files\Office12\Excel.exe;
Execute lib://win\notepad.exe // win is a folder connection referring to c:\windows
```

## Field/Fields

De sleutelwoorden **Field** en **Fields** voor scripts worden gebruikt in opdrachten **Declare**, **Derive**, **Drop**, **Comment**, **Rename** en **Tag/Untag**.

## FlushLog

Met de opdracht **FlushLog** dwingt u af dat Qlik Sense de inhoud van de scriptbuffer naar het scriptlogbestand schrijft.

**Syntaxis:**

```
FlushLog
```

De inhoud van de buffer wordt naar het logbestand geschreven. Deze opdracht kan nuttig zijn bij de foutsporing, aangezien u gegevens ontvangt die anders verloren zouden zijn gegaan als de uitvoering van een script mislukt.

**Voorbeeld:**

```
FlushLog;
```

## Force

Met de opdracht **force** kunt u afdwingen dat Qlik Sense veldnamen en veldwaarden van de daaropvolgende **LOAD**- en **SELECT**-opdrachten interpreteert als geschreven met alleen hoofdletters, met alleen kleine letters, met beginhoofdletters of zoals ze worden weergegeven (gemengd). Met deze opdracht kunt u veldwaarden van tabellen interpreteren volgens verschillende conventies.

De **force**-instructie kan ook veldnamen wijzigen tijdens een lading of selecteren met de volgende gegevensbronnen:

- QVD
- CSV (tekstbestanden)
- XLS
- QVX (bestanden en ODBC-verbindingen)

De **force**-instructie kan alleen veldnamen wijzigen als de gegevens zijn geladen in de compacte modus (geladen met *)

De veldnamen van de volgende gegevensbronnen worden niet beïnvloed door de **force**-instructie.

- JSON
- Parquet
- XML
- XLSX

**Syntaxis:**

```
Force ( capitalization | case upper | case lower | case mixed )
```

Als er niets is gespecificeerd, wordt force mixed case verondersteld. De force-opdracht is geldig tot een nieuwe force-opdracht wordt gegeven.

De opdracht **force** werkt niet in de toegangssectie: alle veldwaarden die zijn geladen zijn niet hoofdlettergevoelig.

Voorbeelden en resultaten

| Voorbeeld | Resultaat |
|---|---|
| Dit voorbeeld laat zien hoe beginhoofdletters kunnen worden afgedwongen.<br>`FORCE Capitalization;`<br>`Capitalization:`<br>`LOAD * Inline [`<br>`ab`<br>`Cd`<br>`eF`<br>`GH`<br>`];` | De tabel **Capitalization** bevat de volgende waarden:<br>`Ab`<br>`Cd`<br>`Ef`<br>`Gh`<br>Alle waarden beginnen met een hoofdletter. |

| Voorbeeld | Resultaat |
| --- | --- |
| Dit voorbeeld laat zien hoe hoofdlettergebruik kan worden afgedwongen.<br>`FORCE Case Upper;`<br>`CaseUpper:`<br>`LOAD * Inline [`<br>`ab`<br>`Cd`<br>`eF`<br>`GH`<br>`];` | De tabel **CaseUpper** bevat de volgende waarden:<br>`AB`<br>`CD`<br>`EF`<br>`GH`<br>Alle waarden zijn in hoofdletters. |
| Dit voorbeeld laat zien hoe het gebruik van kleine letters kan worden afgedwongen.<br>`FORCE Case Lower;`<br>`CaseLower:`<br>`LOAD * Inline [`<br>`ab`<br>`Cd`<br>`eF`<br>`GH`<br>`];` | De tabel **CaseLower** bevat de volgende waarden:<br>`ab`<br>`cd`<br>`ef`<br>`gh`<br>Alle waarden zijn in kleine letters. |

| Voorbeeld | Resultaat |
|---|---|
| Dit voorbeeld laat zien hoe het gebruik van een combinatie van kleine letters en hoofdletters kan worden afgedwongen.<br>`FORCE Case Mixed;`<br>`CaseMixed:`<br>`LOAD * Inline [`<br>`ab`<br>`Cd`<br>`eF`<br>`GH`<br>`];` | De tabel **CaseMixed** bevat de volgende waarden:<br>`ab`<br>`Cd`<br>`eF`<br>`GH`<br>Alle waarden worden weergegeven zoals zij voorkomen in het script. |

**Zie ook:**

## From

Het sleutelwoord **From** voor scripts wordt gebruikt in opdrachten **Load** om te verwijzen naar een bestand en in opdrachten **Select** om te verwijzen naar een databasetabel of -weergave.

## Load

Met de opdracht **LOAD** worden velden geladen uit een bestand, uit gegevens die in het script zijn gedefinieerd, uit een eerder geladen tabel, van een webpagina, uit het resultaat van een daaropvolgende **SELECT**-opdracht of door gegevens automatisch te genereren. U kunt tevens gegevens van analytische verbindingen inladen.

**Syntaxis:**

```
LOAD [ distinct ] fieldlist

[( from file [ format-spec ] |

from_field fieldassource [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 ]
```

Argumenten

| Argument | Beschrijving |
|---|---|
| distinct | U kunt **distinct** als predicaat gebruiken wanneer u alleen unieke records wilt laden. Als er dubbele records zijn, wordt het eerste exemplaar geladen.<br><br>Als u voorgaande loads gebruikt, moet u **distinct** in de eerste laadinstructie plaatsen, aangezien **distinct** alleen van invloed is op de bestemmingstabel. |

| Argument | Beschrijving |
|---|---|
| fieldlist | *fieldlist ::= ( * \| field{, * \| field } )*<br>Een lijst van de velden die geladen moeten worden. Gebruik een * als veldlijst om alle velden in de tabel aan te duiden.<br>*field ::= ( fieldref \| expression ) [***as** *aliasname ]*<br>De velddefinitie moet altijd een literal, een referentie naar een bestaand veld of een uitdrukking bevatten.<br>*fieldref ::= ( fieldname \|@fieldnumber \|@startpos:endpos [ **I** \| **U** \| **R** \| **B** \| **T**] )*<br>*fieldname* is een tekst die identiek is aan een veldnaam in de tabel. Houd er rekening mee dat de veldnaam tussen rechte dubbele aanhalingstekens of vierkante haken moet staan als deze bijvoorbeeld spaties bevat. Soms zijn veldnamen niet expliciet beschikbaar. In dat geval wordt een andere notatie gebruikt:<br><br>@*fieldnumber* geeft het veldnummer in een tabelbestand met scheidingstekens aan. Het moet een positief geheel getal zijn, voorafgegaan door "@". De nummering is altijd vanaf 1 optellend tot het aantal velden.<br><br>@*startpos*:*endpos* geeft de begin- en eindposities van een veld aan in een bestand met een vaste recordlengte. De posities moeten beide positieve gehele getallen zijn. De twee getallen moeten worden voorafgegaan door "@" en gescheiden door een dubbele punt. De nummering is altijd vanaf 1 optellend tot het aantal posities. In het laatste veld wordt **n** gebruikt als eindpositie.<br>• Als @*startpos*:*endpos* direct wordt gevolgd door de tekens **I** of**U**, worden de gelezen bytes geïnterpreteerd als een binair ondertekend (**I**) of niet-ondertekend (**U**) geheel getal (Intel-bytevolgorde). Het aantal gelezen posities moet 1, 2 of 4 zijn.<br>• Als @*startpos*:*endpos* direct wordt gevolgd door het teken **R**, worden de bytes geïnterpreteerd als een binair reëel getal (IEEE 32-bits of 64-bits drijvende komma). Het aantal gelezen posities moet 4 of 8 zijn.<br>• Als @*startpos*:*endpos* direct wordt gevolgd door het teken **B**, worden de gelezen bytes geïnterpreteerd als een BCD (Binary Coded Decimal) volgens de COMP-3-standaard. U kunt een willekeurig aantal bytes opgeven.<br><br>*expression* kan een numerieke functie of een tekenreeksfunctie zijn, gebaseerd op een of meer andere velden in dezelfde tabel. Zie de syntaxis van uitdrukkingen voor meer informatie.<br><br>**as** wordt gebruikt om een nieuwe naam aan een veld toe te wijzen. |

| Argument | Beschrijving |
|---|---|
| from | **from** wordt gebruikt als gegevens moeten worden geladen vanuit een bestand via een map- of webbestandsgegevensverbinding<br><br>*file ::= [ path ] filename*<br><br>**Voorbeeld:** ***'lib://Table Files/'***<br><br>Als er geen pad is opgegeven, zoekt Qlik Sense het bestand in de map die is opgegeven in de opdracht **Directory**. Als er geen opdracht **Directory** is, zoekt Qlik Sense in de werkdirectory, *C:\Users\{user}\Documents\Qlik\Sense\Apps*.<br><br>*In een Qlik Sense-serverinstallatie wordt de werkdirectory opgegeven in Qlik Sense Repository Service. Standaard is dit C:\ProgramData\Qlik\Sense\Apps.*<br><br>De *filename* kan de standaard jokertekens van DOS bevatten ( * en ? ). Daarmee worden alle overeenkomstige bestanden in de opgegeven map geladen.<br>*format-spec ::= ( fspec-item { , fspec-item } )*<br>De opmaakspecificatie bestaat uit een lijst met verschillende opmaakspecificaties, tussen haakjes.<br><br>**Bestaande scriptingmodus**<br><br>In de bestaande scriptmodus, worden tevens de volgende padindelingen ondersteund:<br><br>• absoluut<br><br>**Voorbeeld:** ***c:\data***<br><br>• relatief ten opzichte van de werkmap van de Qlik Sense-app.<br><br>**Voorbeeld:** ***data***<br><br>• URL-adres (HTTP of FTP) dat verwijst naar een locatie op internet of intranet.<br><br>**Voorbeeld:** ***http://www.qlik.com*** |

| Argument | Beschrijving |
|---|---|
| from_field | **from_field** wordt gebruikt als gegevens moeten worden geladen uit een eerder geladen veld.<br>*fieldassource::=***(***tablename, fieldname***)***<br><br>De veldnaam is de naam van de eerder geladen *tablename* en *fieldname*.<br>*format-spec ::= ( fspec-item {, fspec-item } )*<br>De opmaakspecificatie bestaat uit een lijst met verschillende opmaakspecificaties, tussen haakjes. Ga voor meer informatie naar *Opmaakspecificaties (page 176)*.<br><br>***from_field*** *ondersteunt alleen komma's als scheidingsteken in lijsten bij het scheiden van velden in tabellen.* |
| inline | **inline** wordt gebruikt als gegevens moeten worden getypt in het script en niet uit een bestand moeten worden geladen.<br>*data ::= [ text ]*<br><br>Gegevens die u invoert met een **inline**-clausule moeten tussen specifieke tekens worden geplaatst - vierkante haakjes, aanhalingstekens of backticks. De tekst ertussen wordt net zo geïnterpreteerd als de inhoud van een bestand. Waar u bijvoorbeeld een nieuwe regel in een tekstbestand zou invoegen, moet u dat ook doen in de tekst van een **inline**-clausule: u drukt dus op Enter bij het typen van het script.<br><br>In een simpele inline-lading wordt het aantal kolommen bepaald door de eerste regel.<br>*format-spec ::= ( fspec-item {, fspec-item } )*<br>U kunt de inline-lading aanpassen met meerdere van dezelfde indelingspecificatie-items die beschikbaar zijn voor andere geladen tabellen. Deze items worden tussen haakjes getoond. Ga voor meer informatie naar *Opmaakspecificaties (page 176)*.<br><br>Voor meer informatie over inline loads, raadpleegt u Laad de volgende gegevens als een inline load. |
| resident | **resident** wordt gebruikt als gegevens moeten worden geladen uit een eerder geladen tabel.<br>*table label* is een label dat vooraf gaat aan de **LOAD**- of **SELECT**-opdracht(en) waarmee de oorspronkelijke tabel is gemaakt. Het label moet worden opgegeven met een puntkomma aan het eind. |

| Argument | Beschrijving |
|---|---|
| autogenerate | **autogenerate** wordt gebruikt als gegevens automatisch moeten worden gegenereerd door Qlik Sense.<br>*size ::= number*<br><br>*Number* is een geheel getal dat het aantal te genereren records aangeeft.<br><br>De veldenlijst mag geen uitdrukkingen bevatten die gegevens uit een externe gegevensbron of een eerder geladen tabel vereisen, tenzij u verwijst naar een enkele veldwaarde in een eerder geladen tabel met de functie **Peek**. |

| Argument | Beschrijving |
|---|---|
| extension | U kunt gegevens via analytische verbindingen inladen. Als u een functie wilt aanroepen die in de SSE-plugin is gedefinieerd, moet u de clausule **extension** gebruiken of een script evalueren.<br><br>U kunt één tabel naar de SSE-plugin verzenden, waarop één gegevenstabel wordt teruggegeven. Als door de plugin niet de namen worden aangegeven van de velden die worden teruggegeven, krijgen de velden de namen Field1, Field2 enzovoort.<br><br>`Extension pluginname.functionname( tabledescription );`<br><br>• Gegevens inladen met een functie in een SSE-plugin<br>*tabledescription ::= (table { ,tablefield} )*<br>Als u geen tabelvelden aangeeft, worden de velden gebruikt in de volgorde waarin deze zijn ingeladen.<br>• Gegevens inladen door een script in een SSE-plugin te evalueren<br>*tabledescription ::= ( script, table { ,tablefield} )*<br><br>**Afhandeling van gegevenssoorten in de tabelvelddefinitie**<br><br>Gegevenssoorten in analytische verbindingen worden automatisch gedetecteerd. Als de gegevens geen getalswaarden bevatten en ten minste één teksttekenreeks die niet leeg is, wordt het veld als tekstveld beschouwd. In alle andere gevallen wordt het veld als getalsveld beschouwd.<br><br>U kunt de gegevenssoort afdwingen door een veldnaam op te nemen in **String()** voor een tekenreeks of **Mixed()** voor gemengd.<br><br>• Met **String()** maakt u het veld een tekstveld. Als het veld numeriek is, wordt het tekstgedeelte van de dubbele waarde geëxtraheerd. Er wordt niet geconverteerd.<br>• Met **Mixed()** maakt u het veld een veld met dubbele waarden.<br><br>**String()** en **Mixed()** kunnen niet buiten tabelvelddefinities van **extension** worden gebruikt, en u kunt geen andere Qlik Sense -functies in een tabelvelddefinitie gebruiken.<br><br>**Meer over analytische verbindingen**<br><br>U moet analytische verbindingen configureren voordat u deze kunt gebruiken. |
| where | **where** is een clausule die wordt gebruikt om aan te geven of een record wel of niet in de selectie moet worden opgenomen. De selectie wordt opgenomen als *criterion* de waarde True heeft.<br>*criterion* is een logische uitdrukking. |

| Argument | Beschrijving |
|---|---|
| while | **while** is een clausule die aangeeft of een record herhaaldelijk moet worden gelezen. Hetzelfde record wordt gelezen zolang *criterion* de waarde True heeft. Een **while**-clausule is vaak pas nuttig wanneer u deze samen met de functie **IterNo( )** gebruikt.<br><br>*criterion* is een logische uitdrukking. |
| group by | **group by** is een clausule die wordt gebruikt om te definiëren over welke velden de gegevens moeten worden geaggregeerd (gegroepeerd). De aggregatievelden moeten op de een of andere manier in de geladen uitdrukkingen worden opgenomen. U mag geen andere velden dan de aggregatievelden gebruiken buiten aggregatiefuncties in de geladen uitdrukkingen.<br><br>*groupbyfieldlist ::= (fieldname { ,fieldname } )* |
| order by | **order by** is een clausule die wordt gebruikt om de records van een residente tabel te sorteren voordat ze worden verwerkt door de **load**-opdracht. De residente tabel kan worden gesorteerd op één of meer velden in oplopende of aflopende volgorde. Het sorteren gebeurt primair op numerieke waarde en secundair op nationale sorteervolgorde. U kunt deze clausule alleen gebruiken wanneer de gegevensbron een residente tabel is.<br>De volgordevelden geven aan op welk veld de residente tabel wordt gesorteerd. U kunt het veld opgeven met z'n naam of met z'n nummer in de residente tabel (het eerste veld is nummer 1).<br><br>orderbyfieldlist ::= fieldname [ sortorder ] { , fieldname [ sortorder ] }<br><br>*sortorder* is *asc* voor oplopend of *desc* voor aflopend. Als u geen *sortorder* opgeeft, wordt *asc* verondersteld.<br><br>*fieldname*, *path*, *filename* en *aliasname* zijn tekenreeksen die aangeven waar de respectievelijke namen voor staan. Elk veld in de brontabel kan worden gebruikt als *fieldname*. Velden die zijn gemaakt via de as-clausule (*aliasname*) vallen echter buiten het bereik en kunnen niet worden gebruikt binnen dezelfde **load**-opdracht. |

Als u geen gegevensbron opgeeft door middel van de clausule **from**, **inline**, **resident**, **from_fieldextensie** of **autogenerate**, worden gegevens geladen uit de resultaten van de direct volgende **SELECT**- of **LOAD**-instructie. Deze opdracht mag geen prefix hebben.

**Voorbeelden:**

Verschillende bestandstypen laden

Laad een bestand met gegevens met scheidingstekens met standaardopties:

```
LOAD * from data1.csv;
```

Laad een bestand met gegevens met scheidingstekens via een bibliotheekverbinding (DataFiles):

```
LOAD * from 'lib://DataFiles/data1.csv';
```

Laad alle bestanden met gegevens met scheidingstekens via een bibliotheekverbinding (DataFiles):

```
LOAD * from 'lib://DataFiles/*.csv';
```

Laad een bestand met gegevens met scheidingstekens, waarbij een komma is opgegeven als scheidingsteken en ingesloten labels worden gebruikt:

```
LOAD * from 'c:\userfiles\data1.csv' (ansi, txt, delimiter is ',', embedded labels);
```

Laad een bestand met gegevens met scheidingstekens, waarbij een tab is opgegeven als scheidingsteken en ingesloten labels worden gebruikt:

```
LOAD * from 'c:\userfiles\data2.txt' (ansi, txt, delimiter is '\t', embedded labels);
```

Laad een dif-bestand met ingesloten kopteksten:

```
LOAD * from file2.dif (ansi, dif, embedded labels);
```

Laad drie velden vanuit een bestand met vaste recordlengte zonder kopteksten:

```
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);
```

Laad een QVX-bestand, waarbij een absoluut pad wordt opgegeven:

```
LOAD * from C:\qdssamples\xyz.qvx (qvx);
```

#### Webbestanden inladen

Inladen via de standaard-URL die in de gegevensverbinding voor het webbestand is ingesteld:

```
LOAD * from [lib://MyWebFile];
```

Inladen via een specifieke URL in plaats van via de URL die in de gegevensverbinding voor het webbestand is ingesteld:

```
LOAD * from [lib://MyWebFile] (URL is 'http://localhost:8000/foo.bar');
```

Inladen via een specifieke URL die met een dollarteken-uitbreiding in een variabele is ingesteld:

```
SET dynamicURL = 'http://localhost/foo.bar';

LOAD * from [lib://MyWebFile] (URL is '$(dynamicURL)');
```

#### Selecteren van bepaalde velden, velden hernoemen en velden berekenen

Laad slechts drie specifieke velden vanuit een bestand met gegevens met scheidingstekens:

```
LOAD FirstName, LastName, Number from data1.csv;
```

Hernoem het eerste veld tot A en het tweede veld tot B bij het laden van een bestand zonder labels:

```
LOAD @1 as A, @2 as B from data3.txt (ansi, txt, delimiter is '\t', no labels);
```

Laad Name als een samenvoeging van FirstName, een spatie en LastName:

```
LOAD FirstName&' '&LastName as Name from data1.csv;
```

Laad Quantity, Price en Value (het product van Quantity en Price):

```
LOAD Quantity, Price, Quantity*Price as Value from data1.csv;
```

### Bepaalde records selecteren

Laad alleen unieke records. Dubbele records worden verwijderd:

```
LOAD distinct FirstName, LastName, Number from data1.csv;
```

Laad alleen records waarbij het veld Litres een waarde van meer dan nul heeft:

```
LOAD * from Consumption.csv where Litres>0;
```

### Gegevens niet in bestand en automatisch gegenereerde gegevens laden

Laad een tabel met uitgelijnde gegevens en twee velden met de naam CatID en Category:

```
LOAD * Inline

[CatID, Category

0,Regular

1,Occasional

2,Permanent];
```

Laad een tabel met uitgelijnde gegevens en drie velden met de naam UserID, Password en Access:

```
LOAD * Inline [UserID, Password, Access

A, ABC456, User

B, VIP789, Admin];
```

Laad een tabel met 10.000 rijen. Veld A bevat het nummer van de ingelezen record (1,2,3,4,5...) en veld B bevat een willekeurig getal tussen 0 en 1:

```
LOAD RecNo( ) as A, rand( ) as B autogenerate(10000);
```

*De haakjes na autogenerate zijn toegestaan maar niet verplicht.*

### Gegevens laden uit een eerder geladen tabel

Als eerste laden we een bestand met een tabel met scheidingstekens en geven dat de naam tab1:

```
tab1:
```

```
SELECT A,B,C,D from 'lib://DataFiles/data1.csv';
```

Laad velden vanuit de reeds geladen tabel tab1 als tab2:

```
tab2:

LOAD A,B,month(C),A*B+D as E resident tab1;
```

Laad velden uit reeds geladen tabel tab1 maar alleen records waarbij A groter is dan B:

```
tab3:

LOAD A,A+B+C resident tab1 where A>B;
```

Laad velden uit al geladen tabel tab1 geordend volgens A:

```
LOAD A,B*C as E resident tab1 order by A;
```

Laad velden vanuit reeds geladen tabel tab1, geordend volgens het eerste veld en vervolgens het tweede veld:

```
LOAD A,B*C as E resident tab1 order by 1,2;
```

Laad veld vanuit reeds geladen tabel tab1 geordend volgens C aflopend, daarna B in oplopende volgorde, en dan het eerste veld in aflopende volgorde:

```
LOAD A,B*C as E resident tab1 order by C desc, B asc, 1 desc;
```

### Gegevens laden uit eerder geladen velden

Laad veld Types uit eerder geladen tabel Characters als A:

```
LOAD A from_field (Characters, Types);
```

### Gegevens laden uit een volgende tabel (voorafgaande lading)

Laad A, B en berekende velden X en Y vanuit Table1 die is geladen in de volgende instructie **SELECT**:

```
LOAD A, B, if(C>0,'positive','negative') as X, weekday(D) as Y;

SELECT A,B,C,D from Table1;
```

### Gegevens groeperen

Laad velden gegroepeerd (geaggregeerd) op ArtNo:

```
LOAD ArtNo, round(Sum(TransAmount),0.05) as ArtNoTotal from table.csv group by ArtNo;
```

Laad velden gegroepeerd (geaggregeerd) opWeek en ArtNo:

```
LOAD Week, ArtNo, round(Avg(TransAmount),0.05) as WeekArtNoAverages from table.csv group by
Week, ArtNo;
```

### Herhaaldelijk lezen van één record

In dit voorbeeld hebben we een invoerbestand, Grades.csv, met de cijfers voor elke student gecomprimeerd in één veld:

```
Student,Grades

Mike,5234

John,3345

Pete,1234

Paul,3352
```

De cijfers, op een schaal van 1-5, staan voor Math, English, Science en History. We kunnen de cijfers scheiden tot afzonderlijke waarden. Hiervoor lezen we elke record meerdere keren met een **while**-clausule. We gebruiken de functie**IterNo( )** als een teller. Bij elke lezing wordt het cijfer uit de functie **Mid** gehaald en opgeslagen in Grade en wordt het vak geselecteerd op basis van de functie **pick** en opgeslagen in Subject. De laatste **while**-clausule bevat de test waarmee wordt gecontroleerd of alle cijfers zijn gelezen (in dit geval vier per leerling). Dat houdt in dat de volgende leerlingrecord moet worden gelezen.

```
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));
```

Het resultaat is een tabel met de volgende gegevens:

| 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 |

#### Inladen via analytische verbindingen

De onderstaande voorbeeldgegevens worden gebruikt.

```
Values:
Load
 Rand() as A,
 Rand() as B,
 Rand() as C
AutoGenerate(50);
```

### Gegevens inladen met een functie

In deze voorbeelden nemen we aan dat we een plugin met de naam *P* voor analytische verbinding hebben, die een aangepaste functie *Calculate(Parameter1, Parameter2)* bevat. Deze functie geeft de tabel *Results* terug met de velden *Field1* en *Field2*.

```
Load * Extension P.Calculate( Values{A, C} );
```

Hiermee laadt u alle gegevens in die worden teruggegeven nadat de velden A en C naar de functie zijn verzonden.

```
Load Field1 Extension P.Calculate( Values{A, C} );
```

Hiermee laadt u alleen veld Field1 in de velden A en C naar de functie zijn verzonden.

```
Load * Extension P.Calculate( Values );
```

Hiermee laadt u alle gegevens in die worden teruggegeven nadat de velden A en B naar de functie zijn verzonden. Als velden niet gespecificeerd zijn, worden A en B gebruikt als deze het eerst zijn in de volgorde in de tabel.

```
Load * Extension P.Calculate( Values {C, C});
```

Hiermee laadt u alle velden in die worden teruggegeven nadat het veld C naar beide parameters van de functie is verzonden.

```
Load * Extension P.Calculate( Values {String(A), Mixed(B)});
```

Hiermee laadt u alle velden in die worden teruggegeven nadat het veld A geforceerd als tekenreeks en het veld B geforceerd als numeriek naar de functie zijn verzonden.

**Gegevens inladen door een script te evalueren**

```
Load A as A_echo, B as B_echo Extension R.ScriptEval( 'q;', Values{A, B} );
```
Laad de tabel die door het script q wordt teruggegeven als de waarden van A en B worden verzonden.

```
Load * Extension R.ScriptEval( '$(My_R_Script)', Values{A, B} );
```
Hiermee laadt u de tabel in die door het script wordt teruggegeven in de variabele My_R_Script als de waarden A en B worden verzonden.

```
Load * Extension R.ScriptEval( '$(My_R_Script)', Values{B as D, *} );
```
Hiermee laadt u de tabel in die door het script wordt teruggegeven in de variabele My_R_Script als de waarden van B worden herbenoemd tot D, A en C. Als u * gebruikt, verzendt u de overgebleven ongekoppelde velden.

*Bestandsextensies van DataFiles-verbindingen zijn hoofdlettergevoelig. Bijvoorbeeld: .qvd.*

## Opmaakspecificaties

Elke indelingsspecificatie definieert een bepaalde eigenschap van het tabelbestand:

**`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`]

### Tekenset

Tekenset is een bestandsspecificatie voor de **LOAD**-opdracht die de tekenset definieert die wordt gebruikt in het bestand.

De specificaties **ansi**, **oem** en **mac** werden gebruikt in QlikView 12 en werken nog steeds. Zij worden echter niet gegenereerd bij het maken van de **LOAD**-opdracht met Qlik Sense.

**Syntaxis:**

```
utf8 | unicode | ansi | oem | mac | codepage is
```

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| **utf8** | UTF-8-tekenset |
| **unicode** | Unicode-tekenset |
| **ansi** | Windows, codepagina 1252 |

| Argument | Beschrijving |
|---|---|
| **oem** | DOS, OS/2, AS400 en andere |
| **mac** | Codepagina 10000 |
| **codepage is** | Met de specificatie **codepage** is het mogelijk om elke Windows-codepagina te gebruiken als *N*. |

**Beperkingen:**

Conversie van de **oem**-tekenreeks is niet geïmplementeerd voor macOS. Als geen tekenset is opgegeven, wordt onder Windows codepagina 1252 gebruikt.

**Voorbeeld:**

```
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)
```

**Zie ook:**

p *Load (page 163)*

### Tabelopmaak

De tabelindeling is een bestandsspecificatie voor de **LOAD**-opdracht waarmee het bestandstype wordt vastgelegd. Als er niets is gespecificeerd, wordt een *.txt*-bestand verondersteld.

Tabelopmaaktypen

| Waardetype | Beschrijving |
|---|---|
| txt | In een tekstbestand met scheidingstekens worden de kolommen in de tabel gescheiden door een bepaald teken. |

| Waardetype | Beschrijving |
|---|---|
| fix | In een bestand met een vaste recordlengte is elke kolom precies een bepaald aantal tekens breed.<br><br>Veel bestanden met vaste recordlengte bevatten records die via een einde regel van elkaar zijn gescheiden, maar er zijn meer geavanceerde opties om de recordgrootte in bytes op te geven of om records meer dan één regel lang te laten zijn met **Record is**.<br><br>*Als de gegevens tekens met meerdere bytes bevatten, kunnen afbreekposities in velden onjuist uitgelijnd raken aangezien de indeling is gebaseerd op een vaste lengte in bytes.* |
| dif | In een *.dif*-bestand (Data Interchange Format) wordt een speciale tabelindeling gebruikt. |
| biff | Qlik Sense kan ook gegevens in standaard Excel-bestanden interpreteren, door middel van de *biff*-indeling (Binary Interchange File Format). |
| ooxml | Excel 2007 en hogere versies gebruiken de indeling ooxml *.xslx*.<br><br>De specificatie **Tabel is** kan worden gebruikt om de werkbladnaam te definiëren die als een tabel moet worden geladen.<br><br>*Tabel is (page 182)* |
| html | Als de tabel onderdeel vormt van een html-pagina of -bestand, moet html worden gebruikt. |
| xml | xml (Extensible Markup Language) is een veelgebruikte opmaaktaal voor de representatie van gegevensstructuren in tekstindeling.<br><br>De specificatie **Tabel is** kan worden gebruikt om het xml-pad te definiëren dat als een tabel moet worden geladen.<br><br>*Tabel is (page 182)* |
| qvd | De indeling *qvd* is de eigen indeling voor QVD-bestanden die zijn geëxporteerd uit een Qlik Sense-app. |
| qvx | *qvx* is a bestands-/stream-indeling voor snelle uitvoer naar Qlik Sense. |

| Waardetype | Beschrijving |
|---|---|
| parquet | Apache Parquet is een opslagindeling met kolommen, wat zeer efficiënt is voor het opslaan en doorzoeken van grote gegevensverzamelingen.<br><br>Met Parquet-bestanden die geneste gegevens bevatten, kunt u de tabel uit het Parquet-bestand specificeren die moet worden geladen met de specificatie **Table is**. Bijvoorbeeld: `LOAD * FROM [lib://DataFiles/company.parquet] (parquet, table is [company:salesrep.salesrep]);`.<br><br>*Tabel is (page 182)* |
| json | JSON (JavaScript Object Notation) is een taalonafhankelijke gegevensindeling.<br><br>Als de JSON meerdere matrices bevat, kan de specificatie **Tabel is** worden gebruikt om het pad naar de iterator voor de subtabel te definiëren.<br><br>*Tabel is (page 182)* |

### Delimiter is

Voor tabelbestanden met scheidingstekens kan een willekeurig scheidingsteken worden opgegeven via de specificatie **delimiter is**. Deze specificatie is alleen relevant voor .txt-bestanden met scheidingstekens.

**Syntaxis:**

```
delimiter is char
```

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| **char** | Hiermee wordt een enkel teken van de 127 ASCII-tekens gespecificeerd. |

Daarnaast kunnen de volgende waarden worden gebruikt:

Optionele waarden

| Value | Beschrijving |
|---|---|
| **'\t'** | geeft een tabteken weer, met of zonder aanhalingstekens. |
| **'\\'** | geeft een backslash (\) weer. |
| **'spaces'** | geeft alle combinaties van een of meer spaties weer. Niet-afdrukbare tekens met een ASCII-waarde onder de 32, met uitzondering van CR en LF, worden als spaties geïnterpreteerd. |

Als er niets is gespecificeerd, wordt **delimiter is ','** verondersteld.

**Voorbeeld:**

```
LOAD * from a.txt (utf8, txt, delimiter is ',' , embedded labels);
```

**Zie ook:**

p *Load (page 163)*

### No eof

Met de specificatie **no eof** geeft u aan dat het teken voor het einde van het bestand moet worden genegeerd bij het laden van **.txt**-bestanden met scheidingstekens.

**Syntaxis:**

```
no eof
```

Als de specificatie **no eof** wordt gebruikt, worden tekens met codepunt 26, die gewoonlijk het eind van het bestand aangeven, genegeerd en kunnen deze deel uitmaken van een veldwaarde.

Dit is alleen relevant voor tekstbestanden met scheidingstekens.

**Voorbeeld:**

```
LOAD * from a.txt (txt, utf8, embedded labels, delimiter is ' ', no eof);
```

**Zie ook:**

p *Load (page 163)*

### Labels

**Labels** is een bestandsspecificatie voor de opdracht **LOAD** waarmee wordt gedefinieerd waar in een bestand de veldnamen te vinden zijn.

**Syntaxis:**

```
embedded labels|explicit labels|no labels
```

De veldnamen worden in verschillende plaatsen in het bestand aangetroffen. Als de eerste record de veldnamen bevat, moet u **embedded labels** gebruiken. Als er geen veldnamen worden aangetroffen, moet u **no labels** gebruiken. In *dif*-bestanden wordt soms een afzonderlijke koptekstsectie met uitdrukkelijke veldnamen gebruikt. In dat geval moet u **explicit labels** gebruiken. Als er niets wordt opgegeven, wordt aangenomen dat **embedded labels** wordt gebruikt, ook voor *dif*-bestanden.

**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)
```

**Zie ook:**

p *Load (page 163)*

### Header is

Hiermee wordt de grootte van de koptekst in tabelbestanden opgegeven. U kunt een willekeurige koptekstlengte opgeven met behulp van de specificatie **header is**. Een koptekst is een tekstgedeelte dat niet in Qlik Sense wordt gebruikt.

**Syntaxis:**

```
header is n
```

```
header is line
```

```
header is n lines
```

De lengte van de koptekst kan worden opgegeven in bytes (**header is n**) of in regels (**header is line** of **header is n lines**). **n** moet een positief geheel getal zijn dat de lengte van de koptekst aangeeft. Als dit niet is gespecificeerd, wordt **header is 0** verondersteld. De specificatie **header is** is uitsluitend relevant voor tabelbestanden.

**Voorbeeld:**

Dit is een voorbeeld van een gegevensbrontabel die een tekstregel voor de koptekst bevat die niet als gegevens moet worden geïnterpreteerd in Qlik Sense.

```
*Header line
Col1,Col2
a,B
c,D
```

Bij gebruik van de specificatie **header is 1 lines** wordt de eerste regel niet geladen als gegevens. In het voorbeeld geeft de specificatie **embedded labels** aan dat Qlik Sense de eerste niet-uitgesloten regel moet interpreteren als regel met veldlabels.

```
LOAD Col1, Col2
FROM 'lib://files/header.txt'
(txt, embedded labels, delimiter is ',', msq, header is 1 lines);
```

Het resultaat is een tabel met twee velden: Col1 en Col2.

**Zie ook:**

p *Load (page 163)*

### Record is

Voor bestanden met een vaste recordlengte moet de recordlengte worden opgegeven met de specificatie **record is**.

**Syntaxis:**

```
Record is n
Record is line
Record is n lines
```

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| n | Geeft de recordlengte in bytes. |
| line | Geeft de recordlengte op als één regel. |
| n lines | Geeft de recordlengte in regels waarbij n een positief geheel getal is dat de recordlengte voorstelt. |

**Beperkingen:**

De specificatie **record is** is alleen relevant voor **fix**-bestanden.

**Zie ook:**

p *Load (page 163)*

### Tabel is

Voor Excel-, XML-, Parquet- of JSON-bestanden kunt u de tabel specificeren van waaruit u gegevens laadt in de tabelindelingsspecificatie.

**Syntaxis:**

```
Table is table name
```

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| table name | Specificeert de naam van de tabel. De waarde is afhankelijk van de tabelindeling:<br>• Excel: de werkbladnaam.<br>• XML: het pad dat het deel van de XML specificeert dat moet worden geladen.<br>• Parquet: het pad dat de tabel specificeert met de indeling *<node>.<node>.<node>*.<br>Gebruik **Table is** wanneer u een tabel specificeert binnen een geneste structuur. Bijvoorbeeld: u hebt Parquet-gegevens in het volgende schema:<br>`Schema:`<br>`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)`<br>U kunt Phone en de daarin geneste velden laden als een tabel met het argument `Table is [Schema:Phone.Item]`. Dit genereert het sleutelveld *%Key_Phone* met de tabel.<br>• JSON: In JSON-bestanden met matrices, het pad naar de iterator voor de subtabel.<br>U kunt een lading maken van alle tabellen door **alle tabellen** op te geven in plaats van **Tabel is**. Bijvoorbeeld:<br>`LOAD * FROM [lib://DataFiles/Sales.json] (json, all tables);`<br>Als uw gegevens meerdere matrices met identieke inhoud bevatten, gebruik dan het trefwoord **DISTINCT** in de load-instructies voor subtabellen. Bijvoorbeeld:<br>`LOAD DISTINCT * FROM [lib://DataFiles/Sales.json] (json, all tables);` |

### Voorbeeld: 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]);
```

### Voorbeeld: XML

```
LOAD
       city%Table,
       %Key_row_7FAC1F878EC01ECB
FROM [lib://AttachedFiles/cities.xml]
(XmlSimple, table is [root/row/country/city]);
```

### Voorbeeld: Parquet

Het bestand company.parquet bevat het volgende 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)
```

Het volgende zou de inhoud uit het bestand in tabellen laden. De eerste LOAD-instructie laadt de hoofdgroep. De tweede LOAD-instructie laadt de inhoud van de *salesrep*-groep als een tabel. De derde laadt de *headquarter*-groep als een tabel. De vierde laadt de *region*-groep als een tabel.

```
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]
```

### Voorbeeld: 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]);
```

### Beperkingen:

De specificatie **Table is** is alleen relevant voor Excel-, XML, Parquet- of JSON-bestanden.

## Quotes

**Quotes** is een bestandsspecificatie voor de opdracht **LOAD** waarmee wordt gedefinieerd of aanhalingstekens kunnen worden gebruikt en wat de prioriteit is tussen aanhalingstekens en scheidingstekens. Uitsluitend voor tekstbestanden.

**Syntaxis:**

```
no quotes
```

```
msq
```

Als de specificatie wordt weggelaten, kunnen standaard de aanhalingstekens " " of ' ' worden gebruikt, maar alleen als ze de eerste en laatste niet-blanco tekens in een veldwaarde zijn.

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| no quotes | Wordt gebruikt als aanhalingstekens niet worden geaccepteerd in een tekstbestand. |
| msq | Wordt gebruikt voor het opgeven van aanhalingstekens in moderne stijl, waardoor meerregelige inhoud in velden mogelijk wordt. Velden die tekens bevatten die het einde van een regel aangeven, moeten tussen dubbele aanhalingstekens worden geplaatst.<br><br>Eén beperking van de optie msq is dat enkele tekens die een dubbel aanhalingsteken aangeven ('') die als eerste of laatste teken voorkomen in veldinhoud worden geïnterpreteerd als het begin of einde van meerregelige inhoud. Dit kan tot onvoorspelbare resultaten leiden in de gegevensset die wordt geladen. In dat geval moet u standaard aanhalingstekens gebruiken door de specificatie weg te laten. |

### XML

Deze scriptspecificatie wordt gebruikt voor het laden van xml-bestanden. Geldige opties voor de**XML** specificatie worden vermeld in de syntaxis.

*U kunt geen DTD-bestanden laden in Qlik Sense.*

**Syntaxis:**

```
xmlsimple
```

**Zie ook:**

p *Load (page 163)*

### KML

Deze scriptspecificatie wordt gebruikt voor het laden van KML-bestanden voor een kaartvisualisatie.

**Syntaxis:**

```
kml
```

Het KML-bestand kan gebiedsgegevens (bijvoorbeeld landen of regio's) aangeven die worden weergegeven door polygonen, lijngegevens (bijvoorbeeld spoorwegen of wegen) of puntgegevens (bijvoorbeeld steden of plaatsen) die worden weergegeven door punten in de vorm [long, lat].

### URL is

De scriptspecificatie wordt tijdens het laden van een webbestand gebruikt voor het instellen van de URL van een gegevensverbinding hiervoor.

**Syntaxis:**

```
URL is string
```

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| string | Hiermee wordt de URL van het in te laden bestand gespecificeerd. Dit gaat voor de URL die in de gebruikte webbestandverbinding is ingesteld. |

**Beperkingen:**

De specificatie **URL is** is alleen relevant voor webbestanden. U moet een bestaande gegevensverbinding voor een webbestand gebruiken.

**Zie ook:**

p *Load (page 163)*

### userAgent is

Deze scriptspecificatie wordt gebruikt om de browsergebruikersagent in te stellen als er een webbestand wordt ingeladen.

**Syntaxis:**

```
userAgent is string
```

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| string | Hiermee wordt de tekenreeks van de browsergebruikersagent weergegeven. Hiermee negeert u de standaardbrowsergebruikersagent "Mozilla/5.0". |

**Beperkingen:**

De specificatie **userAgent is** is alleen relevant voor webbestanden.

**Zie ook:**

p *Load (page 163)*

## Let

De opdracht **let** is een aanvulling op de opdracht **set** voor het definiëren van scriptvariabelen. In tegenstelling tot de opdracht **set** evalueert de opdracht **let** de uitdrukking rechts van '=' tijdens runtime van het script voordat deze wordt toegewezen aan de variabele.

**Syntaxis:**

```
Let variablename=expression
```

Voorbeelden en resultaten:

| Voorbeeld | Resultaat |
|---|---|
| `Set x=3+4;`<br><br>`Let y=3+4;`<br><br>`z=$(y)+1;` | `$(x)` wordt geëvalueerd als ' 3+4 '<br><br>`$(y)` wordt geëvalueerd als ' 7 '<br><br>`$(z)` wordt geëvalueerd als ' 8 '<br><br>Let op het verschil tussen de **Set**- en **Let**-instructies. De **Set**-instructie wijst de reeks '3+4' toe aan de variabele, terwijl de **Let**-instructie de reeks evalueert en 7 toewijst aan de variabele. |
| Let T=now( ); | `$(T)` krijgt de waarde van de huidige tijd. |

## Loosen Table

Een of meer interne gegevenstabellen van Qlik Sense kunt u tijdens de uitvoering van het script expliciet declareren als losjes gekoppeld met behulp van de opdracht **Loosen Table**. Als een tabel losjes is gekoppeld, zijn alle associaties tussen veldwaarden in de tabel verwijderd. Een vergelijkbaar effect kan worden verkregen door elk veld van de losjes gekoppelde tabel als onafhankelijke, niet-verbonden tabel te laden. Losjes gekoppeld kan handig zijn tijdens het testen om tijdelijk verschillende onderdelen van de gegevensstructuur te isoleren. Een losjes gekoppelde tabel kan worden geïdentificeerd in de tabelweergave aan de hand van de stippellijnen. Het gebruik van een of meer **Loosen Table**-opdrachten in het script zorgt dat Qlik Sense eventuele instellingen negeert voor losjes gekoppelde tabellen voordat het script wordt uitgevoerd.

**Syntaxis:**

```
Loosen Tabletablename [ , tablename2 ...]
```

```
Loosen Tablestablename [ , tablename2 ...]
```

U kunt de syntaxis **Loosen Table** of **Loosen Tables** gebruiken.

*Als er in Qlik Sense circulaire referenties in de gegevensstructuur worden gevonden die niet kunnen worden verbroken door tabellen die interactief of expliciet in het script tot losjes gekoppeld zijn benoemd, worden een of meer extra tabellen gedwongen losjes gekoppeld, totdat er geen circulaire referenties meer bestaan. Als dit gebeurt, wordt er een dialoogvenster weergegeven met een* ***waarschuwing dat er een circulaire referentie*** *is gevonden.*

**Voorbeeld:**

```
Tab1:

SELECT * from Trans;

Loosen Table Tab1;
```

## Map

De opdracht **map ... using** wordt gebruikt voor de toewijzing van een bepaalde veldwaarde of uitdrukking aan de waarden van een specifieke toewijzingstabel. De toewijzingstabel wordt gemaakt met behulp van de opdracht **Mapping**.

**Syntaxis:**

```
Map fieldlist Using  mapname
```

De automatische toewijzing vindt plaats voor velden die zijn geladen na de opdracht **Map … Using** tot aan het einde van het script of totdat een opdracht **Unmap** wordt aangetroffen.

De toewijzing gebeurt als laatste in de keten van gebeurtenissen die tot de opslag van het veld in de interne tabel in Qlik Sense leiden. Toewijzing vindt dus niet steeds plaats wanneer een veldnaam wordt aangetroffen als onderdeel van een uitdrukking, maar wanneer de waarde wordt opgeslagen onder de veldnaam in de interne tabel. Als op het niveau van de uitdrukking toewijzing vereist is, moet echter de functie **Applymap()** worden gebruikt.

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| *fieldlist* | Een door komma's gescheiden lijst met de velden die vanaf dit punt in het script moeten worden toegewezen. Met een * als veldlijst worden alle velden aangeduid. De jokertekens * en ? zijn toegestaan in de veldnamen. Het gebruik van aanhalingstekens rond veldnamen kan nodig zijn bij het gebruik van jokertekens. |
| *mapname* | De naam van een toewijzingstabel die eerder is gelezen in een opdracht **mapping load** of **mapping select**. |

Voorbeelden en resultaten:

| Voorbeeld | Resultaat |
|---|---|
| Map Country Using Cmap; | Maakt het toewijzen van het veld Country mogelijk met de toewijzing Cmap. |
| Map A, B, C Using X; | Maakt het toewijzen van de velden A, B en C mogelijk met de toewijzing X. |
| Map * Using GenMap; | Maakt het toewijzen van alle velden mogelijk met GenMap. |

## NullAsNull

De opdracht **NullAsNull** schakelt de conversie van NULL-waarden naar tekenreekswaarden uit die eerder is ingesteld via een **NullAsValue**-opdracht.

**Syntaxis:**

```
NullAsNull *fieldlist
```

De opdracht **NullAsValue** fungeert als schakeloptie en kan verschillende keren worden in- of uitgeschakeld in het script, met behulp van de opdracht **NullAsValue** of **NullAsNull**.

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| *fieldlist | Een door komma's gescheiden lijst met velden waarvoor **NullAsNull** moet worden ingeschakeld. Met een * als veldlijst worden alle velden aangeduid. De jokertekens * en ? zijn toegestaan in de veldnamen. Het gebruik van aanhalingstekens rond veldnamen kan nodig zijn bij het gebruik van jokertekens. |

**Voorbeeld:**

```
NullAsNull A,B;
LOAD A,B from x.csv;
```

## NullAsValue

De opdracht **NullAsValue** legt vast voor welke velden NULL moet worden omgezet in een waarde.

**Syntaxis:**

```
NullAsValue *fieldlist
```

In Qlik Sense wordt standaard aangenomen dat NULL-waarden ontbrekende of ongedefinieerde entiteiten zijn. In sommige databasecontexten moeten NULL-waarden echter als speciale waarden worden beschouwd in plaats van ontbrekende waarden. Het feit dat NULL-waarden gewoonlijk niet mogen worden gekoppeld aan andere NULL-waarden, is te omzeilen door middel van de opdracht **NullAsValue**.

De opdracht **NullAsValue** fungeert als schakeloptie en werkt bij opeenvolgende laadopdrachten. Deze opdracht kan met behulp van de opdracht **NullAsNull** worden uitgeschakeld.

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| *fieldlist | Een door komma's gescheiden lijst met velden waarvoor **NullAsValue** moet worden ingeschakeld. Met een * als veldlijst worden alle velden aangeduid. De jokertekens * en ? zijn toegestaan in de veldnamen. Het gebruik van aanhalingstekens rond veldnamen kan nodig zijn bij het gebruik van jokertekens. |

**Voorbeeld:**

```
NullAsValue A,B;
Set NullValue = 'NULL';
LOAD A,B from x.csv;
```

## Qualify

De opdracht **Qualify** wordt gebruikt om de kwalificatie van veldnamen in te schakelen. Dit houdt in dat veldnamen de tabelnaam als prefix krijgen.

**Syntaxis:**

```
Qualify *fieldlist
```

De automatische join van velden met dezelfde naam in verschillende tabellen kan worden voorkomen met behulp van de opdracht **qualify**. Hiermee wordt de veldnaam gekwalificeerd met de tabelnaam ervan. Bij kwalificatie worden de veldnamen gewijzigd wanneer ze in een tabel worden aangetroffen. De nieuwe naam heeft de vorm *tablename.fieldname*. *Tablename* is gelijk aan het label van de huidige tabel of, als er geen label is, aan de naam die verschijnt achter **from** in **LOAD**- en **SELECT**-opdrachten.

De kwalificatie vindt plaats voor alle velden die worden geladen na de opdracht **qualify**.

Kwalificatie is standaard altijd uitgeschakeld aan het begin van een script. Kwalificatie van een specifiek veld kan op elk moment worden geactiveerd met behulp van de instructie **qualify**. Kwalificatie kan op elk moment worden uitgeschakeld met behulp van de instructie **Unqualify**.

*De opdracht **qualify** moet niet worden gebruikt in combinatie met gedeeltelijk opnieuw laden.*

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| *fieldlist | Een door komma's gescheiden lijst met velden waarvoor kwalificatie moet worden ingeschakeld. Met een * als veldlijst worden alle velden aangeduid. De jokertekens * en ? zijn toegestaan in de veldnamen. Het gebruik van aanhalingstekens rond veldnamen kan nodig zijn bij het gebruik van jokertekens. |

**Example 1:**

```
Qualify B;

LOAD A,B from x.csv;

LOAD A,B from y.csv;
```

De twee tabellen **x.csv** en **y.csv** zijn alleen gekoppeld via **A**. Dit leidt tot drie velden: , , . A, x.B, y.B.

**Example 2:**

In een onbekende database is het vaak handig om in het begin alleen een enkel veld of een paar velden te koppelen, zoals in het volgende voorbeeld:

```
qualify *;

unqualify TransID;

SQL SELECT * from tab1;

SQL SELECT * from tab2;

SQL SELECT * from tab3;
```

Alleen **TransID** zal worden gebruikt voor associaties tussen de tabellen *tab1*, *tab2* en *tab3*.

## Rem

Met de opdracht **rem** kunt u opmerkingen of commentaar aan het script toevoegen, of scriptopdrachten tijdelijk deactiveren zonder ze te verwijderen.

**Syntaxis:**

```
Rem string
```

Alles wat tussen **rem** en de volgende puntkomma **;** staat, wordt als opmerking beschouwd.

Er zijn twee alternatieve manieren om opmerkingen in uw script op te nemen:

1. U kunt een opmerking op elke willekeurige plaats in het script opnemen - behalve tussen twee aanhalingstekens - door het betreffende deel tussen **/*** en ***/** te plaatsen.
2. Als u **//** in het script typt, wordt alle tekst rechts daarvan op dezelfde rij beschouwd als een opmerking. (Let op de uitzondering //: die als deel van een internetadres kan worden gebruikt.)

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| string | Een willekeurige tekst. |

**Voorbeeld:**

```
Rem ** This is a comment **;
/* This is also a comment */
// This is a comment as well
```

# Rename

Het sleutelwoord **Rename** voor scripts kan worden gebruikt om de naam te wijzigen van tabellen of velden die al zijn geladen.

## Rename field

Met deze scriptfunctie wordt de naam van een of meer bestaande Qlik Sense-velden gewijzigd nadat deze zijn geladen.

*Het wordt niet aanbevolen om een variabele dezelfde naam te geven als een veld of functie in Qlik Sense.*

U kunt de syntaxis **rename field** of **rename fields** gebruiken.

**Syntaxis:**

```
Rename Field (using mapname | oldname to newname{ , oldname to newname })
```

```
Rename Fields (using mapname | oldname to newname{ , oldname to newname })
```

**Argumenten:**

| Argument | Beschrijving |
|---|---|
| mapname | De naam van de eerder geladen toewijzingstabel met een of meer oude en nieuwe veldnaamparen. |
| oldname | De oude veldnaam. |
| newname | De nieuwe veldnaam. |

**Beperkingen:**

U kunt de namen van twee velden niet zo wijzigen dat deze dezelfde naam hebben.

**Example 1:**

```
Rename Field XAZ0007 to Sales;
```

**Example 2:**

```
FieldMap:

Mapping SQL SELECT oldnames, newnames from datadictionary;

Rename Fields using FieldMap;
```

## Rename table

Met deze scriptfunctie wordt de naam van een of meer bestaande interne Qlik Sense-tabellen gewijzigd nadat deze zijn geladen.

U kunt de syntaxis **rename table** of **rename tables** gebruiken.

**Syntaxis:**

```
Rename Table (using mapname | oldname to newname{ , oldname to newname })
Rename Tables (using mapname | oldname to newname{ , oldname to newname })
```

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| mapname | De naam van de eerder geladen toewijzingstabel met een of meer oude en nieuwe tabelnaamparen. |
| oldname | De oude tabelnaam. |
| newname | De nieuwe tabelnaam. |

**Beperkingen:**

Twee tabellen met verschillende namen kunnen niet dezelfde naam krijgen. Het script genereert een fout als u probeert de naam van een tabel te wijzigen zodat deze dezelfde naam krijgt als een bestaande tabel.

**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

De **Search**-opdracht wordt gebruik voor het opnemen of uitsluiten van velden bij slimme zoekacties.

**Syntaxis:**

```
Search Include *fieldlist
Search Exclude *fieldlist
```

U kunt verschillende Search-opdrachten gebruiken om uw selectie van op te nemen velden te verfijnen. De opdrachten worden geëvalueerd van boven naar beneden.

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| *fieldlist | Een door komma's gescheiden lijst van op te nemen of uit te sluiten velden in zoekopdrachten in slimme zoekacties. Met een * als veldlijst worden alle velden aangeduid. De jokertekens * en ? zijn toegestaan in de veldnamen. Het gebruik van aanhalingstekens rond veldnamen kan nodig zijn bij het gebruik van jokertekens. |

**Voorbeeld:**

Zoekvoorbeelden

| Opdracht | Beschrijving |
|---|---|
| `Search Include *;` | Alle velden opnemen in zoekopdrachten in slimme zoekacties. |
| `Search Exclude [*ID];` | Alle velden die eindigen op ID uitsluiten van zoekopdrachten in slimme zoekacties. |
| `Search Exclude '*ID';` | Alle velden die eindigen op ID uitsluiten van zoekopdrachten in slimme zoekacties. |
| `Search Include ProductID;` | Het veld ProductID opnemen in zoekopdrachten in slimme zoekacties. |

Het gecombineerde resultaat van deze drie opdrachten in deze volgorde is dat alle velden die eindigen op ID behalve ProductID worden uitgesloten van zoekopdrachten in slimme zoekacties.

## Section

Met de opdracht **section** kan worden aangegeven of de daarop volgende **LOAD**- en **SELECT**-opdrachten moeten worden beschouwd als gegevens of als een definitie van de toegangsrechten.

**Syntaxis:**

```
Section (access | application)
```

Als er niets is gespecificeerd, wordt **section application** verondersteld. De definitie van **section** is geldig totdat er een nieuwe **section**-opdracht wordt gemaakt.

**Voorbeeld:**

```
Section access;

Section application;
```

## Select

U selecteert velden uit een ODBC-gegevensbron of van een OLE DB-provider met de standaard **SELECT**-opdrachten van SQL. Of de **SELECT**-opdrachten worden geaccepteerd, hangt af van het gebruikte ODBC-stuurprogramma of de gebruikte OLE DB-provider. Voor het gebruik van de **SELECT** -opdracht is een open gegevensverbinding naar de bron vereist.

**Syntaxis:**

```
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 ]
```

Verder kunnen diverse **SELECT**-instructies soms tot één instructie worden samengevoegd door middel van de operator **union**:

```
selectstatement Union selectstatement
```

Omdat de **SELECT**-opdracht wordt geïnterpreteerd door het ODBC-stuurprogramma of de OLE DB-provider, zijn er soms afwijkingen van de algemene SQL-syntaxis, afhankelijk van de mogelijkheden van de ODBC-stuurprogramma's of OLE DB-provider, bijvoorbeeld:

- **as** is niet altijd toegestaan, *aliasname* moet dan direct achter *fieldname* worden geplaatst.
- Soms is **as** juist verplicht bij het gebruik van *aliasname*.
- **distinct**, **as**, **where**, **group by**, **order by** of **union** worden soms niet ondersteund.
- Het ODBC-stuurprogramma accepteert niet altijd alle verschillende soorten aanhalingstekens.

*Deze beschrijving van de SQL-opdracht **SELECT** is niet volledig! **SELECT**-opdrachten kunnen bijvoorbeeld worden genest of worden samengevoegd tot een enkele **SELECT**-opdracht, en het aantal toegestane functies in een uitdrukking is soms heel groot.*

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| distinct | **distinct** wordt gebruikt om waarden die dubbel voorkomen in de geselecteerde velden, maar één keer te laden. |
| distinctrow | **distinctrow** wordt gebruikt om records die dubbel voorkomen in de brontabel, maar één keer te laden. |
| fieldlist | **`fieldlist ::= (*\| field ) {, field }`**<br>Een lijst van de te selecteren velden. Gebruik een sterretje * als veldlijst voor alle velden in de tabel.<br>**`fieldlist ::= field {, field }`**<br>Een lijst van een of meer velden, gescheiden door komma's.<br>**`field ::= ( fieldref\| expression ) [as aliasname ]`**<br>De uitdrukking kan bijvoorbeeld een numerieke of tekenreeksfunctie zijn, gebaseerd op een of meer andere velden. Enkele van de meestal geaccepteerde operatoren en functies zijn: +, -, *, /, & (aaneengeschakelde tekenreeks), sum(fieldname), count (fieldname), avg(fieldname)(average), month(fieldname), etc. Zie de documentatie van het ODBC-stuurprogramma voor meer informatie.<br>**`fieldref ::= [ tablename. ] fieldname`**<br>**tablename**en **fieldname** zijn teksttekenreeksen voor de tabelnaam en veldnaam. Ze moeten tussen rechte dubbele aanhalingstekens worden geplaatst als ze bijvoorbeeld spaties bevatten.<br>De **as**-clausule wordt gebruikt om een nieuwe naam toe te wijzen aan het veld. |
| from | **`tablelist ::= table {, table }`**<br>De lijst van tabellen waaruit de velden worden geselecteerd.<br>**`table ::= tablename [ [as ] aliasname ]`**<br>**tablename** mag tussen aanhalingstekens worden geplaatst, maar dat hoeft niet. |

| Argument | Beschrijving |
|---|---|
| where | **where** is een clausule die wordt gebruikt om aan te geven of een record wel of niet in de selectie moet worden opgenomen.<br>**criterion** is een logische uitdrukking, die zeer complex kan zijn. Enkele geaccepteerde operatoren zijn: numerieke operatoren en functies, **=**, **<>** of **#(**niet gelijk), **>**, **>=**, **<**, **<=**, **and**, **or**, **not**, **exists**, **some**, **all**, **in** en ook nieuwe **SELECT**-instructies. Zie de documentatie van het ODBC-stuurprogramma of de OLE DB-provider voor meer informatie. |
| group by | Met **group by**-clausule worden diverse records geaggregeerd (gegroepeerd) tot een enkele record. Binnen een groep moeten alle records voor een bepaald veld dezelfde waarde hebben, anders kan het veld alleen binnen een uitdrukking worden gebruikt, bijvoorbeeld als som of gemiddelde. De op een of meer velden gebaseerde uitdrukking wordt gedefinieerd in de uitdrukking van het veldsymbool. |
| having | Met **having** worden groepen gekwalificeerd, net zoals bij de **where**-clausule records worden gekwalificeerd. |
| order by | **order by** is een clausule die de sorteervolgorde aangeeft van de tabel die door de **SELECT**-opdracht ontstaat. |
| join | **join** verenigt verschillende tabellen tot één tabel. Veld- en tabelnamen moeten tussen aanhalingstekens worden geplaatst als ze spaties of letters uit de nationale tekenset bevatten. Als het script automatisch door Qlik Sense wordt gegenereerd, worden de aanhalingstekens gebruikt die worden geaccepteerd door het ODBC-stuurprogramma dat of de OLE DB-provider die is opgegeven in de gegevensbrondefinitie van de opdracht **Connect**. |

**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

Met de opdracht **set** legt u scriptvariabelen vast. Deze kunnen worden gebruikt om tekenreeksen, paden, stations en dergelijke te vervangen.

**Syntaxis:**

```
Set variablename=string
```

**Example 1:**

```
Set FileToUse=Data1.csv;
```

**Example 2:**

```
Set Constant="My string";
```

**Example 3:**

```
Set BudgetYear=2012;
```

## Sleep

Met de opdracht **sleep** onderbreekt u de uitvoering van een script voor een bepaalde tijd.

**Syntaxis:**

```
Sleep n
```

**Argumenten:**

| Argument | Beschrijving |
|---|---|
| n | Wordt opgegeven in milliseconden, waarbij *n* een positief geheel getal is dat niet groter is dan 3600000 (1 uur). De waarde mag ook een uitdrukking zijn. |

**Example 1:**

```
Sleep 10000;
```

**Example 2:**

```
Sleep t*1000;
```

## SQL

Met de opdracht **SQL** kunt u via een SQL- of ODBC-verbinding een willekeurige OLE DB-opdracht verzenden.

**Syntaxis:**

```
SQL sql_command
```

Als u SQL-opdrachten verzendt waarmee de database wordt bijgewerkt, wordt er een foutcode geretourneerd als Qlik Sense de ODBC-verbinding in de modus alleen-lezen heeft geopend.

De syntaxis:

```
SQL SELECT * from tab1;
```
is toegestaan en wordt voor de consistentie als voorkeurssyntaxis voor **SELECT** aangeduid. Het prefix SQL voor de **SELECT**-opdracht blijft echter optioneel.

**Argumenten:**

| Argument | Beschrijving |
|---|---|
| *sql_command* | Een geldige SQL-opdracht. |

**Example 1:**

```
SQL leave;
```

**Example 2:**

```
SQL Execute <storedProc>;
```

## SQLColumns

Met de opdracht **sqlcolumns** wordt een set velden geretourneerd waarin de kolommen worden beschreven van een ODBC- of OLE DB-gegevensbron waarmee via **connect** verbinding is gemaakt.

**Syntaxis:**

```
SQLcolumns
```

Als u de velden combineert met de velden die worden verkregen met de opdrachten **sqltables** en **sqltypes**, krijgt u een goed overzicht van een bepaalde database. De twaalf standaardvelden zijn:

TABLE_QUALIFIER

TABLE_OWNER

TABLE_NAME

COLUMN_NAME

DATA_TYPE

TYPE_NAME

PRECISION

LENGTH

SCALE

RADIX

NULLABLE

REMARKS

In een ODBC-referentiehandleiding vindt u een gedetailleerde omschrijving van deze velden.

**Voorbeeld:**

```
Connect to 'MS Access 7.0 Database; DBQ=C:\Course3\DataSrc\QWT.mbd';
SQLcolumns;
```

*Sommige ODBC-stuurprogramma's ondersteunen deze opdracht mogelijk niet. Sommige ODBC-stuurprogramma's produceren mogelijk extra velden.*

## SQLTables

Met de opdracht **sqltables** wordt een set velden geretourneerd waarin de tabellen worden beschreven van een ODBC- of OLE DB-gegevensbron waarmee via **connect** verbinding is gemaakt.

**Syntaxis:**

```
SQLTables
```

Als u de velden combineert met de velden die worden verkregen met de opdrachten **sqlcolumns** en **sqltypes**, krijgt u een goed overzicht van een bepaalde database. De vijf standaardvelden zijn:

TABLE_QUALIFIER

TABLE_OWNER

TABLE_NAME

TABLE_TYPE

REMARKS

In een ODBC-referentiehandleiding vindt u een gedetailleerde omschrijving van deze velden.

**Voorbeeld:**

```
Connect to 'MS Access 7.0 Database; DBQ=C:\Course3\DataSrc\QWT.mbd';
SQLTables;
```

*Sommige ODBC-stuurprogramma's ondersteunen deze opdracht mogelijk niet. Sommige ODBC-stuurprogramma's produceren mogelijk extra velden.*

## SQLTypes

Met de opdracht **sqltypes** wordt een set velden geretourneerd waarin de typen worden beschreven van een ODBC- of OLE DB-gegevensbron waarmee via **connect** verbinding is gemaakt.

**Syntaxis:**

```
SQLTypes
```

Als u de velden combineert met de velden die worden verkregen met de opdrachten **sqlcolumns** en **sqltables**, krijgt u een goed overzicht van een bepaalde database. De vijftien standaardvelden zijn:

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

In een ODBC-referentiehandleiding vindt u een gedetailleerde omschrijving van deze velden.

**Voorbeeld:**

```
Connect to 'MS Access 7.0 Database; DBQ=C:\Course3\DataSrc\QWT.mbd';
SQLTypes;
```

*Sommige ODBC-stuurprogramma's ondersteunen deze opdracht mogelijk niet. Sommige ODBC-stuurprogramma's produceren mogelijk extra velden.*

## Star

Met de opdracht **star** kunt u een tekenreeks opgeven die de set van alle waarden van een veld in een database vertegenwoordigt. De opdracht beïnvloedt de **LOAD**- en **SELECT**-opdrachten die volgen.

**Syntaxis:**

```
Star is[ string ]
```

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| string | Een willekeurige tekst. Houd er rekening mee dat de tekenreeks tussen aanhalingstekens moet worden geplaatst als deze spaties bevat.<br><br>Als er niets is aangegeven, wordt **star is;** verondersteld, d.w.z. dat er geen sterretje beschikbaar is tenzij er expliciet een is opgegeven. Deze definitie is geldig totdat er een nieuwe **star**-opdracht wordt gemaakt. |

Het wordt niet aangeraden om de instructie **Star is** te gebruiken in het gegevensdeel van het script (onder **Section Application**) als er gebruik wordt gemaakt van sectietoegang. Het sterretje wordt echter volledig ondersteund voor de beveiligde velden in het gedeelte **Section Access** van het script. In dit geval hoeft u de expliciete instructie **Star is** niet te gebruiken, aangezien deze altijd impliciet is in sectietoegang.

### Beperkingen

- U kunt het sterretje niet gebruiken in sleutelvelden, oftewel velden waarmee tabellen worden gekoppeld.
- U kunt het sterretje niet gebruiken in velden die worden beïnvloed door de instructie **Unqualify**, aangezien dit mogelijk van invloed is op velden waarmee tabellen worden gekoppeld.
- U kunt het sterretje niet gebruiken in niet-logische tabellen, zoals tabellen voor het laden van informatie of voor het laden van toewijzingen.
- Wanneer u het sterretje gebruikt in een beperkend veld (een veld met een koppeling naar de gegevens) in sectietoegang, stelt het de waarden voor die in dit veld voorkomen in de sectietoegang. Het sterretje stelt geen andere waarden voor die mogelijk voorkomen in de gegevens maar niet worden vermeld in de sectietoegang.
- U kunt het sterretje niet gebruiken in velden die worden beïnvloed door enige vorm van gegevensreductie buiten het gebied **Section Access**.

### Voorbeeld

Het onderstaande voorbeeld is een samenvatting van een load-script voor gegevens met sectietoegang.

```
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

];
```

Het volgende is van toepassing:

- Het symbool *Star* is *.
- De gebruiker *ADMIN* ziet alle velden. Er wordt niets weggelaten.
- De gebruiker *USER1* kan het veld *SALES* niet bekijken.
- De gebruiker *USER2* kan het veld *WAREHOUSE* niet bekijken.
- De gebruiker *USER3* kan het veld *EMPLOYEES* niet bekijken.
- De gebruiker *USER4* wordt tweemaal toegevoegd aan de oplossing om twee velden voor deze gebruiker WEG TE LATEN, namelijk *SALES* en *WAREHOUSE*.
- *USER5* heeft een '*' wat betekent dat alle getoonde velden in OMIT niet beschikbaar zijn. Gebruiker *USER5* kan de velden *SALES*, *WAREHOUSE* en *EMPLOYEES* dus niet zien, maar deze gebruiker kan wel het veld *ORDERS* zien.

## Store

De **Store**-instructie genereert een QVD-, Parquet-, CSV- of TXT-bestand.

**Syntaxis:**

```
Store [ fieldlist from] table into filename [ format-spec ];
```

De opdracht maakt een expliciet genoemd QVD, Parquet of tekstbestand aan.

De opdracht kan alleen velden exporteren uit één gegevenstabel, tenzij u opslaat in Parquet. Als velden uit diverse tabellen moeten worden geëxporteerd in een QVD-, CSV- of TXT-bestand, moet eerst een expliciete join in het script worden opgegeven om de te exporteren gegevenstabel te maken. U kunt meerdere tabellen in één Parquet opslaan door de gegevens in de Parquet-bestanden te nesten.

De tekstwaarden worden in de indeling BOM naar het CSV-bestand in UTF-8geëxporteerd. U kunt een scheidingsteken opgeven, zie **LOAD**. De **store**-opdracht voor een CSV-bestand ondersteunt geen BIFF-export.

> *In sommige gevallen met gegevens die niet goed gevormd zijn, worden velden tussen dubbele aanhalingstekens geplaatst om er zeker van te zijn dat de gegevens correct worden geïnterpreteerd. Dit kan bijvoorbeeld gebeuren als het veld tekens bevat zoals aanhalingstekens, komma's, spaties of regeleinden.*

**Argumenten:**

Opdrachtargumenten opslaan

| Argument | Beschrijving |
|---|---|
| *fieldlist::= ( * \| field) { , field } )* | Een lijst van de te selecteren velden. Met een * als veldlijst worden alle velden aangeduid.<br><br>*field::= fieldname [***as** *aliasname ]*<br><br>*fieldname* is een tekst die identiek is aan een veldnaam in *table*. (Houd er rekening mee dat de veldnaam tussen rechte dubbele aanhalingstekens of vierkante haken moet staan als deze spaties of andere niet-standaardtekens bevat.)<br><br>*aliasname* is een alternatieve naam voor het veld, te gebruiken in het resulterende QVD- of CSV-bestand. |
| *table* | Een scriptlabel representeert een reeds geladen tabel die als gegevensbron wordt gebruikt. |

| Argument | Beschrijving |
| --- | --- |
| *filename* | De naam van het doelbestand inclusief een geldig pad naar een bestaande mapgegevensverbinding.<br><br>**Voorbeeld:** ***'lib://Table Files/target.qvd'***<br><br>In de bestaande scriptmodus, worden tevens de volgende padindelingen ondersteund:<br><br>• absoluut<br><br>**Voorbeeld:** ***c:\data\sales.qvd***<br><br>• relatief ten opzichte van de werkmap van de Qlik Sense-app.<br><br>**Voorbeeld:** ***data\sales.qvd***<br><br>Als er geen pad is opgegeven, bewaart Qlik Sense het bestand in de map die wordt gespecificeerd in de **Directory**-instructie. Als er geen **Directory**-instructie is opgegeven, bewaart Qlik Sense het bestand in de werkmap, *C:\Users\{user}\Documents\Qlik\Sense\Apps*. |

| Argument | Beschrijving |
| --- | --- |
| *format-spec ::=*( ( **txt** \| **qvd** \| **parquet**), **compressie is** *codec*) | U kunt de indelingsspecificatie instellen voor een van deze bestandsindelingen. Als de opmaakspecificatie wordt weggelaten, wordt **qvd** verondersteld.<br>• **txt** voor CSV en TXT-bestanden.<br>• **qvd** voor QVD-bestanden.<br>• **parquet** voor Parquet-bestanden.<br>Als u **parquet** gebruikt, kunt u ook instellen welke compressiecodec met **compressie is** moet worden gebruikt. Als u de compressiecodec met **compressie is** niet opgeeft, wordt `snappy` gebruikt. De volgende compressie-instellingen zijn beschikbaar:<br>• `uncompressed`<br>• `snappy`<br>• `gzip`<br>• `lz4`<br>• `brotli`<br>• `zstd`<br>• `lz4_hadoop`<br>Voorbeeld:<br>`Store mytable into [lib://AttachedFiles/myfile.parquet] (parquet, compression is lz4);` |

**Voorbeelden:**

```
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';
```

## Opslaan in Parquet-bestanden

Parquet is een sterk getypeerde bestandsindeling, waarbij elk veld een afzonderlijk specifiek type gegevens bevat (zoals in32, double, tijdstempel of tekst). Qlik Sense slaat interne gegevens op als een zwak getypeerde dual, waarbij gegevens van verschillende bronnen in dezelfde velden gecombineerd kunnen worden. Omdat maar één deel van de dual opgeslagen kan worden in elk veld in Parquet, is het belangrijk om te weten wat elk veld bevat. Qlik Sense gebruikt standaard het veldtype om te bepalen hoe het veld moet worden opgeslagen. U moet het gegevenstype van uw velden opgeven voor het laden als u gegevens in Parquet-bestanden in een specifieke indeling opslaat. Als u probeert om gegevens op te slaan in niet-compatibele velden in een Parquet-bestand, zoals getallen in een tekstveld of tekst in een tijdstempelveld, krijgt u null-waarden.

Het is mogelijk om het standaardgedrag te wijzigen als u gegevens laadt die u in Parquet wilt opslaan. U kunt instellen dat uw gegevenstype wordt gewijzigd of u kunt gegevens taggen om specifieke kolomtypen in Parquet af te dwingen.

### Gegevens opmaken voor opslag in Parquet

U kunt Qlik Sense-opmaakfuncties gebruiken om uw gegevens te classificeren. **Text()**, **Num()**, **Interval()** of **Timestamp()** kunnen bijvoorbeeld gegevensopmaak afdwingen bij het opslaan van gegevens in Parquet. Qlik Sense kan gegevens opslaan in bijna 20 gegevenstypen, afhankelijk van de veldkenmerken en automatische veldtags. Ga voor meer informatie naar *Interpretatiefuncties (page 1447)*.

#### Voorbeeld: Gegevens opmaken met Num() en Text()

Het volgende voorbeeld toont hoe u gegevens voorbereidt die u wilt opslaan in Parquet. **Num()** wordt toegepast op het getalveld. **Text()** wordt toegepast op tekst en gemengd. In het geval van een gemengd veld, voorkomt **Text()** dat het wordt verwerkt als een getalveld in Parquet en worden de tekstwaarden gewijzigd in null-waarden.

```
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);
```

### Gegevens taggen voor opslag in Parquet

U tagt uw gegevens met `$parquet`-tags om specifieke kolomtypen af te dwingen bij het opslaan van gegevens in Parquet. Elk gegevenstype kan worden afgedwongen door de bijbehorende controletag toe te voegen. Stel dat u een veld als INT32 in Parquet wilt opslaan, tag dit dan met `$parquet-int32` in het load-script. Afhankelijk van het gegevenstype wordt de tekenreeks of de numerieke weergave van de dubbele gegevens opgeslagen.

U kunt de volgende Parquet-controletag gebruiken om velden te taggen voor opslag in Parquet-bestanden.

Parquet-controletags

| Controletag | Dual | Fysiek type | Logisch type | Geconverteerd type |
|---|---|---|---|---|
| `$parquet-boolean` | Getal | BOOLEAANSE WAARDE | NONE | NONE |
| `$parquet-int32` | Getal | INT32 | NONE | NONE |
| `$parquet-int64` | Getal | INT64 | NONE | NONE |
| `$parquet-float` | Getal | FLOAT | NONE | NONE |
| `$parquet-double` | Getal | DOUBLE | NONE | NONE |
| `$parquet-bytearray` | Tekenreeks | BYTE_ARRAY | NONE | UTF8 |
| `$parquet-bytearrayfix` | Getal | FIXED_LEN_BYTE_ARRAY | NONE | DECIMAL |
| `$parquet-decimal` | Getal | INT64 | DECIMAL | DECIMAL |
| `$parquet-date` | Getal | INT32 | DATUM | DATUM |
| `$parquet-time` | Getal | INT64 | TIJD | TIME_MICROS |
| `$parquet-timestamp` | Getal | INT64 | TIMESTAMP | TIMESTAMP_MICROS |
| `$parquet-string` | Tekenreeks | BYTE_ARRAY | TEKENREEKS | UTF8 |
| `$parquet-enum` | Tekenreeks | BYTE_ARRAY | ENUM | ENUM |
| `$parquet-interval` | Getal | FIXED_LEN_BYTE_ARRAY | INTERVAL | INTERVAL |
| `$parquet-json` | Tekenreeks | BYTE_ARRAY | JSON | JSON |
| `$parquet-bson` | Tekenreeks | BYTE_ARRAY | BSON | BSON |
| `$parquet-uuid` | Tekenreeks | FIXED_LEN_BYTE_ARRAY | UUID | NONE |

### Voorbeeld: Gegevens taggen voor opslag in Parquet

In dit voorbeeld worden twee tags gebruikt om de gegevens voor Parquet te definiëren. Het veld *num* is getagd met `$parquet-int32` om het te definiëren als getalveld dat in Parquet als INT32 wordt ingesteld.

```
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);
```

### Geneste gegevens opslaan in Parquet-bestanden

U kunt meerdere tabellen opslaan in een Parquet-bestand door ze te nesten in gestructureerde gegevens. **Store** ondersteunt gestructureerde knooppunten en lijstknooppunten in een sterrenschema. Afzonderlijke tabellen kunnen ook in de geneste modus worden opgeslagen met behulp van de specificatie **Scheidingsteken is**.

Wanneer u tabellen opslaat, kunt u de gewenste tabellen specificeren door ze te scheiden met komma's. Bijvoorbeeld: `STORE Table1, Table2, Table3 INTO [lib://<file location>/<file name>.parquet] (parquet);`. U kunt bepalen welke velden worden opgeslagen door een veldlijst te gebruiken in de **Store**-opdracht. Bijvoorbeeld `STORE Field1, Field2, FROM Table1, Table2 INTO [lib://<file location>/<file name>.parquet] (parquet);`. Alle velden in de veldenlijst moeten in één of meer van de vermelde tabellen staan. De eerste tabel in de **Store**-instructie wordt gebruikt als de feitentabel in het sterrenschema.

Veldnamen worden gebruikt om te beheren hoe groepen worden gemaakt en genest. Standaard worden veldnamen met een punt (`.`) opgesplitst in knooppunten. Het scheidingsteken kan worden gewijzigd door de systeemvariabele *FieldNameDelimiter* in te stellen of door de specificatie **Scheidingsteken is** te gebruiken. De specificatie overschrijft de systeemvariabele..

Veldnamen worden opgesplitst door het scheidingsteken en de onderdelen worden gebruikt om het schema met geneste groepen te maken. `STORE Field1, Field1.Field2, Field1.Field3, Field1.Field4 FROM Table1 INTO [nested.parquet] (parquet, delimiter is '.');` maakt bijvoorbeeld twee groepen (*Group1* en *Group2*) met *Fields1, Field2* en *Field3, Field4*.

> *Groepen en velden mogen niet dezelfde naam hebben in een knooppunt in het schema. `STORE Address, Address.Street INTO [nested.parquet] (parquet, delimiter is '.');` zal bijvoorbeeld mislukken omdat Address dubbelzinnig is en zowel een gegevensveld als een groep is.*

Wanneer geneste gegevens in Parquet worden opgeslagen, worden sleutels tussen tabellen omgezet in link-knooppunten in het schema. De tabellen worden omgezet in gestructureerde knooppunten in het schema. U kunt de standaardtransformatie overschrijven met veldnamen.

#### Voorbeeld: Geneste gegevens opslaan in Parquet-bestanden

```
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;
```

Het resulterende Parquet-bestand heeft het volgende 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)
```

### Beperkingen

Geneste gegevens opslaan in Parquet heeft de volgende beperkingen:

- Opslag ondersteunt geen kaartknooppunten.
- Tijdens het opslaan worden sleutelvelden die zijn gegeneerd vanuit het laden van geneste Parquet-bestanden niet toegevoegd.
- U kunt gegevens uit tabellen niet samen opslaan als ze niet zijn gekoppeld aan sleutelvelden.
- Het geneste bestand denormaliseert het gegevensmodel. Waarden waar niet naar wordt verwezen, worden niet opgeslagen en waarden waar meerdere keren naar wordt verwezen, worden gekopieerd.

### Beperkingen

- **STORE** load-script-instructies worden niet ondersteund door Connectoren van internetopslagprovider in Qlik Sense Enterprise on Windows en Qlik Sense Desktop.

## Table/Tables

De sleutelwoorden **Table** en **Tables** voor scripts worden gebruikt in opdrachten **Drop**, **Comment** en **Rename**, alsmede als indelingsspecificatie in **Load**-opdrachten.

## Tag

Met de scriptopdracht kunnen tags aan één of meer velden of tabellen worden toegewezen. Als wordt geprobeert om een tag toe te wijzen aan een veld of tabel die niet bestaat in de app, wordt de tag genegeerd. Als conflicterende exemplaren van een veld- of tagnaam zijn aangetroffen, wordt de laatste waarde gebruikt.

**Syntaxis:**

```
Tag [field|fields] fieldlist with tagname
```

```
Tag [field|fields] fieldlist using mapname
```

```
Tag table tablelist with tagname
```

Argumenten

| Argument | Beschrijving |
|---|---|
| fieldlist | Eén of meerdere velden die getagd moeten worden, in een door komma's gescheiden lijst. |
| mapname | De naam van een toewijzingstabel die eerder is geladen in een **mapping Load**- of **mapping Select**-opdracht. |
| tablelist | Een door komma's gescheiden lijst van de tabellen die getagd moeten worden. |
| tagname | De naam van de tag die moet worden toegepast op het veld. |

**Example 1:**

```
tagmap:

mapping LOAD * inline [

a,b

Alpha,MyTag

Num,MyTag

];
```

```
tag fields using tagmap;
```

**Example 2:**

```
tag field Alpha with 'MyTag2';
```

## Trace

Met de opdracht **trace** wordt eventueel een tekenreeks naar het venster **Voortgang scriptuitvoering** en in het scriptlogbestand geschreven. Dit is zeer handig bij het opsporen van fouten. Met behulp van $-uitbreidingen van variabelen die zijn berekend vóór de **trace**-opdracht, kunt u het bericht aanpassen.

**Syntaxis:**

```
Trace string
```

**Example 1:**

De volgende instructie kan worden gebruikt meteen na de Load-instructie waarmee de 'Main'-tabel wordt geladen.

```
Trace Main table loaded;
```

Hiermee wordt de tekst ‘Main table loaded’ weergegeven in het scriptuitvoeringsvenster en in het logbestand.

**Example 2:**

De volgende instructies kunnen worden gebruikt meteen na de Load-instructie waarmee de 'Main'-tabel wordt geladen.

```
Let MyMessage = NoOfRows('Main') & ' rows in Main table';

Trace $(MyMessage);
```

Hiermee wordt in het scriptuitvoeringsvenster en in het logbestand een tekst weergegeven die het aantal rijen aangeeft, bijvoorbeeld ‘265,391 rows in Main table’ .

## Unmap

Met de opdracht **Unmap** wordt de toewijzing van veldwaarden die is vastgelegd door een eerdere opdracht **Map … Using**, uitgeschakeld voor velden die daarna worden geladen.

**Syntaxis:**

```
Unmap *fieldlist
```

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| *fieldlist | Een door komma's gescheiden lijst met velden die vanaf dit punt in het script niet meer moeten worden toegewezen. Met een * als veldlijst worden alle velden aangeduid. De jokertekens * en ? zijn toegestaan in de veldnamen. Het gebruik van aanhalingstekens rond veldnamen kan nodig zijn bij het gebruik van jokertekens. |

Voorbeelden en resultaten:

| Voorbeeld | Resultaat |
|---|---|
| Unmap Country; | De toewijzing van het veld Country wordt uitgeschakeld. |
| Unmap A, B, C; | De toewijzing van de velden A, B en C wordt uitgeschakeld. |
| Unmap * ; | De toewijzing van alle velden wordt uitgeschakeld. |

## Unqualify

Met de opdracht **Unqualify** wordt de kwalificatie van veldnamen die eerder was ingeschakeld met de opdracht **Qualify**, uitgeschakeld.

**Syntaxis:**

```
Unqualify *fieldlist
```

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| *fieldlist | Een door komma's gescheiden lijst met velden waarvoor kwalificatie moet worden ingeschakeld. Met een * als veldlijst worden alle velden aangeduid. De jokertekens * en ? zijn toegestaan in de veldnamen. Het gebruik van aanhalingstekens rond veldnamen kan nodig zijn bij het gebruik van jokertekens.<br><br>In de documentatie voor de opdracht **Qualify** vindt u meer informatie. |

**Example 1:**

In een onbekende database is het vaak handig om in het begin alleen een enkel veld of een paar velden te koppelen, zoals in het volgende voorbeeld:

```
qualify *;
unqualify TransID;
SQL SELECT * from tab1;
SQL SELECT * from tab2;
SQL SELECT * from tab3;
```

Eerst wordt de kwalificatie voor alle velden uitgeschakeld.

Vervolgens wordt de kwalificatie uitgeschakeld voor **TransID**.
Alleen **TransID** zal worden gebruikt voor associaties tussen de tabellen *tab1*, *tab2* en *tab3*. Alle andere velden worden gekwalificeerd met de tabelnaam.

## Untag

Met de scriptopdracht kunnen tags van één of meer velden of tabellen worden verwijderd. Als wordt geprobeert om een tag te verwijderen van een veld of tabel die niet bestaat in de app, wordt de tag niet verwijderd.

**Syntaxis:**

```
Untag [field|fields] fieldlist with tagname

Untag [field|fields] fieldlist using mapname

Untag table tablelist with tagname
```

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| fieldlist | Een of meerdere velden waarvan de extra info moet worden verwijderd, in een door komma's gescheiden lijst. |
| mapname | De naam van een toewijzingstabel die eerder is geladen in een opdracht mapping **LOAD** of mapping **SELECT**. |
| tablelist | Een door komma's gescheiden lijst van de tabellen die moeten worden ontdaan van extra info. |
| tagname | De naam van de tag met extra info die moet worden verwijderd uit het veld. |

**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 Werkdirectory

Als u verwijst naar een bestand in een scriptopdracht en het pad wordt weggelaten, zoekt Qlik Sense in deze volgorde naar het bestand:

1. De directory die is opgegeven door een opdracht **Directory** (alleen ondersteund in de bestaande scriptmodus).
2. Als er geen opdracht **Directory** is, zoekt Qlik Sense in de werkdirectory.

### Qlik Sense Desktop werkdirectory

In Qlik Sense Desktop is *C:\Users\{user}\Documents\Qlik\Sense\Apps* de werkdirectory.

### Qlik Sense werkdirectory

In een Qlik Sense-serverinstallatie wordt de werkdirectory opgegeven in Qlik Sense Repository Service. Standaard is dit *C:\ProgramData\Qlik\Sense\Apps*. Raadpleeg de Help van Qlik Beheerconsole voor meer informatie.

# 4 Werken met variabelen in de editor voor het laden van gegevens

Een variabele in Qlik Sense is een container waarin een statische waarde of een berekening wordt opgeslagen, bijvoorbeeld een numerieke of alfanumerieke waarde. Als u de variabele in de app gebruikt, wordt een eventuele wijziging die wordt aangebracht in de variabele overal toegepast waar de variabele wordt gebruikt. U kunt variabelen definiëren in het overzicht Variabelen of in het script met de editor voor het laden van gegevens. U kunt de waarde van variabelen instellen met de instructies **Let** of **Set** in het load-script voor gegevens.

*U kunt tevens werken met de Qlik Sense-variabelen uit het overzicht van variabelen bij de bewerking van een werkblad.*

## 4.1 Overzicht

Als het eerste teken van een variabelewaarde het isgelijkteken '=' is, probeert Qlik Sense de waarde als formule (Qlik Sense-uitdrukking) te interpreteren. Vervolgens wordt niet de formuletekst, maar het resultaat weergegeven of geretourneerd.

Bij het gebruik van de variabele wordt deze vervangen door de waarde ervan. Variabelen kunnen in het script worden gebruikt voor uitbreiding met een dollarteken en in verschillende besturingsopdrachten. Dat is zeer handig als dezelfde tekenreeks vaak in het script wordt herhaald, bijvoorbeeld een pad.

Bepaalde speciale systeemvariabelen worden aan het begin van de scriptuitvoering door Qlik Sense ingesteld, ongeacht hun eerdere waarden.

## 4.2 Een variabele definiëren

Variabelen bieden de mogelijkheid om statische waarden of het resultaat van een berekening op te slaan. Gebruik de volgende syntaxis bij het definiëren van een variabele:

```
set variablename = string
```

of

```
let variable = expression
```

De **Set**-instructie wordt gebruikt voor tekenreekstoewijzing. Het wijst de tekst rechts van het isgelijkteken toe aan de variabele. De **Let**-instructie evalueert een uitdrukking rechts van het isgelijkteken tijdens de runtime van het script en wijst het resultaat toe aan de uitdrukking van de variabele.

Variabelen zijn hoofdlettergevoelig.

*Het wordt niet aanbevolen om een variabele dezelfde naam te geven als een veld of functie in Qlik Sense.*

**Voorbeelden:**

`set x = 3 + 4; //` de variabele zal de tekenreeks '3 + 4' ophalen als de waarde.

`let x = 3 + 4; //` retourneert 7 als de waarde.

`set x = Today(); //` retourneert 'Today()' als de waarde.

`let x = Today(); //` retourneert de datum van vandaag als de waarde, bijvoorbeeld '9/27/2021'.

### Variabelen een naam geven

Aanbevolen wordt om te overwegen of u een gestandaardiseerde naamgevingsconventie wilt gebruiken voor de variabelen die u in een app maakt. U kunt er bijvoorbeeld voor zorgen dat alle namen van uw variabelen beginnen met *v*. Voorbeeld: *vUserText*. Dit zorgt ervoor dat variabelen snel worden herkend als variabelen en worden onderscheiden van metingen, velden en functies.

## 4.3 Een variabele verwijderen

Als u een variabele uit het script verwijdert en de gegevens opnieuw laadt, blijft de variabele in de app. Als u de variabele volledig uit de app wilt verwijderen, moet u de variabele ook uit het variabelendialoogvenster verwijderen.

## 4.4 De waarde van een variabele laden als een veldwaarde

Als u de waarde van een variabele wil laden als een veldwaarde in een **LOAD**-opdracht en het resultaat van de dollaruitbreiding is tekst in plaats van numeriek of een uitdrukking, moet u de uitgebreide variabele tussen enkele aanhalingstekens zetten.

**Voorbeeld:**

In dit voorbeeld wordt de systeemvariabele die de lijst met scriptfouten bevat in een tabel geladen. Zoals u ziet, zijn voor uitbreiding van in de clausule geen aanhalingstekens vereist, terwijl bij de uitbreiding van aanhalingstekens zijn vereist.ScriptErrorCountScriptErrorList

```
IF $(ScriptErrorCount) >= 1 THEN
   LOAD '$(ScriptErrorList)' AS Error AutoGenerate 1;
END IF
```

## 4.5 Berekeningen met variabelen

Er zijn diverse manieren om variabelen te gebruiken met berekende waarden in Qlik Sense, en het resultaat is afhankelijk van de wijze waarop u de variabele definieert en aanroept in een uitdrukking.

In dit voorbeeld laden we enkele inline-gegevens:

```
LOAD * INLINE [
    Dim, Sales
    A, 150
    A, 200
    B, 240
    B, 230
    C, 410
    C, 330
];
```

Eerst definiëren we twee variabelen:

```
Let vSales  = 'Sum(Sales)' ;
Let vSales2  = '=Sum(Sales)' ;
```

In de tweede variabele plaatsen we een isgelijkteken voor de uitdrukking. Hierdoor wordt de variabele berekend voordat de variabele wordt uitgebreid en de uitdrukking wordt geëvalueerd.

Als u de variabele vSales zonder toevoeging gebruikt, is het resultaat de tekenreeks Sum(Sales) en wordt er geen berekening uitgevoerd.

Als u een dollar-tekenuitbreiding toevoegt en $(vSales) in de uitdrukking aanroept, wordt de variabele uitgebreid en wordt de som van Sales weergegeven.

En tot slot als u $(vSales2) aanroept, wordt de variabele eerst berekend en daarna uitgebreid. Dit betekent dat het weergegeven resultaat de totale som van Sales is. Het verschil tussen =$(vSales) en =$(vSales2) metinguitdrukkingen wordt duidelijk uit de tabel met de resultaten:

Resultaten

| Dim | $(vSales) | $(vSales2) |
|---|---|---|
| A | 350 | 1560 |
| B | 470 | 1560 |
| C | 740 | 1560 |

Zoals u ziet, is het resultaat van $(vSales) de deelsom voor een dimensiewaarde, terwijl het resultaat van $(vSales2) de totale som is.

De volgende scriptvariabelen zijn beschikbaar:

- *Foutvariabelen (page 293)*
- *Variabelen voor getalinterpretatie (page 227)*
- *Systeemvariabelen (page 218)*
- *Variabelen voor afhandeling van waarden (page 225)*

## 4.6 Systeemvariabelen

Systeemvariabelen waarvan sommige door het systeem zijn gedefinieerd, geven informatie over het systeem en de Qlik Sense-app.

## Overzicht van systeemvariabelen

Sommige functies worden na het overzicht nader beschreven. Bij deze functies kunt u ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

### CreateSearchIndexOnReload

Deze variabele wordt gebruikt om aan te geven of zoekindexbestanden moeten worden gemaakt wanneer gegevens opnieuw worden geladen.

```
CreateSearchIndexOnReload
```

### Floppy

Retourneert de stationsletter van het eerst gevonden diskettestation. Dit is normaal gesproken *a:*. Dit is een door het systeem gedefinieerde variabele.

```
Floppy
```

> *Deze variabele wordt niet ondersteund in de standaardmodus.*

### CD

Retourneert de stationsletter van het eerst gevonden cd-rom-station. Als er geen cd-rom-station is gevonden, wordt *c:* geretourneerd. Dit is een door het systeem gedefinieerde variabele.

```
CD
```

> *Deze variabele wordt niet ondersteund in de standaardmodus.*

### HidePrefix

Alle veldnamen die beginnen met deze teksttekenreeks worden op dezelfde manier als de systeemvelden verborgen. Dit is een door de gebruiker gedefinieerde variabele.

```
HidePrefix
```

### HideSuffix

Alle veldnamen die eindigen met deze teksttekenreeks worden op dezelfde manier als de systeemvelden verborgen. Dit is een door de gebruiker gedefinieerde variabele.

```
HideSuffix
```

### Include

De variabele **Include/Must_Include** specificeert een bestand dat tekst bevat die in het script moet worden opgenomen en moet worden geëvalueerd als scriptcode. Deze wordt niet gebruikt om gegevens toe te voegen. U kunt delen van uw scriptcode opslaan in een apart tekstbestand en dit in meerdere apps hergebruiken. Dit is een door de gebruiker gedefinieerde variabele.

```
$(Include=filename)
```

```
$(Must_Include=filename)
```

### OpenUrlTimeout

Deze variabele bepaalt de time-out in seconden die Qlik Sense moet aanhouden bij het ophalen van gegevens uit URL-bronnen (bijvoorbeeld -pagina's). HTML pagina's). Als deze variabele niet is ingesteld, is de time-out ongeveer 20 minuten.

```
OpenUrlTimeout
```

### QvPath

Retourneert het pad naar het Qlik Sense-programma. Dit is een door het systeem gedefinieerde variabele.

```
QvPath
```

*Deze variabele wordt niet ondersteund in de standaardmodus.*

### QvRoot

Retourneert de hoofdmap van het Qlik Sense-programma. Dit is een door het systeem gedefinieerde variabele.

```
QvRoot
```

*Deze variabele wordt niet ondersteund in de standaardmodus.*

### QvWorkPath

Retourneert het pad naar de huidige Qlik Sense-app. Dit is een door het systeem gedefinieerde variabele.

```
QvWorkPath
```

*Deze variabele wordt niet ondersteund in de standaardmodus.*

### QvWorkRoot

Retourneert de hoofdmap van de huidige Qlik Sense-app. Dit is een door het systeem gedefinieerde variabele.

```
QvWorkRoot
```

*Deze variabele wordt niet ondersteund in de standaardmodus.*

### StripComments

Als deze variabele wordt ingesteld op 0, worden de opmerkingen in het script tussen /*..*/ en na  // niet verwijderd. Als deze variabele niet is ingesteld, worden opmerkingen altijd verwijderd.

```
StripComments
```

### Verbatim

Normaliter worden alle veldwaarden automatisch ontdaan van voorafgaande of opvolgende spaties (ASCII 32) voordat ze in de Qlik Sense-database worden geladen. Als u deze variabele instelt op 1, worden de spaties niet verwijderd. Tabs (ASCII 9) en harde spaties (ANSI 160) worden nooit verwijderd.

```
Verbatim
```

### WinPath

Retourneert het pad naar Windows. Dit is een door het systeem gedefinieerde variabele.

```
WinPath
```

*Deze variabele wordt niet ondersteund in de standaardmodus.*

### WinRoot

Retourneert de hoofdmap van Windows. Dit is een door het systeem gedefinieerde variabele.

```
WinRoot
```

*Deze variabele wordt niet ondersteund in de standaardmodus.*

### CollationLocale

Geeft op welke landinstellingen moeten worden gebruikt voor sorteervolgorde en zoekafstemming. De waarde is de cultuurnaam van een landinstelling, bijvoorbeeld 'en-US'.Dit is een door het systeem gedefinieerde variabele.

```
CollationLocale
```

## CreateSearchIndexOnReload

Deze variabele wordt gebruikt om aan te geven of zoekindexbestanden moeten worden gemaakt wanneer gegevens opnieuw worden geladen.

**Syntaxis:**

```
CreateSearchIndexOnReload
```

U kunt instellen of u wilt dat zoekindexbestanden worden gemaakt tijdens het opnieuw laden van gegevens of na de eerste zoekopdracht van de gebruiker. Het voordeel van het maken van zoekindexbestanden tijdens het opnieuw laden van gegevensbestanden is dat u niet zoals de eerste gebruiker van de zoekopdracht moet wachten. De afweging is dat het langer duurt om gegevens opnieuw te laden wanneer een zoekindex wordt gemaakt.

Als deze variabele wordt weggelaten, worden tijdens het opnieuw laden van de gegevens geen zoekindexbestanden gemaakt.

*Voor sessieapps worden geen zoekindexbestanden gemaakt tijdens het opnieuw laden van gegevens, ongeacht de instelling van deze variabele.*

*Nieuwe apps bevatten `set CreateSearchIndexOnReload=1` in de standaard `SET`-instructies. Bij het maken van scripts in het Analyse-activiteitencentrum bevatten de SET-uitdrukkingen geen `CreateSearchIndexOnReload` als scriptbedrijfsmiddelen om gegevens te bewaren, waardoor een indexwaarde niet nodig is.*

**Example 1: Zoekindexbestanden maken tijdens het opnieuw laden van gegevens**

```
set CreateSearchIndexOnReload=1;
```

**Example 2: Zoekindexbestanden maken na de eerste zoekopdracht**

```
set CreateSearchIndexOnReload=0;
```

## HidePrefix

Alle veldnamen die beginnen met deze teksttekenreeks worden op dezelfde manier als de systeemvelden verborgen. Dit is een door de gebruiker gedefinieerde variabele.

**Syntaxis:**

```
HidePrefix
```

**Voorbeeld:**

```
set HidePrefix='_' ;
```

Bij gebruik van deze opdracht worden de veldnamen die beginnen met een liggend streepje niet getoond in de lijst met veldnamen als de systeemvelden verborgen zijn.

## HideSuffix

Alle veldnamen die eindigen met deze teksttekenreeks worden op dezelfde manier als de systeemvelden verborgen. Dit is een door de gebruiker gedefinieerde variabele.

**Syntaxis:**

```
HideSuffix
```

**Voorbeeld:**

```
set HideSuffix='%';
```

Bij gebruik van deze opdracht worden de veldnamen die eindigen met een procentteken niet getoond in de lijst met veldnamen als de systeemvelden verborgen zijn.

## Include

De variabele **Include/Must_Include** specificeert een bestand dat tekst bevat die in het script moet worden opgenomen en moet worden geëvalueerd als scriptcode. Deze wordt niet gebruikt om gegevens toe te voegen. U kunt delen van uw scriptcode opslaan in een apart tekstbestand en dit in meerdere apps hergebruiken. Dit is een door de gebruiker gedefinieerde variabele.

*In Qlik Sense en Qlik Sense Desktop ondersteunt deze variabele alleen verwijzingen naar verbindingen voor mapgegevens. Verwijzingen naar bestanden in aanbieders van cloud-gebaseerde opslag worden niet ondersteund.*

**Syntaxis:**

```
$(Include=filename)
```

```
$(Must_Include=filename)
```

Er zijn twee versies van de variabele:

- **Include** genereert geen fout als het bestand niet wordt gevonden. De bewerking mislukt zonder melding.
- **Must_Include** genereert een fout als het bestand niet wordt gevonden.

Als u geen pad opgeeft, is de bestandsnaam gerelateerd aan de werkmap voor de Qlik Sense-app. U kunt ook een absoluut bestandspad opgegeven of een pad naar een lib:// mapverbinding. Zet geen spatie voor of na het gelijkteken.

*De constructie **set Include =**filename is niet van toepassing.*

**Voorbeelden:**

```
$(Include=abc.txt);
```

```
$(Must_Include=lib://DataFiles/abc.txt);
```

### Beperkingen

#### Beperkte cross-compatibiliteit tussen UTF-8 gecodeerde bestanden onder Windows versus Linux.

Het gebruik van UTF-8 met BOM (Byte Order Mark) is optioneel. BOM kan interfereren met het gebruik van UTF-8 in software die geen niet-ASCII bytes verwacht aan het begin van een bestand, maar die anders wel met de tekststroom overweg kan.

- Windows systemen gebruiken BOM in UTF-8 om aan te geven dat een bestand UTF-8 gecodeerd is, ook al is er geen dubbelzinnigheid in de byte-opslag.
- Unix/Linux gebruiken UTF-8 voor Unicode, maar gebruiken de BOM niet omdat dit interfereert met de syntaxis voor opdrachtbestanden.

Dit heeft een aantal consequenties voor Qlik Sense.

- In Windows wordt elk bestand dat begint met een UTF-8 BOM beschouwd als een UTF-8 scriptbestand. Anders wordt ANSI-codering verondersteld.
- In Linux is de standaard 8 bit codepagina UTF-8. Daarom werkt UTF-8, ook al bevat het geen BOM.

Zodoende kan de overdraagbaarheid niet worden gegarandeerd. Het is niet altijd mogelijk om onder Windows een bestand te maken dat door Linux kan worden weergegeven en omgekeerd. Er is geen onderlinge compatibiliteit tussen de twee systemen wat betreft UTF-8 gecodeerde bestanden, omdat de BOM verschillend wordt behandeld.

## OpenUrlTimeout

Deze variabele bepaalt de time-out in seconden die Qlik Sense moet aanhouden bij het ophalen van gegevens uit URL-bronnen (bijvoorbeeld -pagina's). HTML pagina's). Als deze variabele niet is ingesteld, is de time-out ongeveer 20 minuten.

**Syntaxis:**

```
OpenUrlTimeout
```

**Voorbeeld:**

```
set OpenUrlTimeout=10;
```

## StripComments

Als deze variabele wordt ingesteld op 0, worden de opmerkingen in het script tussen /*..*/ en na // niet verwijderd. Als deze variabele niet is ingesteld, worden opmerkingen altijd verwijderd.

**Syntaxis:**

```
StripComments
```

Bepaalde databasestuurprogramma's geven tussen de tekens /*..*/ hints voor het optimaliseren van **SELECT**-opdrachten. Als dit het geval is, moeten de opmerkingen niet worden verwijderd voordat de **SELECT**-opdracht naar het databasestuurprogramma wordt gestuurd.

*Het wordt aanbevolen deze variabele meteen na de betreffende opdrachten terug te zetten op 1.*

**Voorbeeld:**

```
set StripComments=0;
SQL SELECT * /* <optimization directive> */ FROM Table ;
set StripComments=1;
```

### Verbatim

Normaliter worden alle veldwaarden automatisch ontdaan van voorafgaande of opvolgende spaties (ASCII 32) voordat ze in de Qlik Sense-database worden geladen. Als u deze variabele instelt op 1, worden de spaties niet verwijderd. Tabs (ASCII 9) en harde spaties (ANSI 160) worden nooit verwijderd.

**Syntaxis:**

```
Verbatim
```

**Voorbeeld:**

```
set Verbatim = 1;
```

## 4.7 Variabelen voor afhandeling van waarden

In dit hoofdstuk worden variabelen beschreven die worden gebruikt voor de afhandeling van NULL en andere waarden.

### Overzicht van variabelen voor afhandeling van waarden

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

**NullDisplay**

Het gedefinieerde symbool vervangt alle NULL-waarden van ODBC en connectoren op het laagste gegevensniveau. Dit is een door de gebruiker gedefinieerde variabele.

```
NullDisplay
```

**NullInterpret**

Als het gedefinieerde symbool in een tekstbestand, CSV-bestand of een uitgelijnde uitdrukking staat, wordt het geïnterpreteerd als NULL. Dit is een door de gebruiker gedefinieerde variabele.

```
NullInterpret
```

**NullValue**

Als de opdracht **NullAsValue** wordt gebruikt, vervangt het gedefinieerde symbool alle NULL-waarden in de velden waarvoor **NullAsValue** is opgegeven, door de opgegeven tekenreeks.

```
NullValue
```

**OtherSymbol**
Geeft aan dat een symbool voor een **LOAD/SELECT**-uitdrukking moet worden behandeld zoals 'alle andere waarden'. Dit is een door de gebruiker gedefinieerde variabele.

```
OtherSymbol
```

## NullDisplay

Het gedefinieerde symbool vervangt alle NULL-waarden van ODBC en connectoren op het laagste gegevensniveau. Dit is een door de gebruiker gedefinieerde variabele.

**Syntaxis:**

```
NullDisplay
```

**Voorbeeld:**

```
set NullDisplay='<NULL>';
```

## NullInterpret

Als het gedefinieerde symbool in een tekstbestand, CSV-bestand of een uitgelijnde uitdrukking staat, wordt het geïnterpreteerd als NULL. Dit is een door de gebruiker gedefinieerde variabele.

**Syntaxis:**

```
NullInterpret
```

**Voorbeelden:**

```
set NullInterpret=' ';

set NullInterpret =;

set NullInterpret ='';
```

## NullValue

Als de opdracht **NullAsValue** wordt gebruikt, vervangt het gedefinieerde symbool alle NULL-waarden in de velden waarvoor **NullAsValue** is opgegeven, door de opgegeven tekenreeks.

**Syntaxis:**

```
NullValue
```

**Voorbeeld:**

```
NullAsValue Field1, Field2;
set NullValue='<NULL>';
```

## OtherSymbol

Geeft aan dat een symbool voor een **LOAD/SELECT**-uitdrukking moet worden behandeld zoals 'alle andere waarden'. Dit is een door de gebruiker gedefinieerde variabele.

**Syntaxis:**

```
OtherSymbol
```

**Voorbeeld:**

```
set OtherSymbol='+';
LOAD * inline
[X, Y
a, a
b, b];
LOAD * inline
[X, Z
a, a
+, c];
```

De veldwaarde Y='b' is nu gekoppeld aan Z='c' via het andere symbool.

# 4.8 Variabelen voor getalinterpretatie

Variabelen voor getalinterpretatie worden door het systeem vastgesteld. De variabelen staan aan het begin van het load-script en passen instellingen voor de getalnotatie toe op het moment dat het script wordt uitgevoerd. Ze kunnen worden verwijderd, bewerkt of gedupliceerd.

Variabelen voor getalinterpretatie worden automatisch gegenereerd volgens de huidige landinstellingen van het besturingssysteem wanneer een nieuwe app wordt gemaakt. In Qlik Sense Desktop is dit afhankelijk van de instellingen van het besturingssysteem van de computer. In Qlik Sense is dit op basis van de instellingen van het besturingssysteem van de server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Valutanotatie

### MoneyDecimalSep

Het gedefinieerde scheidingsteken voor decimalen vervangt het decimaalteken voor valuta dat is ingesteld door uw landinstellingen.

```
MoneyDecimalSep
```

### MoneyFormat

Het gedefinieerde symbool vervangt het valutasymbool dat is ingesteld door uw landinstellingen.

```
MoneyFormat
```

### MoneyThousandSep

Het gedefinieerde scheidingsteken voor duizendtallen vervangt het cijfergroeperingssymbool voor valuta dat is ingesteld door uw landinstellingen.

```
MoneyThousandSep
```

## Getalnotatie

### DecimalSep

Het gedefinieerde scheidingsteken voor decimalen vervangt het decimaalteken dat is ingesteld door uw landinstellingen.

```
DecimalSep
```

### ThousandSep

Het gedefinieerde scheidingsteken voor duizendtallen vervangt het cijfergroeperingssymbool van het besturingssysteem (landinstellingen).

```
ThousandSep
```

### NumericalAbbreviation

De numerieke afkorting bepaalt welke afkorting u moet gebruiken als het voorvoegsel dat de schaal van numerieke waarden aangeeft, bijvoorbeeld M voor mega of een miljoen ($10^6$), en µ voor micro ($10^{-6}$).

```
NumericalAbbreviation
```

## Tijdnotatie

### DateFormat

Deze omgevingsvariabele definieert de datumopmaak die standaard wordt gebruikt in de app. De opmaak wordt gebruikt om datums te interpreteren en op te maken. Als de variabele niet is gedefinieerd, wordt de datumopmaak van de regionale instellingen van het besturingssysteem opgehaald wanneer het script wordt uitgevoerd.

```
DateFormat
```

### TimeFormat

De gedefinieerde notatie vervangt de tijdnotatie van het besturingssysteem (landinstellingen).

```
TimeFormat
```

### TimestampFormat

De gedefinieerde notatie vervangt de datum- en tijdnotatie van het besturingssysteem (landinstellingen).

```
TimestampFormat
```

### MonthNames

De gedefinieerde notatie vervangt de naamgevingsconventie voor maanden in de landinstellingen.

```
MonthNames
```

### LongMonthNames

De gedefinieerde notatie vervangt de lange naamgevingsconventie voor maanden in de landinstellingen.

```
LongMonthNames
```

### DayNames

De gedefinieerde notatie vervangt de naamgevingsconventie voor weekdagen die zijn ingesteld door uw landinstellingen.

```
DayNames
```

### LongDayNames

De gedefinieerde notatie vervangt de lange naamgevingsconventie voor weekdagen in de landinstellingen.

```
LongDayNames
```

### FirstWeekDay

Geheel getal dat definieert welke dag moet worden gebruikt als eerste dag van de week.

```
FirstWeekDay
```

### BrokenWeeks

Deze instelling definieert of weken worden gesplitst of niet.

```
BrokenWeeks
```

### ReferenceDay

De instelling definieert welke dat in januari is ingesteld als referentiedag voor het definiëren van week 1.

```
ReferenceDay
```

### FirstMonthOfYear

De instelling definieert welke maand moet worden gebruikt als eerste maand van het jaar. Deze kan worden gebruikt voor het definiëren van boekjaren die een maandelijkse verschuiving hebben en bijvoorbeeld op 1 april starten.

> *Deze instelling wordt momenteel niet gebruikt, maar is gereserveerd voor toekomstig gebruik.*

Geldige instellingen zijn 1 (januari) t/m 12 (december). De standaardinstelling is 1.

**Syntaxis:**

```
FirstMonthOfYear
```

**Voorbeeld:**

```
Set FirstMonthOfYear=4; //Sets the year to start in April
```

## BrokenWeeks

Deze instelling definieert of weken worden gesplitst of niet.

**Syntaxis:**

```
BrokenWeeks
```

In Qlik Sense worden de landinstellingen opgehaald wanneer de app wordt gemaakt en de bijbehorende instellingen worden als omgevingsvariabelen opgeslagen in het script.

Een Noord-Amerikaanse app-ontwikkelaar krijgt vaak `Set BrokenWeeks=1;` in het script, wat overeenkomt met gebroken weken. Een Europese app-ontwikkelaar krijgt vaak `Set BrokenWeeks=0;` in het script, wat overeenkomt met ongebroken weken.

Ongebroken weken betekent dat:

- In sommige jaren begint week 1 in december en in andere jaren week loopt de laatste week van het voorgaande jaar door tot in januari.
- Volgens ISO 8601 heeft week 1 altijd ten minste 4 dagen in januari. In Qlik Sense kan dit worden geconfigureerd met de variabele `ReferenceDay`.

Gebroken weken betekent dat:

- De laatste week van het jaar loopt nooit door tot in januari.
- Week 1 begint op 1 januari en is, in de meeste gevallen, geen volle week.

De volgende waarden kunnen worden gebruikt:

- 0 (= ongebroken weken gebruiken)
- 1 (= gebroken weken gebruiken)

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

**Voorbeelden:**

Als u ISO-instellingen wilt voor weken en weeknummers, moet u het volgende opnemen in het script:

```
Set FirstWeekDay=0;
Set BrokenWeeks=0;   //(use unbroken weeks)
Set ReferenceDay=4;
```

Als u US-instellingen wilt, moet u het volgende opnemen in het script:

```
Set FirstWeekDay=6;
Set BrokenWeeks=1;   //(use broken weeks)
Set ReferenceDay=1;
```

## DateFormat

Deze omgevingsvariabele definieert de datumnotatie die als standaard wordt gebruikt in de app en op datumretourfuncties zoals `date()` en `date#()`. De notatie wordt gebruikt om datums te interpreteren en op te maken. Als de variabele niet is gedefinieerd, wordt de datumnotatie die is ingesteld door uw regionale instellingen opgehaald wanneer het script wordt uitgevoerd.

**Syntaxis:**

```
DateFormat
```

Voorbeelden functie DateFormat

| Voorbeeld | Resultaat |
|---|---|
| `Set DateFormat='M/D/YY'; //(US format)` | Dit gebruik van de functie `DateFormat` definieert de datum als de Amerikaanse notatie, maand/dag/jaar. |
| `Set DateFormat='DD/MM/YY'; //(UK date format)` | Dit gebruik van de functie `DateFormat` definieert de datum als de Britse notatie, dag/maand/jaar. |
| `Set DateFormat='YYYY/MM/DD'; //(ISO date format)` | Dit gebruik van de functie `DateFormat` definieert de datum als de ISO-notatie, jaar/maand/dag. |

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 – Standaardsysteemvariabelen

Load-script en resultaten

**Overzicht**

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling van datums.
- De functie `DateFormat` die de Amerikaanse datumnotatie gebruikt.

In dit voorbeeld wordt een gegevensset geladen in de tabel '`Transactions`'. Deze bevat een veld `date`. De Amerikaanse `DateFormat`-definitie wordt gebruikt. Dit patroon wordt gebruikt voor de omzetting van impliciete tekst naar een datum wanneer de tekstdatums worden geladen.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `month`

Maak deze meting:

```
=sum(amount)
```

Resultatentabel

| date | month | =sum(amount) |
|---|---|---|
| 01/01/2022 | Jan | 1000 |
| 02/01/2022 | Feb | 2123 |
| 03/01/2022 | Mar | 4124 |
| 04/01/2022 | Apr | 2431 |

De definitie `DateFormat` MM/DD/YYYY wordt gebruikt voor de impliciete omzetting van tekst in datums en dit is waarom het veld `date` als een datum wordt geïnterpreteerd. Dezelfde notatie wordt gebruikt voor het weergeven van de datum, zoals in de resultatentabel wordt getoond.

## Voorbeeld 2 – Systeemvariabele wijzigen

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensverzameling uit het vorige voorbeeld.
- De `DateFormat` functie die de notatie 'DD/MM/JJJJ' gebruikt.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `month`

Maak deze meting:

```
=sum(amount)
```

Resultatentabel

| date | month | =sum(amount) |
|---|---|---|
| 01/01/2022 | Jan | 1000 |
| 02/01/2022 | Jan | 2123 |
| 03/01/2022 | Jan | 4124 |
| 04/01/2022 | Jan | 2431 |

Omdat de definitie `DateFormat` is ingesteld op 'DD/MM/JJJJ' ziet u dat de twee getallen achter het eerste '/'-symbool zijn geïnterpreteerd als maand, waardoor alle records afkomstig zijn uit de maand januari.

## Voorbeeld 3 – Interpretatie van datum

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met datums in numerieke indeling.
- De `DateFormat` variabele die de notatie 'DD/MM/JJJJ' gebruikt.
- De `date()` variabele.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `month`

Maak deze meting:

```
=sum(amount)
```

Resultatentabel

| date | month | =sum(amount) |
|---|---|---|
| 06/03/2022 | Jun | 1000 |

| date | month | =sum(amount) |
|---|---|---|
| 06/04/2022 | Jun | 2123 |
| 06/05/2022 | Jun | 4124 |
| 06/07/2022 | Jun | 2431 |

In het load-script gebruikt u de `date()` functie om de numerieke datum om te zetten in een datumnotatie. Omdat u in de functie geen specifieke notatie als tweede argument opgeeft, wordt de `DateFormat` gebruikt. Dit resulteert in een datumveld met de notatie 'MM/DD/JJJJ'.

## Voorbeeld 4 – Notatie buitenlandse datum

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling van datums.
- De variabele `DateFormat`, die de notatie 'DD/MM/JJJJ' gebruikt maar niet wordt omsloten door slashes.

### Load-script

```
// 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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `month`

Maak deze meting:

```
=sum(amount)
```

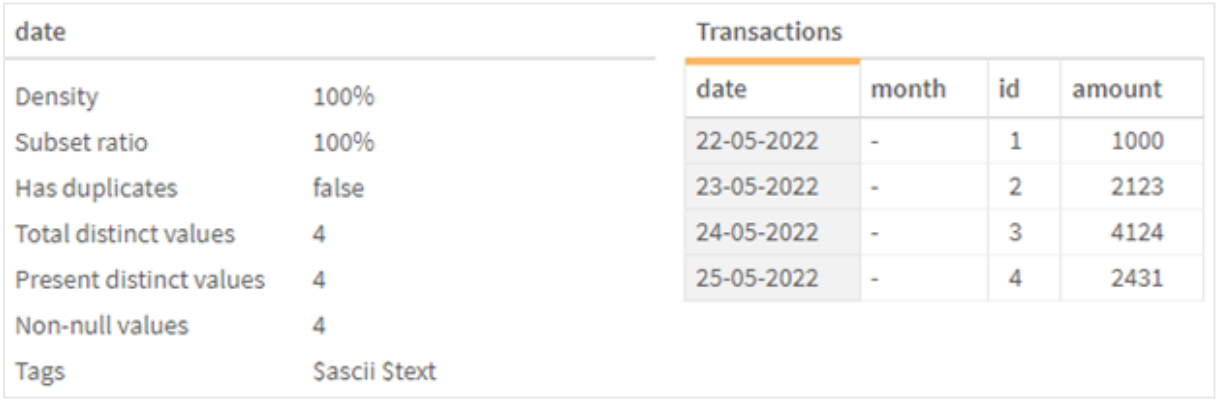
Resultatentabel

| date | month | =sum(amount) |
|---|---|---|
| 22-05-2022 | - | 1000 |
| 23-05-2022 | - | 2123 |
| 24-05-2022 | - | 4124 |
| 25-05-2022 | - | 2431 |

In het initiële load-script wordt `DateFormat` gebruikt als de standaard 'MM/DD/JJJJ'. Omdat het veld `date` in de gegevensverzameling Transacties deze notatie niet heeft, wordt het veld niet als datum geïnterpreteerd. Dit is te zien in de resultatentabel waarin de veldwaarden `month` null zijn.

U kunt de geïnterpreteerde gegevenstypen verifiëren in de gegevensmodelviewer door de eigenschappen van 'Labels' van het veld `date` te controleren:

*Voorbeeld van de tabel `Transactions`. Let op de 'Labels' voor het veld `date` die aangeven dat de tekstuele invoergegevens niet impliciet zijn geconverteerd naar een datum/tijdstempel.*

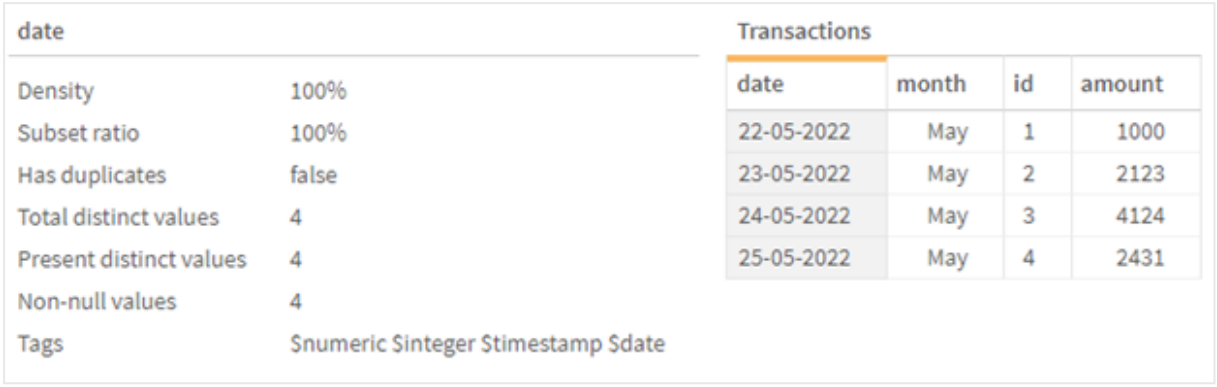



Dit kan worden opgelost door de systeemvariabele `DateFormat` in te schakelen:

```
// SET DateFormat='DD/MM/YYYY';
```

Verwijder de dubbele slashes en laad de gegevens opnieuw.

*Voorbeeld van de tabel `Transactions`. Let op de 'Labels' voor het veld `date` die aangeven dat de tekstuele invoergegevens impliciet zijn geconverteerd naar een datum/tijdstempel.*



date

| | |
|---|---|
| Density | 100% |
| Subset ratio | 100% |
| Has duplicates | false |
| Total distinct values | 4 |
| Present distinct values | 4 |
| Non-null values | 4 |
| Tags | $numeric $integer $timestamp $date |

Transactions

| date | month | id | amount |
|---|---|---|---|
| 22-05-2022 | May | 1 | 1000 |
| 23-05-2022 | May | 2 | 2123 |
| 24-05-2022 | May | 3 | 4124 |
| 25-05-2022 | May | 4 | 2431 |

## DayNames

De gedefinieerde notatie vervangt de naamgevingsconventie voor weekdagen die zijn ingesteld door uw landinstellingen.

**Syntaxis:**

```
DayNames
```

Bij het wijzigen van de variabele is een puntkomma ; vereist om de afzonderlijke waarden te scheiden.

Voorbeelden van functie DayName

| Functievoorbeeld | Resultaatdefinitie |
|---|---|
| `Set DayNames='Mon;Tue;Wed;Thu;Fri;Sat;Sun';` | Dit gebruik van de functie Dagnamen definieert dagnamen in hun afgekorte vorm. |
| `Set DayNames='M;Tu;W;Th;F;Sa;Su';` | Dit gebruik van de functie Dagnamen definieert dagnamen op hun eerste letters. |

De functie `DayNames` wordt vaak gebruikt in combinatie met de volgende functies:

Gerelateerde functies

| Functie | Interactie |
|---|---|
| *weekday (page 1197)* | Scriptfunctie om `DayNames` als veldwaarden te retourneren. |
| *Date (page 1391)* | Scriptfunctie om `DayNames` als veldwaarden te retourneren. |
| *LongDayNames (page 248)* | Lange vormwaarden van `DayNames`. |

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege

landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – Standaardsysteemvariabelen

Load-script en resultaten

### Overzicht

In dit voorbeeld zijn de datums in de gegevensverzameling ingesteld in de notatie MM/DD/JJJJ.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met datums, die wordt geladen in een tabel met de naam `Transactions`.
- Een veld `date`.
- De standaard `DayNames`-definitie.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- dayname

Maak deze meting:

```
sum(amount)
```

Resultatentabel

| date | dayname | sum(amount) |
|---|---|---|
| 01/01/2022 | Za | 1000 |
| 02/01/2022 | Di | 2123 |
| 03/01/2022 | Di | 4124 |
| 04/01/2022 | Vr | 2431 |

In het load-script wordt de functie `WeekDay` gebruikt met het veld `date` als het opgegeven argument. In de resultatentabel geeft de uitvoer van deze functie `WeekDay` de dagen van de week weer in de notatie van de definitie `DayNames`.

## Voorbeeld 2 – Systeemvariabele wijzigen

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad. Dezelfde gegevensset en het scenario uit het eerste voorbeeld worden gebruikt.

Aan het begin van het script wordt de definitie `DayNames` echter gewijzigd om de afgekorte dagen van de week in het Afrikaans te gebruiken.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `dayname`

Maak deze meting:

```
sum(amount)
```

Resultatentabel

| date | dayname | sum(amount) |
|---|---|---|
| 01/01/2022 | Za | 1000 |
| 02/01/2022 | Di | 2123 |
| 03/01/2022 | Di | 4124 |
| 04/01/2022 | Vr | 2431 |

In de resultatentabel geeft de uitvoer van deze functie `WeekDay` de dagen van de week weer in de notatie van de definitie `DayNames`.

Het is belangrijk om te onthouden dat als de taal voor de `DayNames` wordt gewijzigd zoals in dit voorbeeld, de `LongDayNames` nog steeds de dagen van de week in het Engels bevat. Dit zou ook moeten worden gewijzigd als beide variabelen in de toepassing worden gebruikt.

## Voorbeeld 3 – Datumfunctie

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met datums, die wordt geladen in een tabel met de naam `Transactions`.
- Een veld `date`.
- De standaard `DayNames`-definitie.

### Load-script

```
SET DayNames='Mon;Tue;Wed;Thu;Fri;Sat;Sun';

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
];
```

**Resultaten**

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `dayname`

Maak deze meting:

```
sum(amount)
```

Resultatentabel

| date | dayname | sum(amount) |
|---|---|---|
| 01/01/2022 | Za | 1000 |
| 02/01/2022 | Di | 2123 |
| 03/01/2022 | Di | 4124 |
| 04/01/2022 | Vr | 2431 |

De standaard `DayNames`-definitie wordt gebruikt. In het load-script wordt de functie `Date` gebruikt met het veld `date` als het eerste argument. Het tweede argument is `WWW`. Deze opmaak converteert het resultaat naar de waarden die zijn opgeslagen in de definitie `DayNames`. Dit wordt weergegeven in de uitvoer van de resultatentabel.

## DecimalSep

Het gedefinieerde scheidingsteken voor decimalen vervangt het decimaalteken dat is ingesteld door uw landinstellingen.

Qlik Sense interpreteert tekst automatisch als getallen wanneer een herkenbaar nummerpatroon wordt aangetroffen. De systeemvariabelen `ThousandSep` en `DecimalSep` bepalen de samenstelling van de patronen die worden toegepast bij het parseren van tekst als getallen. De variabelen `ThousandSep` en `DecimalSep` stellen het standaardpatroon voor getalnotatie in bij het visualiseren van numerieke inhoud in front-end diagrammen en tabellen. Dat wil zeggen dat het rechtstreeks van invloed is op de opties voor **Getalnotatie** voor elke front-end-uitdrukking.

Uitgaande van een duizendtal scheidingsteken van komma ',' en een decimaal scheidingsteken van '.', zijn dit voorbeelden van patronen die impliciet zouden worden geconverteerd naar numerieke equivalente waarden:

```
0,000.00
```

`0000.00`

`0,000`

Dit zijn voorbeelden van patronen die als tekst ongewijzigd zouden blijven; dat wil zeggen, niet geconverteerd naar numeriek:

`0.000,00`

`0,00`

**Syntaxis:**

**`DecimalSep`**

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `Set DecimalSep='.';` | Hiermee stelt u '.' in als het scheidingsteken voor decimalen. |
| `Set DecimalSep=',';` | Hiermee stelt u ',' in als het scheidingsteken voor decimalen. |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld – Effect van het instellen van variabelen voor het scheiden van getallen op verschillende invoergegevens

Load-script en resultaten

**Overzicht**

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling van sommen en datums met de sommen in verschillende opmaakpatronen.

- Een tabel met de naam `Transactions`.
- De variabele `DecimalSep` die is ingesteld op '.'.
- De variabele `ThousandSep` die is ingesteld op ','.
- De variabele `delimiter` die is ingesteld als het teken '|' om de verschillende velden op een regel te scheiden.

### Load-script

```
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 '|');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie `amount`.

Maak deze meting:

```
=sum(amount)
```

Resultatentabel

| Amount | =Sum(amount) |
|---|---|
| Totalen | 20814.7086 |
| 1.000-45 | |
| 3.756,178 | |
| 4124,35 | |
| 23.344 | 23.344 |
| 2431.36 | 2431.36 |

| Amount | =Sum(amount) |
|---|---|
| 2431.84 | 2431.84 |
| 3,454.356 | 3454.356 |
| 3554.284 | 3554.284 |
| 4132.5246 | 4132.5246 |
| 4,787 | 4787 |

Elke waarde die niet als getal wordt geïnterpreteerd, blijft als tekst en wordt standaard links uitgelijnd. Alle succesvol geconverteerde waarden worden rechts uitgelijnd, waarbij het oorspronkelijke invoerformaat behouden blijft.

De uitdrukkingskolom toont het numerieke equivalent, dat standaard is opgemaakt met alleen een scheidingsteken voor decimalen '.'. Dit kan worden overschreven met de vervolgkeuzelijstinstelling **Getalnotatie** in de uitdrukkingsconfiguratie.

## FirstWeekDay

Geheel getal dat definieert welke dag moet worden gebruikt als eerste dag van de week.

**Syntaxis:**

```
FirstWeekDay
```

Maandag is de eerste dag van de week volgens ISO 8601, de internationale standaard voor de aanduiding van termen betreffende kalender, datum en tijd. Maandag wordt ook gebruikt als eerste dag van de week in een aantal landen, zoals het Verenigd Koninkrijk, Frankrijk, Duitsland en Zweden.

Maar in andere landen, zoals in de Verenigde Staten en Canada, wordt zondag gezien als de eerste dag van de week.

In Qlik Sense worden de landinstellingen opgehaald wanneer de app wordt gemaakt en de bijbehorende instellingen worden als omgevingsvariabelen opgeslagen in het script.

Een Noord-Amerikaanse app-ontwikkelaar krijgt vaak `Set FirstWeekDay=6;` in het script, wat overeenkomt met zondag. Een Europese app-ontwikkelaar krijgt vaak `Set FirstWeekDay=0;` in het script, wat overeenkomt met maandag.

Waarden die kunnen worden ingesteld voor FirstWeekDay

| Value | Dag |
|---|---|
| 0 | Maandag |
| 1 | Dinsdag |
| 2 | Woensdag |
| 3 | Donderdag |

| Value | Dag |
|---|---|
| 4 | Vrijdag |
| 5 | Zaterdag |
| 6 | Zondag |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

#### Voorbeelden:

Als u ISO-instellingen wilt voor weken en weeknummers, moet u het volgende opnemen in het script:

```
Set FirstWeekDay=0;  // Monday as first week day
Set BrokenWeeks=0;
Set ReferenceDay=4;
```

Als u US-instellingen wilt, moet u het volgende opnemen in het script:

```
Set FirstWeekDay=6;  // Sunday as first week day
Set BrokenWeeks=1;
Set ReferenceDay=1;
```

## Voorbeeld 1 – Maakt gebruik van de standaardwaarde (script)

### Load-script en resultaten

#### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

In dit voorbeeld gebruikt het load-script de standaardwaarde van de Qlik Sense-systemvariabele, `FirstWeekDay=6`. Deze gegevens bevatten gegevens voor de eerste 14 dagen van 2020.

### Load-script

```
// 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,
        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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `week`
- `weekday`

Resultatentabel

| Date | week | weekday |
|---|---|---|
| 01/01/2021 | 1 | Wo |
| 01/02/2021 | 1 | Do |
| 01/03/2021 | 1 | Vr |
| 01/04/2021 | 1 | Za |
| 01/05/2021 | 2 | Zo |
| 01/06/2020 | 2 | Ma |

| Date | week | weekday |
|---|---|---|
| 01/07/2020 | 2 | Di |
| 01/08/2020 | 2 | Wo |
| 01/09/2020 | 2 | Do |
| 01/10/2020 | 2 | Vr |
| 01/11/2020 | 2 | Za |
| 01/12/2020 | 3 | Zo |
| 01/13/2020 | 3 | Ma |
| 01/14/2020 | 3 | Di |

Omdat de standaardinstellingen worden gebruikt, wordt de `FirstWeekDay` systeemvariabele ingesteld op 6. In de resultatentabel is te zien dat elke nieuwe week op zondag begint (5 en 12 januari).

## Voorbeeld 2 – De FirstWeekDay variabele wijzigen (script)

Load-script en resultaten

### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

In dit voorbeeld bevatten de gegevens de eerste 14 dagen van 2020. Aan het begin van het script stellen we de `FirstWeekDay` variabele in op 3.

### Load-script

```
// 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
];
```

**Resultaten**

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `week`
- `weekday`

Resultatentabel

| Date | week | weekday |
|---|---|---|
| 01/01/2021 | 52 | Wo |
| 01/02/2021 | 1 | Do |
| 01/03/2021 | 1 | Vr |
| 01/04/2021 | 1 | Za |
| 01/05/2021 | 1 | Zo |
| 01/06/2020 | 1 | Ma |
| 01/07/2020 | 1 | Di |
| 01/08/2020 | 1 | Wo |
| 01/09/2020 | 2 | Do |
| 01/10/2020 | 2 | Vr |
| 01/11/2020 | 2 | Za |
| 01/12/2020 | 2 | Zo |
| 01/13/2020 | 2 | Ma |
| 01/14/2020 | 2 | Di |

Omdat de `FirstWeekDay` systeemvariabele is ingesteld op 3, begint de eerste dag van de week op donderdag. In de resultatentabel is te zien dat elke nieuwe week begint op donderdag (2 en 9 januari).

## LongDayNames

De gedefinieerde notatie vervangt de lange naamgevingsconventie voor weekdagen in de landinstellingen.

**Syntaxis:**

```
LongDayNames
```

Het volgende voorbeeld van de `LongDayNames`-functie definieert de dagnamen:

```
Set LongDayNames='Monday;Tuesday;Wednesday;Thursday;Friday;Saturday;Sunday';
```

Bij het wijzigen van de variabele is een puntkomma ; vereist om de afzonderlijke waarden te scheiden.

De functie `LongDayNames` kan worden gebruikt in combinatie met de functie *Date (page 1391)* die `DayNames` als veldwaarden teruggeeft.

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – Standaardsysteemvariabele

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met datums, die wordt geladen in een tabel met de naam `Transactions`.
- Een veld `date`.
- De standaard `LongDayNames`-definitie.

### Load-script

```
SET LongDayNames='Monday;Tuesday;Wednesday;Thursday;Friday;Saturday;Sunday';

Transactions:
LOAD
date,
Date(date,'WWWW') 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
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `dayname`

Maak deze meting:

```
=sum(amount)
```

Resultatentabel

| date | dayname | =sum(amount) |
|---|---|---|
| 01/01/2022 | Zaterdag | 1000 |
| 02/01/2022 | Dinsdag | 2123 |
| 03/01/2022 | Dinsdag | 4124 |
| 04/01/2022 | Vrijdag | 2431 |

In het load-script wordt om een veld genaamd `dayname` te maken de functie `Date` gebruikt met het veld `date` als het eerste argument. Het tweede argument in de functie is de opmaak `WWWW`.

Door deze opmaak te gebruiken, worden de waarden van het eerste argument geconverteerd naar de overeenkomstige naam van de volledige dag die is ingesteld in de variabele `LongDayNames`. In de resultatentabel geven de veldwaarden van ons aangemaakte veld `dayname` dit weer.

## Voorbeeld 2 – Systeemvariabele wijzigen

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Dezelfde gegevensset en het scenario uit het eerste voorbeeld worden gebruikt. Aan het begin van het script wordt de definitie `LongDayNames` echter gewijzigd om de dagen van de week in het Spaans te gebruiken.

#### Load-script

```
SET LongDayNames='Lunes;Martes;Miércoles;Jueves;Viernes;Sábado;Domingo';

Transactions:
LOAD
date,
```

```
Date(date,'WWWW') 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
];
```

**Resultaten**

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `dayname`

Maak deze meting:

```
=sum(amount)
```

Resultatentabel

| date | dayname | =sum(amount) |
|---|---|---|
| 01/01/2022 | Sábado | 1000 |
| 02/01/2022 | Martes | 2123 |
| 03/01/2022 | Martes | 4124 |
| 04/01/2022 | Viernes | 2431 |

In het load-script wordt de variabele `LongDayNames` gewijzigd om de dagen van de week in het Spaans weer te geven.

Vervolgens maakt u een veld genaamd `dayname`, dat de functie `Date` is die wordt gebruikt met het veld `date` als het eerste argument.

Het tweede argument in de functie is de opmaak `WWWW`. Met behulp van deze opmaak converteert Qlik Sense de waarden van het eerste argument in de overeenkomende volledige dagnaam die is ingesteld in de variabele `LongDayNames`.

In de resultatentabel geven de veldwaarden van ons gemaakte veld `dayname` de dagen van de week weer, geschreven in het Spaans en voluit.

## LongMonthNames

De gedefinieerde notatie vervangt de lange naamgevingsconventie voor maanden in de landinstellingen.

**Syntaxis:**

```
LongMonthNames
```

Bij het wijzigen van de variabele moet de ; worden gebruikt om de afzonderlijke waarden te scheiden.

Het volgende voorbeeld van de functie LongMonthNames definieert maandnamen voluit:

```
Set
LongMonthNames='January;February;March;April;May;June;July;August;September;October;November;D
ecember';
```

De functie `LongMonthNames` wordt vaak gebruikt in combinatie met de volgende functies:

Gerelateerde functies

| Functie | Interactie |
|---|---|
| *Date (page 1391)* | Scriptfunctie om `DayNames` als veldwaarden te retourneren. |
| *LongDayNames (page 248)* | Lange vormwaarden van `DayNames`. |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – Standaardsysteemvariabelen

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met datums die wordt geladen in een tabel genaamd `Transactions`.
- Een veld `date`.
- De standaard `LongMonthNames`-definitie.

### Load-script

```
SET
LongMonthNames='January;February;March;April;May;June;July;August;September;October;November;D
ecember';
```

```
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
01/07/2022,7,2854.83
01/08/2022,8,3554.28
01/09/2022,9,3756.17
01/10/2022,10,3454.35
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies.

- `date`
- `monthname`

Maak deze meting

```
=sum(amount)
```

Resultatentabel

| date | monthname | sum(amount) |
|---|---|---|
| 01/01/2022 | Januari | 1000.45 |
| 01/02/2022 | Januari | 2123.34 |
| 01/03/2022 | Januari | 4124.35 |
| 01/04/2022 | Januari | 2431.36 |
| 01/05/2022 | Januari | 4787.78 |
| 01/06/2022 | Januari | 2431.84 |
| 01/07/2022 | Januari | 2854.83 |
| 01/08/2022 | Januari | 3554.28 |
| 01/09/2022 | Januari | 3756.17 |
| 01/10/2022 | Januari | 3454.35 |

De standaard `LongMonthNames`-definitie wordt gebruikt. In het load-script wordt om een veld genaamd `month` te maken de functie `Date` gebruikt met het veld `date` als het eerste argument. Het tweede argument in de functie is de opmaak .MMMM

Met behulp van deze opmaak converteert Qlik Sense de waarden van het eerste argument in de overeenkomende volledige maandnaam die is ingesteld in de variabele `LongMonthNames`. In de resultatentabel geven de veldwaarden van ons aangemaakte veld `month` dit weer.

## Voorbeeld 2 – Systeemvariabele wijzigen

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met datums die wordt geladen in een tabel genaamd `Transactions`.
- Een veld `date`.
- De variabele `LongMonthNames` die is aangepast om de afgekorte dagen van de week in het Spaans te gebruiken.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg `sum(amount)` toe als een meting en deze velden als dimensies:

- `date`
- `monthname`

Maak deze meting:

```
=sum(amount)
```

Resultatentabel

| 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 |

In het load-script wordt de variabele `LongMonthNames` gewijzigd om de maanden van het jaar in het Spaans weer te geven. Vervolgens, om een veld genaamd `monthname` te maken, wordt de functie `Date` gebruikt met het veld `date` als het eerste argument. Het tweede argument in de functie is de opmaak `MMMM`.

Met behulp van deze opmaak converteert Qlik Sense de waarden van het eerste argument in de overeenkomende volledige maandnaam die is ingesteld in de variabele `LongMonthNames`. In de resultatentabel geven de velwaarden van ons gemaakt veld `monthname` de maandnaam in het Spaans weer.

## MoneyDecimalSep

Het gedefinieerde scheidingsteken voor decimalen vervangt het decimaalteken voor valuta dat is ingesteld door uw landinstellingen.

*Standaard toont Qlik Sense getallen en tekst anders in tabeldiagrammen. Getallen worden rechts uitgelijnd en tekst wordt links uitgelijnd. Dit maakt het eenvoudiger om problemen bij de omzetting van tekst-naar-getallen te vinden. Tabellen op deze pagina die resultaten van Qlik Sense weergeven, gebruiken deze opmaak.*

**Syntaxis:**

```
MoneyDecimalSep
```

Qlik Sense-applicaties zullen tekstvelden die voldoen aan deze indeling interpreteren als geldwaarden. Het tekstveld moet het valutasymbool bevatten dat is gedefinieerd in de systeemvariabele `MoneyFormat`. `MoneyDecimalSep` is vooral nuttig wanneer gegevensbronnen worden verwerkt die zijn ontvangen met verschillende regionale instellingen.

Het volgende voorbeeld geeft een mogelijk gebruik van de systeemvariabele `MoneyDecimalSep` weer:

```
Set MoneyDecimalSep='.';
```

Deze functie wordt vaak gebruikt in combinatie met de volgende functies:

Gerelateerde functies

| Functie | Interactie |
|---|---|
| `MoneyFormat` | Bij de interpretatie van een tekstveld, wordt het symbool `MoneyFormat` gebruikt als onderdeel van de interpretatie. Voor getalnotatie wordt de notatie `MoneyFormat` door Qlik Sense gebruikt in diagramobjecten. |
| `MoneyThousandSep` | Bij interpretatie van een tekstveld, moet ook aan de functie `MoneyThousandSep` worden voldaan. |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 - MoneyDecimalSep puntnotatie (.)

### Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met datums die wordt geladen in een tabel genaamd `Transactions`.
- Verstrekte gegevens waarbij het geldveld de tekstnotatie met een punt '.' heeft die als scheidingsteken voor decimalen wordt gebruikt. Elk record wordt bovendien voorafgegaan door het symbool $, met uitzondering van het laatste record, dat vooraf wordt gegaan door het symbool £.

Vergeet niet dat de systeemvariabele `MoneyFormat` de dollar ($) definieert als standaardvaluta.

### Load-script

```
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'
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:`amount`.

Voeg de volgende metingen toe:

- `isNum(amount)`
- `sum(amount)`

Controleer de onderstaande resultaten die de juiste interpretatie van alleen alle dollar-waarden ($) laat zien.

Resultatentabel

| amount | =isNum(amount) | =Sum(amount) |
|---|---|---|
| Totalen | 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 |

| amount | =isNum(amount) | =Sum(amount) |
|---|---|---|
| $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 |

De bovenstaande resultatentabel laat zien hoe het veld `amount` correct is geïnterpreteerd voor alle waarden met het dollar-prefix ($) en dat het `amount` met het pond-prefix (£) niet is omgezet in een geldelijke waarde.

## Voorbeeld 2 - MoneyDecimalSep komma-notatie (,)

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset die wordt geladen in de tabel `Transactions`.
- Verstrekte gegevens waarbij het geldveld de tekstnotatie met een komma ',' heeft die als scheidingsteken voor decimalen wordt gebruikt. Elk record heeft ook het dollar-prefix ($) met uitzondering van het laatste record, dat het onjuiste scheidingsteken voor decimalen gebruikt, de punt '.'.

Vergeet niet dat de systeemvariabele `MoneyFormat` de dollar ($) definieert als standaardvaluta.

### Load-script

```
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'
];
```

**Resultaten**

Alineatekst voor Resultaten.

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:`amount`.

Voeg de volgende metingen toe:

- `isNum(amount)`
- `sum(amount)`

Controleer de onderstaande resultaten voor een demonstratie van de juiste interpretatie van alle waarden met uitzondering van het bedrag waarbij de punt '.' als scheidingsteken voor decimalen wordt gebruikt. In dat geval had een punt moeten zijn gebruikt in plaats van een komma.

Resultatentabel

| amount | =isNum(amount) | =Sum(amount) |
|---|---|---|
| Totalen | 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

Deze systeemvariabele definieert het opmaakpatroon dat wordt gebruikt door Qlik voor automatische vertaling van tekst naar getal, waarbij het getal een valutasymbool als prefix krijgt. Het definieert tevens hoe metingen waarbij de opmaaknotatie-eigenschappen zijn ingesteld op Geld, in diagramobjecten worden weergegeven.

Het gedefinieerde symbool als onderdeel van het opmaakpatroon in de systeemvariabele `MoneyFormat` vervangt het valutasymbool dat is ingesteld door uw landinstellingen.

> *Standaard toont Qlik Sense getallen en tekst anders in tabeldiagrammen. Getallen worden rechts uitgelijnd en tekst wordt links uitgelijnd. Dit maakt het eenvoudiger om problemen bij de omzetting van tekst-naar-getallen te vinden. Tabellen op deze pagina die resultaten van Qlik Sense weergeven, gebruiken deze opmaak.*

**Syntaxis:**

```
MoneyFormat
```

```
Set MoneyFormat='$ #,##0.00; ($ #,##0.00)';
```

Deze opmaak wordt weergegeven in diagramobjecten wanneer de eigenschap `Number Formatting` van een numeriek veld is ingesteld op `Money`. Als het valutasymbool van het tekstveld overeenkomt met het symbool dat is ingesteld in de variabele `MoneyFormat` en numerieke tekstvelden worden geïnterpreteerd door Qlik Sense, interpreteert Qlik Sense dit veld als een geldelijke waarde

Deze functie wordt vaak gebruikt in combinatie met de volgende functies:

Gerelateerde functies

| Functie | Interactie |
|---|---|
| *MoneyDecimalSep (page 255)* | Voor getalnotatie wordt `MoneyDecimalSep` gebruikt in de veldindeling van objecten. |
| *MoneyThousandSep (page 264)* | Voor getalnotatie wordt `MoneyThousandSep` gebruikt in de veldindeling van objecten. |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 - MoneyFormat

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat een gegevensverzameling die wordt geladen in een tabel met de naam `Transactions`. De standaarddefinitie voor de `MoneyFormat`-variabele wordt gebruikt.

### Load-script

```
SET MoneyThousandSep=',';
SET MoneyDecimalSep='.';
SET MoneyFormat='$###0.00;-$###0.00';

Transactions:
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `amount`

Voeg deze meting toe:

```
=Sum(amount)
```

Selecteer onder **Getalnotatie Geld** om `Sum(amount)` in te stellen als een geldelijke waarde.

Resultatentabel

| date | Amount | =Sum(amount) |
|---|---|---|
| Totalen | | $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 |

De standaard `MoneyFormat`-definitie wordt gebruikt. Dit ziet er als volgt uit: `$###0.00;-$###0.00`. In de resultatentabel geeft de opmaak van het `amount`-veld het valutasymbool en de decimale punt weer en er zijn decimalen toegevoegd.

## Voorbeeld 2 - MoneyFormat met scheidingsteken voor duizendtallen en meerdere invoerindelingen

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met gemengde-invoerindeling die is geladen in de tabel `Transactions` met scheidingstekens voor duizendtallen en scheidingstekens voor decimalen.
- De `MoneyFormat`-definitie wordt aangepast met een komma als scheidingsteken voor duizendtallen.
- Een van de rijen met gegevens heeft komma's als scheidingsteken voor duizendtallen op de verkeerde plek. Hier is te zien hoe dit bedrag tekst is en niet kan worden geïnterpreteerd als getal.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `amount`

Voeg deze meting toe:

```
=Sum(amount)
```

Selecteer onder **Getalnotatie Geld** om `Sum(amount)` in te stellen als een geldelijke waarde.

Resultatentabel

| date | Amount | =Sum(amount) |
|---|---|---|
| Totalen | | $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 |

Aan het begin van het script kan de systeemvariabele `MoneyFormat` worden aangepast met een komma als scheidingsteken voor duizendtallen. In de Qlik Sense-tabel is te zien hoe de indeling dit scheidingsteken bevat. Daarnaast is de rij met het onjuiste scheidingsteken niet correct geïnterpreteerd en blijft de inhoud tekst. Daarom telt deze rij niet mee in het totaal.

## MoneyThousandSep

Het gedefinieerde scheidingsteken voor duizendtallen vervangt het cijfergroeperingssymbool voor valuta dat is ingesteld door uw landinstellingen.

*Standaard toont Qlik Sense getallen en tekst anders in tabeldiagrammen. Getallen worden rechts uitgelijnd en tekst wordt links uitgelijnd. Dit maakt het eenvoudiger om problemen bij de omzetting van tekst-naar-getallen te vinden. Tabellen op deze pagina die resultaten van Qlik Sense weergeven, gebruiken deze opmaak.*

**Syntaxis:**

```
MoneyThousandSep
```

Qlik Sense-applicaties zullen tekstvelden die voldoen aan deze indeling interpreteren als geldwaarden. Het tekstveld moet het valutasymbool bevatten dat is gedefinieerd in de systeemvariabele `MoneyFormat`. `MoneyThousandSep` is vooral nuttig wanneer gegevensbronnen worden verwerkt die zijn ontvangen met verschillende regionale instellingen.

Het volgende voorbeeld geeft een mogelijk gebruik van de systeemvariabele `MoneyThousandSep` weer:

```
Set MoneyDecimalSep=',';
```

Deze functie wordt vaak gebruikt in combinatie met de volgende functies:

Gerelateerde functies

| Functie | Interactie |
|---|---|
| `MoneyFormat` | Bij de interpretatie van een tekstveld, wordt het symbool `MoneyFormat` gebruikt als onderdeel van de interpretatie. Voor getalnotatie wordt de indeling `MoneyFormat` door Qlik Sense gebruikt in diagramobjecten. |
| `MoneyDecimalSep` | Bij interpretatie van een tekstveld, moet ook aan de functie `MoneyDecimalSep` worden voldaan. |

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 - MoneyThousandSep komma-notatie (,)

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in de tabel `Transactions`.
- Verstrekte gegevens waarbij het geldveld de tekstindeling met een komma heeft die als scheidingsteken voor duizendtallen wordt gebruikt. Elk record heeft ook de prefix $.

Vergeet niet dat de systeemvariabele `MoneyFormat` de dollar ($) definieert als standaardvaluta.

### Load-script

```
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'
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:`amount`.

Voeg de volgende metingen toe:

- `isNum(amount)`
- `sum(amount)`

Bekijk de onderstaande resultaten. De tabel laat de juiste interpretatie zien van alle waarden met een komma als scheidingsteken voor duizendtallen.

Het `amount`-veld is correct geïnterpreteerd voor alle waarden, met uitzondering van één waarde waarbij een punt als scheidingsteken voor duizendtallen is gebruikt.

Resultatentabel

| amount | =isNum(amount) | =Sum(amount) |
|---|---|---|
| Totalen | 0 | $161645330590.00 |
| $3.454.343.566 | 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 |
| $24,313,369,837 | -1 | $24313369837.00 |
| $24,33,884,663 | -1 | $24313884663.00 |
| $35,545,828,255 | -1 | $35545828255.00 |
| $37,565,817,436 | -1 | $37565817436.00 |

## Voorbeeld 2 - MoneyThousandSep punt-notatie (.)

#### Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset die wordt geladen in de tabel `Transactions`.
- Verstrekte gegevens waarbij het geldveld een tekstindeling met een punt heeft die als scheidingsteken voor duizendtallen wordt gebruikt. Elk record heeft ook de prefix $.

Vergeet niet dat de systeemvariabele `MoneyFormat` de dollar ($) definieert als standaardvaluta.

### Load-script

```
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'
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:`amount`.

Voeg de volgende metingen toe:

- `isNum(amount)`
- `sum(amount)`

Bekijk de onderstaande resultaten voor de juiste interpretatie van alle waarden met een punt als scheidingsteken voor duizendtallen.

Het `amount`-veld is correct geïnterpreteerd voor alle waarden, met uitzondering van één waarde waarbij een komma als scheidingsteken voor duizendtallen is gebruikt.

Resultatentabel

| amount | =isNum(amount) | =Sum(amount) |
|---|---|---|
| Totalen | 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

De gedefinieerde notatie vervangt de naamgevingsconventie voor maanden in de landinstellingen.

**Syntaxis:**

```
MonthNames
```

Bij het wijzigen van de variabele moet de ; worden gebruikt om de afzonderlijke waarden te scheiden.

Voorbeelden van functies

| Voorbeeld | Resultaten |
|---|---|
| `Set MonthNames='Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec';` | Dit gebruik van de functie `MonthNames` definieert maandnamen in het Engels en hun afgekorte vorm. |
| `Set MonthNames='Enero;Feb;Marzo;Abr;Mayo;Jun;Jul;Agosto;Set;Oct;Nov;Dic';` | Dit gebruik van de functie `MonthNames` definieert maandnamen in het Spaans en hun afgekorte vorm. |

De functie `MonthNames` kan worden gebruikt in combinatie met de volgende functies:

Gerelateerde functies

| Functie | Interactie |
|---|---|
| *month (page 1036)* | Scriptfunctie om waarden gedefinieerd in `MonthNames` als veldwaarden te retourneren |
| *Date (page 1391)* | Scriptfunctie om waarden gedefinieerd in `MonthNames` als veldwaarden te retourneren op basis van een opgegeven opmaakargument |
| *LongMonthNames (page 251)* | Lange vormwaarden van `MonthNames` |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – Standaardsysteemvariabelen

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met datums die wordt geladen in een tabel genaamd `Transactions`.
- Een veld `date`.
- De standaard `MonthNames`-definitie.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `monthname`

Maak deze meting:

```
=sum(amount)
```

Resultatentabel

| 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 |

De standaard `MonthNames`-definitie wordt gebruikt. In het load-script wordt de functie `Month` gebruikt met het veld `date` als het opgegeven argument.

In de resultatentabel geeft de uitvoer van deze functie `Month` de maanden van het jaar weer in de notatie van de definitie `MonthNames`.

## Voorbeeld 2 – Systeemvariabele wijzigen

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met datums die wordt geladen in een tabel genaamd `Transactions`.
- Een veld `date`.
- De variabele `MonthNames` die is aangepast om de afgekorte maanden in het Spaans te gebruiken.

### Load-script

```
                            Set
MonthNames='Enero;Feb;Marzo;Abr;Mayo;Jun;Jul;Agosto;Set;Oct;Nov;Dic';

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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `monthname`

Maak deze meting:

```
=sum(amount)
```

Resultatentabel

| date | monthname | sum(amount) |
|---|---|---|
| 01/01/2022 | Enero | 1000.45 |
| 01/02/2022 | Enero | 2123.34 |

| date | monthname | sum(amount) |
|---|---|---|
| 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 |

In het load-script wordt eerst de variabele `MonthNames` gewijzigd om de maanden van het jaar afgekort in het Spaans weer te geven. De functie `Month` wordt gebruikt met het veld `date` als het opgegeven argument.

In de resultatentabel geeft de uitvoer van deze functie `Month` de maanden van het jaar weer in de notatie van de definitie `MonthNames`.

Het is belangrijk om te onthouden dat als de taal voor de variabele `MonthNames` wordt gewijzigd zoals in dit voorbeeld, de variabele `LongMonthNames` nog steeds de maanden van het jaar in het Engels bevat. De variabele `LongMonthNames` zou moeten worden gewijzigd als beide variabelen worden gebruikt in de toepassing.

## Voorbeeld 3 – Datumfunctie

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met datums die wordt geladen in een tabel genaamd `Transactions`.
- Een veld `date`.
- De standaard `MonthNames`-definitie.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `monthname`

Maak deze meting:

```
=sum(amount)
```

Resultatentabel

| 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 |

De standaard `MonthNames`-definitie wordt gebruikt. In het load-script wordt de functie `Date` gebruikt met het veld `date` als het eerste argument. Het tweede argument is `MMM`.

Met behulp van deze opmaak converteert Qlik Sense de waarden van het eerste argument in de overeenkomende maandnaam die is ingesteld in de variabele `MonthNames`. In de resultatentabel geven de veldwaarden van ons aangemaakte veld `month` dit weer.

## NumericalAbbreviation

De numerieke afkorting bepaalt welke afkorting u moet gebruiken als het voorvoegsel dat de schaal van numerieke waarden aangeeft, bijvoorbeeld M voor mega of een miljoen ($10^6$), en µ voor micro ($10^{-6}$).

**Syntaxis:**

```
NumericalAbbreviation
```

Stel de variabele `NumericalAbbreviation` in als tekenreeks met een lijst van afkortingen voor gedefinieerde paren, gescheiden door een puntkomma. Elke afkorting voor een gedefinieerd paar moet een schaal bevatten (de decimale exponent) en worden gescheiden met een dubbele punt, bijvoorbeeld `6:M` voor een miljoen.

De standaardinstelling is `'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'`.

**Voorbeelden:**

Deze instelling wijzigt het voorvoegsel voor duizend naar t en het voorvoegsel voor een miljard naar B. deze optie is handig voor financiële applicaties waar afkortingen worden verwacht zoals t$, M$ en 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

De instelling bepaalt welke dag in januari moet worden ingesteld als referentiedag om week 1 te definiëren. Met andere woorden, deze instelling schrijft voor hoeveel dagen in week 1 datums binnen januari moeten zijn.

**Syntaxis:**

```
ReferenceDay
```

`ReferenceDay` stelt in hoeveel dagen er in de eerste week van het jaar zitten. `ReferenceDay` kan worden ingesteld op elke waarde tussen `1` en `7`. Elke waarde buiten het bereik `1-7` wordt geïnterpreteerd als het middelpunt van de week (`4`), wat overeenkomt met `ReferenceDay` ingesteld op `4`.

Als u geen waarde selecteert voor de instelling `ReferenceDay`, dan zal de standaardwaarde `ReferenceDay=0` weergeven, wat zal worden geïnterpreteerd als het middelpunt van de week (`4`), zoals te zien is in de waardentabel `ReferenceDay` hieronder.

De functie `ReferenceDay` wordt vaak gebruikt in combinatie met de volgende functies:

Gerelateerde functies

| Variabele | Interactie |
|---|---|
| *BrokenWeeks (page 229)* | Als de Qlik Sense-app met ononderbroken weken werkt, wordt de variabele-instelling `ReferenceDay` afgedwongen. Als er echter gebroken weken worden gebruikt, begint week 1 op 1 januari en eindigt deze samen met de variabele-instelling `FirstWeekDay` en negeert de vlag `ReferenceDay`. |

| Variabele | Interactie |
|---|---|
| *FirstWeekDay (page 244)* | Geheel getal dat definieert welke dag moet worden gebruikt als eerste dag van de week. |

Met Qlik Sense kunnen de volgende waarden worden ingesteld voor `ReferenceDay`:

ReferenceDay-waarden

| Waarde | Referentiedag |
|---|---|
| 0 (standaard) | 4 januari |
| 1 | 1 januari |
| 2 | januari 2 |
| 3 | 3 januari |
| 4 | 4 januari |
| 5 | 5 januari |
| 6 | 6 januari |
| 7 | 7 januari |

In het volgende voorbeeld definieert `ReferenceDay = 3` 3 januari als de referentiedag:

```
SET ReferenceDay=3; //(set January 3 as the reference day)
```

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

**Voorbeelden:**

Als u ISO-instellingen wilt voor weken en weeknummers, moet u het volgende opnemen in het script:

```
Set FirstWeekDay=0;
Set BrokenWeeks=0;
Set ReferenceDay=4;    // Jan 4th is always in week 1
```

Als u US-instellingen wilt, moet u het volgende opnemen in het script:

```
Set FirstWeekDay=6;
Set BrokenWeeks=1;
Set ReferenceDay=1;     // Jan 1st is always in week 1
```

## Voorbeeld 1 - Load-script dat gebruikmaakt van de standaardwaarde; ReferenceDay=0

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- De variabele `ReferenceDay` die is ingesteld op `0`.
- De variabele `BrokenWeeks` die is ingesteld op `0`, waardoor de app ononderbroken weken moet gebruiken.
- Een gegevensverzameling met datums van eind 2019 tot begin 2020.

### Load-script

```
SET BrokenWeeks = 0;
SET ReferenceDay = 0;

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
01/07/2020,3000
01/08/2020,5000
01/09/2020,9000
01/10/2020,5000
01/11/2020,7000
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `week`
- `weekday`

Resultatentabel

| 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 |
| 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 |

Week 52 eindigt op zaterdag 28 december. Omdat `ReferenceDay` vereist dat 4 januari in week 1 wordt opgenomen, begint week 1 daarom op 29 december en eindigt op zaterdag 4 januari.

## Voorbeeld - ReferenceDay-variabele ingesteld op 5

### Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- De variabele `ReferenceDay` die is ingesteld op `5`.
- De variabele `BrokenWeeks` die is ingesteld op `0`, waardoor de app ononderbroken weken moet gebruiken.
- Een gegevensverzameling met datums van eind 2019 tot begin 2020.

### Load-script

```
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
01/07/2020,3000
01/08/2020,5000
01/09/2020,9000
01/10/2020,5000
01/11/2020,7000
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `week`
- `weekday`

Resultatentabel

| date | week | weekday |
|---|---|---|
| 12/27/2019 | 52 | Fri |
| 12/28/2019 | 52 | Sat |
| 12/29/2019 | 53 | Sun |

| date | week | weekday |
|---|---|---|
| 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 |

Week 52 eindigt op zaterdag 28 december. De variabele `BrokenWeeks` dwingt de app om ononderbroken weken te gebruiken. De referentiedagwaarde van `5` vereist dat 5 januari wordt opgenomen in week 1.

Dit is echter acht dagen na afloop van week 52 van het voorgaande jaar. Daarom begint week 53 op 29 december en eindigt op 4 januari. Week 1 begint op zondag 5 januari.

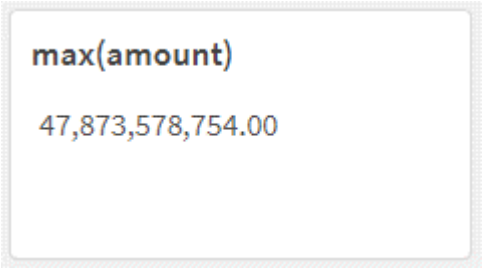
## ThousandSep

Het gedefinieerde scheidingsteken voor duizendtallen vervangt het cijfergroeperingssymbool van het besturingssysteem (landinstellingen).

**Syntaxis:**

```
ThousandSep
```

*Qlik Sense-object dat de variabele `ThousandSep` (met scheidingsteken voor duizendtallen) gebruikt*




Qlik Sense-apps zullen tekstvelden die voldoen aan deze indeling interpreteren als getallen. Deze opmaak wordt weergegeven in diagramobjecten wanneer de eigenschap van een **numeriek veld** is ingesteld op **Getal**.

`ThousandSep` is handig wanneer gegevensbronnen worden verwerkt die vanuit verschillende regionale instellingen zijn ontvangen.

> *Als de `ThousandSep`-variabele wordt aangepast nadat objecten al zijn aangemaakt en zijn opgemaakt in de applicatie, moet de gebruiker ieder relevant veld opnieuw opmaken door de eigenschap **Getal** van **Getalnotatie** te deselecteren en vervolgens weer te selecteren.*

De volgende voorbeelden geven mogelijke toepassingen weer van de systeemvariabele `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)
```

Deze onderwerpen kunnen u helpen bij het werken met deze functie:

Gerelateerde onderwerpen

| Onderwerp | Beschrijving |
|---|---|
| *DecimalSep (page 241)* | In gevallen van tekstveldinterpretatie, moeten de door deze functie verstrekte instellingen voor het scheidingsteken voor decimalen ook worden gerespecteerd. Voor getalopmaak wordt **DecimalSep** waar nodig door Qlik Sense gebruikt. |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – Standaardsysteemvariabelen

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in de tabel `Transactions`.
- Gebruik de standaarddefinitie voor de `ThousandSep`-variabele.

### Load-script

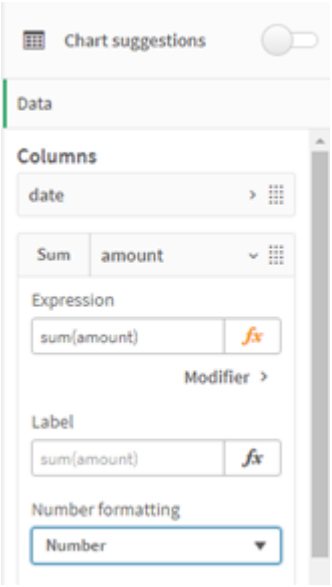
```
Transactions:
Load
date,
id,
amount
Inline
[
date,id,amount
01/01/2022,1,10000000441
01/02/2022,2,21237492432
01/03/2022,3,41249475336
01/04/2022,4,24313369837
01/05/2022,5,47873578754
01/06/2022,6,24313884663
01/07/2022,7,28545883436
01/08/2022,8,35545828255
01/09/2022,9,37565817436
01/10/2022,10,3454343566
];
```

### Resultaten

### Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:`date`.
2. Voeg de volgende meting toe:
   `=sum(amount)`
3. Selecteer de meting in het eigenschappenvenster onder **Gegevens**.
4. Onder **Getalnotatie** selecteert u **Getal**.

*Getalnotatie aanpassen voor een diagrammeting*




Resultatentabel

| 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 |

In dit voorbeeld wordt de standaarddefinitie `ThousandSep` gebruikt die is ingesteld op een komma als scheidingsteken. In de resultatentabel geeft de opmaak van het bedragveld een komma weer tussen duizendtallen.

## Voorbeeld 2 – Systeemvariabele wijzigen

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset als uit het eerste voorbeeld, geladen in de tabel `Transactions`.
- Aanpassing van de `ThousandSep`-definitie aan het begin van het script om een '*' weer te geven als het scheidingsteken voor duizendtallen. Dit is een extreem voorbeeld en wordt uitsluitend gegeven om de functionaliteit van de variabele te laten zien.

De aanpassing die in dit voorbeeld wordt gebruikt, is extreem en komt zelden voor, maar wordt hier getoond om de functionaliteit van de variabele te demonstreren.

### Load-script

```
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
];
```

### Resultaten

### Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:`date`.
2. Voeg de volgende meting toe:
   `=sum(amount)`
3. Selecteer de meting in het eigenschappenvenster onder **Gegevens**.
4. Onder **Getalnotatie** selecteert u **Aangepast**.

Resultatentabel

| date | =sum(amount) |
|---|---|
| 01/01/2022 | 10000000441.00 |

| date | =sum(amount) |
|---|---|
| 01/02/2022 | 21237492432.00 |
| 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 |

Aan het begin van het script kan de systeemvariabele `ThousandSep` worden gewijzigd in een '*'. In de resultatentabel geeft de opmaak van het bedragveld een '*' weer tussen duizendtallen.

## Voorbeeld 3 – Tekstinterpretatie

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset die wordt geladen in de tabel `Transactions`.
- Gegevens waarbij het numerieke veld een tekstindeling heeft met een komma die als scheidingsteken voor duizendtallen wordt gebruikt.
- Gebruik van de standaardsysteemvariabele `ThousandSep`.

### Load-script

```
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'
];
```

**Resultaten**

**Doe het volgende:**

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:`date`.
2. Voeg de volgende meting toe:
   `=sum(amount)`
3. Selecteer de meting in het eigenschappenvenster onder **Gegevens**.
4. Onder **Getalnotatie** selecteert u **Getal**.
5. Voeg de volgende meting toe om te bepalen of het bedragveld al dan niet een numerieke waarde is:
   `=isnum(amount)`

Resultatentabel

| 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 |
| 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 |

Zodra de gegevens zijn geladen, zien we dat Qlik Sense het bedragveld als een numerieke waarde heeft geïnterpreteerd omdat de gegevens voldoen aan de `ThousandSep`-variabele. Dit wordt gedemonstreerd door de `isnum()`-functie die iedere invoer evalueert tot -1 of `TRUE`.

*In Qlik Sense wordt de booleaanse waarde Waar vertegenwoordigd door -1 en de waarde Onwaar door 0.*

## TimeFormat

De gedefinieerde notatie vervangt de tijdnotatie van het besturingssysteem (landinstellingen).

**Syntaxis:**

```
TimeFormat
```

**Voorbeeld:**

```
Set TimeFormat='hh:mm:ss';
```

## TimestampFormat

De gedefinieerde notatie vervangt de datum- en tijdnotatie van het besturingssysteem (landinstellingen).

**Syntaxis:**

```
TimestampFormat
```

**Voorbeeld:**

In de volgende voorbeelden wordt *1983-12-14T13:15:30Z* als invoer voor de tijdstempel gebruikt om de resultaten van verschillende **SET TimestampFormat**-opdrachten weer te geven. **YYYYMMDD** wordt gebruikt als datumnotatie en **h:mm:ss TT** als tijdnotatie. Bovenaan het load-script voor gegevens wordt de datumnotatie bepaald met de opdracht **SET DateFormat** en de tijdnotatie met de opdracht **SET TimeFormat**.

Resultaten

| Voorbeeld | Resultaat |
|---|---|
| `SET TimestampFormat='YYYYMMDD';` | 19831214 |
| `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 |

### Voorbeelden: load-script

Voorbeeld: load-script

In het eerste load-script wordt *SET TimestampFormat='DD/MM/YYYY h:mm:ss[.fff] TT'* gebruikt. In het tweede load-script wordt de opmaak van de tijdstempel gewijzigd in *SET TimestampFormat='MM/DD/YYYY hh:mm:ss[.fff]'*. De verschillende resultaten geven weer hoe de opdracht **SET TimeFormat** werkt met verschillende indelingen voor tijdgegevens.

De onderstaande tabel geeft de gegevensverzameling weer die is gebruikt in de daaropvolgende load-scripts. De tweede kolom van de tabel geeft de opmaak van de verschillende tijdstempels in de gegevensverzameling weer. De eerste vijf tijdstempels volgen de ISO 8601-regels. De zesde niet.

### Gegevensverzameling

*Tabel met de tijdgegevens die zijn gebruikt en de opmaak voor elke tijdstempel in de gegevensverzameling.*

| 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 |

Maak een nieuwe sectie aan in de **Editor voor laden van gegevens**, voeg vervolgens het voorbeeldscript toe en voer het uit. Voeg vervolgens ten minste de velden die in de resultatenkolom staan aan een werkblad in uw app toe om het resultaat te bekijken.

### Load-script

```
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';

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, 0.333333333333333, 2038593, M, Blue
];
```

### Resultaten

*Qlik Sense-tabel met de resultaten van de variabele voor TimestampFormat-interpretatie die wordt gebruikt in het load-script. De laatste tijdstempel in de gegevensverzameling geeft geen correcte datum als resultaat.*

| 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 | - |

Het volgende load-script gebruikt dezelfde gegevensverzameling. Het gebruikt echter *SET TimestampFormat='MM/DD/YYYY hh:mm:ss[.fff]'* voor overeenkomst met de zesde tijdstempel die niet voldoet aan ISO 8601.

Vervang het vorige voorbeeldscript in de **Editor voor laden van gegevens** met het onderstaande en voer het uit. Voeg vervolgens ten minste de velden die in de resultatenkolom staan aan een werkblad in uw app toe om het resultaat te bekijken.

### Load-script

```
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]';

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, 0.333333333333333, 2038593, M, Blue
];
```

**Resultaten**

*Qlik Sense-tabel met de resultaten van de variabele voor TimestampFormat-interpretatie die wordt gebruikt in het load-script.*

| 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 Direct Discovery-variabelen

## Direct Discovery-systeemvariabelen

### DirectCacheSeconds

U kunt een cachelimiet voor de Direct Discovery-queryresultaten instellen voor visualisaties. Wanneer deze tijdslimiet is bereikt, maakt Qlik Sense de cache leeg als nieuwe Direct Discovery-query's worden gemaakt. Qlik Sense zoekt in de brongegevens naar de brongegevens voor de selecties en maakt de cache opnieuw voor de aangegeven tijdslimiet. Het resultaat voor elke combinatie van selecties wordt apart in de cache opgenomen. Met andere woorden, de cache wordt voor elke selectie apart vernieuwd, zodat de ene selectie de cache alleen vernieuwt voor de geselecteerde velden en een tweede selectie de cache vernieuwt voor de relevante velden. Als de tweede selectie velden bevat die zijn vernieuwd in de eerste selectie, worden zij niet opnieuw bijgewerkt in de cache als de cachelimiet nog niet is bereikt.

De Direct Discovery-cache is niet van toepassing op **tabel**visualisaties. Bij tabelselecties wordt telkens een query op de gegevensbron uitgevoerd.

De grenswaarde moet worden ingesteld in seconden. De standaard cachelimiet is 1800 seconden (30 minuten).

De waarde die wordt gebruikt voor **DirectCacheSeconds** is de waarde die is ingesteld tijdens de uitvoering van de **DIRECT QUERY**-opdracht. De waarde kan niet worden gewijzigd tijdens de runtime.

**Voorbeeld:**

```
SET DirectCacheSeconds=1800;
```

### DirectConnectionMax

U kunt asynchrone, parallelle aanroepen uitvoeren op de database door gebruik te maken van de voorziening voor verbindingsgroepering. De syntaxis van het laadscript voor het instellen van de groeperingsvoorziening is als volgt:

```
SET DirectConnectionMax=10;
```

De numerieke instelling geeft het maximumaantal databaseverbindingen op dat door de Direct Discovery-code moet worden gebruikt tijdens het bijwerken van een werkblad. De standaardinstelling is 1.

> *Deze variabele moet met zorg worden gebruikt. Bekend is dat bij instelling op een waarde groter dan 1 problemen ontstaan bij het maken van een verbinding met Microsoft SQL Server.*

**DirectUnicodeStrings**
Direct Discovery kan ondersteuning bieden voorde selectie van uitgebreide Unicode-gegevens door gebruik te maken van de SQL-standaardindeling voor uitgebreide tekenreeksliterals (N'<uitgebreide tekenreeks>') zoals vereist bij sommige databases (met name SQL Server). Het gebruik van deze syntaxis kan worden ingeschakeld voor Direct Discovery met de scriptvariabele **DirectUnicodeStrings**.

Als deze variabele wordt ingesteld op 'waar', wordt het gebruik van de standaard ANSI-markering voor brede tekens "N" vóór de tekenreeksliterals mogelijk gemaakt. Niet alle databases ondersteunen deze standaard. De standaardinstelling is 'false'.

**DirectDistinctSupport**
Als een **DIMENSION**-veldwaarde wordt geselecteerd in een Qlik Sense-object, wordt een query gegenereerd voor de brondatabase. Wanneer de query groepering vereist, gebruikt Direct Discovery het trefwoord **DISTINCT** om alleen unieke waarden te selecteren. Sommige databases vereisen echter het trefwoord **GROUP BY**. Stel **DirectDistinctSupport** in op 'false' om **GROUP BY** te genereren in plaats van **DISTINCT** in query's voor unieke waarden.

```
SET DirectDistinctSupport='false';
```
If DirectDistinctSupport is ingesteld op true, wordt **DISTINCT** gebruikt. Als deze niet is ingesteld, wordt standaard gebruikgemaakt van **DISTINCT**.

**DirectEnableSubquery**
In scenario's met meerdere tabellen met hoge cardinaliteit, is het mogelijk om subquery's te genereren in de SQL-query in plaats van een grote IN-clausule te genereren. Dit wordt geactiveerd door **DirectEnableSubquery** in te stellen op 'true'. De standaardwaarde is 'false'.

> *Als **DirectEnableSubquery** is ingeschakeld, kunt u geen tabellen laden die zich niet in de Direct Discovery-modus bevinden.*

```
SET DirectEnableSubquery='true';
```

## Teradata-variabelen voor query banding

Teradata-query banding is een functie waarbij ondernemingstoepassingen in staat worden gesteld samen te werken met de onderliggende Teradata-database om betere boekhoudvoorzieningen, prioriteitsstelling en werklastbeheer te bieden. Via query banding kunt u metagegevens, zoals gebruikersreferenties, aan een query koppelen.

Er zijn twee variabelen beschikbaar. Beide zijn tekenreeksen die worden geëvalueerd en naar de database verzonden.

SQLSessionPrefix

Deze tekenreeks wordt verzonden wanneer er een verbinding met de database wordt gemaakt.

```
SET SQLSessionPrefix = 'SET QUERY_BAND = ' & Chr(39) & 'Who=' & OSuser() & ';' & Chr(39) & '
FOR SESSION;';
```

Als **OSuser()** bijvoorbeeld *WA\sbt* retourneert, wordt deze geëvalueerd tot `SET QUERY_BAND = 'Who=WA\sbt;' FOR SESSION;`, die naar de database wordt verzonden bij het tot stand brengen van de verbinding.

SQLQueryPrefix

Deze tekenreeks wordt verzonden voor elke afzonderlijke query.

```
SET SQLSessionPrefix = 'SET QUERY_BAND = ' & Chr(39) & 'Who=' & OSuser() & ';' & Chr(39) & '
FOR TRANSACTION;';
```

## Direct Discovery-tekenvariabelen

### DirectFieldColumnDelimiter

U kunt het teken dat wordt gebruikt als veldscheidingsteken instellen in **Direct Query**-opdrachten voor databases die een ander teken dan een komma vereisen als veldscheidingsteken. Het opgegeven teken moet worden omringd door enkele aanhalingstekens in de **SET**-instructie.

```
SET DirectFieldColumnDelimiter= '|'
```

### DirectStringQuoteChar

U kunt een teken gebruiken om tekenreeksen van aanhalingstekens te voorzien in een gegenereerde query. De standaard is een enkel aanhalingsteken. Het opgegeven teken moet worden omringd door enkele aanhalingstekens in de **SET**-instructie.

```
SET DirectStringQuoteChar= '"';
```

### DirectIdentifierQuoteStyle

U kunt opgeven dat niet-ANSI-aanhalingstekens worden gebruikt voor id's in gegenereerde query's. Momenteel vormen GoogleBQ-aanhalingstekens de enige niet-ANSI-aanhalingstekens die beschikbaar zijn. De standaardinstelling is ANSI. Er kan gebruik worden gemaakt van hoofdletters, kleine letters en gemengde tekens (ANSI, ansi, Ansi).

```
SET DirectIdentifierQuoteStyle="GoogleBQ";
```

Zo wordt bijvoorbeeld gebruikgemaakt van ANSI-aanhalingstekens in de volgende **SELECT**-opdracht:

```
SELECT [Quarter] FROM [qvTest].[sales] GROUP BY [Quarter]
```

Als **DirectIdentifierQuoteStyle** is ingesteld op "GoogleBQ", wordt in de **SELECT**-opdracht als volgt gebruikgemaakt van aanhalingstekens:

```
SELECT [Quarter] FROM [qvTest.sales] GROUP BY [Quarter]
```

### DirectIdentifierQuoteChar

U kunt een teken gebruiken het gebruik van aanhalingstekens voor id's in een gegenereerde query te besturen. Deze kan worden ingesteld op één teken (zoals een dubbel aanhalingsteken) of twee tekens (zoals een stel vierkante haken). De standaard is een dubbel aanhalingsteken.

```
SET DirectIdentifierQuoteChar='[]';
```

```
SET DirectIdentifierQuoteChar='``';

SET DirectIdentifierQuoteChar=' ';

SET DirectIdentifierQuoteChar='""';
```

### DirectTableBoxListThreshold

Als Direct Discovery-velden worden gebruikt in een **Table**visualisatie, wordt een drempelwaarde ingesteld om het aantal weergegeven rijen te beperken. De standaard drempelwaarde is 1000. De instelling voor de standaard drempelwaarde kan worden gewijzigd door de variabele **DirectTableBoxListThreshold** in te stellen in het load-script. Bijvoorbeeld:

```
SET DirectTableBoxListThreshold=5000;
```

De ingestelde drempelwaarde geldt alleen voor **Tabel**visualisaties die Direct Discovery-velden bevatten. **Tabel**visualisaties die alleen velden in het geheugen bevatten worden niet beperkt door de instelling **DirectTableBoxListThreshold**.

Er worden geen velden weergegeven in de **Tabel**visualisatie totdat de selectie minder records heeft dat de drempelwaarde.

## Direct Discovery-variabelen voor getalinterpretatie

### DirectMoneyDecimalSep

Het gedefinieerde decimaalscheidingsteken vervangt het decimaalteken voor valuta in de SQL-opdracht die wordt gegenereerd voor het laden van gegevens met Direct Discovery. Dit teken moet overeenkomen met het teken dat wordt gebruikt in **DirectMoneyFormat**.

Standaardwaarde is `'.'`

**Voorbeeld:**

```
Set DirectMoneyDecimalSep='.';
```

### DirectMoneyFormat

Het gedefinieerde teken vervangt de valutanotatie in de SQL-opdracht die wordt gegenereerd voor het laden van gegevens met Direct Discovery. Het valutateken voor het scheidingsteken voor duizendtallen moet niet worden opgenomen.

Standaardwaarde is `'#.0000'`

**Voorbeeld:**

```
Set DirectMoneyFormat='#.0000';
```

### DirectTimeFormat

De gedefinieerde tijdnotatie vervangt de tijdnotatie in de SQL-opdracht die wordt gegenereerd voor het laden van gegevens met Direct Discovery.

**Voorbeeld:**

```
Set DirectTimeFormat='hh:mm:ss';
```

**DirectDateFormat**
De gedefinieerde datumnotatie vervangt de datumnotatie in de SQL-opdracht die wordt gegenereerd voor het laden van gegevens met Direct Discovery.

**Voorbeeld:**

```
Set DirectDateFormat='MM/DD/YYYY';
```

**DirectTimeStampFormat**
De gedefinieerde indeling vervangt de datum- en tijdnotatie in het SQL-opdracht die is gegenereerd in het SQL-opdracht voor het laden van gegevens laden met behulp van Direct Discovery.

**Voorbeeld:**

```
Set DirectTimestampFormat='M/D/YY hh:mm:ss[.fff]';
```

## 4.10 Foutvariabelen

Na uitvoering van het script zijn alle foutvariabelen voorzien van een waarde. De eerste variabele, ErrorMode, wordt ingevoerd door de gebruiker en de laatste drie door Qlik Sense met informatie over fouten in het script.

### Overzicht van foutvariabelen

Elke variabele wordt na het overzicht verder omschreven. U kunt ook in de syntaxis op de variabelenaam klikken om meteen naar de details van die specifieke variabele te gaan.

Raadpleeg de Qlik Sense online Help voor nadere details over de variabele.

**ErrorMode**
Deze foutvariabele bepaalt welke actie in Qlik Sense wordt uitgevoerd als bij uitvoering van het script een fout optreedt.

```
ErrorMode
```

**ScriptError**
Deze foutvariabele retourneert de foutcode van de laatst uitgevoerde scriptopdracht.

```
ScriptError
```

**ScriptErrorCount**
De foutvariabele retourneert het totale aantal opdrachten die fouten hebben veroorzaakt bij de uitvoering van het huidige script. Deze variabele wordt voorafgaand aan de uitvoering van het script altijd op de beginwaarde 0 ingesteld.

```
ScriptErrorCount
```

**ScriptErrorList**
Deze foutvariabele bevat een samengestelde lijst van alle scriptfouten die tijdens de laatste uitvoering van het script zijn opgetreden. Elke fout staat op een nieuwe regel.

**ScriptErrorList**

## ErrorMode

Deze foutvariabele bepaalt welke actie in Qlik Sense wordt uitgevoerd als bij uitvoering van het script een fout optreedt.

**Syntaxis:**

```
ErrorMode
```

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| **ErrorMode=1** | De standaardinstelling. De uitvoering van het script wordt gestopt en de gebruiker wordt gevraagd om actie te ondernemen (niet-batchmodus). |
| **ErrorMode =0** | De fout wordt simpelweg genegeerd in Qlik Sense en het script wordt verder uitgevoerd bij de volgende scriptopdracht. |
| **ErrorMode =2** | In Qlik Sense wordt een foutbericht "Uitvoering van script is mislukt..." gegenereerd zodra de fout optreedt, zonder dat de gebruiker eerst wordt gevraagd om actie te ondernemen. |

**Voorbeeld:**

```
set ErrorMode=0;
```

## ScriptError

Deze foutvariabele retourneert de foutcode van de laatst uitgevoerde scriptopdracht.

**Syntaxis:**

```
ScriptError
```

Elke keer dat de scriptopdracht correct is uitgevoerd, wordt deze variabele teruggezet op 0. Als een fout optreedt, wordt een interne Qlik Sense-foutcode ingesteld. Foutcodes zijn dubbele waarden met een numerieke component en een tekstcomponent. De volgende foutcodes komen voor:

Foutcodes van scripts

| Foutcode | Beschrijving |
|---|---|
| 0 | Geen fout. Dubbele waardetekst is leeg. |
| 1 | Algemene fout. |
| 2 | Syntaxisfout. |

| Foutcode | Beschrijving |
|---|---|
| 3 | Algemene ODBC-fout. |
| 4 | Algemene OLE DB-fout. |
| 5 | Algemene aangepaste databasefout. |
| 6 | Algemene XML-fout. |
| 7 | Algemene HTML-fout. |
| 8 | Bestand niet gevonden. |
| 9 | Database niet gevonden. |
| 10 | Tabel niet gevonden. |
| 11 | Veld niet gevonden. |
| 12 | Bestand heeft verkeerde indeling. |
| 16 | Semantische fout. |

**Voorbeeld:**

```
set ErrorMode=0;

LOAD * from abc.qvf;

if ScriptError=8 then

exit script;

//no file;

end if
```

## ScriptErrorCount

De foutvariabele retourneert het totale aantal opdrachten die fouten hebben veroorzaakt bij de uitvoering van het huidige script. Deze variabele wordt voorafgaand aan de uitvoering van het script altijd op de beginwaarde 0 ingesteld.

**Syntaxis:**

```
ScriptErrorCount
```

## ScriptErrorList

Deze foutvariabele bevat een samengestelde lijst van alle scriptfouten die tijdens de laatste uitvoering van het script zijn opgetreden. Elke fout staat op een nieuwe regel.

Syntaxis:

```
ScriptErrorList
```

# 5 Scriptuitdrukkingen

Uitdrukkingen kunnen zowel in **LOAD**- als in **SELECT**-opdrachten worden gebruikt. De hier beschreven syntaxis en functies zijn van toepassing op de **LOAD**-opdracht en niet op de **SELECT**-opdracht, omdat deze laatste wordt geïnterpreteerd door het ODBC-stuurprogramma en niet door Qlik Sense. De meeste ODBC-stuurprogramma's zijn echter in staat een aantal van de onderstaande functies te interpreteren.

Uitdrukkingen bestaan uit functies, velden en operatoren die worden gecombineerd volgens een speciale syntaxis.

Alle uitdrukkingen in een Qlik Sense-script retourneren, afhankelijk van wat van toepassing is, een getal en/of een tekenreeks. Logische functies en operators leveren 0 op voor False en -1 voor True. Conversies van getal naar tekenreeks en vice versa zijn daarbij geïmpliceerd. Logische operatoren en functies interpreteren 0 als False en al het andere als True.

De algemene syntaxis voor een uitdrukking is:

Algemene syntaxis

| Uitdrukking | Velden | Operator |
|---|---|---|
| expression ::= (constant | constant | \| |
| expression ::= (constant | fieldref | \| |
| expression ::= (constant | operator1 expression | \| |
| expression ::= (constant | expression operator2 expression | \| |
| expression ::= (constant | function | \| |
| expression ::= (constant | ( expression ) | ) |

waarbij:

- **constant** een tekenreeks (tekst, datum of tijd) is die tussen enkele rechte aanhalingstekens staat, of een getal. Constanten worden geschreven zonder scheidingsteken voor duizendtallen en met een punt als scheidingsteken voor decimalen.
- **fieldref** is een veldnaam van de geladen tabel.
- **operator1** is een unaire operator (werkt op één uitdrukking, de uitdrukking rechts van de operator).
- **operator2** is een binaire operator (werkt op twee uitdrukkingen, een aan elke zijde).
- **function ::= functionname( parameters)**
- **parameters ::= expression { , expression }**

Het aantal en de typen parameters zijn niet willekeurig. Deze zijn afhankelijk van de gebruikte functie.

Uitdrukkingen en functies kunnen naar believen worden genest, en zolang de uitdrukking een interpreteerbare waarde retourneert, geeft Qlik Sense geen foutberichten weer.

# 6 Diagramuitdrukkingen

Een diagramuitdrukking (visualisatie) is een combinatie van functies, velden en rekenkundige operatoren (+ * / =) en andere metingen. Uitdrukkingen worden gebruikt voor het verwerken van gegevens in de app om een resultaat op te leveren dat kan worden bekeken in een visualisatie. Hun gebruik beperkt zich niet tot metingen. U kunt visualisaties bouwen die dynamischer en krachtiger zijn, met uitdrukkingen voor titels, subtitels, voetnoten en zelfs dimensies.

Dit betekent bijvoorbeeld dat de titel van een visualisatie geen statische tekst hoeft te zijn, maar kan worden gemaakt op basis van een uitdrukking waarvan het resultaat verandert afhankelijk van de uitgevoerde selecties.

*Voor gedetailleerde naslaginformatie over scriptfuncties en diagramfuncties, raadpleegt u de Scriptsyntaxis en diagramfuncties.*

## 6.1 Het aggregatiebereik definiëren

Er zijn gewoonlijk twee factoren die samen bepalen welke records worden gebruikt voor het definiëren van de waarde van aggregatie in een uitdrukking. Bij het werken in visualisaties zijn deze factoren:

- Dimensionale waarde (bij aggregatie in een diagramuitdrukking)
- Selecties

Samen definiëren deze factoren het bereik van de aggregatie. U komt mogelijk situaties tegen waarbij u de selectie, de dimensie of beide buiten beschouwing wilt laten bij de berekening. In diagramfuncties kunt u dit bereiken door de kwalificatie TOTAL, set-analyse of een combinatie van beide te gebruiken. Samengevat:

- Om de dimensionale groepering te negeren of aan te passen: gebruik de TOTAL-kwalificaties
- De selectie negeren of aanpassen: gebruik set-analyse

Aggregatie: methode en beschrijving

| Methode | Beschrijving |
|---|---|
| Kwalificatie TOTAL | Bij gebruik van de kwalificatie TOTAL binnen uw aggregatiefunctie wordt de dimensionale waarde buiten beschouwing gelaten.<br><br>De total-kwalificatie kan binnen een aggregatie worden gebruikt om het bereik van de aggregatie opnieuw te definiëren door de dimensionale waarden te negeren. De aggregatie wordt in plaats daarvan uitgevoerd op alle mogelijke veldwaarden.<br><br>Na de kwalificatie **TOTAL** kan een lijst met een of meer veldnamen tussen punthaken worden opgegeven.<br><br>Bijvoorbeeld:<br><br>`Sum(TOTAL <Quarter> {<Year={2013}, Quarter={'Q2'}>} Amount)`<br><br>Deze veldnamen moeten een subset zijn van de diagramdimensies. In dit geval wordt bij de berekening geen rekening gehouden met de diagramdimensies met uitzondering van de aangegeven dimensievariabelen. Er wordt één waarde geretourneerd voor elke combinatie van veldwaarden in de opgegeven dimensievelden. Ook velden die momenteel niet een dimensie zijn in een diagram, kunnen worden opgenomen in de lijst. Dit kan nuttig zijn bij groepsdimensies waarbij de dimensievelden niet vast zijn. Als alle dimensies in de groep worden aangegeven, geeft de functie consistente aggregatieresultaten als het drill-downniveau verandert.<br><br>Zie *Voorbeeld: TOTAL-kwalificatie met uitzondering, en set-analyse (page 302)* voor een voorbeeld van het toevoegen van uitzonderingen tussen punthaken. |
| Set-analyse | Bij gebruik van set-analyse binnen uw aggregatie wordt de selectie buiten beschouwing gelaten. De aggregatie wordt uitgevoerd op alle waarden verdeeld over de dimensies. |
| TOTAL-kwalificatie en set-analyse | Bij gebruik van de **TOTAL** -kwalificatie en set-analyse binnen uw aggregatie worden de selectie en de dimensies buiten beschouwing gelaten, met uitzondering van de dimensies die binnen punthaken staan na de **TOTAL** -kwalificatie |
| ALL -kwalificatie | Bij gebruik van de **ALL**-kwalificatie binnen uw aggregatie worden de selectie en de dimensies buiten beschouwing gelaten. Het equivalent kan worden bereikt met de instructie {1} voor set-analyse en de **TOTAL** -kwalificatie:<br><br>`=sum(All Sales)`<br><br>`=sum({1} Total Sales)` |

## Voorbeeld: TOTAL-kwalificatie

In het volgende voorbeeld wordt getoond hoe TOTAL kan worden gebruikt voor het berekenen van een relatief aandeel. Ervan uitgaande dat Q2 is geselecteerd, wordt met TOTAL de som van alle waarden berekend, waarbij de dimensies buiten beschouwing worden gelaten.

Voorbeeld: kwalificatie total

| 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% |

*Als u de getallen wilt weergeven als een percentage, in het eigenschappenvenster, selecteert u voor de meting die u wilt weergeven als een percentagewaarde, onder **Getalnotatie**, de optie **Getal** en kiest u onder **Opmaak** de optie **Eenvoudig** en een van de %-indelingen.*

## Voorbeeld: set-analyse

In het volgende voorbeeld wordt getoond hoe set-analyse kan worden gebruikt om een vergelijking tussen gegevensverzamelingen te maken voordat er iets is geselecteerd. Ervan uitgaande dat Q2 is geselecteerd, wordt via set-analyse met de set-definitie {1} de som van alle waarden berekend, waarbij selecties buiten beschouwing worden gelaten maar de som wordt verdeeld over de dimensies.

Voorbeeld: set-analyse

| 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% |

## Voorbeeld: TOTAL-kwalificatie en set-analyse ({1}-id)

In het volgende voorbeeld wordt getoond hoe set-analyse en de kwalificatie TOTAL kunnen worden gecombineerd om een vergelijking tussen gegevensverzamelingen te maken voordat er iets is geselecteerd en over alle dimensies heen. Ervan uitgaande dat Q2 is geselecteerd, wordt via set-analyse met de set-definitie (1) en de kwalificatie TOTAL de som van alle waarden berekend waarbij eventuele selecties en de dimensies buiten beschouwing worden gelaten.

Voorbeeld: kwalificatie TOTAL en set-analyse

| Year | Quarter | Sum (Amount) | Sum({1} TOTAL Amount) | Sum(Amount)/Sum({1} TOTAL Amount) |
|---|---|---|---|---|
| | | 3000 | 10800 | 27,8% |
| 2012 | Q2 | 1700 | 10800 | 15,7% |
| 2013 | Q2 | 1300 | 10800 | 12% |

## Voorbeeld: TOTAL-kwalificatie en set-analyse (gegevens vergelijken met een referentiewaarde)

Het volgende voorbeeld laat zien hoe u set-analyse en de TOTAL-kwalificatie kunt gebruiken om een referentiekolom in uw tabel te maken voor eenvoudige vergelijking van statistieken op rijniveau. Om de gegevens gemakkelijker te begrijpen en te gebruiken, labelen we de metingkolommen.

De gegevens van het tweede kwartaal van 2013 worden gebruikt als referentie om te vergelijken met alle andere waarden in de tabel.

**Doe het volgende:**

1. Voeg een tabel toe aan uw werkblad.
2. Voeg de volgende velden als dimensies toe:
    - *Year*
    - *Quarter*
3. Voeg de volgende meting toe:
   `Sum(Amount)`
4. In de eigenschappen voor de meting die u zojuist heb toegevoegd, geeft u de meting het volgende label: *Total Sales*
5. Voeg de volgende meting toe:
   `Sum(TOTAL {<Year={2013}, Quarter={'Q2'}>} Amount)`
6. In de eigenschappen voor de meting die u zojuist heb toegevoegd, geeft u de meting het volgende label: *Reference Quarter (2013-Q2) Total Sales*
7. Voeg de volgende meting toe:
   `round((sum(Amount) / sum(total {<Year={2013}, Quarter={'Q2'}>} Amount))*100,'0.01') & '%'`
8. In de eigenschappen voor de meting die u zojuist heb toegevoegd, geeft u de meting het volgende label: *Rounded Quotient - Total Sales Compared to Reference Quarter (2013-Q2)*

Zonder selecties toe te passen, zou uw tabel er als volgt uit moeten zien:

Voorbeeld: TOTAL-kwalificatie en set-analyse (referentiekolom)

| 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% |

## Voorbeeld: TOTAL-kwalificatie met uitzondering, en set-analyse

Het volgende voorbeeld laat zien hoe de TOTAL-kwalificatie gebruikt moet worden, maar met een uitzondering op deze kwalificatie tussen punthaken. De voorbeelduitdrukkingen maken ook gebruik van set-analyse.

**Doe het volgende:**

1. Voeg een tabel toe aan uw werkblad.
2. Voeg de volgende velden als dimensies toe:
    - *Year*
    - *Quarter*
3. Voeg de volgende meting toe:
   `Sum(Amount)`
4. In de eigenschappen voor de meting die u zojuist heb toegevoegd, geeft u de meting het volgende label: *Total Sales*
5. Voeg de volgende meting toe:
   `Sum(TOTAL <Quarter> {<Year={2013}, Quarter={'Q2'}>} Amount)`
6. In de eigenschappen voor de meting die u zojuist heb toegevoegd, geeft u de meting het volgende label: *Reference Quarter (2013-Q2) Total Sales*

Voorbeelden: TOTAL-kwalificatie met uitzondering, en set-analyse

| Year | Quarter | Total Sales | Reference Quarter (2013-Q2) Total Sales |
|---|---|---|---|
| 2012 | Q1 | 1100 | 0 |
| 2012 | Q2 | 1700 | 1300 |
| 2012 | Q3 | 1400 | 0 |

| Year | Quarter | Total Sales | Reference Quarter (2013-Q2) Total Sales |
|---|---|---|---|
| 2012 | Q4 | 1800 | 0 |
| 2013 | Q1 | 1000 | 0 |
| 2013 | Q2 | 1300 | 1300 |
| 2013 | Q3 | 1100 | 0 |
| 2013 | Q4 | 1400 | 0 |

De uitdrukking van de referentiekolom wordt als volgt geïnterpreteerd:

- De set-uitdrukking `{<Year={2013}, Quarter={'Q2'}>}` beperkt de invoergegevens tot records met een `Year` van 2013 en een `Quarter` van Q2.
- De `TOTAL`-kwalificatie zorgt ervoor dat alle dimensies in de tabel genegeerd worden. Met de `<Quarter>`-specificatie wordt echter een uitzondering gemaakt voor de dimensie `Quarter`.

### Gegevens die worden gebruikt in voorbeelden op deze pagina

Gegevens die worden gebruikt in voorbeelden:

```
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 Set-analyse

Als u een selectie maakt in een app, definieert u een subset van records in de gegevens. Aggregatiefuncties zoals `Sum()`, `Max()`, `Min()`, `Avg()` en `Count()` worden berekend op basis van deze subset.

Dit betekent dat uw selectie het bereik van de aggregatie definieert; deze definieert de set met records op basis waarvan berekeningen worden uitgevoerd.

Set-analyse biedt een manier om een bereik te definiëren dat verschilt van de set met records die wordt gedefinieerd door de huidige selectie. Dit nieuwe bereik kan ook als een alternatieve selectie worden beschouwd.

Dit kan nuttig zijn als u de huidige selectie wilt vergelijken met een specifieke waarde, bijvoorbeeld een waarde van vorig jaar of het wereldwijde marktaandeel.

## Ingestelde uitdrukkingen

Set-uitdrukkingen kunnen binnen en buiten aggregatiefuncties worden gebruikt en worden tussen accolades geplaatst.

**Voorbeeld: Binnenste set-uitdrukking**

```
Sum( {$<Year={2021}>} Sales )
```

**Voorbeeld: Buitenste set-uitdrukking**

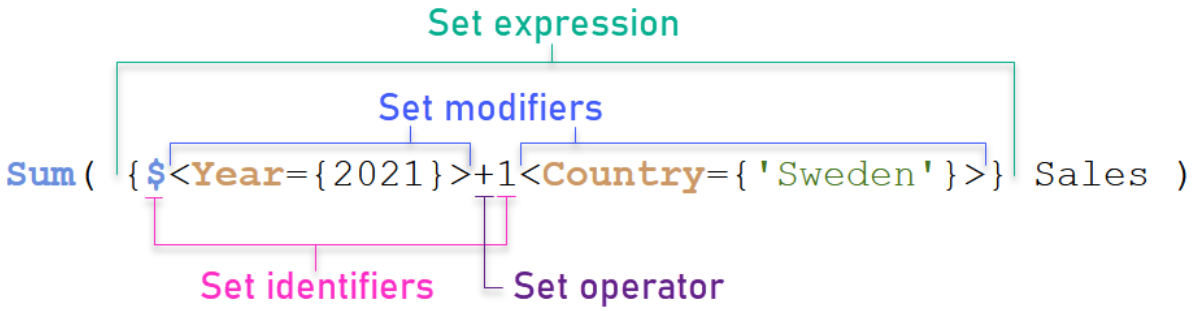
```
{<Year={2021}>} Sum(Sales) / Count(distinct Customer)
```

Een set-uitdrukking bestaat uit een combinatie van de volgende elementen:

- Id's Een set-id vertegenwoordigt een selectie die ergens anders is gedefinieerd. Deze vertegenwoordigt ook een specifieke set met records in de gegevens. Dit kan de huidige selectie, een selectie van een bladwijzer of een selectie van een alternatieve status zijn. Een simpele set-uitdrukking bestaat uit een enkele id, zoals een dollarteken, {$}, oftewel alle records in de huidige selectie.
  Voorbeelden: `$`, `1`, `BookMark1`, `State2`
- Operatoren Een set-operator kan worden gebruikt om samenvoegingen, verschillen of snijpunten tussen verschillende set-id's te maken. Op deze manier kunt u een subset of een superset maken van de selecties die zijn gedefinieerd door de set-id's.
  Voorbeelden: `+`, `-`, `*`, `/`
- **Modificaties**. Een set-modificatie kan worden toegevoegd aan de set-id om de selectie te wijzigen. Een modicatie kan ook afzonderlijk worden gebruikt om de standaard-id te wijzigen. Een modificatie moet tussen punthaken `<...>` worden geplaatst.
  Voorbeelden: `<Year={2020}>`, `<Supplier={ACME}>`
- **Vlag voor behoud van lege set**. Een geavanceerd element dat kan worden gebruikt in elke buitenste setuitdrukking, behalve de laatste, in een keten van buitenste setuitdrukkingen. Deze vlag heft het impliciet wissen op van sets die gedefinieerd zijn in deze buitenste setuitdrukking, waardoor de configuratie van geavanceerde uitdrukkingen vereenvoudigd wordt.
  Syntaxis: `&`

De elementen worden gecombineerd tot set-uitdrukkingen.

*Elementen in een set-uitdrukking*




De bovenstaande set-uitdrukking is bijvoorbeeld gebaseerd op de aggregatie `Sum(Sales)`.

De eerste operand retourneert verkopen voor het jaar `2021` van de huidige selectie. Dit wordt aangegeven door de set-id `$` en de modificatie die een selectie van het jaar `2021` bevat. De tweede operand retourneert `Sales` voor `Sweden` en negeert de huidige selectie, die wordt aangegeven door de set-id `1`.

Tot slot retourneert de uitdrukking een set met records die toebehoort aan een van de twee set-operanden, zoals wordt aangegeven door de set-operator `+`.

## Voorbeelden

Voorbeelden waarin de bovenstaande elementen van de set-uitdrukking worden gecombineerd, zijn beschikbaar in de volgende onderwerpen:

## Natuurlijke sets

Een set-uitdrukking vertegenwoordigt over het algemeen zowel een set met records in het gegevensmodel als een selectie die deze subset van gegevens definieert. In dit geval wordt de set een natuurlijke set genoemd.

Set-id's, met of zonder set-modificaties, vertegenwoordigen altijd natuurlijke sets.

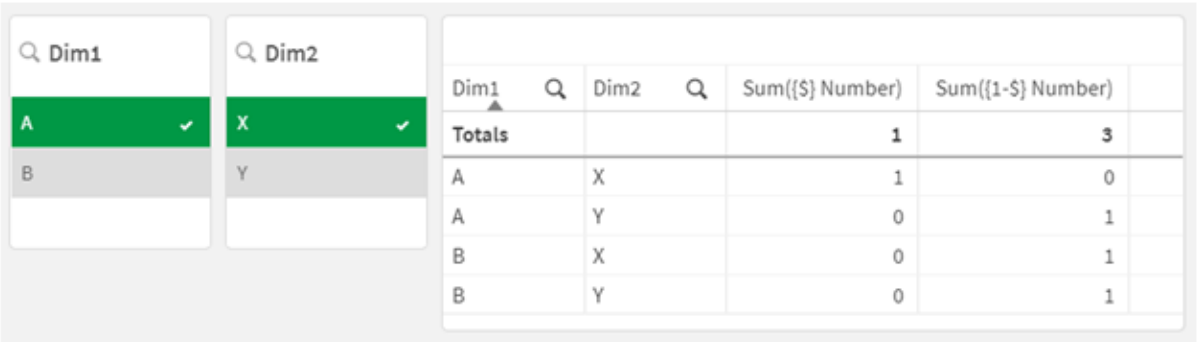
Een set-uitdrukking die gebruikmaakt van set-operators vertegenwoordigt ook een subset van de records, maar deze kan over het algemeen nog steeds niet worden omschreven met behulp van selecties met veldwaarden. Een dergelijke uitdrukking wordt een niet-natuurlijke set genoemd.

De set die wordt verstrekt door `{1-$}` kan niet altijd worden gedefinieerd door een selectie. Dit is daarom geen natuurlijke set. Dit kunt u zien door de volgende gegevens te laden, deze aan een tabel toe te voegen en dan selecties te maken met behulp van filtervakken.

```
Load * Inline
[Dim1, Dim2, Number
A, X, 1
A, Y, 1
B, X, 1
B, Y, 1];
```

Als u selecties maakt voor `Dim1` en `Dim2` ziet u de weergave zoals in de onderstaande tabel.

*Tabel met natuurlijke en niet-natuurlijke sets*

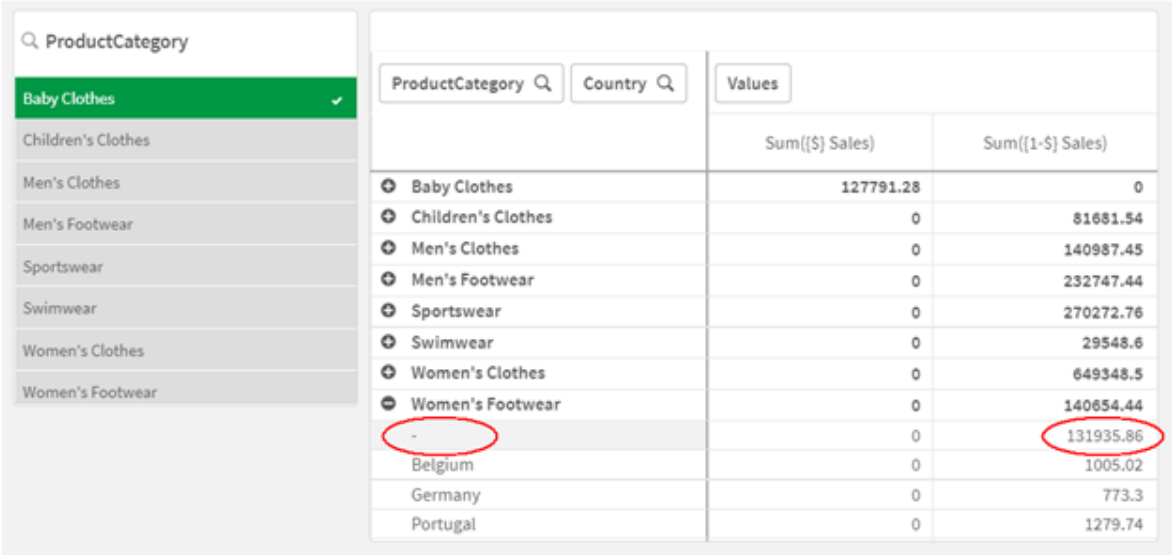



De set-uitdrukking in de eerste meting gebruikt een natuurlijke set: deze komt overeen met de selectie die is gemaakt `{$}`.

De tweede meting is anders. Deze gebruikt `{1-$}`. Het is niet mogelijk om een selectie te maken die overeenkomt met deze set, daarom is dit een niet-natuurlijke set.

Dit onderscheid heeft een aantal gevolgen:

- Set-modificaties kunnen alleen worden toegepast op set-id's. Ze kunnen niet worden toegepast op een willekeurige set-uitdrukking. Het is bijvoorbeeld niet mogelijk om de volgende set-uitdrukking te gebruiken:
  `{ (BM01 * BM02) <Field={x,y}> }`
  Hier geven de normale (ronde) haken aan dat de snijpunten tussen `BM01` en `BM02` geëvalueerd moeten worden voordat de set-modificatie wordt toegepast. De reden is dat er geen elementset aanwezig is die kan worden gewijzigd.
- U kunt geen niet-natuurlijke set gebruiken in de elementfuncties van `P()` en `E()`. Deze functies retourneren een elementset, maar het is niet mogelijk om de elementset af te leiden van een niet-natuurlijke set.
- Aan een meting die een niet-natuurlijke set gebruikt kan niet altijd de juiste dimensiewaarde worden toegewezen als het gegevensmodel veel tabellen bevat. In het volgende diagram worden enkele uitgezonderde verkoopcijfers bijvoorbeeld toegewezen aan de correcte Country, terwijl anderen NULL en Country bevatten.
  *Diagram met een niet-natuurlijke set*




  Of de toewijzing correct wordt uitgevoerd is afhankelijk van het gegevensmodel. In dit geval kan het aantal niet worden toegewezen als het betrekking heeft op een land dat niet in de selectie is opgenomen.

| Id | Beschrijving |
|---|---|
| 1 | Vertegenwoordigt de volledige set van alle records in de toepassing, ongeacht eventueel uitgevoerde selecties. |

| Id | Beschrijving |
|---|---|
| $ | Vertegenwoordigt de records van de huidige selectie. De set-uitdrukking **{$}** staat dus gelijk aan het niet aangeven van een set-uitdrukking. |
| $1 | Vertegenwoordigt de vorige selectie. $2 vertegenwoordigt de vorige selectie op één na, enzovoort. |
| $_1 | Vertegenwoordigt de volgende (voorwaartse) selectie. $_2 vertegenwoordigt de volgende selectie min één, enzovoort. |
| BM01 | U kunt elke gewenste bladwijzer-id of bladwijzernaam gebruiken. |
| MyAltState | U kunt verwijzen naar de selecties die zijn uitgevoerd in een alternatieve state met behulp van de naam van de state. |

| Voorbeeld | Resultaat |
|---|---|
| sum ({1} Sales) | Retourneert de totale verkopen voor de app, waarbij selecties wel wordt genegeerd maar de dimensie niet. |
| sum ({$} Sales) | Retourneert de verkopen voor de huidige selectie, oftewel hetzelfde als sum(Sales). |
| sum ({$1} Sales) | Retourneert de verkopen voor de vorige selectie. |
| sum ({BM01} Sales) | Retourneert de verkopen voor de bladwijzer met de naam *BM01*. |

| Voorbeeld | Resultaat |
|---|---|
| sum({$<OrderDate = DeliveryDate>} Sales) | Retourneert de verkopen voor de huidige selectie waarbij OrderDate = DeliveryDate. |
| sum({1<Region = {US}>} Sales) | Retourneert de verkopen voor de regio US, waarbij de huidige selectie buiten beschouwing wordt gelaten. |
| sum({$<Region = >} Sales) | Retourneert de verkopen voor de selectie, maar met de selectie in *Region* verwijderd. |
| sum({<Region = >} Sales) | Retourneert hetzelfde als in het voorbeeld hierboven. Wanneer de te modificeren set wordt weggelaten, wordt $ verondersteld. |
| sum({$<Year={2000}, Region={“U*”}>} Sales) | Retourneert de verkopen voor de huidige selectie, maar met nieuwe selecties in zowel *Year* als *Region*. |

## Set-Id's

Een set-id vertegenwoordigt een set met records in de gegevens. Dit kunnen alle gegevens zijn of een subset van de gegevens. Het is een set met records die door een selectie wordt gedefinieerd. Dit kan de huidige selectie, alle gegevens (geen selectie), een selectie van een bladwijzer of een selectie van een alternatieve status zijn.

In het voorbeeld `Sum( {$<Year = {2009}>} Sales )` is de id het dollarteken: `$`. Deze vertegenwoordigt de huidige selectie. Maar ook alle mogelijke records. Deze set kan vervolgens worden gewijzigd door het modificatiegedeelte van de set-uitdrukking: de selectie `2009` in `Year` wordt toegevoegd.

De $ set-id is hetzelfde als geen set-id vermelden. Door het voorbeeld hierboven is de uitdrukking `Sum( {$<Year = {2009}>} Sales )` bijvoorbeeld gelijk aan `Sum( {<Year = {2009}>} Sales )`.

In een complexere set-uitdrukking kunnen twee id's samen met een operator worden gebruikt om een samenvoeging, een verschil of een snijpunt van twee recordsets te vormen.

De volgende tabel bevat een aantal algemene id's.

Voorbeelden met algemene id's

| Id | Beschrijving |
|---|---|
| `1` | Vertegenwoordigt de volledige set van alle records in de toepassing, ongeacht eventueel uitgevoerde selecties. |
| `$` (of geen set-id) | Vertegenwoordigt de records van de huidige selectie in de standaard-state. De set-uitdrukking `{$}` staat dus meestal gelijk aan het niet aangeven van een set-uitdrukking. |
| `$1` | Vertegenwoordigt de vorige selectie in de standaard-state. `$2` vertegenwoordigt de vorige selectie min één, enzovoort. |
| `$_1` | Vertegenwoordigt de volgende (vooruit) selectie. `$_2` vertegenwoordigt de volgende selectie min één, enzovoort. |
| `BM01` | U kunt elke gewenste bladwijzer-id of bladwijzernaam gebruiken. |
| `AltState` | U kunt naar een alternatieve state verwijzen door middel van de naam van de state. |
| `AltState::BM01` | Een bladwijzer bevat de selecties van alle states en u kunt naar een specifieke bladwijzer verwijzen door de bladwijzernaam in aanmerking te nemen. |

De volgende tabel bevat voorbeelden met verschillende id's.

Voorbeelden met verschillende id's

| Voorbeeld | Resultaat |
|---|---|
| `Sum ({1} Sales)` | Retourneert de totale verkopen voor de app, waarbij selecties wel wordt genegeerd maar de dimensie niet. |
| `Sum ({$} Sales)` | Retourneert de verkopen voor de huidige selectie, oftewel hetzelfde als `Sum(Sales)`. |
| `Sum ({$1} Sales)` | Retourneert de verkopen voor de vorige selectie. |
| `Sum ({BM01} Sales)` | Retourneert de verkopen voor de bladwijzer met de naam `BM01`. |

## Set-operatoren

Set-operatoren worden gebruikt voor het opnemen, uitsluiten of het doorkruisen van gegevensverzamelingen. Alle operatoren gebruiken sets als operanden en leveren een set op als resultaat.

U kunt set-operatoren in twee verschillende situaties gebruiken:

- Om de bewerking Set uit te voeren voor set-id's die een set met records in de gegevens vertegenwoordigen.
- Om de bewerking Set uit te voeren voor elementsets, voor de veldwaarden of in een set-modificatie.

De volgende tabel bevat de operatoren die kunnen worden gebruikt in set-uitdrukkingen.

Operatoren

| Operator | Beschrijving |
|---|---|
| + | Samenvoegen. Deze binaire bewerking retourneert een set die bestaat uit de records of elementen die horen bij een van de twee set-operanden. |
| - | Uitsluiten. Deze binaire bewerking retourneert een set records of elementen die horen bij de eerste maar niet bij de andere van de twee set-operanden. Daarnaast levert het de complementerende set op als het wordt gebruikt als unaire operator. |
| * | Doorsnede. Deze binaire bewerking retourneert een set die bestaat uit de records of elementen die horen bij beide set-operanden. |
| / | Symmetrisch verschil (XOR). Deze binaire bewerking retourneert een set die bestaat uit de records of elementen die horen bij een van de twee set-operanden, maar niet bij beide set-operanden. |

De volgende tabel bevat voorbeelden met operatoren.

Voorbeelden met operatoren

| Voorbeeld | Resultaat |
|---|---|
| `Sum ({1-$} Sales)` | Retourneert de verkoop voor alles behalve de huidige selectie. |
| `Sum ({$*BM01} Sales)` | Retourneert de verkoop voor de gemeenschappelijke elementen voor de selectie en bladwijzer #160;`BM01`. |
| `Sum ({-($+BM01)} Sales)` | Retourneert de verkoop voor alles behalve de selectie en bladwijzer `BM01`. |
| `Sum ({$<Year={2009}>+1<Country={'Sweden'}>} Sales)` | Retourneert de verkopen voor het jaar 2009 die zijn gekoppeld aan de huidige selecties en voegt de volledige verzameling gegevens die zijn gekoppeld aan het land `Sweden` voor alle jaren toe. |
| `Sum ({$<Country={"S*"}+{"*land"}>} Sales)` | Retourneert de verkopen voor landen die beginnen met `S` of eindigen met `land`. |

## Set-modificaties

Set-uitdrukkingen worden gebruikt om het bereik van een berekening te definiëren. Het belangrijkste onderdeel van de set-uitdrukking is de set-modificator die een selectie definieert. Deze wordt gebruik om de gebruikersselectie of de selectie in de set-id te wijzigen. Het resultaat definieert een nieuw bereik voor de berekening.

De set-modificatie bestaat uit een of meer veldnamen, elk gevolgd door een selectie die op het veld wordt toegepast. De modificatie is omsloten door punthaken: < >

Bijvoorbeeld:

- `Sum ( {$<Year = {2015}>} Sales )`
- `Count ( {1<Country = {Germany}>} distinct OrderID )`
- `Sum ( {$<Year = {2015}, Country = {Germany}>} Sales )`

### Elementsets

Een elementset kan als volgt worden gedefinieerd:

- Een lijst met waarden
- Een zoekopdracht
- Een verwijzing naar een ander veld
- Een setfunctie

Als de definitie van een elementset wordt verwijderd, zorgt de set-modificatie dat alle selecties in dit veld worden gewist. Bijvoorbeeld:

```
Sum( {$<Year = >} Sales )
```

### Voorbeelden: diagramuitdrukkingen voor set-modificaties op basis van elementsets

Voorbeelden - Diagramuitdrukkingen

#### Load-script

Laad de volgende gegevens als een inline-load in de editor voor laden van gegevens om de onderstaande voorbeelden van diagramuitdrukkingen te maken.

```
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
];
```

### Diagramuitdrukkingen

Maak een tabel in een Qlik Sense-werkblad met de volgende diagramuitdrukkingen.

Tabel - Set-modificaties op basis van elementsets

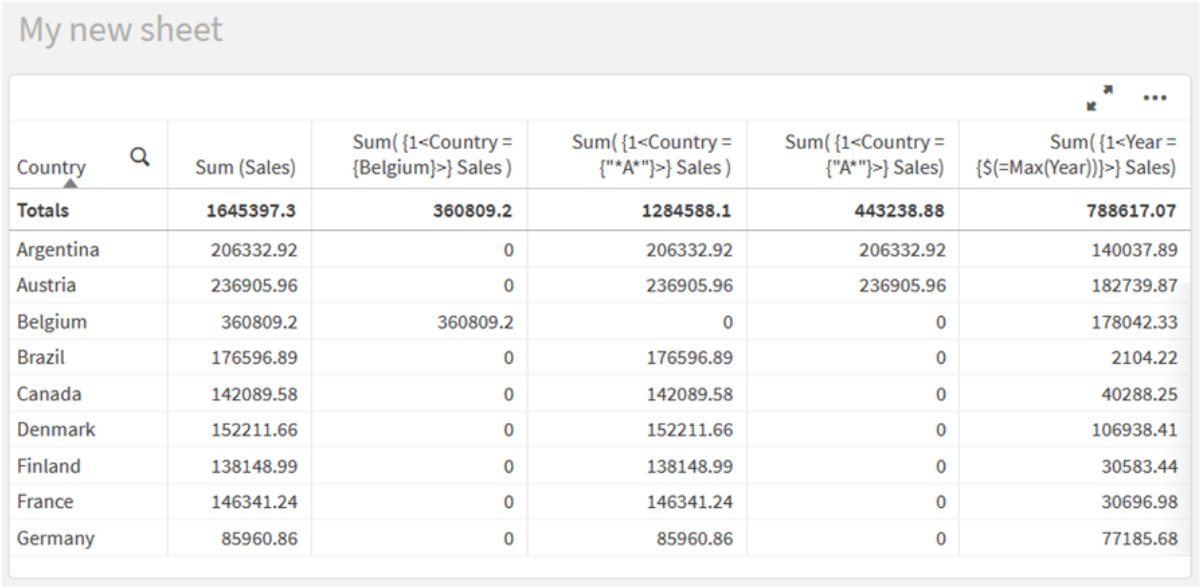
| Land | Sum(Sales) | Som ({1<Land={België}>} Verkoop) | Som ({1<Land={"*A*"}>} Verkoop) | Som ({1<Land={"A*"}>} Verkoop) | Sum({1<Year={$(=Max (Year))}>} Sales) |
|---|---|---|---|---|---|
| Totalen | 1645397.3 | 360809.2 | 1284588.1 | 443238.88 | 788617.07 |
| Argentinië | 206332.92 | 0 | 206332.92 | 206332.92 | 140037.89 |
| Oostenrijk | 236905.96 | 0 | 236905.96 | 236905.96 | 182739.87 |
| België | 360809.2 | 360809.2 | 0 | 0 | 178042.33 |
| Brazilië | 176596.89 | 0 | 176596.89 | 0 | 2104.22 |
| Canada | 142089.58 | 0 | 142089.58 | 0 | 40288.25 |
| Denemarken | 152211.66 | 0 | 152211.66 | 0 | 106938.41 |
| Finland | 138148.99 | 0 | 138148.99 | 0 | 30583.44 |
| Frankrijk | 146341.24 | 0 | 146341.24 | 0 | 30696.98 |
| Duitsland | 85960.86 | 0 | 85960.86 | 0 | 77185.68 |

### Uitleg

- Dimensies:
  - `Country`
- Metingen:
  - `Sum(Sales)`
    Sum `Sales` zonder set-uitdrukking.
  - `Sum({1<Country={Belgium}>}Sales)`
    Selecteer `Belgium` en vervolgens de som die overeenkomt met `Sales`.

- `Sum({1<Country={"*A*"}>}Sales)`
  Selecteer alle landen die een `A` bevatten en vervolgens de som die overeenkomt met `Sales`.
- `Sum({1<Country={"A*"}>}Sales)`
  Selecteer alle landen die beginnen met een `A` en vervolgens de som die overeenkomt met `Sales`.
- `Sum({1<Year={$(=Max(Year))}>}Sales)`
  Bereken de `Max(Year)`, dit is `2015`, en vervolgens de som die overeenkomt met `Sales`.

*Set-modificaties op basis van elementsets*


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) |
|---|---|---|---|---|---|
| **Totals** | **1645397.3** | **360809.2** | **1284588.1** | **443238.88** | **788617.07** |
| 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 |

## Weergegeven waarden

Het meest voorkomende voorbeeld van een elementset is een elementset die is gebaseerd op een lijst van veldwaarden die tussen accolades zijn geplaatst. Bijvoorbeeld:

- `{$<Country = {Canada, Germany, Singapore}>}`
- `{$<Year = {2015, 2016}>}`

De binnenste accolades definiëren de elementset. De individuele waarden worden gescheiden door komma's.

## Aanhalingstekens en hoofdlettergevoeligheid

Als de waarden lege waarden of speciale tekens bevatten, moeten de waarden tussen aanhalingstekens worden geplaatst. Bij het gebruik van enkele aanhalingstekens is er sprake van een hoofdlettergevoelige overeenkomst met de waarde van één veld. Bij het gebruik van dubbele aanhalingstekens is er sprake van een niet-hoofdlettergevoelige overeenkomst met één of meerdere veldwaarden. Bijvoorbeeld:

- `<Country = {'New Zealand'}>`
  Komt alleen overeen met `New Zealand`.
- `<Country = {"New Zealand"}>`
  Komt overeen met `New Zealand`, `NEW ZEALAND` en `new zealand`.

Datums moeten tussen aanhalingstekens worden geplaatst. U moet de datumindeling gebruiken van het desbetreffende veld. Bijvoorbeeld:

- `<ISO_Date = {'2021-12-31'}>`
- `<US_Date = {'12/31/2021'}>`
- `<UK_Date = {'31/12/2021'}>`

Dubbele aanhalingstekens kunnen worden vervangen door vierkante haken of accent graves.

## Zoekopdrachten

U kunt elementsets ook maken met behulp van zoekopdrachten. Bijvoorbeeld:

- `<Country = {"C*"}>`
- `<Ingredient = {"*garlic*"}>`
- `<Year = {">2015"}>`
- `<Date = {">12/31/2015"}>`

Jokertekens kunnen worden gebruikt voor zoekopdrachten in tekst: een asterisk (`*`) vertegenwoordigt een willekeurig aantal tekens en een vraagteken (`?`) vertegenwoordigt één teken. Relationele operatoren kunnen worden gebruikt om numerieke zoekopdrachten te definiëren.

U moet bij zoekopdrachten altijd dubbele aanhalingstekens gebruiken. Zoekopdrachten zijn niet hoofdlettergevoelig.

## Uitbreidingen met dollarteken

Uitbreidingen met dollarteken zijn nodig als u een berekening binnen uw elementenset wilt gebruiken. Als u bijvoorbeeld alleen naar het laatst mogelijke jaar wilt kijken, gebruikt u:

```
<Year = {$(=Max(Year))}>
```

## Geselecteerde waarden in andere velden

Modificaties kunnen worden gebaseerd op de geselecteerde waarden van een ander veld. Bijvoorbeeld:

```
<OrderDate = DeliveryDate>
```

Deze modificatie neemt de geselecteerde waarden uit `DeliveryDate` en past deze toe als selectie op `OrderDate`. Als er veel afzonderlijke waarden zijn – meer dan een paar honderd – vormt deze bewerking een te intensieve belasting van de processor en u dient deze dan te vermijden.

## Functies elementset

De elementset kan ook worden gebaseerd op de set-functies `P()` (mogelijke waarden) en `E()` (uitgesloten waarden).

Als u bijvoorbeeld de landen wilt selecteren waarin het product `Cap` is verkocht, kunt u het volgende gebruiken:

```
<Country = P({1<Product={Cap}>} Country)>
```

Als u de landen wilt bekijken waar het product `Cap` niet is verkocht, kunt u het volgende gebruiken:

```
<Country = E({1<Product={Cap}>} Country)>
```

## Set-modificaties met zoekopdrachten

U kunt elementsets maken door middel van zoekopdrachten met set-modificaties.

Bijvoorbeeld:

- `<Country = {"C*"}>`
- `<Year = {">2015"}>`
- `<Ingredient = {"*garlic*"}>`

Zoekopdrachten moeten altijd tussen dubbele aanhalingstekens, vierkante haakjes of accent graves worden geplaatst. U kunt een lijst gebruiken met een combinatie van letterlijke tekenreeksen (enkele aanhalingstekens) en zoekopdrachten (dubbele aanhalingstekens). Bijvoorbeeld:

```
<Product = {'Nut', "*Bolt", Washer}>
```

### Tekstzoekopdrachten

Jokertekens kunnen worden gebruikt voor tekstzoekopdrachten:

- Een asterisk (*) vertegenwoordigt een willekeurig aantal tekens.
- Een vraagteken (?) vertegenwoordigt één teken.
- Een accent circonflexe (^) vertegenwoordigt het begin van een woord.

Bijvoorbeeld:

- `<Country = {"C*", "*land"}>`
  Moet overeenkomen met alle landen die beginnen met een `C` of eindigen op `land`.
- `<Country = {"*^z*"}>`
  Hiermee worden alle landen aangegeven die overeenkomen met een woord dat begint met `z`, zoals `New Zealand`.

### Numerieke zoekopdrachten

U kunt numerieke zoekopdrachten uitvoeren met behulp van de volgende relationele operatoren: `>`, `>=`, `<`, `<=`

Een numerieke zoekopdracht begint altijd met een van deze operatoren. Bijvoorbeeld:

- `<Year = {">2015"}>`
  Moet overeenkomen met 2016 en opeenvolgende jaren.
- `<Date = {">=1/1/2015<1/1/2016"}>`
  Moet overeenkomen met alle datums in 2015. Houd rekening met de syntaxis voor het beschrijven van een periode tussen twee datums. De datumindeling moet overeenkomen met de datumindeling van het desbetreffende veld.

### Zoekopdrachten voor uitdrukkingen

U kunt zoekopdrachten voor uitdrukkingen gebruiken om geavanceerdere zoekopdrachten uit te voeren. Een aggregatie wordt vervolgens geëvalueerd voor elke veldwaarde in het zoekveld. Alle waarden waarvoor de zoekuitdrukking een waarde oplevert, worden geselecteerd.

Een zoekopdracht voor een uitdrukking begint altijd met een isgelijkteken: =

Bijvoorbeeld:

```
<Customer = {"=Sum(Sales)>1000"}>
```

Hiermee worden alle klanten geretourneerd met verkopen die hoger zijn dan 1000. `Sum(Sales)` wordt berekend op basis van de huidige selectie. Dit betekent dat als u een selectie in een ander veld hebt, zoals het veld `Product`, u de klanten krijgt die alleen voldoen aan de verkoopvoorwaarde voor de geselecteerde producten.

Als u wilt dat de voorwaarde onafhankelijk is van de selectie, moet u de set-analyse in de zoekreeks gebruiken. Bijvoorbeeld:

```
<Customer = {"=Sum({1} Sales)>1000"}>
```

De uitdrukkingen na het isgelijkteken worden geïnterpreteerd als een booleaanse waarde. Dit betekent dat als het iets anders evalueert, elk getal dat niet gelijk is aan nul als waar wordt geïnterpreteerd en nul en tekenreeksen als onwaar.

### Aanhalingstekens

Gebruik aanhalingstekens als de zoekreeksen lege of speciale karakters bevatten. Bij het gebruik van enkele aanhalingstekens is er sprake van een hoofdlettergevoelige overeenkomst met de waarde van één veld. Bij het gebruik van dubbele aanhalingstekens is er sprake van een niet-hoofdlettergevoelige zoekopdracht die mogelijk overeenkomt met meerdere veldwaarden.

Bijvoorbeeld:

- `<Country = {'New Zealand'}>`
  Moet alleen overeenkomen met `New Zealand`.
- `<Country = {"New Zealand"}>`
  Moet overeenkomen met `New Zealand`, `NEW ZEALAND` en `new zealand`

Dubbele aanhalingstekens kunnen worden vervangen door vierkante haken of accent graves.

> *In vorige versies van Qlik Sense was er geen onderscheid tussen enkele en dubbele aanhalingstekens en alle aanhalingstekens werden als zoekopdrachten behandeld. Om achterwaartse compatibiliteit te behouden, blijven apps die gemaakt zijn met oudere versies van Qlik Sense werken zoals ze in eerdere versies deden. Apps gemaakt met Qlik Sense van november 2017 of later maken onderscheid tussen de twee soorten aanhalingstekens.*

## Voorbeelden: diagramuitdrukkingen voor set-modificaties met zoekopdrachten

Voorbeelden - Diagramuitdrukkingen

**Load-script**

Laad de volgende gegevens als een inline-load in de editor voor laden van gegevens om de onderstaande voorbeelden van diagramuitdrukkingen te maken.

```
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];
```

**Voorbeeld 1: diagramuitdrukkingen voor set-modificaties met tekstzoekopdrachten.**

Maak een tabel in een Qlik Sense-werkblad met de volgende diagramuitdrukkingen.

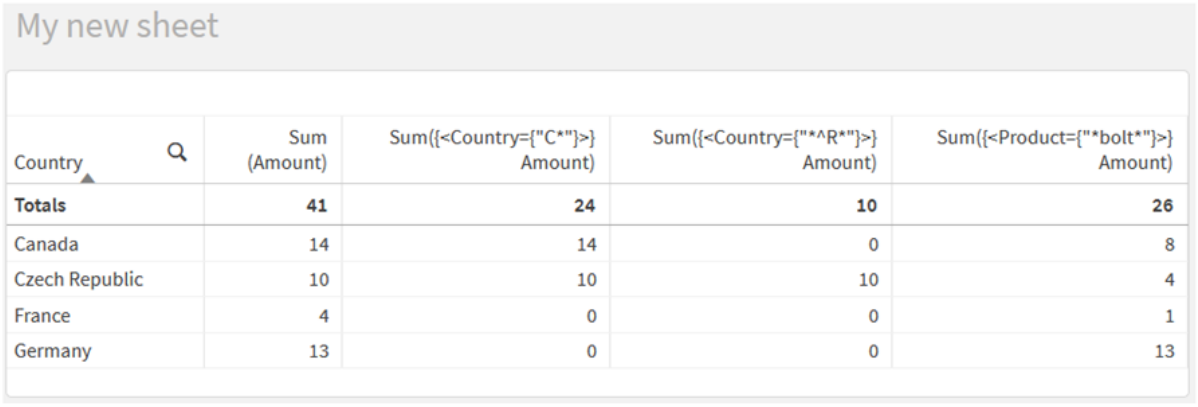
Tabel - Set-modificaties met tekstzoekopdrachten

| Land | Sum (Amount) | Sum({<Country={"C*"}>} Amount) | Sum({<Country={"*^R*"}>} Amount) | Sum({<Product={"*bolt*"}>} Amount) |
|---|---|---|---|---|
| **Totalen** | **41** | **24** | **10** | **26** |
| Canada | 14 | 14 | 0 | 8 |
| Tsjechië | 10 | 10 | 10 | 4 |
| Frankrijk | 4 | 0 | 0 | 1 |
| Duitsland | 13 | 0 | 0 | 13 |

**Uitleg**

- Dimensies:
  - `Country`
- Metingen:
  - `Sum(Amount)`
    Sum `Amount` zonder set-uitdrukking.
  - `Sum({<Country={"C*"}>}Amount)`
    Sum `Amount` voor alle landen die beginnen met `C`, zoals `Canada` en `Czech Republic`.
  - `Sum({<Country={"*^R*"}>}Amount)`
    Sum `Amount` voor alle landen die een woord bevatten dat begint met `R`, zoals `Czech Republic`.
  - `Sum({<Product={"*bolt*"}>}Amount)`
    Sum `Amount` voor alle producten die de tekenreeks `bolt` bevatten, zoals `Bolt` en `Anchor bolt`.

*Set-modificaties met tekstzoekopdrachten*


My new sheet

| Country | Sum (Amount) | Sum({<Country={"C*"}>} Amount) | Sum({<Country={"*^R*"}>} Amount) | Sum({<Product={"*bolt*"}>} Amount) |
|---|---|---|---|---|
| **Totals** | **41** | **24** | **10** | **26** |
| Canada | 14 | 14 | 0 | 8 |
| Czech Republic | 10 | 10 | 10 | 4 |
| France | 4 | 0 | 0 | 1 |
| Germany | 13 | 0 | 0 | 13 |

### Voorbeeld 2: diagramuitdrukkingen met numerieke zoekopdrachten

Maak een tabel in een Qlik Sense-werkblad met de volgende diagramuitdrukkingen.

Tabel - Set-modicaties met numerieke zoekopdrachten

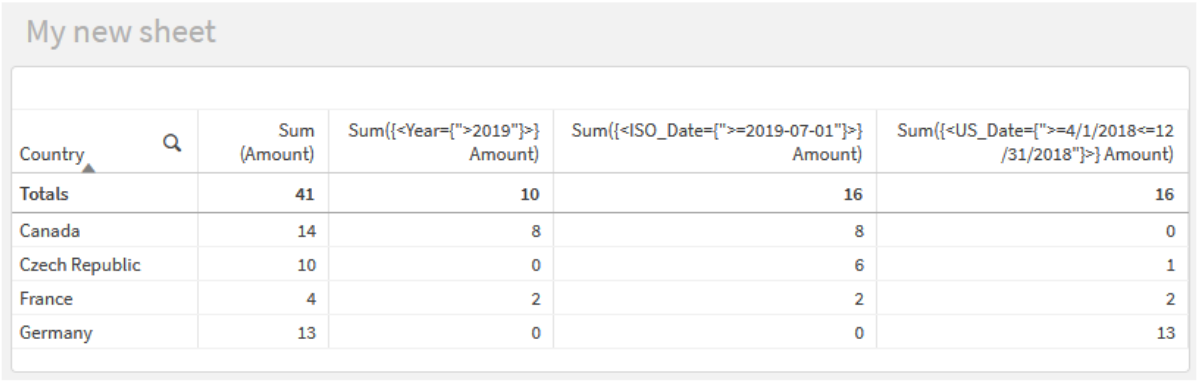
| Land | Sum (Amount) | Sum({<Year={">2019"}>} Amount) | Sum({<ISO_Date={">=2019-07-01"}>} Amount) | Sum({<US_Date={">=4/1/2018<=12/31/2018"}>} Amount) |
|---|---|---|---|---|
| **Totalen** | **41** | **10** | **16** | **16** |
| Canada | 14 | 8 | 8 | 0 |
| Tsjechië | 10 | 0 | 6 | 1 |
| Frankrijk | 4 | 2 | 2 | 2 |
| Duitsland | 13 | 0 | 0 | 13 |

### Uitleg

- Dimensies:
  - `Country`
- Metingen:
  - `Sum(Amount)`
    Sum `Amount` zonder set-uitdrukking.
  - `Sum({<Year={">2019"}>}Amount)`
    Sum `Amount` voor alle jaren na `2019`.
  - `Sum({<ISO_Date={">=2019-07-01"}>}Amount)`
    Sum `Amount` voor alle datums op of na `2019-07-01`. De indeling van de datum in de zoekopdracht moet overeenkomen met de indeling van het veld.
  - `Sum({<US_Date={">=4/1/2018<=12/31/2018"}>}Amount)`

Sum `Amount` voor alle datums van `4/1/2018` tot `12/31/2018`, inclusief de begin- en einddatum. De indeling van de datums in de zoekopdracht moet overeenkomen met de indeling van het veld.

*Set-modicaties met numerieke zoekopdrachten*




| Country | Sum (Amount) | Sum({<Year={">2019"}>} Amount) | Sum({<ISO_Date={">=2019-07-01"}>} Amount) | Sum({<US_Date={">=4/1/2018<=12/31/2018"}>} 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 |

### Voorbeeld 3: diagramuitdrukkingen met zoekopdrachten voor uitdrukkingen

Maak een tabel in een Qlik Sense-werkblad met de volgende diagramuitdrukkingen.

Table - Set modifiers with expression searches

| Country | Sum (Amount) | Sum({<Country={"=Sum (Amount)>10"}>} Amount) | Sum({<Country={"=Count(distinct Product)=1"}>} Amount) | Sum({<Product={"=Count (Amount)>3"}>} 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 |

### Uitleg

- Dimensies:
  - `Country`
- Metingen:
  - `Sum(Amount)`
    Sum `Amount` zonder set-uitdrukking.
  - `Sum({<Country={"=Sum(Amount)>10"}>}Amount)`
    Sum `Amount` voor alle landen die een aggregeerde som hebben van `Amount` die groter is dan `10`.
  - `Sum({<Country={"=Count(distinct Product)=1"}>}Amount)`

Sum `Amount` voor alle landen die zijn gekoppeld met precies één afzonderlijk product.

- `Sum({<Product={"=Count(Amount)>3"}>}Amount)`
  Sum `Amount` voor alle landen die meer dan drie transacties in de gegevens hebben.

*Set-modificaties met zoekopdrachten voor uitdrukkingen*

My new sheet

| Country | Sum (Amount) | Sum({<Country={"=Sum(Amount)>10"}>} Amount) | Sum({<Country={"=Count(distinct Product)=1"}>} Amount) | Sum({<Product={"=Count(Amount)>3"}>} 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 |

| Voorbeelden | Resultaten |
|---|---|
| sum( {$-1<Product = {"*Internal*", "*Domestic*"}>} Sales ) | Retourneert de verkoop voor de huidige selectie, exclusief transacties met betrekking tot producten met de reeks 'Internal' of 'Domestic' in de productnaam. |
| sum( {$<Customer = {"=Sum ({1<Year = {2007}>} Sales ) > 1000000"}>} Sales ) | Retourneert de verkoop voor de huidige selectie, maar met een nieuwe selectie in het veld 'Customer': alleen klanten die in 2007 een totale verkoop van meer dan 1000000 hadden. |

## Set-modificaties met uitbreiding via dollartekens

Uitdrukkingen met dollarteken zijn constructies die worden berekend voordat de uitdrukking wordt geparseerd en beoordeeld. Het resultaat wordt vervolgens in de uitdrukking ingevoegd in plaats van `$(...)`. De berekening van de uitdrukking wordt vervolgens uitgevoerd met behulp van het resultaat van de uitdrukking met dollarteken.

De uitdrukkingseditor toont een voorbeeld van de uitdrukking met dollarteken zodat u kunt controleren waarin de uitdrukking resulteert.

*Voorbeeld van de uitdrukking met dollarteken in de uitdrukkingseditor*

Gebruik uitbreidingen met dollarteken als u een berekening binnen uw elementset wilt gebruiken.

Als u bijvoorbeeld alleen wilt kijken naar het laatst mogelijke jaar, kunt de volgende constructie gebruiken:

```
<Year = {$(=Max(Year))}>
```

`Max(Year)` wordt eerst berekend en het resultaat wordt ingevoegd in de uitdrukking in plaats van `$(…)`.

Het resultaat na de uitdrukking met dollarteken is de volgende uitdrukking:

```
<Year = {2021}>
```

De uitdrukking in de uitdrukking met dollarteken wordt berekend op basis van de huidige selectie. Als u een selectie in een ander veld hebt, beïnvloedt dit het resultaat van de uitdrukking.

Als u wilt dat de berekening onafhankelijk is van de selectie, moet u de set-analyse in de uitdrukking met dollarteken gebruiken. Bijvoorbeeld:

```
<Year = {$(=Max({1} Year))}>
```

### Strings (tekenreeksen)

Als u wilt dat de uitdrukking met dollarteken resulteert in een tekenreeks, gelden de normale regels voor aanhalingstekens. Bijvoorbeeld:

```
<Country = {'$(=FirstSortedValue(Country,Date)'}>
```

Het resultaat na de uitdrukking met dollarteken is de volgende uitdrukking:

```
<Country = {'New Zealand'}>
```

Er ontstaat een fout in de syntaxis als u geen aanhalingstekens gebruikt.

### Numbers (getallen)

Als u wilt dat de uitdrukking met dollarteken resulteert in een getal, moet u zorgen dat de uitdrukking dezelfde opmaak krijgt als het veld. U moet de uitdrukking daarom soms in een opmaakfunctie verpakken.

Bijvoorbeeld:

```
<Amount = {$(=Num(Max(Amount), '###0.00'))}>
```

Het resultaat na de uitdrukking met dollarteken is de volgende uitdrukking:

```
<Amount = {12362.00}>
```

Gebruik een hash om af te dwingen dat een uitdrukking altijd een decimale komma gebruikt en geen scheidingsteken voor duizendtallen. Bijvoorbeeld:

```
<Amount = {$(#=Max(Amount))}>
```

### Datums

Als u wilt dat de uitdrukking met dollarteken resulteert in een datum, moet u zorgen dat de uitdrukking de juiste opmaak heeft. U moet de uitdrukking daarom soms in een opmaakfunctie verpakken.

Bijvoorbeeld:

```
<Date = {'$(=Date(Max(Date)))'}>
```

Het resultaat na de uitdrukking met dollarteken is de volgende uitdrukking:

```
<Date = {'12/31/2015'}>
```

Net als bij tekenreeksen moet u de juiste aanhalingstekens gebruiken.

Een veelvoorkomende gebruikscase is om uw berekening te beperken tot de laatste maand (of jaar). In dat geval kunt u een numerieke zoekopdracht gebruiken in combinatie met de functie `AddMonths()`.

Bijvoorbeeld:

```
<Date = {">=$(=AddMonths(Today(),-1))"}>
```

Het resultaat na de uitdrukking met dollarteken is de volgende uitdrukking:

```
<Date = {">=9/31/2021"}>
```

Hiermee worden alle evenementen geselecteerd die de afgelopen maand hebben plaatsgevonden.

### Voorbeeld: diagramuitdrukkingen voor set-modificaties met uitdrukkingen met dollarteken

#### Voorbeeld - diagramuitdrukkingen

**Load-script**

Laad de volgende gegevens als een inline-load in de editor voor laden van gegevens om de onderstaande voorbeelden van diagramuitdrukkingen te maken.

```
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];
```

### Diagramuitdrukkingen met uitdrukkingen met dollarteken

Maak een tabel in een Qlik Sense-werkblad met de volgende diagramuitdrukkingen.

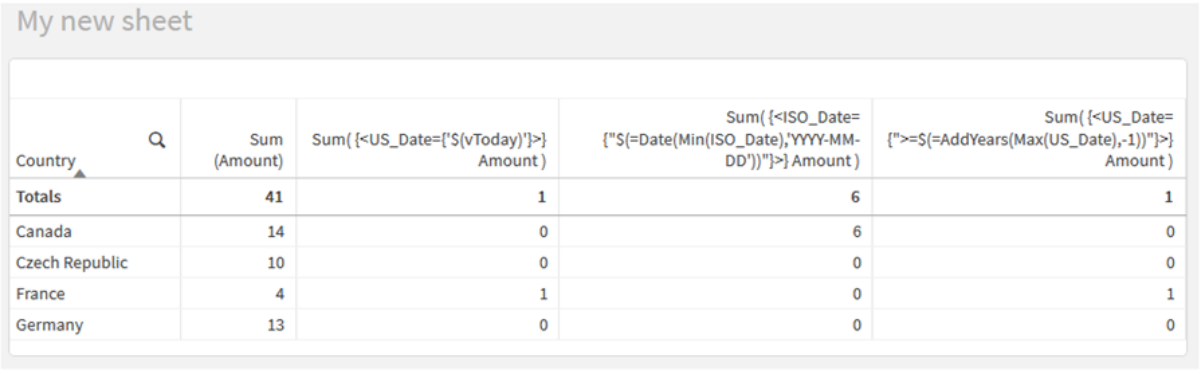
Tabel - Set-modificaties met uitbreiding via dollartekens

| Land | 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) |
|---|---|---|---|---|
| **Totalen** | **41** | **1** | **6** | **1** |
| Canada | 14 | 0 | 6 | 0 |
| Tsjechië | 10 | 0 | 0 | 0 |
| Frankrijk | 4 | 1 | 0 | 1 |
| Duitsland | 13 | 0 | 0 | 0 |

### Uitleg

- Dimensies:
  - `Country`
- Metingen:
  - `Sum(Amount)`
    Sum `Amount` zonder set-uitdrukking.
  - `Sum({<US_Date={'$(vToday)'}>}Amount)`
    Sum `Amount` voor alle records waarbij `US_Date` gelijk is aan de variabele`vToday`.
  - `Sum({<ISO_Date={"$(=Date(Min(ISO_Date),'YYYY-MM-DD'))"}>}Amount)`
    Sum `Amount` voor alle records waarbij `ISO_Date` gelijk is aan de eerst (laagst) mogelijke `ISO_Date`. De functie `Date()` is nodig om te garanderen dat de indeling van de datum overeenkomt met de indeling van het veld.
  - `Sum({<US_Date={">=$(=AddYears(Max(US_Date),-1))"}>}Amount)`
    Sum `Amount` voor alle records met een `US_Date` na of op de datum, één jaar door de laatste (hoogst) mogelijke `US_Date`. De functie `AddYears()` retourneert een datum in de indeling die is opgegeven door de variabele `DateFormat` en deze moet overeenkomen met de indeling van het veld `US_Date`.

*Set-modificaties met uitbreiding via dollartekens*


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 |

| Voorbeelden | Resultaten |
|---|---|
| sum( {$<Year = {$(#vLastYear)}>} Sales ) | Retourneert de verkoop van het vorige jaar in relatie tot de huidige selectie. Hier wordt de variabele vLastYear die het relevante jaar bevat, gebruikt in een uitbreiding via een dollarteken. |
| sum( {$<Year = {$(#=Only(Year)-1)}>} Sales ) | Retourneert de verkoop van het vorige jaar in relatie tot de huidige selectie. Een uitbreiding met een dollarteken wordt gebruikt om het vorige jaar te berekenen. |

## Set-modificaties met set-operatoren

Set-operatoren worden gebruikt voor het opnemen, uitsluiten of het doorkruisen van verschillende elementsets. Ze combineren de verschillende methoden voor het definiëren van elementsets.

De operatoren zijn gelijk aan de operatoren die voor set-id's worden gebruikt.

Operatoren

| Operator | Beschrijving |
|---|---|
| + | Samenvoegen. Deze binaire bewerking retourneert een set die bestaat uit de records of elementen die horen bij een van de twee set-operanden. |
| - | Uitsluiten. Deze binaire bewerking retourneert een set records of elementen die horen bij de eerste maar niet bij de andere van de twee set-operanden. Daarnaast levert het de complementerende set op als het wordt gebruikt als unaire operator. |
| * | Doorsnede. Deze binaire bewerking retourneert een set die bestaat uit de records of elementen die horen bij beide set-operanden. |
| / | Symmetrisch verschil (XOR). Deze binaire bewerking retourneert een set die bestaat uit de records of elementen die horen bij een van de twee set-operanden, maar niet bij beide set-operanden. |

De volgende twee modificaties definiëren bijvoorbeeld dezelfde set met veldwaarden:

- `<Year = {1997, "20*"}>`
- `<Year = {1997} + {"20*"}>`

Beide uitdrukkingen selecteren `1997` en de jaren die beginnen met `20`. In andere woorden: dit is de samenvoeging van de twee voorwaarden.

Bij set-operatoren kunnen complexere definities worden gebruikt. Bijvoorbeeld:

```
<Year = {1997, "20*"} - {2000}>
```

Met deze uitdrukking worden dezelfde jaren als hierboven geselecteerd, maar wordt het jaar `2000`uitgesloten.

.

## Voorbeelden: diagramuitdrukkingen voor set-modificaties met set-operatoren

### Voorbeelden - Diagramuitdrukkingen

#### Load-script

Laad de volgende gegevens als een inline-load in de editor voor laden van gegevens om de onderstaande voorbeelden van diagramuitdrukkingen te maken.

```
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];
```

#### Diagramuitdrukkingen

Maak een tabel in een Qlik Sense-werkblad met de volgende diagramuitdrukkingen.

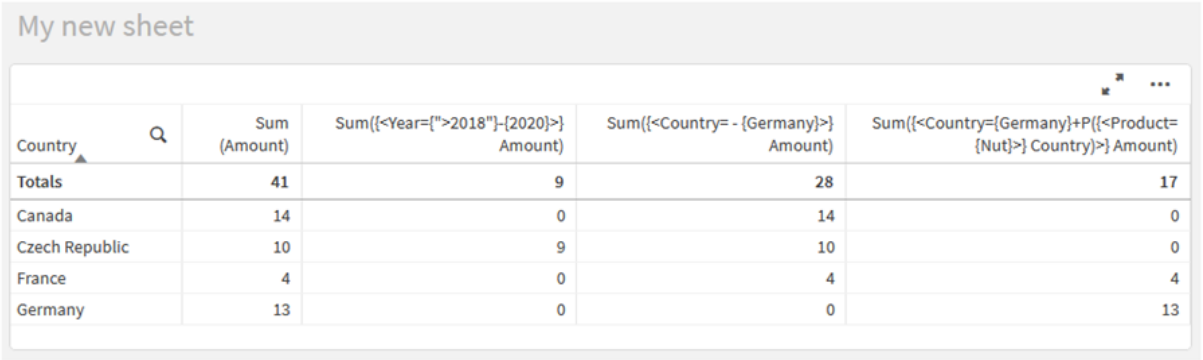
Tabel - Set-modificaties met set-operatoren

| Land | Sum (Amount) | Sum({<Year={">2018"}-{2020}>} Amount) | Sum ({<Country=-{Germany}>} Amount) | Sum({<Country={Germany}+P({<Product={Nut}>}Country)>} Amount) |
|---|---|---|---|---|
| **Totalen** | **41** | **9** | **28** | **17** |
| Canada | 14 | 0 | 14 | 0 |
| Tsjechië | 10 | 9 | 10 | 0 |
| Frankrijk | 4 | 0 | 4 | 4 |
| Duitsland | 13 | 0 | 0 | 13 |

### Uitleg

- Dimensies:
  - `Country`
- Metingen:
  - `Sum(Amount)`
    Sum `Amount` zonder set-uitdrukking.
  - `Sum({<Year={">2018"}-{2020}>}Amount)`
    Sum `Amount` voor alle jaren na `2018`, behalve `2020`.
  - `Sum({<Country=-{Germany}>}Amount)`
    Sum `Amount` voor alle landen, behalve `Germany`. Let op de unaire operator voor uitsluitingen.
  - `Sum({<Country={Germany}+P({<Product={Nut}>}Country)>}Amount)`
    Sum `Amount` voor `Germany` en alle landen die zijn gekoppeld aan het product `Nut`.

*Set-modificaties met set-operatoren*



| Voorbeelden | Resultaten |
|---|---|
| sum( {$<Product = Product + {OurProduct1} - {OurProduct2} >} Sales ) | Retourneert de verkoop voor de huidige selectie, maar met het product “OurProduct1” toegevoegd aan de lijst van geselecteerde producten en “OurProduct2” hieruit verwijderd. |
| sum( {$<Year = Year + ({“20*”,1997} - {2000}) >} Sales ) | Retourneert de verkoop voor de huidige selectie, maar met extra selecties in het veld “Year”: 1997 en alle jaren die beginnen met “20”, maar niet 2000.<br><br>Opmerking: als 2000 in de huidige selectie is opgenomen, blijft dit jaar nog opgenomen na de modificatie. |
| sum( {$<Year = (Year + {“20*”,1997}) - {2000} >} Sales ) | Retourneert bijna hetzelfde als hierboven, maar hier wordt het jaar 2000 uitgesloten, ook als het in de huidige selectie is opgenomen. Het voorbeeld laat zien dat het gebruik van haakjes om een volgorde aan te geven van groot belang kan zijn. |
| sum( {$<Year = {“*”} - {2000}, Product = {“*bearing*”} >} Sales ) | Retourneert de verkoop voor de huidige selectie, maar met een nieuwe selectie in “Year”: alle jaren behalve 2000 en alleen voor producten die de tekenreeks ‘lager’ bevatten. |

## Set-modificaties met impliciete set-operatoren

De standaard manier voor het schrijven van selecties in een set-modificatie is het gebruik van een isgelijkteken. Bijvoorbeeld:

```
Year = {">2015"}
```

De uitdrukking rechts van het isgelijkteken in de set-modificatie wordt een elementset genoemd. Deze definieert een set met afzonderlijke veldwaarden, oftewel een selectie.

Met deze notatie wordt een nieuwe selectie gedefinieerd en wordt de huidige selectie in het veld genegeerd. Dus als de set-id een selectie in dit veld bevat, wordt de oude selectie vervangen door de selectie in de elementset.

Als u uw selectie wilt baseren op de huidige selectie in het veld, moet u een andere uitdrukking gebruiken.

Als u bijvoorbeeld de oude selectie in acht wilt nemen en de vereiste wilt toevoegen dat het jaar na 2015 moet worden meegenomen, kunt u het volgende schrijven:

```
Year = Year * {">2015"}
```

De asterisk is een set-operator die een snijpunt definieert. U krijgt hierdoor het snijpunt tussen de huidige selectie in `Year` en de aanvullende vereiste dat het jaar na `2015` moet worden meegenomen. U kunt dit ook op de volgende manier schrijven:

```
Year *= {">2015"}
```

De toewijzingsoperator (*=) definieert impliciet een snijpunt.

Zo kunnen impliciete samenvoegingen, uitsluitingen en symmetrische verschillen met behulp van de volgende tekens worden gedefinieerd: `+=`, `-=`, `/=`

## Voorbeelden: diagramuitdrukkingen voor set-modificaties met impliciete set-operatoren

Voorbeelden - Diagramuitdrukkingen

### Load-script

Laad de volgende gegevens als een inline-load in de editor voor laden van gegevens om de onderstaande voorbeelden van diagramuitdrukkingen te maken.

```
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];
```

### Diagramuitdrukkingen met impliciete set-operatoren

Maak een tabel in een Qlik Sense-werkblad met de volgende diagramuitdrukkingen.

Selecteer `Canada` en `Czech Republic` uit een lijst met landen.

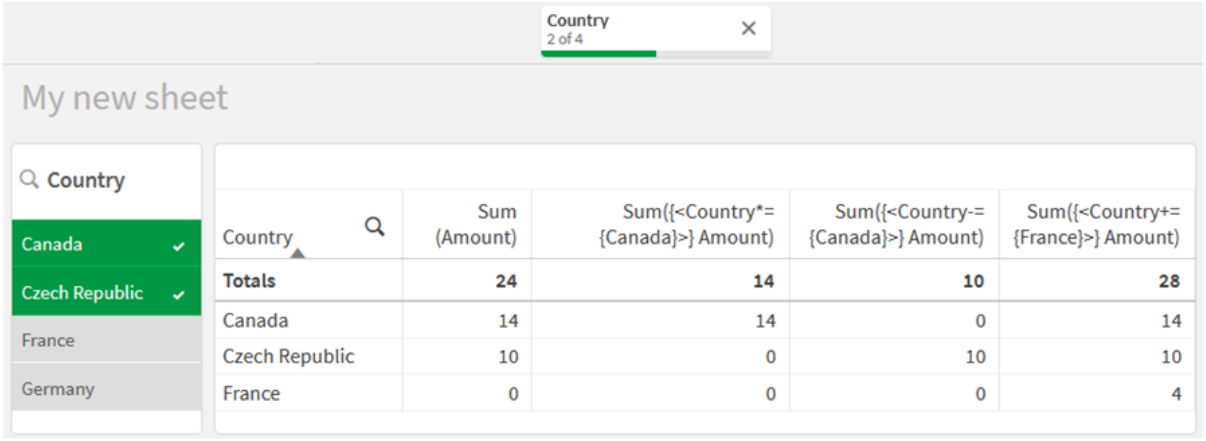
Tabel - Diagramuitdrukkingen met impliciete set-operatoren

| Land | Sum (Amount) | Sum({<Country*= {Canada}>} Amount) | Sum({<Country-= {Canada}>} Amount) | Sum({<Country+= {France}>} Amount) |
|---|---|---|---|---|
| **Totalen** | **24** | **14** | **10** | **28** |
| Canada | 14 | 14 | 0 | 14 |
| Tsjechië | 10 | 0 | 10 | 10 |
| Frankrijk | 0 | 0 | 0 | 4 |

### Uitleg

- Dimensies:
  - `Country`
- Metingen:
  - `Sum(Amount)`
    Sum `Amount` voor de huidige selectie. Houd er rekening mee dat alleen `Canada` en `Czech Republic` niet-nulwaarden bevatten.
  - `Sum({<Country*={Canada}>}Amount)`
    Sum `Amount` voor de huidige selectie, samen met de vereiste dat `Country` `Canada` moet zijn. Als `Canada` geen onderdeel uitmaakt van de gebruikersselectie, retourneert de set-uitdrukking een lege set en bevatten alle rijen in de kolom de waarde `0`.
  - `Sum({<Country-={Canada}>}Amount)`
    Sum `Amount` voor de huidige selectie, maar eerst wordt `Canada` uitgesloten van de selectie `Country`. Als `Canada` geen onderdeel uitmaakt van de gebruikersselectie, worden in de set-uitdrukking geen getallen gewijzigd.
  - `Sum({<Country+={France}>}Amount)`
    Sum `Amount` voor de huidige selectie, maar eerst wordt `France` toegevoegd aan de selectie `Country`. Als `France` al onderdeel uitmaakt van de gebruikersselectie, worden in de set-uitdrukking geen getallen gewijzigd.

*Set-modificaties met impliciete set-operatoren.*




| Country | Sum (Amount) | Sum({<Country*= {Canada}>} Amount) | Sum({<Country-= {Canada}>} Amount) | Sum({<Country+= {France}>} Amount) |
|---|---|---|---|---|
| **Totals** | **24** | **14** | **10** | **28** |
| Canada | 14 | 14 | 0 | 14 |
| Czech Republic | 10 | 0 | 10 | 10 |
| France | 0 | 0 | 0 | 4 |

| Voorbeelden | Resultaten |
|---|---|
| sum( {$<Product += {OurProduct1, OurProduct2} >} Sales ) | Retourneert de verkoop voor de huidige selectie, maar gebruikt een impliciete binding om de producten 'OurProduct1' en 'OurProduct2' toe te voegen aan de lijst met geselecteerde producten. |

| Voorbeelden | Resultaten |
|---|---|
| sum( {$<Year += {"20*",1997} - {2000} >} Sales ) | Retourneert de verkoop voor de huidige selectie, maar gebruikt een impliciete binding om een aantal jaren aan de selectie toe te voegen: 1997 en alle jaren die beginnen met "20", maar niet 2000.<br><br>Opmerking: als 2000 in de huidige selectie is opgenomen, blijft dit jaar nog opgenomen na de modificatie. Hetzelfde als `<Year=Year + ({"20*",1997}-{2000})>`. |
| sum( {$<Product *= {OurProduct1} >} Sales ) | Retourneert de verkoop voor de huidige selectie, maar alleen voor de gezamenlijk geselecteerde huidige producten en het product 'OurProduct1'. |

## Set-modificaties die gebruikmaken van set-functies

Soms moet u een set met veldwaarden definiëren met behulp van een geneste set-definitie. U wilt mogelijk alle klanten selecteren die een specifiek product hebben geselecteerd, zonder het product te selecteren.

In dergelijke gevallen gebruikt u de elementsetfuncties `P()` en `E()`. Deze retourneren de elementsets van mogelijke waarden en uitgesloten waarden van een veld. Binnen de haakjes kunt u het desbetreffende veld opgeven en een set-uitdrukking die het bereik definieert. Bijvoorbeeld:

```
P({1<Year = {2021}>} Customer)
```

Deze retourneert een set van klanten met transacties in 2021. U kunt deze vervolgens gebruiken in een set-modificatie. Bijvoorbeeld:

```
Sum({<Customer = P({1<Year = {2021}>} Customer)>} Amount)
```

Deze set-uitdrukking selecteert vervolgens deze klanten, maar beperkt de selectie niet tot 2021.

Deze functies kunnen niet worden gebruikt in andere uitdrukkingen.

Bovendien kunnen alleen natuurlijke sets worden gebruikt in de elementsetfuncties. Met andere woorden een set records die met een eenvoudige selectiebewerking kan worden gedefinieerd.

Bijvoorbeeld, de set van `{1-$}` kan niet altijd door middel van een selectie worden gedefinieerd en is dan ook geen natuurlijke set. Als deze functies op niet-natuurlijke sets worden gebruikt, worden er onverwachte resultaten geretourneerd.

### Voorbeelden: dagramuitdrukkingen voor set-modificaties die gebruikmaken van setfuncties

Voorbeelden - Diagramuitdrukkingen

**Load-script**

Laad de volgende gegevens als een inline-load in de editor voor laden van gegevens om de onderstaande voorbeelden van diagramuitdrukkingen te maken.

```
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];
```

### Diagramuitdrukkingen

Maak een tabel in een Qlik Sense-werkblad met de volgende diagramuitdrukkingen.

Tabel - Set-modificaties die gebruikmaken van setfuncties

| Land | Sum (Amount) | Sum({<Country=P ({<Year= {2019}>}Country)>} Amount) | Sum({<Product=P ({<Year= {2019}>}Product)>} Amount) | Sum({<Country=E ({<Product= {Washer}>}Country)>} Amount) |
|---|---|---|---|---|
| **Totalen** | **41** | **10** | **17** | **13** |
| Canada | 14 | 0 | 6 | 0 |
| Tsjechië | 10 | 10 | 10 | 0 |
| Frankrijk | 4 | 0 | 1 | 0 |
| Duitsland | 13 | 0 | 0 | 13 |

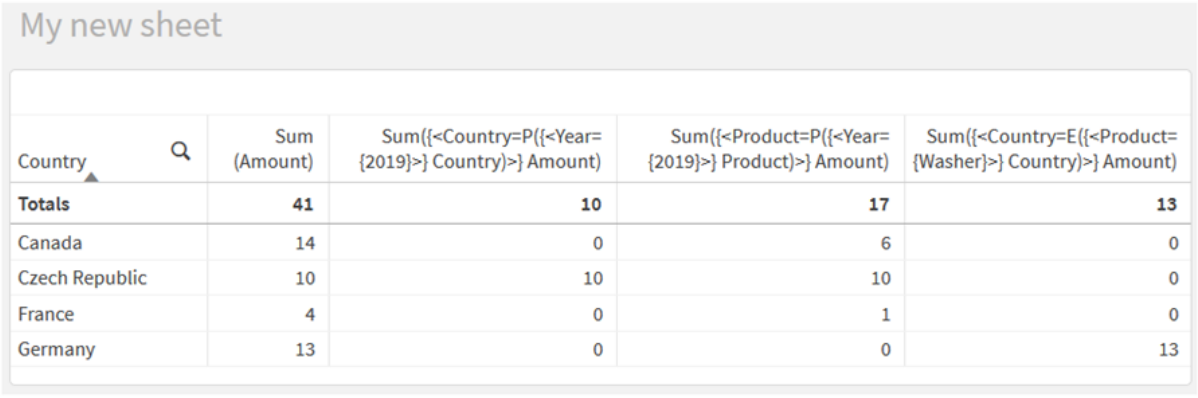
### Uitleg

- Dimensies:
  - `Country`
- Metingen:
  - `Sum(Amount)`
    Sum `Amount` zonder set-uitdrukking.
  - `Sum({<Country=P({<Year={2019}>} Country)>} Amount)`
    Sum `Amount` voor de landen die zijn gekoppeld aan het jaar `2019`. Hiermee wordt de berekening voor `2019` echter niet beperkt.
  - `Sum({<Product=P({<Year={2019}>} Product)>} Amount)`

Sum `Amount` voor de producten die zijn gekoppeld aan het jaar `2019`. Hiermee wordt de berekening voor `2019` echter niet beperkt.

- `Sum({<Country=E({<Product={Washer}>} Country)>} Amount)`
  Sum `Amount` voor de landen die niet zijn gekoppeld aan het product `Washer`.

*Set-modificaties die gebruikmaken van set-functies*


My new sheet

| Country | Sum (Amount) | Sum({<Country=P({<Year={2019}>} Country)>} Amount) | Sum({<Product=P({<Year={2019}>} Product)>} Amount) | Sum({<Country=E({<Product={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 |

| Voorbeelden | Resultaten |
|---|---|
| sum( {$<Customer = P({1<Product= {'Shoe'}>} Customer)>} Sales ) | Retourneert de verkoop voor de huidige selectie, maar alleen de klanten die ooit het product 'Shoe' gekocht hebben. De elementfunctie P( ) retourneert hier een lijst met mogelijke klanten; de klanten die worden geïmpliceerd door de selectie 'Shoe' in het veld Product. |
| sum( {$<Customer = P({1<Product= {'Shoe'}>})>} Sales ) | Hetzelfde als hierboven. Als het veld in de elementfunctie wordt weggelaten, retourneert de functie de mogelijke waarden van het veld dat is opgegeven in de outer-toewijzing. |
| sum( {$<Customer = P({1<Product= {'Shoe'}>} Supplier)>} Sales ) | Retourneert de verkoop voor de huidige selectie, maar alleen de klanten die ooit het product 'Shoe' geleverd hebben. De klant is dus ook een leverancier. De elementfunctie P( ) retourneert hier een lijst met mogelijke leveranciers; de leveranciers die worden geïmpliceerd door de selectie 'Shoe' in het veld Product. De lijst met leveranciers wordt vervolgens gebruikt als selectie in het veld Customer. |
| sum( {$<Customer = E({1<Product= {'Shoe'}>})>} Sales ) | Retourneert de verkoop voor de huidige selectie, maar alleen de klanten die nooit het product 'Shoe' hebben gekocht. De elementfunctie E( ) retourneert hier een lijst met uitgesloten klanten; de klanten die worden uitgesloten door de selectie 'Shoe' in het veld Product. |

## Vlag voor behoud van lege set

U kunt het symbool & (de vlag voor behoud van lege set) aan het begin van sommige buitenste setuitdrukkingen gebruiken om het impliciet wissen van selectiesets tijdens gebruikersanalyse te overschrijven. Met de vlag voor behoud van lege set kunt u wijzigen hoe aggregaties worden berekend en weergegeven wanneer gebruikers selecties maken, zonder dat u lange, geavanceerde setuitdrukkingen hoeft te herschrijven.

In de volgende setuitdrukking wordt bijvoorbeeld de vlag voor behoud van lege set gebruikt:

```
{&<Country*={'Sweden', 'Denmark'}>} {<ProductCategory={'shirts'}>} Sum( Sales )
```

### Achtergrond

#### Gebruiksvoorwaarden

De vlag voor behoud van lege set kan worden gebruikt in setuitdrukkingen die een van de volgende componenten bevatten:

- Ten minste één binnenste en één buitenste setuitdrukking
- Twee of meer buitenste setuitdrukkingen

Gebruik de vlag voor behoud van lege set wanneer u *Impliciet wissen van selectiesets (page 332)* wilt vermijden.

#### Impliciet wissen van selectiesets

Impliciet wissen van selectiesets is een intern gedrag dat soms ongewenst kan zijn voor eindgebruikersanalyse wanneer complexe ketens van setuitdrukkingen worden gebruikt.Impliciet wissen van selectiesets kan optreden voor setuitdrukkingen die zijn gestructureerd zoals beschreven in de *Gebruiksvoorwaarden (page 332)*.

Het ongewenste gedrag is als volgt:

1. Eén of meer buitenste setuitdrukkingen in de keten – specifiek de eerste (of enige) en alle volgende buitenste setuitdrukkingen behalve de laatste – definiëren gegevensverzamelingen. Bijvoorbeeld: `Country*={'Sweden', 'Denmark'}>}`
2. Er gebeurt iets waardoor een lege set wordt geproduceerd voor een of meer van deze buitenste setuitdrukkingscomponenten. Meestal wordt er een conflicterende selectie gemaakt in de app.
3. Voor de lege sets worden de sets die door deze uitdrukkingen zijn gedefinieerd, gewist zodra de uitdrukking wordt geëvalueerd. Dit betekent dat de gegevens die in de aggregaties worden gebruikt, niet langer worden gereduceerd tot de gedefinieerde sets, maar in plaats daarvan worden alle gegevens gebruikt.
   Aan de andere kant worden de sets die zijn gedefinieerd in buitenste setuitdrukkingen die niet-lege sets produceren, evenals de laatste buitenste setuitdrukking in de keten, gerespecteerd tijdens de evaluatie.

Dit gedrag resulteert in aggregatiewaarden die onverwacht kunnen zijn en kan leiden tot inconsistente analyse. Om het impliciet wissen van selectiesets te omzeilen, kunt u de vlag voor het behouden van lege sets gebruiken. Zie: *De oplossing: De vlag voor het behouden van lege sets gebruiken (page 333)*.

### De oplossing: De vlag voor het behouden van lege sets gebruiken

Gebruik het symbool & (de vlag voor het behouden van lege sets) om het impliciet wissen van selectiesets op te heffen. Deze vlag zorgt ervoor dat de sets die zijn gedefinieerd in alle buitenste setuitdrukkingen, behalve de laatste, in een keten niet worden gewist, maar in plaats daarvan worden gerespecteerd gedurende de evaluatie van de gehele uitdrukking.

De vlag voor het behouden van lege sets biedt:

- Flexibiliteit en tijdsbesparing wanneer u geavanceerde, lange setuitdrukkingen hebt die u intact wilt houden, in plaats van ze te herschrijven.
- De mogelijkheid om aggregatielogica aan te passen aan geavanceerde gebruiksscenario's.

### Syntaxis

In een buitenste setuitdrukking wordt de vlag voor het behouden van lege sets geplaatst als één enkel teken & aan het begin van de uitdrukking – dat wil zeggen, vóór de setmodificatoren, identificatoren en operatoren.

**Voorbeelden:**

Voorbeelden – vlag voor het behouden van lege sets

| Uitdrukking | Uitdrukking zonder vlag voor het behouden van lege sets |
|---|---|
| `{&<Country*={'Sweden', 'Denmark'}>} {<ProductCategory={'shirts'}>} Sum( Sales )` | `{<Country={'Sweden'}>} {<ProductCategory={'shirts'}>} Sum (Sales)` |
| `{& <Year=, Product={XI345}>} {<Region={Europe}>} Sum (Sales)` | `{<Year=, Product={XI345}>} {<Region= {Europe}>} Sum (Sales)` |

### Details en voorbeelden

Zie *Ketens van meervoudige buitenste en binnenste set-uitdrukkingen (page 336)* voor meer diepgaande informatie.

## Binnenste en buitenste set-uitdrukkingen

Set-uitdrukkingen kunnen binnen en buiten aggregatiefuncties worden gebruikt en worden tussen accolades geplaatst.

Als u een set-uitdrukking binnen een aggregatiefunctie gebruikt, kan deze er zo uit komen te zien:

**Voorbeeld: Binnenste set-uitdrukking**

```
Sum( {$<Year={2021}>} Sales )
```

Gebruik een set-uitdrukking buiten de aggregatiefunctie als u uitdrukkingen met meerdere aggregaties hebt en niet dezelfde set-uitdrukking in elke aggregatiefunctie wilt schrijven.

Als u een buitenste set-uitdrukking gebruikt, moet deze aan het begin van het bereik worden geplaatst.

#### Voorbeeld: Buitenste set-uitdrukking

```
{<Year={2021}>} Sum(Sales) / Count(distinct Customer)
```

Als u een set-uitdrukking buiten de aggregatiefunctie gebruikt, kunt u deze ook toepassen op bestaande mastermetingen.

#### Voorbeeld: Buitenste set-uitdrukking toegepast op mastermeting

```
{<Year={2021}>} [Master Measure]
```

Een set-uitdrukking die buiten aggregatiefuncties wordt gebruikt, heeft invloed op de volledige uitdrukking, tenzij deze tussen haakjes is geplaatst. In dat geval definiëren de haakjes het bereik. In het onderstaande voorbeeld van een statische bereik wordt de set-uitdrukking alleen toegepast op de aggregatie binnen de haakjes.

#### Voorbeeld: Statisch bereik toepassen

```
( {<Year={2021}>} Sum(Amount) / Count(distinct Customer) ) - Avg(CustomerSales)
```

## Regels

#### Statisch bereik

De set-uitdrukking is van invloed op de volledige uitdrukking, tenzij deze tussen haakjes is geplaatst. Indien dit het geval is, definiëren de haakjes het statische bereik.

#### Positie

De set-uitdrukking moet aan het begin van het statische bereik worden geplaatst.

#### Context

De context is de selectie die relevant is voor de uitdrukking. Normaal gesproken is de context altijd de standaardstate van de huidige selectie. Maar als een object op een alternatieve state is ingesteld, is de context de alternatieve state van de huidige selectie.

U kunt ook een context definiëren in de vorm van een buitenste set-uitdrukking.

#### Overname

Binnenste set-uitdrukkingen hebben voorrang op buitenste set-uitdrukkingen. Als de binnenste set-uitdrukking een set-id bevat, overschrijft deze de context die is gedefinieerd in de buitenste set-uitdrukking. Anders worden de binnenste en buitenste set-uitdrukkingen beide geëvalueerd.

#### Example 1: Binnste set-uitdrukking met set-id

```
{<Year={2023}>} Sum(Sales) / Count({1} distinct OrderNumber)
```

In de bovenstaande uitdrukking bestaat de binnenste set-uitdrukking `Count({1} distinct OrderNumber)` uit de set-id `{1}`. Wanneer deze binnenste set-uitdrukking wordt geëvalueerd, wordt het bereik dat is gedefinieerd in de buitenste set-uitdrukking `{<Year={2023}>}` niet toegepast.

### Example 2: Binnenste set-uitdrukkingen met set-id's

```
{<Year={2023}>} Sum ({<Status={'Confirmed'}>} Sales_Stream1) + Sum ({<UpdatedStatus=
{'Confirmed'}>} Sales_Stream2)
```

In de bovenstaande uitdrukking bevatten de binnenste set-uitdrukkingen `Sum ({<Status={'Confirmed'}>} Sales_Stream1)` en `Sum ({<UpdatedStatus={'Confirmed'}>} Sales_Stream2)` geen set-id's. Daarom worden de buitenste set-uitdrukkingen `{<Year={2023}>}` en beide binnenste set-uitdrukkingen allemaal toegepast bij het evalueren van het resultaat.

#### Toewijzing van elementset

De toewijzing van de elementset bepaalt hoe de twee selecties worden samengevoegd. Als een normaal gelijkteken wordt gebruikt, heeft de selectie in de binnenste set-uitdrukking voorrang. Anders wordt de impliciete set-operator gebruikt.

- `{<Field={value}>}` - deze binnenste selectie vervangt elke buitenste selectie in "`Field`".
- `{<Field+={value}>}` - deze binnenste selectie wordt samengevoegd met de buitenste selectie in "`Field`", met behulp van de verenigingsoperator.
- `{<Field*={value}>}` - deze binnenste selectie wordt samengevoegd met de buitenste selectie in "`Field`", met behulp van de doorsnede-operator.

#### Overname in meerdere stappen

Overname kan in meerdere stappen worden uitgevoerd. Voorbeelden:

- Huidige selectie → `Sum(Amount)`
  De aggregatiefunctie gebruikt de context, dat is hier de huidige selectie.
- Huidige selectie → `{<Set1>} Sum(Amount)`
  `Set1` neemt de huidige selectie over en het resultaat is de context voor de aggregatiefunctie.
- Huidige selectie → `{<Set1>} ({<Set2>} Sum(Amount))`
  `Set2` neemt `Set1` over, die vervolgens de huidige selectie overneemt en het resultaat is de context voor de aggregatiefunctie.

#### Overname voor ketens van binnenste en buitenste set-uitdrukkingen

Er zijn bijkomende overwegingen voor overname bij het gebruik van meerdere set-uitdrukkingen in complexe ketens, vooral wanneer een buitenste set-uitdrukking gevolgd wordt door andere buitenste set-uitdrukkingen en/of een binnenste set-uitdrukking. Ga voor meer informatie naar *Ketens van meervoudige buitenste en binnenste set-uitdrukkingen (page 336)*.

#### De functie Aggr()

De functie `Aggr()` maakt een geneste aggregatie die twee afzonderlijke aggregaties bevat. In het onderstaande voorbeeld wordt een `Count()` berekend voor elke waarde van `Dim`, en de resulterende matrix is geaggregeerd met behulp van de functie `Sum()`.

#### Voorbeeld:

`Sum(Aggr(Count(X),Dim))`
`Count()` is de binnenste aggregatie en `Sum()` is de buitenste aggregatie.

- De binnenste aggregatie neemt geen context over van de buitenste aggregatie.
- De binnenste aggregatie neemt de context van de functie `Aggr()` over, die mogelijk een set-uitdrukking bevat.
- Zowel de functie `Aggr()` als de buitenste aggregatiefunctie nemen de context van een buitenste set-uitdrukking over.

## Ketens van meervoudige buitenste en binnenste set-uitdrukkingen

Het is mogelijk om meer dan één buitenste set-uitdrukking te hebben. In dit voorbeeld hebben we twee buitenste setuitdrukkingen en één binnenste:

```
{<Year={2021}>} {<Region={"Europe"}>} Sum( {$<Product={"XI345"}>} Sales )
```

De set-uitdrukkingen worden van links naar rechts toegepast. Dit betekent dat als twee set-uitdrukkingen conflicterende sets gebruiken, de meest rechtse uitdrukking voorrang heeft.

Hoewel er geen limiet is aan het aantal buitenste set-uitdrukkingen die op deze manier aan elkaar geketend zijn, is het aantal binnenste set-uitdrukkingen beperkt tot één. Als er meer dan één binnenste set-uitdrukking naast elkaar staat, zal er geen fout optreden, maar zal alleen de meest rechtse gebruikt worden bij de evaluatie.

### Uitdrukkingovername instellen

Als een aggregatie-uitdrukking meer dan één set-uitdrukking bevat als onderdeel van een keten, worden de set-uitdrukkingen één voor één van links naar rechts geëvalueerd. Het resultaat van elke set-uitdrukking is een groep sets, mogelijk één voor elke dimensie. De sets worden doorgegeven aan de volgende set-uitdrukking, die verdere wijzigingen kan aanbrengen voordat deze wordt doorgegeven aan de volgende set.

Zolang elke set-uitdrukking op verschillende dimensies werkt, zullen er geen conflicten tussen ze zijn, en zullen alle sets behouden blijven tot het einde van de keten (en toegepast worden tijdens het aggregeren). Er is één uitzondering op deze regel, die soms van toepassing is als een dimensie in een set-uitdrukking een lege set heeft opgeleverd. Zie het volgende voor meer informatie:

- *Lege selectiesets (page 336)*
- *Impliciete selectieset wissen (page 337)*

### Lege selectiesets

Wanneer een dimensie in een set-uitdrukking wordt geëvalueerd, is het mogelijk dat de set geen gegevens bevat. Dit wordt een lege set genoemd. Lege sets kunnen in veel scenario's voorkomen, zoals:

- Onjuiste uitdrukkingdefinitie. Als u bijvoorbeeld een waarde 2025 probeert te zoeken met een setuitdrukking `{<OrderYear={‘2025’}>}`, wordt de uitdrukking geschreven als `{<OrderYear={‘0025’}>}`.
- Tijdens de analyse wordt een selectie gemaakt in een andere dimensie dan die waarnaar in de uitdrukking wordt verwezen en deze selectie resulteert in geen beschikbare gegevens voor de ingestelde dimensie van de uitdrukking. Een gebruiker maakt bijvoorbeeld een selectie voor `Region='Europe'` en een set uitdrukkingsfilters voor `{<ProductCategory={‘Shirts’}>} Sum (Sales)`. Als er geen overhemden in Europa worden verkocht, bevat de uitdrukking een lege set.

- De lege set wordt expliciet gedefinieerd in de uitdrukking. Dit is niet gebruikelijk of nuttig, maar technisch mogelijk. Bijvoorbeeld: `{<Year={}>}`

### Impliciete selectieset wissen

Als het resultaat na het toepassen van een set-uitdrukking de lege set is voor een van de dimensies (zie *Lege selectiesets (page 336)*) en er wordt hierna nog een set-uitdrukking toegepast, dan wordt de selectie voor die dimensie gewist (teruggezet naar de volledige set) vlak voordat die volgende set-uitdrukking wordt toegepast. Een gewiste selectie betekent dat alle waarden worden opgenomen. Dit wissen gebeurt alleen als er nog één of meer buitenste set-uitdrukkingen worden toegepast na de set-uitdrukking waarbij de lege set wordt geproduceerd.

Hier volgt een voorbeeld van wanneer een gewiste selectieset zal plaatsvinden:

```
{<Year={}>} {<Region={"Europe"}>} Sum( Sales )
```

`Year={}` zal de lege set voor de dimensie `Year` maken. Dit betekent dat de selectieset voor `Year` wordt gewist wanneer de set wordt doorgegeven aan de volgende set-uitdrukking, die `Region={Europe}` bevat. Het wissen van de selectie gebeurt alleen voor een enkele dimensie. Sets in andere dimensies, inclusief sets binnen dezelfde set-uitdrukking, worden nog steeds zonder wijziging doorgegeven.

Hier volgt een voorbeeld van een gewiste selectieset:

```
{<Year={}, Product={"XI345"}>} {<Region={"Europe"}>} Sum(Sales)
```

In het bovenstaande voorbeeld gebeurt het volgende:

- De selectiesets `Product={"XI345"}` en `Region={"Europe"}` worden toegepast.
- `Year={}` wordt niet toegepast. Dit komt doordat die set gewist wordt wanneer de laatste set-uitdrukking—`{<Region={"Europe"}>}`—wordt toegepast.

Door een component van de set-uitdrukking naar de laatste buitenste set-uitdrukking te verplaatsen, wordt voorkomen dat er impliciet gewist wordt. Neem dit voorbeeld:

```
{<Product={"XI345"}>} {<Year={}, Region={"Europe"}>} Sum(Sales)
```

Hier hebben we de uitdrukking `Year` verplaatst naar de laatste set-uitdrukking in de keten. Hierdoor wordt het impliciet wissen voorkomen en wordt de lege set voor de dimensie `Year` daadwerkelijk gebruikt voor de `Sum`-aggregatie.

In elk geval kan ongewenst gedrag optreden wanneer selectiesets gewist worden. Om de analyse-ervaring te verbeteren en het schrijven van ketens van set-uitdrukkingen te vergemakkelijken, kunt u de vlag voor behoud van lege set gebruiken. Ga voor meer informatie naar *Vlag voor behoud van lege set (buitenste set-uitdrukkingen) (page 337)*.

## Vlag voor behoud van lege set (buitenste set-uitdrukkingen)

Gebruik het symbool `&` (de vlag voor behoud van lege set) om het evaluatiemodel voor buitenste set-uitdrukkingen te vereenvoudigen. Bijvoorbeeld:

```
{& <Year={}, Product={XI345}>} {<Region={Europe}>} Sum( Sales )
```

In dit voorbeeld definieert `Year={}` een lege selectieset voor `Year`. Zonder het symbool `&` zou de uitdrukking `{<Region={Europe}>}` ervoor zorgen dat de lege selectieset `Year` gewist wordt, waardoor alle gegevens in de aggregatieberekening worden gebruikt. Het symbool `&` dwingt de aggregatie om de lege selectieset `Year` te respecteren, wat resulteert in een begrijpelijkere uitvoer.

Voor volledige details over de vlag voor het behouden van lege sets, zie *Vlag voor behoud van lege set (page 332)*.

## Buitenste set-uitdrukkingen en uitbreidingen met dollarteken

Wanneer één (of meer) uitbreidingen met dollarteken worden gebruikt in een grotere uitdrukking, zal elke uitbreiding met dollarteken afzonderlijk worden geëvalueerd voordat het resultaat wordt ingevoegd in de rest van de uitdrukking. Een implicatie van deze evaluatiemethode is dat set-uitdrukkingen buiten een door een dollarteken uitgebreide uitdrukking het resultaat niet zullen beïnvloeden. Dit creëert een effect waarbij er sprake is van een dubbele evaluatie van informatie binnen een uitdrukking.

### Voorbeeld - Hoe uitdrukkingen die een buitenste set-uitdrukking en uitbreiding met dollarteken bevatten, worden geëvalueerd

Stel dat u deze voorbeelduitdrukking hebt:

```
{<Year={2024}>} $(=Sum(Sales))
```

Binnen de bovenstaande uitdrukking is er een set-uitdrukking om *Year*-waarden van *2024* te selecteren: `{<Year={2024}>}`. Als de volledige voorbeelduitdrukking is geëvalueerd, wordt deze set-uitdrukking niet toegepast.

In plaats daarvan wordt de uitdrukking met dollarteken `$(=Sum(Sales))` eerst geëvalueerd.

Stel voor dit voorbeeld dat `Sum(Sales)` een resultaat van 1500311 oplevert. Bij het evalueren van de uitdrukking `{<Year={2024}>} $(=Sum(Sales))` wordt dit getal ingevoegd in de omringende uitdrukking. Deze evaluatievolgorde resulteert in de volgende gedeeltelijk geëvalueerde configuratie (merk op dat dit geen bruikbare Qlik Sense-uitdrukking is, en alleen wordt getoond om de evaluatievolgorde aan te tonen):

```
{<Year={2024}>} 1500311
```

Vervolgens wordt de uitdrukking `{<Year={2024}>} 1500311` in zijn geheel geëvalueerd. De set-uitdrukking `{<Year={2024}>}` heeft geen enkel effect omdat de uitdrukking rechts ervan gewoon een getal is, en geen enkel type geaggregeerde uitdrukking.

Daarom zou, met behulp van de waarde `Sum(Sales)` die hierboven in het voorbeeld is gegeven, het geëvalueerde resultaat voor de hele uitdrukking `{<Year={2024}>} $(=Sum(Sales))` 1500311 zijn.

### Een uitdrukking opbouwen uit delen

Het gebruik van uitbreidingen met dollarteken in een uitdrukking resulteert in een dubbel evaluatie-effect. U kunt van dit dubbele evaluatie-effect profiteren door een uitdrukking op te bouwen met behulp van delen.

#### Voorbeeld

Stel dat u deze voorbeelduitdrukking hebt:

```
{<Year={2024}>} $('Sum(' & 'Sales' & ')')
```

In de eerste evaluatieronde wordt de uitbreiding met dollarteken `$('Sum(' & 'Sales' & ')')` geëvalueerd. In deze stap worden de drie tekenreeksen gecombineerd tot een enkele tekenreeks `'Sum(Sales)'`. Aangezien dit een tekenreeks is en geen uitdrukking, wordt de uitdrukking binnenin op dit moment niet geëvalueerd.

In dit stadium van de evaluatie wordt de uitdrukking door Qlik Sense als volgt geïnterpreteerd:

```
{<Year={2024}>} Sum(Sales)
```

In dit geval wordt de buitenste uitdrukking voor set-analyse toegepast op de aggregatie `Sum`, omdat we nu een normale uitdrukking hebben geproduceerd, in plaats van alleen een getal (zoals in het eerste voorbeeld). Merk op dat de aanhalingstekens zijn gebruikt in de eerste evaluatieronde, zodat de uitdrukking behouden bleef (en aaneenschakeld is met behulp van `&`) voor de tweede evaluatieronde.

## Variabele opzoeken

Wanneer u een uitbreiding met dollarteken gebruikt om een variabele op te zoeken, wordt de inhoud van de variabele ingevoegd in de omringende uitdrukking voordat die inhoud wordt geëvalueerd.

### Voorbeeld

Stel dat u deze voorbeelduitdrukking hebt:

```
{<Year={2024}>} $(Var1)
```

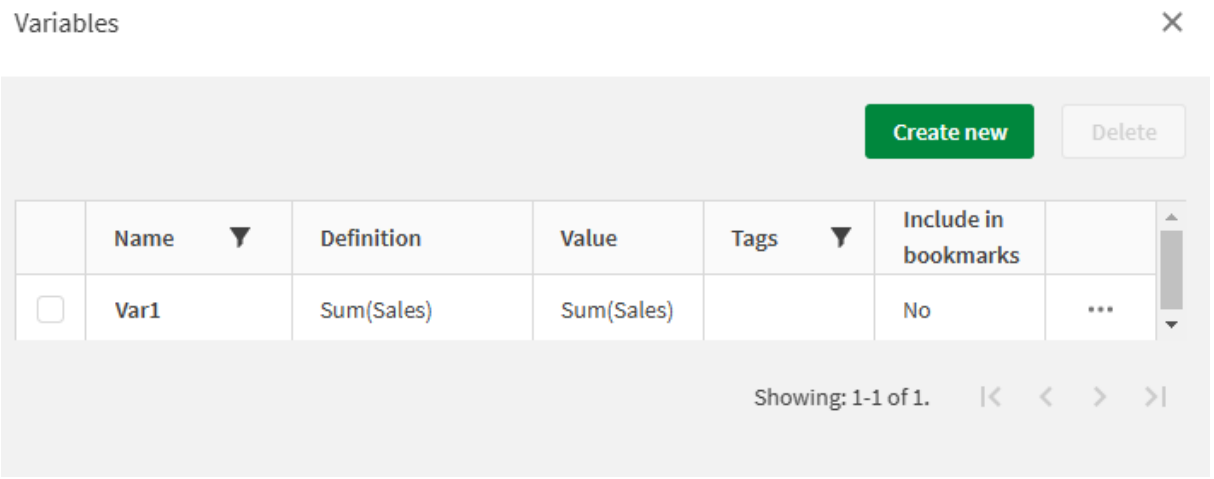
In dit voorbeeld wordt de inhoud van de variabele *Var1* ingevoegd in de omringende uitdrukking voordat die inhoud wordt geëvalueerd.

#### Resultaat A - Variabele definitie zonder gelijkteken

Als *Var1* wordt gedefinieerd als *Sum(Sales)*, zou het er na uitbreiding als volgt uitzien (houd er rekening mee dat de onderstaande configuratie een gedeeltelijk geëvalueerde uitdrukking is):

```
{<Year={2024}>} Sum(Sales)
```

*Het dialoogvenster Variabelen toont de definitie en waarde van de variabele Var1, als er geen gelijkteken aan het begin van de definitie van de variabele staat.*

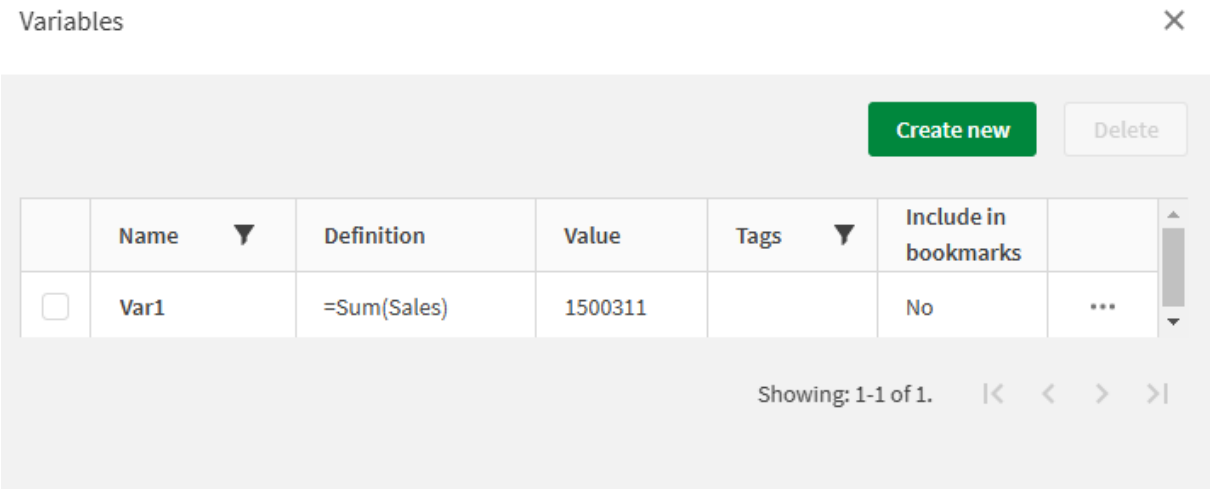



In dit voorbeeld wordt dus de buitenste set-uitdrukking `{<Year={2024}>}` toegepast.

**Resultaat B - Variabele definitie zonder gelijkteken**

Aan de andere kant, als de definitie in *Var1* begint met een gelijkteken - namelijk *=Sum(Sales)* - wordt de buitenste set-uitdrukking niet toegepast. Dit komt doordat het gelijkteken ervoor zorgt dat de variabele uitdrukking van tevoren en geïsoleerd wordt berekend. Dit leidt tot een resultaat dat lijkt op het eerste voorbeeld, waarin er een buitenste set-uitdrukking is die moet worden geëvalueerd naast een getal, wat geen bruikbaar resultaat oplevert.

*Het dialoogvenster Variabelen toont de definitie en waarde van de variabele Var1, als er een gelijkteken aan het begin van de definitie van de variabele staat.*



# Zelfstudie - Een set-uitdrukking maken

U kunt set-uitdrukkingen maken in Qlik Sense om gegevensanalyse te ondersteunen. In deze context wordt de analyse vaak set-analyse genoemd. Set-analyse biedt een manier om een bereik te definiëren dat verschilt van de set met records die wordt gedefinieerd door de huidige selectie in een app.

## Wat u gaat leren

Deze zelfstudie bevat de gegevens en diagramuitdrukkingen om set-uitdrukkingen te maken met set-Id's, set-operatoren en set-modificaties.

## Wie moet deze zelfstudie volgen

Deze zelfstudie is bedoeld voor app-ontwikkelaars die vertrouwd zijn met het werken met de scripteditor en diagramuitdrukkingen.

## Wat u eerst moet doen

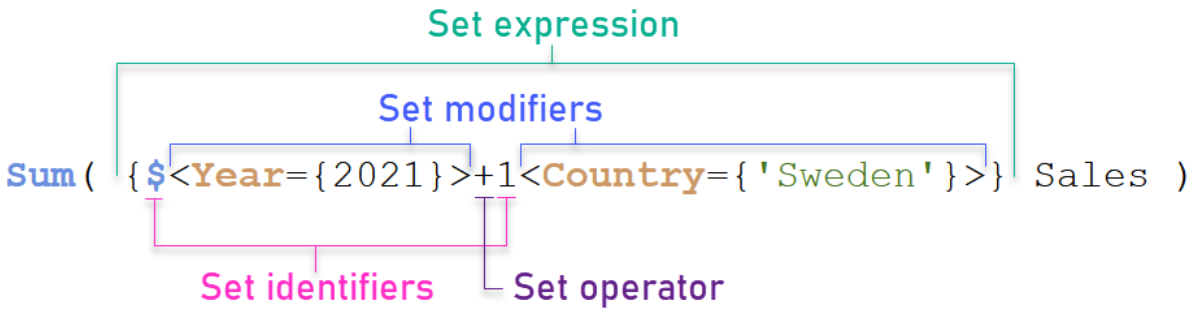
Een toewijzing voor Professional-toegang tot Qlik Sense Enterprise, waarmee u gegevens kunt laden en apps kunt maken.

- Set-analyse deel 1: Inleiding voor beginners
- Set-analyse, deel 2

## Elementen in een set-uitdrukking

Set-uitdrukkingen zijn ingesloten in een aggregatiefunctie, zoals `Sum()`, `Max()`, `Min()`, `Avg()` of `Count()`. Set-uitdrukkingen worden gemaakt van bouwblokken die we elementen noemen. Deze elementen zijn set-id's, set-operatoren en set-modificaties.

*Elementen in een set-uitdrukking*




De bovenstaande set-uitdrukking is bijvoorbeeld gebaseerd op de aggregatie `Sum(Sales)`. De set-uitdrukking is ingesloten tussen de buitenste accolades: `{ }`

De eerste operand in de uitdrukking is: `$<Year={2021}>`

Deze operand retourneert verkopen voor het jaar `2021` voor de huidige selectie. De modificatie, `<Year={2021}>`, bevat de selectie van het jaar 2021. De set-id `$` geeft aan dat de set-uitdrukking is gebaseerd op de huidige selectie.

De tweede operand in de uitdrukking is: `1<Country={'Sweden'}>`

Deze operand retourneert Sales voor Sweden. De modificatie, `<Country={'Sweden'}>`, bevat de selectie van het land Sweden. De set-id `1` geeft aan dat de selecties die in de app zijn gemaakt, worden genegeerd.

Tot slot geeft de set-operator `+` aan dat de uitdrukking een set retourneert die bestaat uit de records die kunnen toebehoren aan elk van de twee set-operanden.

## Zelfstudie - Een set-uitdrukking maken

Voer de volgende procedures uit om de set-uitdrukkingen te maken die in deze zelfstudie worden getoond.

### Een nieuwe app maken en gegevens laden

**Doe het volgende:**

1. Maak een nieuwe app.
2. Klik op **Scripteditor**. U kunt ook het navigatiemenu van de app openen en **Editor voor laden van gegevens** selecteren.
3. Maak een nieuwe sectie in de **Editor voor het laden van gegevens**.
4. Kopieer de volgende gegevens en plak deze in de nieuwe sectie: *Gegevens voor zelfstudie van set-uitdrukkingen (page 349)*
5. Klik op **Gegevens laden**. De gegevens worden geladen als een inline load.

### Set-uitdrukkingen maken met modificaties

De set-modificatie bestaat uit een of meer veldnamen, elk gevolgd door een selectie die op het veld wordt toegepast. De modificatie is omsloten door punthaken. Bijvoorbeeld in deze set-uitdrukking:

```
Sum ( {<Year = {2015}>} Sales )
```

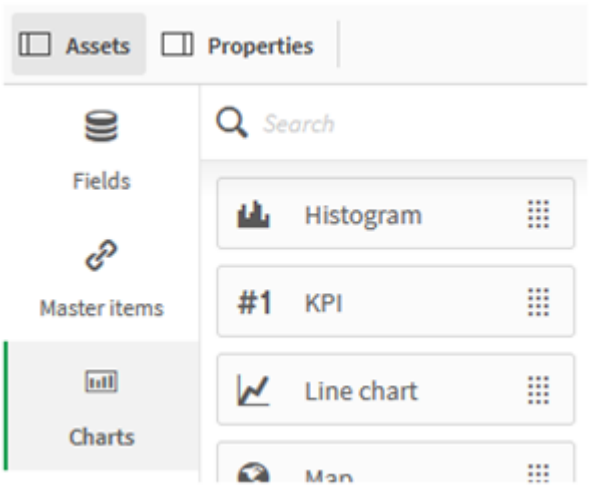
De modificatie is:

```
<Year = {2015}>
```

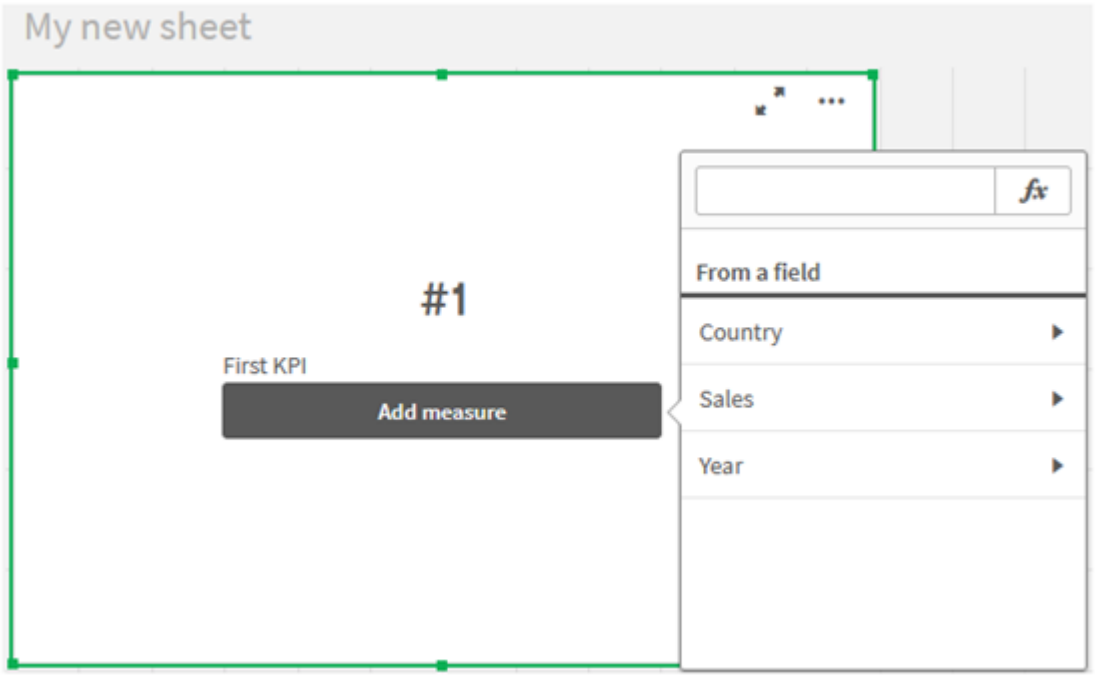
Deze modificatie geeft aan dat gegevens uit het jaar 2015 worden geselecteerd. De accolades die de modificatie omsluiten, duiden op een set-uitdrukking.

**Doe het volgende:**

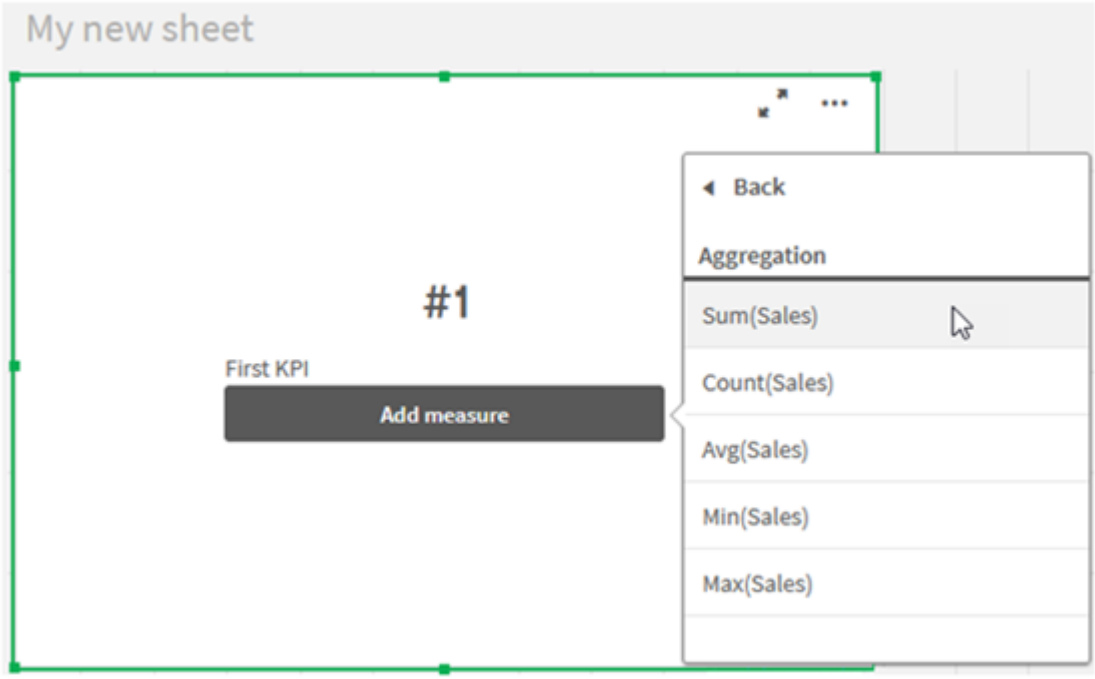
1. Open in een werkblad het venster **Bedrijfsmiddelen** in de navigatiebalk en klik vervolgens op **Diagrammen**.



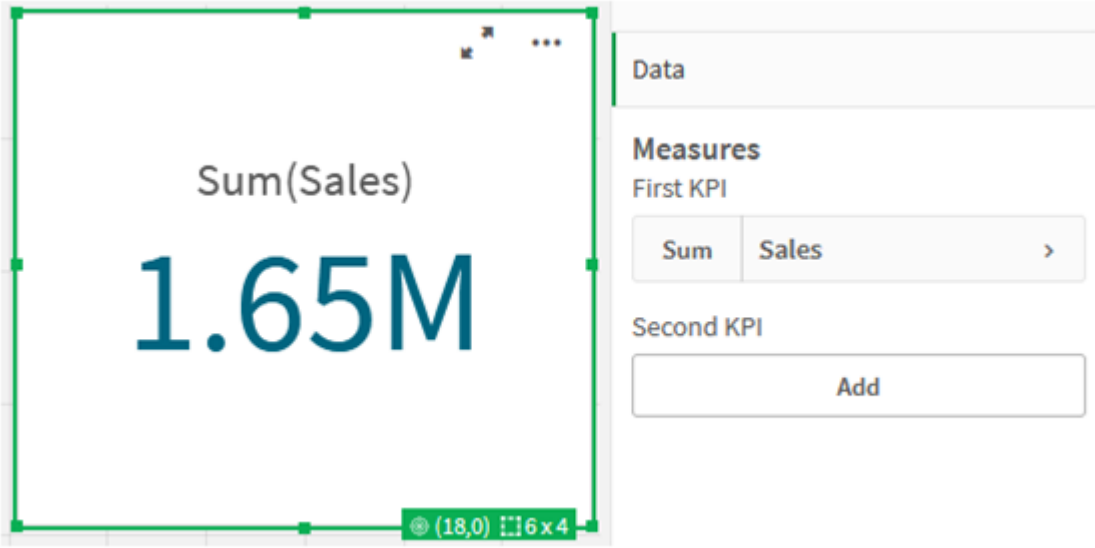

2. Sleep een **KPI** naar een werkblad en klik vervolgens op **Meting toevoegen**.



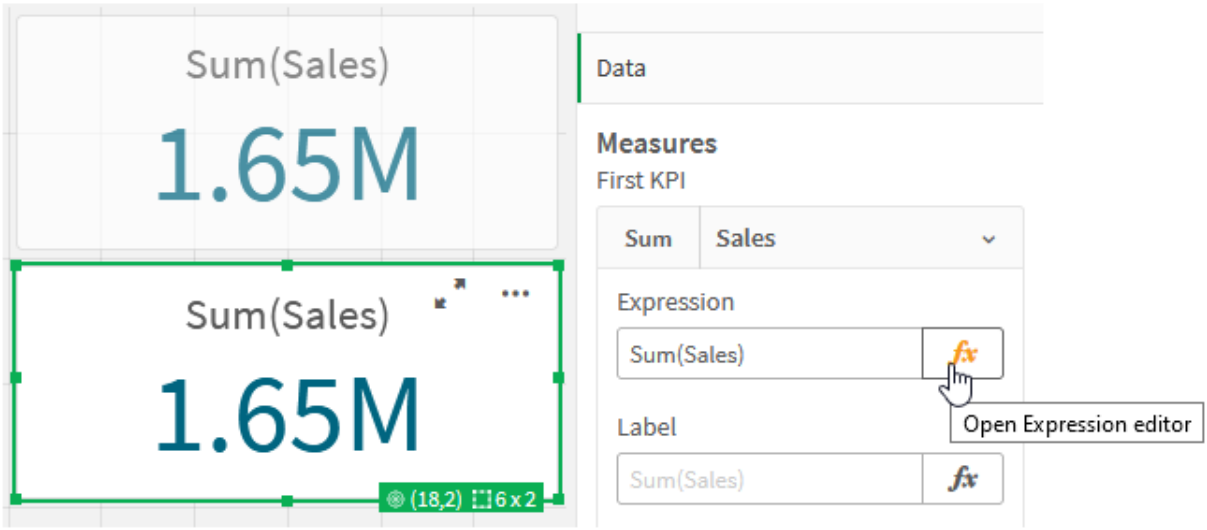

3. Klik op `Sales` en selecteer vervolgens `Sum(Sales)` voor de aggregatie.



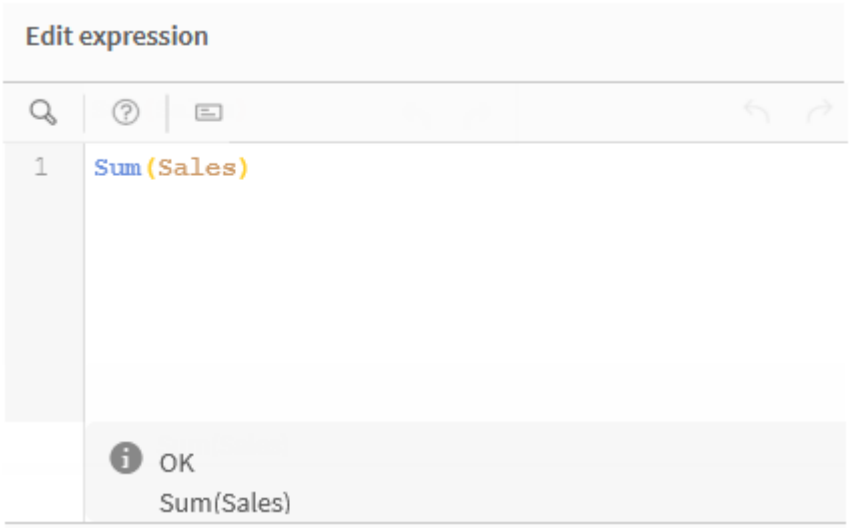

De KPI geeft de som van verkopen voor alle jaren weer.



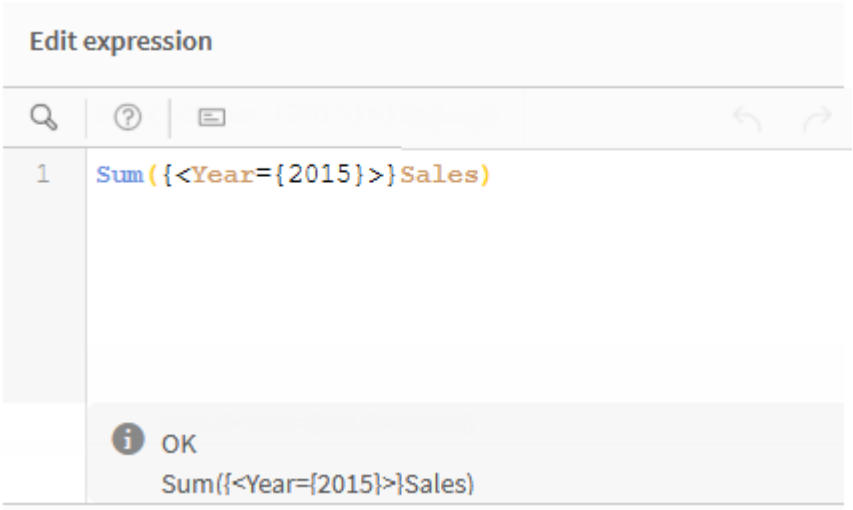

4. Kopieer en plak de KPI om een nieuwe KPI te maken.
5. Klik op de nieuwe KPI, klik op **Verkoop** onder **Metingen** en klik vervolgens op **Uitdrukkingseditor openen**.



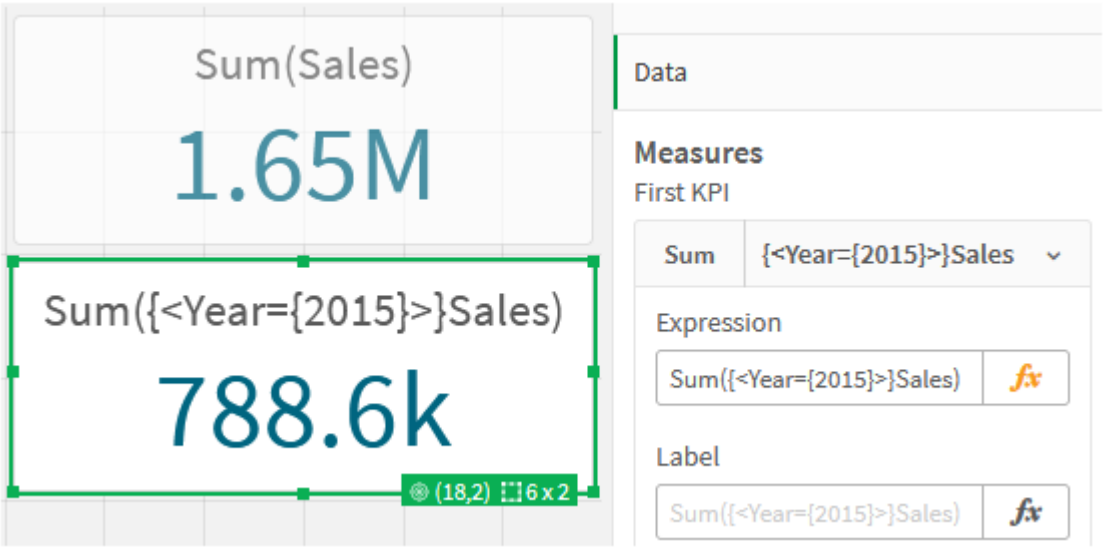

De uitdrukkingseditor wordt geopend met de aggregatie `Sum(Sales)`.




6. Maak in de uitdrukkingseditor een uitdrukking voor de som van Sales alleen voor 2015:
   i. Voeg accolades toe om een set-uitdrukking aan te duiden: `Sum({}Sales)`
   i. Voeg punthaken toe om een set-modificatie aan te duiden: `Sum({<>}Sales)`
   ii. Voeg tussen punthaken het selectieveld toe, in dit geval is het veld `Year`, gevolgd door een gelijkteken. Sluit vervolgens 2015 in tussen nog een paar accolades. De resulterende set-modificatie is: `{<Year={2015}>}`.
   De volledige uitdrukking is:
   `Sum({<Year={2015}>}Sales)`

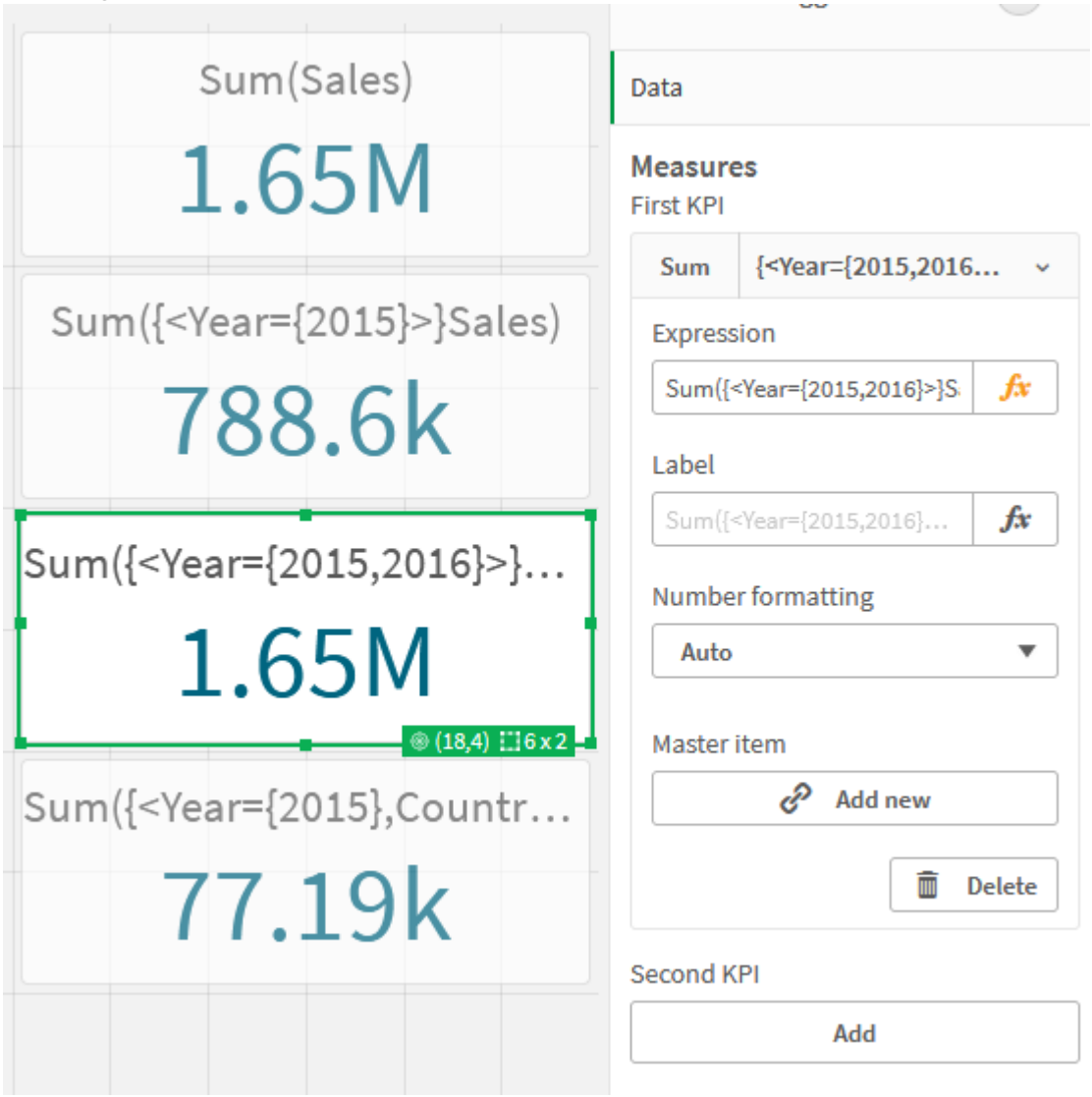



iii. Klik op **Toepassen** om de uitdrukking op te slaan en de uitdrukkingseditor te sluiten. De som van Sales voor 2015 wordt weergegeven in de KPI.




7. Maak nog twee KPI's met de volgende uitdrukkingen:
   `Sum({<Year={2015,2016}>}Sales)`
   De modificatie in het bovenstaande is `<Year={2015,2016}>`. De uitdrukking retourneert de som van Sales voor 2015 en 2016
   `Sum({<Year={2015},Country={'Germany'}>} Sales)`
   De modificatie in het bovenstaande is `<Year={2015}, Country={'Germany'}>`. De uitdrukking retourneert de som van Sales voor 2015, waarbij 2015 kruist met Germany.

*KPI's die gebruikmaken van set-modificaties*




## Set-id's toevoegen

De bovenstaande set-uitdrukkingen gebruiken huidige selecties als basis, omdat geen id werd gebruikt. Voeg vervolgens id's toe om het gedrag op te geven wanneer u selecties maakt.

**Doe het volgende:**

Maak of kopieer op uw werkblad de volgende set-uitdrukkingen:

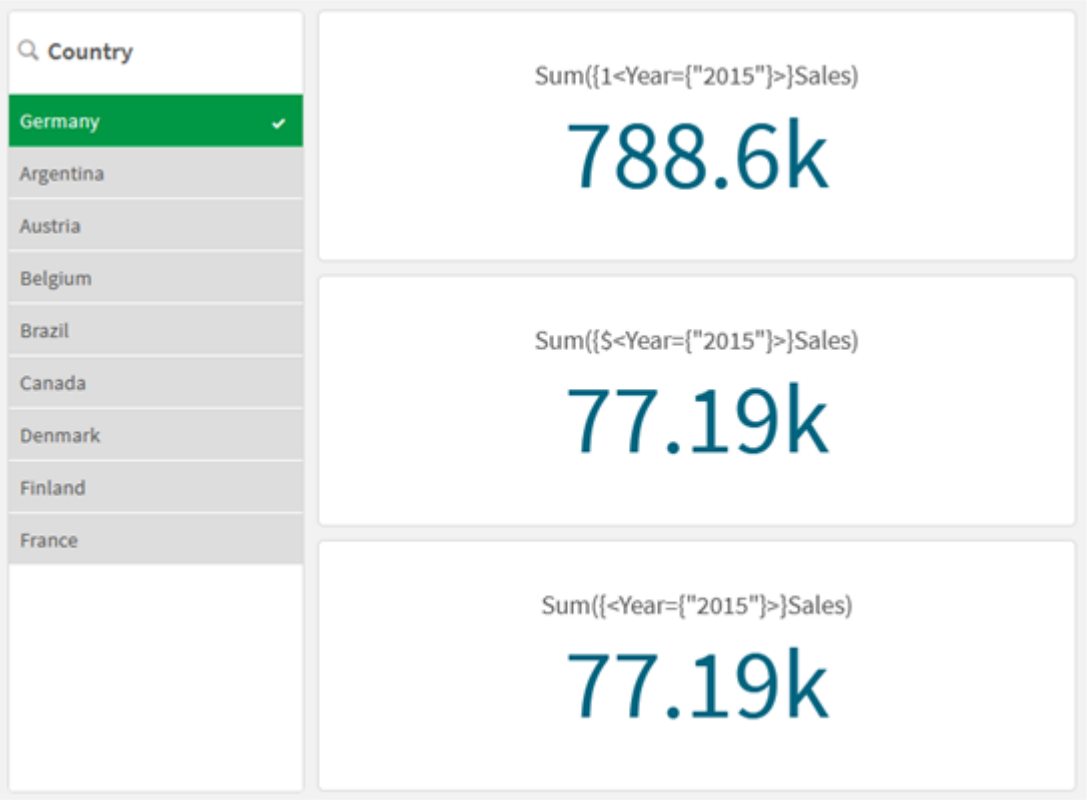
```
Sum({$<Year={"2015"}>}Sales)
```

De id `$` baseert de set-uitdrukking op de huidige selecties die in de gegevens zijn gemaakt. Dit is ook het standaardgedrag wanneer een id niet wordt gebruikt.

```
Sum({1<Year={"2015"}>}Sales)
```

Met de id `1` houdt de aggregatie van `Sum(Sales)` op 2015 geen rekening met de huidige selectie. De waarde van de aggregatie wijzigt niet wanneer de gebruiker andere selecties maakt. Als hieronder bijvoorbeeld Germany is geselecteerd, wordt de waarde voor het aggregatietotaal van 2015 niet gewijzigd.

*KPI's die gebruikmaken van set-modificators en id's*




## Operatoren toevoegen

Set-operatoren worden gebruikt voor het opnemen, uitsluiten of het doorkruisen van gegevensverzamelingen. Alle operatoren gebruiken sets als operanden en leveren een set op als resultaat.

U kunt set-operatoren in twee verschillende situaties gebruiken:

- Om de bewerking Set uit te voeren voor set-id's die een set met records in de gegevens vertegenwoordigen.
- Om de bewerking Set uit te voeren voor elementsets, voor de veldwaarden of in een set-modificatie.

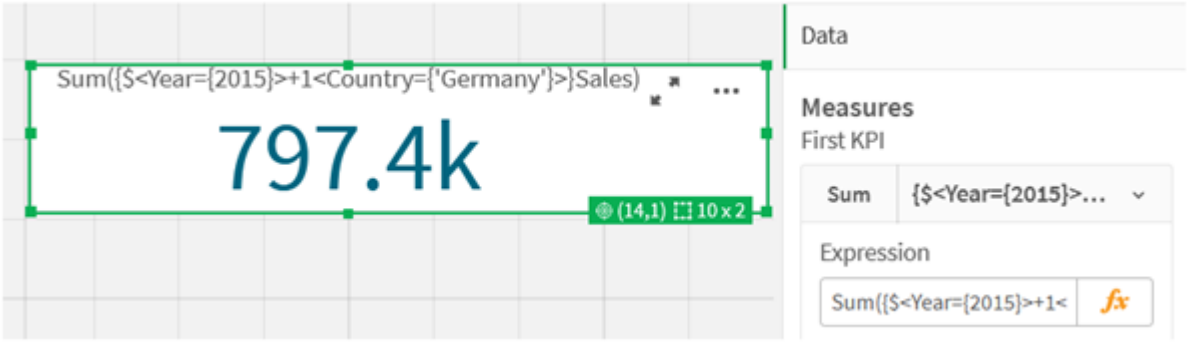
**Doe het volgende:**

Maak of kopieer op uw werkblad de volgende set-uitdrukking:

```
Sum({$<Year={2015}>+1<Country={'Germany'}>}Sales)
```

Hier bewerkstelligt de plustekenoperator `(+)` een samenvoeging (union) van de gegevensverzamelingen voor 2015 en Germany. Zoals hierboven werd aangegeven met betrekking tot set-id's, betekent de dollarteken-id `($)` dat de huidige selecties voor de eerste operand, `<Year={2015}>` worden gebruikt. De id `1` betekent dat de selectie wordt genegeerd voor de tweede operand, `<Country={'Germany'}>`.

*KPI die gebruikmaakt van de operator plusteken (+)*

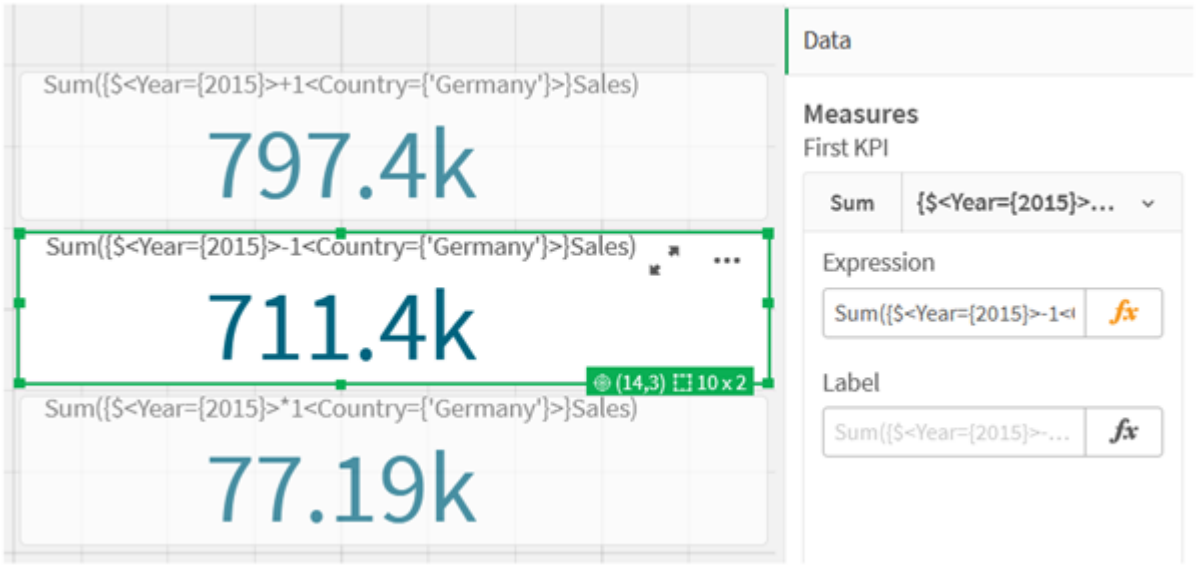



Gebruik anders een minteken `(-)` om een gegevensverzameling te retourneren die bestaat uit de records die behoren bij 2015 maar niet bij Germany. Of gebruik een sterretje `(*)` om een verzameling te retourneren van de records die behoren bij beide verzamelingen.

```
Sum({$<Year={2015}>-1<Country={'Germany'}>}Sales)
```

```
Sum({$<Year={2015}>*1<Country={'Germany'}>}Sales)
```

*KPI's die operators gebruiken*




## Gegevens voor zelfstudie van set-uitdrukkingen

### Load-script

Laad de volgende gegevens als een inline-load en maak vervolgens de diagramuitdrukkingen in de zelfstudie.

```
//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
];
```

## Syntaxis voor set-uitdrukkingen

De volledige syntaxis (het optionele gebruik van standaard haakjes voor de definitie van prioriteit is niet inbegrepen) wordt beschreven met Backus-Naur-formalisme:

```
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 Algemene syntaxis voor diagramuitdrukkingen

De volgende algemene syntaxisstructuur kan worden gebruikt voor diagramuitdrukkingen, met veel optionele parameters:

```
expression ::= ( constant | expressionname | operator1 expression | expression operator2
expression | function | aggregation function | (expression ) )
```

waarbij:

**constant** een tekenreeks (tekst, datum of tijd) is die tussen enkele rechte aanhalingstekens staat, of een getal. Constanten worden geschreven zonder scheidingsteken voor duizendtallen en met een punt als scheidingsteken voor decimalen.

**expressionname** is de naam (het label) is van een andere uitdrukking in hetzelfde diagram.

**operator1** is een unaire operator (werkt op één uitdrukking, de uitdrukking rechts van de operator).

**operator2** is een binaire operator (werkt op twee uitdrukkingen, een aan elke zijde).

```
function ::= functionname ( parameters )
parameters ::= expression { , expression }
```
Het aantal en de typen parameters zijn niet willekeurig. Deze zijn afhankelijk van de gebruikte functie.

```
aggregationfunction ::= aggregationfunctionname ( parameters2 )
parameters2 ::= aggrexpression { , aggrexpression }
```
Het aantal en de typen parameters zijn niet willekeurig. Deze zijn afhankelijk van de gebruikte functie.

## 6.4 Algemene syntaxis voor aggregaties

De volgende algemene syntaxisstructuur kan worden gebruikt voor aggregaties, met veel optionele parameters:

```
aggrexpression ::= ( fieldref | operator1 aggrexpression | aggrexpression operator2
aggrexpression | functioninaggr | ( aggrexpression ) )
```
**fieldref** is een veldnaam.

```
functionaggr ::= functionname ( parameters2 )
```
Uitdrukkingen en functies kunnen zo naar wens worden genest. Zolang een **fieldref** is omsloten door precies één aggregatiefunctie en de uitdrukking een interpreteerbare waarde oplevert, verschijnt er geen foutmelding in Qlik Sense.

# 7 Operatoren

In dit hoofdstuk komen de operatoren die u in Qlik Sense kunt gebruiken aan de orde. Er zijn twee soorten operatoren:

- Unaire operatoren (hebben slechts één operand)
- Binaire operatoren (hebben twee operanden)

De meeste operatoren zijn binair.

De volgende operatoren kunnen worden gedefinieerd:

- Bit-operatoren
- Logische operatoren
- Numerieke operatoren
- Relationele operatoren
- Tekenreeksoperatoren

## 7.1 Bit-operatoren

Alle bit-operatoren zetten de operanden om in (of kappen ze af tot) ondertekende gehele getallen (32-bits) en retourneren het resultaat op dezelfde wijze. Alle bewerkingen worden bit voor bit uitgevoerd. Als een operand niet kan worden geïnterpreteerd als een getal, zal de bewerking NULL retourneren.

Bit-operatoren

| Operator | Volledige naam | Beschrijving |
|---|---|---|
| bitnot | Inverse op bitniveau. | Unaire operator. De bewerking retourneert de logische inverse van de operand die bit voor bit wordt uitgevoerd.<br>**Voorbeeld:**<br>`bitnot 17` retourneert -18 |
| bitand | AND op bitniveau. | De bewerking levert de logische AND op van de operanden die bit voor bit worden uitgevoerd.<br>**Voorbeeld:**<br>`17 bitand 7` retourneert 1 |

| Operator | Volledige naam | Beschrijving |
|---|---|---|
| bitor | OR op bitniveau. | De bewerking levert de logische OR op van de operanden die bit voor bit worden uitgevoerd.<br>**Voorbeeld:**<br>`17 bitor 7` retourneert 23 |
| bitxor | XOR op bitniveau. | De bewerking retourneert de logische exclusieve OR van de operanden die bit voor bit worden uitgevoerd.<br>**Voorbeeld:**<br>`17 bitxor 7` retourneert 22 |
| >> | Bit naar rechts verschuiven. | De bewerking retourneert de eerste operand die naar rechts is verschoven. Het aantal stappen wordt gedefinieerd in de tweede operand.<br>**Voorbeeld:**<br>`8 >> 2` retourneert 2 |
| << | Bit naar links verschuiven. | De bewerking retourneert de eerste operand die naar links is verschoven. Het aantal stappen wordt gedefinieerd in de tweede operand.<br>**Voorbeeld:**<br>`8 << 2` retourneert 32 |

## 7.2 Logische operatoren

Alle logische operatoren interpreteren de operanden logisch en geven True (-1) of False (0) als resultaat.

Logische operatoren

| Operator | Beschrijving |
|---|---|
| not | Logische inverse. Een van de weinige unaire operatoren. De bewerking retourneert de logische inverse van de operand. |
| and | Logische EN. De bewerking retourneert de logische EN van de operanden. |
| or | Logische OF. De bewerking retourneert de logische OF van de operanden. |
| Xor | Logische uitsluitende OF. De bewerking retourneert de logische OF van de operanden. Met andere woorden, net als de logische OF maar met dat verschil dat het resultaat False is als beide operanden True zijn. |

## 7.3 Numerieke operatoren

Alle numerieke operatoren gebruiken de numerieke waarden van de operanden en geven een numerieke waarde als resultaat.

Numerieke operatoren

| Operator | Beschrijving |
|---|---|
| + | Teken voor positief getal (unaire operator) of rekenkundige optelling. De binaire bewerking retourneert de som van de twee operanden. |
| - | Teken voor negatief getal (unaire operator) of rekenkundige aftrekking. De unaire bewerking retourneert de operand vermenigvuldigd met -1. De binaire bewerking retourneert het verschil van de twee operanden. |
| * | Rekenkundige vermenigvuldiging. De bewerking retourneert het product van de twee operanden. |
| / | Rekenkundige deling. De bewerking retourneert de verhouding van de twee operanden. |

## 7.4 Relationele operatoren

Alle relationele operatoren vergelijken de waarden van operanden en retourneren als resultaat True (-1) of False (0). Alle relationele operatoren zijn binair.

Relationele operatoren

| Operator | Beschrijving |
|---|---|
| < | Kleiner dan. Er wordt een numerieke vergelijking gemaakt als beide operanden numeriek geïnterpreteerd kunnen worden. De operator retourneert de logische waarde van het resultaat van de vergelijking. |
| <= | Kleiner dan of gelijk aan. Er wordt een numerieke vergelijking gemaakt als beide operanden numeriek geïnterpreteerd kunnen worden. De operator retourneert de logische waarde van het resultaat van de vergelijking. |
| > | Groter dan. Er wordt een numerieke vergelijking gemaakt als beide operanden numeriek geïnterpreteerd kunnen worden. De operator retourneert de logische waarde van het resultaat van de vergelijking. |
| >= | Groter dan of gelijk aan. Er wordt een numerieke vergelijking gemaakt als beide operanden numeriek geïnterpreteerd kunnen worden. De operator retourneert de logische waarde van het resultaat van de vergelijking. |

| Operator | Beschrijving |
|---|---|
| **=** | Gelijk aan. Er wordt een numerieke vergelijking gemaakt als beide operanden numeriek geïnterpreteerd kunnen worden. De operator retourneert de logische waarde van het resultaat van de vergelijking. |
| **<>** | Niet gelijk aan. Er wordt een numerieke vergelijking gemaakt als beide operanden numeriek geïnterpreteerd kunnen worden. De operator retourneert de logische waarde van het resultaat van de vergelijking. |
| **precedes** | In tegenstelling tot de operator **<** wordt er geen poging gedaan de argumentwaarden numeriek te interpreteren voordat er wordt vergeleken. De bewerking retourneert Waar als de waarde links van de operator een tekstrepresentatie heeft die in een ASCII-vergelijking vóór de tekstrepresentatie van de waarde aan de rechterkant komt.<br><br>**Voorbeeld:**<br><br>`'1 ' precedes ' 2'` retourneert FALSE<br><br>`' 1' precedes ' 2'` retourneert TRUE<br><br>aangezien de ASCII-waarde van een spatie (' ') van minder waarde is dan de ASCII-waarde van een getal.<br><br>Vergelijk dit met:<br><br>`'1 ' < ' 2'` retourneert TRUE<br><br>`' 1' < ' 2'` retourneert TRUE |

| Operator | Beschrijving |
|---|---|
| **follows** | In tegenstelling tot de operator **<** wordt er geen poging gedaan de argumentwaarden numeriek te interpreteren voordat er wordt vergeleken. De bewerking retourneert Waar als de waarde links van de operator een tekstrepresentatie heeft die in een ASCII-vergelijking vóór de tekstrepresentatie van de waarde aan de rechterkant komt.<br><br>**Voorbeeld:**<br><br>`' 2' follows '1'` retourneert FALSE<br><br>' 2' follows ' 1' retourneert TRUE<br><br>aangezien de ASCII-waarde van een spatie (' ') van minder waarde is dan de ASCII-waarde van een getal.<br><br>Vergelijk dit met:<br><br>`' 2' > ' 1'` retourneert TRUE<br><br>`' 2' > '1 '` retourneert TRUE |

## 7.5 Tekenreeksoperatoren

Er zijn twee tekenreeksoperatoren. De eerste maakt gebruik van de tekenreekswaarden van de operanden en retourneert een tekenreeks. De andere vergelijkt de operanden en retourneert een booleaanse waarde waarmee de overeenkomst wordt aangegeven.

### &

Tekenreeksverbinding. De bewerking retourneert een tekenreeks, die bestaat uit de achter elkaar geplaatste tekenreeksen van de twee operanden.

**Voorbeeld:**

`'abc' & 'xyz'` retourneert 'abcxyz'

### like

Tekenreeksvergelijking met jokertekens. De bewerking retourneert de booleaanse waarde True (-1) als de tekenreeks vóór de operator overeenkomt met de tekenreeks achter de operator. De tweede tekenreeks kan de jokertekens * (elk aantal willekeurige tekens) of ? (een enkel willekeurig karakter) bevatten.

**Voorbeeld:**

`'abc' like 'a*'` retourneert True (-1)

`'abcd' like 'a?c*'` retourneert True (-1)

`'abc' like 'a??bc'` retourneert False (0)

# 8 Script- en diagramfuncties

Transformeer en aggregeer gegevens met functies in load-scripts voor gegevens en diagramuitdrukkingen.

Veel functies kunnen op dezelfde wijze worden gebruikt in zowel scripts voor het laden van gegevens als diagramuitdrukkingen, maar er is een aantal uitzonderingen:

- Sommige functies kunnen alleen worden gebruikt in scripts voor het laden van gegevens, aangeduid met - scriptfunctie.
- Sommige functies kunnen alleen worden gebruikt in diagramuitdrukkingen, aangeduid met - diagramfunctie.
- Sommige functies kunnen worden gebruikt in zowel scripts voor het laden van gegevens als diagramuitdrukkingen, maar met verschillen in parameters en toepassing. Deze worden beschreven in aparte onderwerpen aangeduid met - scriptfunctie of - diagramfunctie.

## 8.1 Analytische verbindingen voor uitbreidingen aan de serverzijde (SSE)

Functies die zijn ingeschakeld door analytische verbindingen zijn alleen zichtbaar als u de analytische verbindingen hebt geconfigureerd en Qlik Sense is gestart. U kunt analytische verbindingen configureren in de QMC. Zie Analytische verbindingen maken.

U kunt de analytische verbindingen configureren in de . Zie in de gids het onderwerp "Analytische verbindingen maken".QMCQlik Sense-sites beheren

In Qlik Sense Desktop configureert u de analytische verbindingen door het bestand *Settings.ini* te bewerken, zie het onderwerp “Configuring analytic connections in Qlik Sense Desktop” (“Analytische verbindingen configureren in ”) in de handleiding Qlik Sense Desktop.

## 8.2 Aggregatiefuncties

De functiegroep met zogeheten aggregatiefuncties bestaat uit functies die meerdere veldwaarden als invoer aannemen en een enkel resultaat per groep retourneren, waarbij de groepering wordt gedefinieerd door een diagramdimensie of een **group by**-clausule in de scriptopdracht.

Aggregatiefuncties zijn onder meer **Sum()**, **Count()**, **Min()**, **Max()** en nog veel meer.

De meeste aggregatiefuncties kunnen zowel worden gebruikt in het script voor het laden van gegevens als in diagramuitdrukkingen, maar de syntaxis verschilt.

**Beperkingen:**

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Bij het toewijzen van een naam aan een entiteit moet u niet dezelfde naam geven aan meer dan één veld, variabele of meting. Er geldt een strikte volgorde van prioriteit voor het oplossen van conflicten tussen entiteiten met identieke namen. Deze volgorde wordt weerspiegeld in objecten of contexten waarin deze entiteiten worden gebruikt. Deze volgorde van prioriteit is als volgt:

- Binnen een aggregatiefunctie krijgt een veld voorrang op een variabele. Metinglabels zijn niet relevant in aggregaties en krijgen geen prioriteit.
- Buiten een aggregatiefunctie krijgt een metinglabel voorrang op een variabele, die weer voorrang krijgt op een veld.
- Daarnaast kan een meting buiten een aggregatie opnieuw gebruikt geworden door te verwijzen naar het label, tenzij het label is berekend. In die situatie krijgt de meting minder prioriteit om het risico te voorkomen dat de uitdrukking naar zichzelf verwijst, en in dit geval wordt de naam altijd eerst geïnterpreteerd als een metinglabel, vervolgens als een veldnaam en daarna als een variabelenaam.

## Aggregatiefuncties gebruiken in het script voor het laden van gegevens

Aggregatiefuncties kunnen alleen worden gebruikt binnen de opdrachten **LOAD** en **SELECT**.

## Aggregatiefuncties gebruiken in diagramuitdrukkingen

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Een aggregatiefunctie aggregeren over de set mogelijke records die wordt gedefinieerd door de selectie. Maar er kan een alternatieve set records worden gedefinieerd door gebruik te maken van een set-uitdrukking bij set-analyse.

## Hoe aggregaties worden berekend

Een aggregatie doorloopt de records van een specifieke tabel en aggregeert de records in die tabel. Zo wordt bijvoorbeeld met **Aantal**(<Field>) het aantal records geteld in de tabel waar <Field> voorkomt. Als u alleen de unieke veldwaarden wilt aggregeren, moet u de **distinct**-clausule gebruiken, zoals **Aantal** (**distinct** <Field>).

Als de aggregatiefunctie velden uit verschillende tabellen bevat, doorloopt de aggregatiefunctie de records van het vectorproduct van de velden. Dit zorgt voor prestatieverlies, en om die reden moet u zulke aggregaties vermijden, met name als u grote aantallen gegevens gebruikt.

## Aggregatie van sleutelvelden

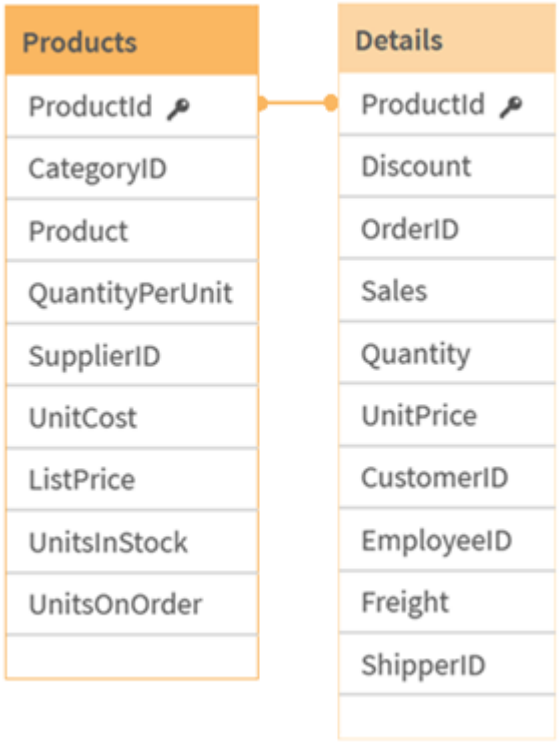
Door de manier waarop aggregaties worden berekend, kunt u geen sleutelvelden aggregeren omdat daarbij niet duidelijk is welke tabel voor de aggregatie moet worden gebruikt. Als bijvoorbeeld het veld <Key> twee tabellen aan elkaar koppelt, is niet duidelijk of **Count**(<Key>) het aantal records van de eerste of tweede tabel moet retourneren.

Maar als u het component **distinct** gebruikt, is de aggregatie op de juiste manier gedefinieerd en kan deze vervolgens worden berekend voor een sleuteveld dat binnen twee tabellen is gekoppeld.

Als u een sleutelveld in een aggregatiefunctie gebruikt zonder het component **afzonderlijk** retourneert Qlik Sense een aantal waar u mogelijk niets aan hebt. De oplossing is in dit geval om het component **distinct** of een kopie van de sleutel te gebruiken – een kopie die alleen in één tabel voorkomt.

In de volgende tabellen is *ProductId* bijvoorbeeld de sleutel tussen de tabellen.

*ProductId-sleutel tussen Products- en Details-tabellen*




*Count(ProductId)* kan zowel geteld worden in de tabel *Products* (die maar één record per product heeft – *ProductId* is de primaire sleutel) of het kan in de *Details*-tabel geteld worden (die waarschijnlijk meerdere records per product heeft). Als u het aantal verschillende producten wilt tellen, moet u *Count(distinct ProductId)* gebruiken. Als u het aantal rijen in een specifieke tabel wilt tellen, moet u de sleutel niet gebruiken.

### Aggregaties van sleutelvelden in drie of meer tabellen

Het prefix **distinct** werkt alleen met sleutelvelden die maximaal twee tabellen koppelen. Bij het groeperen van een aggregatie met een sleutelveld dat in drie of meer tabellen aanwezig is, wordt NULL geretourneerd voor een veld dat frequentiegegevens voor een veld vereist. In het geval van een sleutelveld dat drie of meer tabellen koppelt, moet in plaats daarvan een niet-sleutelkopie van het veld worden gebruikt.

## Basisaggregatiefuncties

### Overzicht basisaggregatiefuncties

De basisaggregatiefuncties zijn een groep van de meest gebruikte aggregatiefuncties.

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

### Basisaggregatiefuncties gebruiken in het script voor het laden van gegevens

#### FirstSortedValue

**FirstSortedValue()** retourneert de waarde van de uitdrukking die is opgegeven in **value** die overeenkomt met het resultaat van het sorteren van **sort_weight**-argument, bijvoorbeeld de naam van het product met de laagste prijs per eenheid. U kunt de nde waarde in de sorteervolgorde opgeven in **rank**. Als meer dan één uitkomst dezelfde **sort_weight** voor de opgegeven **rank** deelt, retourneert de functie NULL. De gesorteerde waarden worden herhaald over een aantal records, zoals is vastgelegd door een **group by**-clausule, of geaggregeerd over een volledige gegevensverzameling als er geen **group by**-clausule is gedefinieerd.

```
FirstSortedValue ([ distinct ] expression, sort_weight [, rank ])
```

#### Max

**Max()** retourneert de hoogste numerieke waarde van de geaggregeerde gegevens in de uitdrukking, zoals gedefinieerd door een **group by**-clausule. Door een **rank** n op te geven, wordt de n-de hoogste waarde gezocht.

```
Max ( expression[, rank])
```

#### Min

**Min()** retourneert de laagste numerieke waarde van de geaggregeerde gegevens in de uitdrukking, zoals gedefinieerd door een **group by**-clausule. Door een **rank** n op te geven, wordt de n-de laagste waarde gezocht.

```
Min ( expression[, rank])
```

#### Mode

**Mode()** retourneert de vaakst voorkomende waarde, de moduswaarde, van de geaggregeerde gegevens in de uitdrukking, zoals gedefinieerd door een **group by**-clausule. De functie **Mode()** kan zowel numerieke waarden als tekstwaarden retourneren.

```
Mode (expression )
```

### Only

**Only()** retourneert een waarde als er één, en niet meer dan één, mogelijk resultaat is voor de geaggregeerde gegevens. Als records slechts één waarde bevatten, wordt die waarde geretourneerd, anders wordt NULL geretourneerd. Gebruik de **group by**-clausule om meerdere records te evalueren. De functie **Only()** kan numerieke waarden en tekstwaarden retourneren.

```
Only (expression )
```

### Sum

**Sum()** berekent het totaal van de waarden die zijn geaggregeerd in de uitdrukking, zoals gedefinieerd door een **group by**-clausule.

```
Sum ([distinct]expression)
```

## Basisaggregatiefuncties gebruiken in diagramuitdrukkingen

Aggregatiefuncties voor diagrammen kunnen alleen worden gebruikt op velden in diagramuitdrukkingen. De argumentuitdrukking van een aggregatiefunctie mag niet nog een andere aggregatiefunctie bevatten.

### FirstSortedValue

**FirstSortedValue()** retourneert de waarde van de uitdrukking die is opgegeven in **value** die overeenkomt met het resultaat van het sorteren van **sort_weight**-argument, bijvoorbeeld de naam van het product met de laagste prijs per eenheid. U kunt de nde waarde in de sorteervolgorde opgeven in **rank**. Als meer dan één uitkomst dezelfde **sort_weight** voor de opgegeven **rank** deelt, retourneert de functie NULL.

```
FirstSortedValue - diagramfunctie([{SetExpression}] [DISTINCT] [TOTAL [<fld
{,fld}>]] value, sort_weight [,rank])
```

### Max

**Max()** retourneert de hoogste waarde van de geaggregeerde gegevens. Door een **rank** n op te geven, wordt de n-de hoogste waarde gezocht.

```
Max - diagramfunctie([{SetExpression}] [DISTINCT] [TOTAL [<fld {,fld}>]] expr
[,rank])
```

### Min

**Min()** retourneert de laagste waarde van de geaggregeerde gegevens. Door een **rank** n op te geven, wordt de n-de laagste waarde gezocht.

```
Min - diagramfunctie([{SetExpression}] [DISTINCT] [TOTAL [<fld {,fld}>]] expr
[,rank])
```

### Mode

**Mode()** retourneert de vaakst voorkomende waarde, de moduswaarde, in de geaggregeerde gegevens. De functie **Mode()** kan zowel tekstwaarden als numerieke waarden verwerken.

```
Mode - diagramfunctie ({[SetExpression] [TOTAL [<fld {,fld}>]]} expr)
```

Only

**Only()** retourneert een waarde als er één, en niet meer dan één, mogelijk resultaat is voor de geaggregeerde gegevens. Als u bijvoorbeeld zoekt naar het enige product waarvan de prijs per eenheid =9 en er zijn meerdere producten met een prijs per eenheid van 9, dan wordt het resultaat NULL geretourneerd.

```
Only - diagramfunctie([{SetExpression}] [DISTINCT] [TOTAL [<fld {,fld}>]]
expr)
```

Sum

**Sum()** berekent het totaal van de waarden die worden geleverd door de uitdrukking of het veld voor de geaggregeerde gegevens.

```
Sum - diagramfunctie([{SetExpression}] [DISTINCT] [TOTAL [<fld {,fld}>]]
expr])
```

## FirstSortedValue

**FirstSortedValue()** retourneert de waarde van de uitdrukking die is opgegeven in **value** die overeenkomt met het resultaat van het sorteren van **sort_weight**-argument, bijvoorbeeld de naam van het product met de laagste prijs per eenheid. U kunt de nde waarde in de sorteervolgorde opgeven in **rank**. Als meer dan één uitkomst dezelfde **sort_weight** voor de opgegeven **rank** deelt, retourneert de functie NULL. De gesorteerde waarden worden herhaald over een aantal records, zoals is vastgelegd door een **group by**-clausule, of geaggregeerd over een volledige gegevensverzameling als er geen **group by**-clausule is gedefinieerd.

**Syntaxis:**

```
FirstSortedValue ([ distinct ] value, sort-weight [, rank ])
```

**Retourgegevenstypen:** dual

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value Expression | De functie vindt de waarde van de uitdrukking **value** die overeenkomt met het resultaat van de sortering **sort_weight**. |
| sort-weight Expression | De uitdrukking die de gegevens bevat die moeten worden gesorteerd. De eerste (laagste) waarde van **sort_weight** wordt gevonden, op basis waarvan de overeenkomstige waarde van de uitdrukking **value** wordt bepaald. Als u een minteken vóór **sort_weight** plaatst, retourneert de functie de laatste (hoogste) gesorteerde waarde. |
| rank Expression | Door een **rank** "n" op te geven die groter is dan 1, krijgt u de n-de gesorteerde waarde. |
| distinct | Als het woord **DISTINCT** voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd. |

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens ten minste de velden die in de resultatenkolom staan aan een werkblad in onze app toe om het resultaat te bekijken.

Als u dezelfde weergave wilt als in de onderstaande resultatenkolom, schakelt u in het eigenschappenvenster onder Sorteren over van Automatisch naar Aangepast en schakelt u numeriek en alfabetisch sorteren uit.

Scriptvoorbeelden

| Voorbeeld | Resultaat |
|---|---|
| `Temp:`<br>`LOAD * inline [`<br>`Customer\|Product\|OrderNumber\|UnitSales\|CustomerID`<br>`Astrida\|AA\|1\|10\|1`<br>`Astrida\|AA\|7\|18\|1`<br>`Astrida\|BB\|4\|9\|1`<br>`Astrida\|CC\|6\|2\|1`<br>`Betacab\|AA\|5\|4\|2`<br>`Betacab\|BB\|2\|5\|2`<br>`Betacab\|DD\|12\|25\|2`<br>`Canutility\|AA\|3\|8\|3`<br>`Canutility\|CC\|13\|19\|3`<br>`Divadip\|AA\|9\|16\|4`<br>`Divadip\|AA\|10\|16\|4`<br>`Divadip\|DD\|11\|10\|4`<br>`] (delimiter is '\|');`<br><br>`FirstSortedValue:`<br>`LOAD Customer,FirstSortedValue(Product, UnitSales)`<br>`as MyProductWithSmallestOrderByCustomer Resident`<br>`Temp Group By Customer;` | `Customer`<br>`MyProductWithSmallestOrderByCustomer`<br>`Astrida CC`<br>`Betacab AA`<br>`Canutility AA`<br>`Divadip DD`<br>De functie sorteert UnitSales van kleinst naar grootst en zoekt daarbij de waarde van Customer met de kleinste waarde van UnitSales, de kleinste bestelling.<br><br>Omdat CC overeenkomst met de kleinste bestelling (waarde van UnitSales=2) voor klant Astrida. AA komt overeen met de kleinste bestelling (4) voor klant Betacab, AA komt overeen met de grootste bestelling (8) voor klant Canutility en DD komt overeen met de grootste bestelling (10) voor klant Divadip.. |

| Voorbeeld | Resultaat |
| --- | --- |
| Stel dat de tabel **Temp** is geladen zoals in het vorige voorbeeld:<br>`LOAD Customer,FirstSortedValue(Product, -UnitSales) as MyProductWithLargestOrderByCustomer Resident Temp Group By Customer;` | `Customer MyProductWithLargestOrderByCustomer`<br>`Astrida AA`<br>`Betacab DD`<br>`Canutility CC`<br>`Divadip -`<br>Er staat een minteken voor het argument sort_weight, dus de functie sorteert de grootste eerst.<br>Omdat AA overeenkomt met de grootste bestelling (waarde van UnitSales:18) voor klant Astrida, DD overeenkomt met de grootste bestelling (12) voor klant Betacab en CC overeenkomt met de grootste bestelling (13) voor klant Canutility. Er zijn twee identieke waarden voor de grootste bestelling (16) voor klant Divadip, dus dit resulteert in een null-waarde. |
| Stel dat de tabel **Temp** is geladen zoals in het vorige voorbeeld:<br>`LOAD Customer,FirstSortedValue(distinct Product, -UnitSales) as MyProductWithSmallestOrderByCustomer Resident Temp Group By Customer;` | `Customer MyProductWithLargestOrderByCustomer`<br>`Astrida AA`<br>`Betacab DD`<br>`Canutility CC`<br>`Divadip AA`<br>Dit is hetzelfde als het vorige voorbeeld, behalve dat de kwalificatie distinct is gebruikt. Dit veroorzaakt een dubbel resultaat voor Divadip dat kan worden genegeerd, wat een null-waarde retourneert. |

## FirstSortedValue - diagramfunctie

**FirstSortedValue()** retourneert de waarde van de uitdrukking die is opgegeven in **value** die overeenkomt met het resultaat van het sorteren van **sort_weight**-argument, bijvoorbeeld de naam van het product met de laagste prijs per eenheid. U kunt de nde waarde in de sorteervolgorde opgeven in **rank**. Als meer dan één uitkomst dezelfde **sort_weight** voor de opgegeven **rank** deelt, retourneert de functie NULL.

**Syntaxis:**

```
FirstSortedValue([{SetExpression}] [DISTINCT] [TOTAL [<fld {,fld}>]] value, sort_weight [,rank])
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **value** | Uitvoerveld. De functie vindt de waarde van de uitdrukking **value** die overeenkomt met het resultaat van de sortering **sort_weight**. |
| **sort_weight** | Invoerveld. De uitdrukking die de gegevens bevat die moeten worden gesorteerd. De eerste (laagste) waarde van **sort_weight** wordt gevonden, op basis waarvan de overeenkomstige waarde van de uitdrukking **value** wordt bepaald. Als u een minteken vóór **sort_weight** plaatst, retourneert de functie de laatste (hoogste) gesorteerde waarde. |
| **rank** | Door een **rank** "n" op te geven die groter is dan 1, krijgt u de n-de gesorteerde waarde. |
| **SetExpression** | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| **DISTINCT** | Als het woord **DISTINCT** voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd. |
| **TOTAL** | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `FirstSortedValue (Product, UnitPrice)` | Retourneert het `Product` met de laagste `UnitPrice`. |
| `FirstSortedValue (Product, UnitPrice, 2)` | Retourneert het `Product` met de op één na laagste `UnitPrice`. |
| `FirstSortedValue (Customer, -UnitPrice, 2)` | Retourneert de `Customer` met het `Product` met de op één na hoogste `UnitPrice`. |
| `FirstSortedValue (Customer, -UnitPrice*UnitSales, 2)` | Retourneert de `Customer`, met de op één na hoogste verkooporderwaarde, `UnitPrice` vermenigvuldigd met de `UnitSales`. |

## Voorbeeld - Basisprincipes voor FirstSortedValue

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Customer`
  - `Product`
  - `UnitSales`
  - `UnitPrice`

### Load-script

```
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 '|');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als metingen:

- `=FirstSortedValue (Product, UnitPrice)`, om het product met de laagste eenheidsprijs te berekenen.
- `=FirstSortedValue (Product, UnitPrice, 2)`, om het product met de op één na laagste eenheidsprijs te berekenen.
- `=FirstSortedValue (Customer, -UnitPrice, 2)`, om de klant met het product met de op één na hoogste eenheidsprijs te berekenen.
- `=FirstSortedValue (Customer, UnitPrice, 3)`, om de klant met het product met de op twee na hoogste eenheidsprijs te berekenen.
- `=FirstSortedValue (Customer, -UnitPrice*UnitSales, 2)`, om de klant met de op één na hoogste verkooporderwaarde te berekenen, `UnitPrice` vermenigvuldigd met de `UnitSales`.

Results table

| FirstSortedValue (Product, UnitPrice) | FirstSortedValue (Product, UnitPrice, 2) | FirstSortedValue (Customer, -UnitPrice, 2) | FirstSortedValue (Customer, UnitPrice, 3) | FirstSortedValue (Customer, -UnitPrice*UnitSales, 2) |
|---|---|---|---|---|
| BB | BB | Betacab | - | Canutility |

De eerste kolom retourneert de waarde `BB`, omdat dit het product is dat de laagste `UnitPrice` (`9`) heeft.

De tweede kolom retourneert de waarde `BB`, omdat dit het product is dat de laagste `UnitPrice` (`10`) heeft.

De derde kolom retourneert waarde `Betacab`, dat is de `Customer` met het `Product` dat de op één na hoogste `UnitPrice` (`20`) heeft.

De vierde kolom retourneert `NULL`, omdat er twee waarden van `Customer` (`Astrida` en `Canutility`) zijn met dezelfde rangschikking (op twee na laagste) `UnitPrice` (`15`).

De vijfde kolom retourneert de `Canutility`, dit is de `Customer` met de op één na hoogste verkooporderwaarde, `UnitPrice` vermenigvuldigd met de `UnitSales` (`120`).

## Voorbeeld - scenario FirstSortedValue

Diagramuitdrukking

### Overzicht

Een gegevensverzameling bevat verkoopcijfers voor de eerste drie maanden van het jaar. De regiomanager wil de best en slechtst verkopende producten identificeren om toekomstige voorraden te plannen.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Date`
  - `SalesProduct`
  - `Category`
  - `Sales`

### Load-script

```
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
];
```

**Resultaten**

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Category`

Maak de volgende metingen:

- `=FirstSortedValue(SalesProduct, -Sales)`, om het best verkopende product in elke categorie te berekenen.
- `=FirstSortedValue(TOTAL SalesProduct, -Sales)`, om het best verkopende product te berekenen.
- `=FirstSortedValue(TOTAL SalesProduct, Sales)`, om het slechtst verkopende product te berekenen.

Results table

| Category | FirstSortedValue (SalesProduct, -Sales) | FirstSortedValue(TOTAL SalesProduct, -Sales) | FirstSortedValue(TOTAL SalesProduct, Sales) |
|---|---|---|---|
| Fruit | Apple | Apple | Carrot |
| Vegetable | Carrot | Apple | Carrot |

De resultaten geven aan dat `Apple` en `Carrot` de best verkopende producten zijn in hun respectieve categorieën, `Fruit` en `Vegetable`. In alle categorieën was `Apple` echter het best verkopende product en `Carrot` het slechtst verkopende product.

## Max

**Max()** retourneert de hoogste numerieke waarde van de geaggregeerde gegevens in de uitdrukking, zoals gedefinieerd door een **group by**-clausule. Door een **rank** n op te geven, wordt de n-de hoogste waarde gezocht.

**Syntaxis:**

```
Max ( expr [, rank])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| expr Expression | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| rank Expression | De standaardwaarde van **rank** is 1. Dit komt overeen met de hoogste waarde. Als u bij **rank** 2 opgeeft, wordt de op één na hoogste waarde geretourneerd. Als u bij **rank** 3 opgeeft, wordt de op twee na hoogste waarde geretourneerd, enzovoort. |

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens ten minste de velden die in de resultatenkolom staan aan een werkblad in onze app toe om het resultaat te bekijken.

Als u dezelfde weergave wilt als in de onderstaande resultatenkolom, schakelt u in het eigenschappenvenster onder Sorteren over van Automatisch naar Aangepast en schakelt u numeriek en alfabetisch sorteren uit.

**Voorbeeld:**

```
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;
```

Resultaattabel

| Customer | MyMax |
|---|---|
| Astrida | 18 |
| Betacab | 5 |
| Canutility | 8 |

**Voorbeeld:**

Stel dat de tabel **Temp** is geladen zoals in het vorige voorbeeld:

```
LOAD Customer, Max(UnitSales,2) as MyMaxRank2 Resident Temp Group By Customer;
```

Resultaattabel

| Customer | MyMaxRank2 |
|---|---|
| Astrida | 10 |
| Betacab | 4 |
| Canutility | - |

## Max - diagramfunctie

**Max()** retourneert de hoogste waarde van de geaggregeerde gegevens. Door een **rank** n op te geven, wordt de n-de hoogste waarde gezocht.

*Bekijk eventueel ook **FirstSortedValue** en **RangeMax**, die een functionaliteit hebben die vergelijkbaar is met die van de functie **Max**.*

**Syntaxis:**

```
Max([{SetExpression}] [TOTAL [<fld {,fld}>]] expr [,rank])
```

**Retourgegevenstypen:** numeriek

Argumenten

| Argument | Beschrijving |
|---|---|
| **expr** | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| **rank** | De standaardwaarde van **rank** is 1. Dit komt overeen met de hoogste waarde. Als u bij **rank** 2 opgeeft, wordt de op één na hoogste waarde geretourneerd. Als u bij **rank** 3 opgeeft, wordt de op twee na hoogste waarde geretourneerd, enzovoort. |
| **SetExpression** | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| **TOTAL** | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `Max(UnitSales)` | Retourneert de hoogste waarde in het veld `UnitSales`. |
| `Max(UnitSales*UnitPrice)` De waarde van een order wordt berekend door het aantal verkochte eenheden in (`UnitSales`) te vermenigvuldigen met de prijs per eenheid. | Retourneert de hoogste waarde van het resultaat van de berekening van alle mogelijke waarden van (`UnitSales`) * (`UnitPrice`). |
| `Max(UnitSales, 2)` | Retourneert de waarde voor de op één na hoogste waarde. |
| `Max(TOTAL UnitSales)` | De kwalificatie `TOTAL` betekent dat de hoogst mogelijke waarde wordt gevonden, waarbij de diagramdimensies buiten beschouwing worden gelaten. Voor een diagram met `Customer` als dimensie, zorgt de kwalificatie `TOTAL` ervoor dat de maximale waarde in de volledige gegevensverzameling wordt geretourneerd, in plaats van de maximale `UnitSales` voor elke klant. |
| `Max({1} TOTAL UnitSales)` | De set-analyseuitdrukking `{1}` definieert de set records die moeten worden geëvalueerd als `ALL`, ongeacht welke selectie wordt gemaakt. Als er bijvoorbeeld een specifieke klant wordt geselecteerd, zal deze nog steeds de maximale `UnitSales` voor de volledige gegevensverzameling retourneren. |

## Voorbeeld - Basisprincipes voor Max

Diagramuitdrukking

### Overzicht

Een gegevensverzameling bevat verkoopcijfers van producten per maand. Wij willen de maximale verkoopwaarde voor elk product identificeren en de maand waarin dit gebeurde.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Customer`
  - `UnitSales`
  - `UnitPrice`

### Load-script

```
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 '|');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Customer`

Maak de volgende metingen:

- `=Max(UnitSales)`, om de maximale `UnitSales` waarde te berekenen.
- `=Max(UnitSales*UnitPrice)`, om de maximumwaarde na vermenigvuldiging van `UnitSales` met `UnitPrice` rij per rij te berekenen.
- `=Max(UnitSales, 2)`, om de op één na hoogste waarde te berekenen.
- `=Max(TOTAL UnitSales)`, om het hoogste totaal te berekenen, ongeacht de dimensie van het diagram.
- `=Max({1} TOTAL UnitSales)`, om het hoogste totaal te berekenen dat alle selecties negeert.

Results table

| Customer | Max (UnitSales) | Max (UnitSales*UnitPrice) | Max (UnitSales, 2) | Max(TOTAL UnitSales) | Max({1} TOTAL UnitSales) |
|---|---|---|---|---|---|
| **Totals** | **10** | **150** | **9** | **10** | **10** |
| Astrida | 10 | 150 | 9 | 10 | 10 |
| Betacab | 5 | 50 | 2 | 10 | 10 |
| Canutility | 8 | 120 | - | 10 | 10 |

Volg in dit voorbeeld het volgende:

- In de eerste meting (`Max(UnitSales)`) is de hoogste waarde voor `UnitSales` voor de klant `Astrida` `10`. De cel `Totals` in deze kolom retourneert `10` omdat deze waarde ook de hoogste `UnitSales` waarde in de hele gegevensverzameling is.

- In de tweede meting (`Max(UnitSales*UnitPrice)`) wordt `150` geretourneerd voor `Astrida`. Dit is de hoogste waarde voor die klant bij vermenigvuldiging van `UnitSales` met `UnitPrice`. De cel `Totals` in deze kolom retourneert `150` omdat deze waarde ook de hoogst berekende waarde in de hele gegevensverzameling is.
- In de derde meting (`Max(UnitSales, 2)`) is de op één na hoogste `UnitSales` waarde voor de klant `Astrida` `9` . Deze waarde is ook de op één na hoogste waarde van alle drie klanten, daarom retourneert de rij `Totals` ook `9`.
- De vierde meting (`Max(TOTAL UnitSales)`) retourneert de waarde `10`. Dit is de hoogste gevonden waarde, onafhankelijk van de dimensie van de klant, daarom retourneren alle rijen dezelfde waarde. Als u echter de klant `Betacab` selecteert, retourneert de tabel alleen waarden voor die klant, in dit geval `5` omdat dit de hoogste `UnitSales` van alle producten voor `Betacab` is.
- De vijfde meting (`Max({1} TOTAL UnitSales)`) retourneert de waarde `10`, ongeacht welke `Customer` is geselecteerd, omdat de gewijzigde set `{1}` wordt gebruikt. Als u bijvoorbeeld `Betacab` selecteert, blijft deze kolom de waarde `10` retourneren.

## Voorbeeld - Max-scenario dat de hoogste verkoop per maand berekent

Diagramuitdrukking

### Overzicht

Een gegevensverzameling bevat verkoopcijfers van producten per maand. Een verkoopmanager wil de maximale verkoopwaarde voor elk product identificeren en de maand waarin dit gebeurde.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Product`
  - `Sales`
  - `Month`

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `Product`
- `Month`

Maak de volgende metingen:

- `=If(Sales = Max(TOTAL <Product> Sales), Sales)`, om de maximale verkoop voor elk product te berekenen. Selecteer **Uitbreidingsmodules** > **Gegevensverwerking** in het eigenschappenvenster van de meting. Schakel het selectievakje **Inclusief nulwaarden** uit om eventuele nullwaarden te verwijderen.

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 |

De resultaten van de meting retourneren de maximale verkoopwaarde voor elk product en de maand waarin deze waarde werd bereikt. Bijvoorbeeld, `Product B` had de hoogste verkoop (`800`) in `2024-02-01`.

---

**Zie ook:**

p *FirstSortedValue - diagramfunctie (page 365)*
p *RangeMax (page 1614)*

## Min

**Min()** retourneert de laagste numerieke waarde van de geaggregeerde gegevens in de uitdrukking, zoals gedefinieerd door een **group by**-clausule. Door een **rank** n op te geven, wordt de n-de laagste waarde gezocht.

**Syntaxis:**

```
Min ( expr [, rank])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| expr Expression | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| rank Expression | De standaardwaarde van **rank** is 1. Dit komt overeen met de laagste waarde. Als u bij **rank** 2 opgeeft, wordt de op één na laagste waarde geretourneerd. Als u bij **rank** 3 opgeeft, wordt de op twee na laagste waarde geretourneerd, enzovoort. |

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens ten minste de velden die in de resultatenkolom staan aan een werkblad in onze app toe om het resultaat te bekijken.

Als u dezelfde weergave wilt als in de onderstaande resultatenkolom, schakelt u in het eigenschappenvenster onder Sorteren over van Automatisch naar Aangepast en schakelt u numeriek en alfabetisch sorteren uit.

**Voorbeeld:**

```
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;
```

Resultaattabel

| Customer | MyMin |
|---|---|
| Astrida | 2 |
| Betacab | 4 |
| Canutility | 8 |

**Voorbeeld:**

Stel dat de tabel **Temp** is geladen zoals in het vorige voorbeeld:

```
LOAD Customer, Min(UnitSales,2) as MyMinRank2 Resident Temp Group By Customer;
```

Resultaattabel

| Customer | MyMinRank2 |
|---|---|
| Astrida | 9 |
| Betacab | 5 |
| Canutility | - |

## Min - diagramfunctie

**Min()** retourneert de laagste waarde van de geaggregeerde gegevens. Door een **rank** n op te geven, wordt de n-de laagste waarde gezocht.

*Bekijk eventueel ook **FirstSortedValue** en **RangeMin**, die een functionaliteit hebben die vergelijkbaar is met die van de functie **Min**.*

**Syntaxis:**

```
Min({[SetExpression] [TOTAL [<fld {,fld}>]]} expr [,rank])
```

**Retourgegevenstypen:** numeriek

Argumenten

| Argument | Beschrijving |
|---|---|
| **expr** | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| **rank** | De standaardwaarde van **rank** is 1. Dit komt overeen met de laagste waarde. Als u bij **rank** 2 opgeeft, wordt de op één na laagste waarde geretourneerd. Als u bij **rank** 3 opgeeft, wordt de op twee na laagste waarde geretourneerd, enzovoort. |
| **SetExpression** | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| **TOTAL** | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd. Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `Min(UnitSales)` | Retourneert de laatste niet-null-waarde in het veld `UnitSales`. |
| `Min(UnitSales*UnitPrice)` De waarde van een order wordt berekend door het aantal verkochte eenheden (`UnitSales`) te vermenigvuldigen met de prijs per eenheid. | Retourneert de laagste niet-null-waarde van het resultaat van de berekening van alle mogelijke waarden van (`UnitSales`) * (`UnitPrice`). |
| `Min(UnitSales, 2)` | Retourneert de waarde voor de op twee na laagste waarde in `UnitSales` (na de NULL-waarden). |
| `Min(TOTAL UnitSales)` | De kwalificatie `TOTAL` betekent dat de laagst mogelijke waarde wordt gevonden, waarbij de diagramdimensies buiten beschouwing worden gelaten. Voor een diagram met `Customer` als dimensie zorgt de kwalificatie `TOTAL` ervoor dat de minimale waarde in de volledige gegevensverzameling wordt geretourneerd, in plaats van de minimale `UnitSales` voor elke klant. |
| `Min({1} TOTAL UnitSales)` | De set-analyseuitdrukking `{1}` definieert de set records die moeten worden geëvalueerd als `ALL`, ongeacht welke selectie wordt gemaakt. Als er bijvoorbeeld een specifieke klant wordt geselecteerd, zal deze nog steeds de minimale `UnitSales` voor de volledige gegevensverzameling retourneren. |

## Voorbeeld - Basisprincipes voor Min

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Customer`
  - `UnitSales`
  - `UnitPrice`

### Load-script

```
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 '|');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Customer`

Maak de volgende metingen:

- `=Min(UnitSales)`, om de minimale `UnitSales` waarde te berekenen.
- `=Min(UnitSales*UnitPrice)`, om de minimumwaarde na vermenigvuldiging van `UnitSales` met `UnitPrice` rij per rij te berekenen.
- `=Min(UnitSales, 2)`, om de op één na laagste waarde te berekenen.
- `=Min(TOTAL UnitSales)`, om het laagste totaal te berekenen, ongeacht de dimensie van het diagram.
- `=Min({1} TOTAL UnitSales)`, om het laagste totaal te berekenen dat alle selecties negeert.

Results table

| Customer | Min (UnitSales) | Min (UnitSales*UnitPrice) | Min (UnitSales, 2) | Min(TOTAL UnitSales) | Min({1} TOTAL UnitSales) |
|---|---|---|---|---|---|
| **Totals** | **2** | **40** | **4** | **2** | **2** |
| Astrida | 4 | 64 | 9 | 2 | 2 |
| Betacab | 2 | 40 | 5 | 2 | 2 |
| Canutility | 8 | 120 | - | 2 | 2 |

Volg in dit voorbeeld het volgende:

- In de eerste meting `Min(UnitSales)` is de laagste waarde voor `UnitSales` voor de klant `Astrida 4`. De cel `Totals` in deze kolom retourneert `2` omdat deze waarde de laagste `UnitSales` waarde in de hele gegevensverzameling is.
- In de tweede meting (`Min(UnitSales*UnitPrice)`) wordt `64` geretourneerd voor `Astrida`. Dit is de laagste waarde voor die klant bij vermenigvuldiging van `UnitSales` met `UnitPrice`. De cel `Totals` in deze kolom retourneert `40` omdat deze waarde ook de laagst berekende waarde in de hele gegevensverzameling is.
- In de derde meting (`Min(UnitSales, 2)`) is de op één na laagste `UnitSales` waarde voor de klant `Astrida 9`. De waarde `4` wordt geretourneerd in de rij `Totals`, omdat dit de op één na laagste `UnitSales` waarde van alle klanten is.

- De vierde meting `Min(TOTAL UnitSales)` retourneert de waarde `2`. Dit is de laagst mogelijke niet-null-waarde, onafhankelijk van de dimensie van de klant, daarom retourneren alle rijen dezelfde waarde. Als u echter de klant `Betacab` selecteert, retourneert de tabel alleen waarden voor die klant, in dit geval `2` omdat dit de laagste `UnitSales` van alle producten voor `Betacab` is.
- De vijfde meting `Min({1} TOTAL UnitSales)`, retourneert `2` ongeacht de waarde die in het veld `Customer` is geselecteerd, aangezien de meting het laagste totaal berekent. Als u bijvoorbeeld `Betacab` selecteert, blijft deze kolom de waarde `2` retourneren.

## Voorbeeld - Min-scenario dat de laagste verkoop per maand berekent

Diagramuitdrukking

### Overzicht

Een gegevensverzameling bevat verkoopcijfers van producten per maand. Een verkoopmanager wil de minimale verkoopwaarde voor elk product identificeren en de maand waarin dit gebeurde.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Product`
  - `Sales`
  - `Month`

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `Product`
- `Month`

Maak de volgende meting:

- `=If(Sales = Min(TOTAL <Product> Sales), Sales)`, om de minimale verkoop voor elk product te berekenen. Selecteer **Uitbreidingsmodules** > **Gegevensverwerking** in het eigenschappenvenster. Schakel het selectievakje **Inclusief nulwaarden** uit.

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 |

De resultaten van de meting retourneren de minimale verkoopwaarde voor elk product en de maand waarin deze waarde werd bereikt. Bijvoorbeeld, `Product B` had de laagste verkoop (`300`) in `2024-01-01`.

**Zie ook:**

p *FirstSortedValue - diagramfunctie (page 365)*
p *RangeMin (page 1618)*

## Mode

**Mode()** retourneert de vaakst voorkomende waarde, de moduswaarde, van de geaggregeerde gegevens in de uitdrukking, zoals gedefinieerd door een **group by**-clausule. De functie **Mode()** kan zowel numerieke waarden als tekstwaarden retourneren.

**Syntaxis:**

```
Mode ( expr)
```

**Retourgegevenstypen:** dual

Argumenten

| Argument | Beschrijving |
|---|---|
| expr<br>Expression | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |

**Beperkingen:**

Als meerdere waarden even vaak voorkomen, is het resultaat NULL.

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens ten minste de velden die in de resultatenkolom staan aan een werkblad in onze app toe om het resultaat te bekijken.

Als u dezelfde weergave wilt als in de onderstaande resultatenkolom, schakelt u in het eigenschappenvenster onder Sorteren over van Automatisch naar Aangepast en schakelt u numeriek en alfabetisch sorteren uit.

Scriptvoorbeelden

| Voorbeeld | Resultaat |
|---|---|
| Temp:<br>LOAD * inline [<br>Customer\|Product\|OrderNumber\|UnitSales\|CustomerID<br>Astrida\|AA\|1\|10\|1<br>Astrida\|AA\|7\|18\|1<br>Astrida\|BB\|4\|9\|1<br>Astrida\|CC\|6\|2\|1<br>Betacab\|AA\|5\|4\|2<br>Betacab\|BB\|2\|5\|2<br>Betacab\|DD<br>Canutility\|DD\|3\|8<br>Canutility\|CC<br>] (delimiter is '\|');<br><br>Mode:<br>LOAD Customer, Mode(Product) as<br>MyMostOftenSoldProduct Resident Temp Group By<br>Customer; | MyMostOftenSoldProduct<br><br>AA<br><br>omdat AA het enige product is dat vaker dan één keer is verkocht. |

## Mode - diagramfunctie

**Mode()** retourneert de vaakst voorkomende waarde, de moduswaarde, in de geaggregeerde gegevens. De functie **Mode()** kan zowel tekstwaarden als numerieke waarden verwerken.

**Syntaxis:**

```
Mode({[SetExpression] [TOTAL [<fld {,fld}>]]} expr)
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **expr** | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| **SetExpression** | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| **TOTAL** | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `Mode(UnitSales)` | Retourneert de meest voorkomende waarde in `UnitSales`.<br>Retourneert `NULL (-)`. Geen enkele waarde komt vaker voor dan de andere. |
| `Mode(Customer)` | Retourneert de meest voorkomende waarde in `Customer`.<br>Retourneert `NULL (-)`. Geen enkele waarde komt vaker voor dan de andere. |
| `Mode (TOTAL UnitSales)` | De kwalificatie `TOTAL` retourneert de meest voorkomende waarde in `Unitsales`, waarbij de diagramdimensies buiten beschouwing worden gelaten. |
| `Mode({1} TOTAL UnitSales)` | De set-analyseuitdrukking `{1}` definieert de set records die moeten worden geëvalueerd als `ALL`, ongeacht welke selectie wordt gemaakt. Als er bijvoorbeeld een specifieke klant wordt geselecteerd, zal deze nog steeds de meest voorkomende waarde van `UnitSales` voor de volledige gegevensverzameling retourneren. |

## Voorbeeld - Basisprincipes voor Mode

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Customer`
  - `UnitSales`
  - `UnitPrice`

### Load-script

```
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 '|');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Customer`

Maak de volgende metingen:

- `=Mode(UnitSales)`, om de meest voorkomende `UnitSales` waarde te berekenen.
- `=Mode(Customer)`, om de meest voorkomende `Customer` waarde te berekenen.
- `=Mode(TOTAL UnitSales)`, om de meest voorkomende `UnitSales` waarde in de hele gegevensverzameling te berekenen, zonder rekening te houden met de diagramdimensies. Als er een selectie is gemaakt, wordt de waarde aangepast om die selectie weer te geven.
- `=Mode({1} TOTAL UnitSales)`, om de meest voorkomende `UnitSales` waarde te berekenen, waarbij selecties worden genegeerd. Als er een selectie is gemaakt, blijft de waarde constant.

Results table

| Customer | Mode (UnitSales) | Mode (Customer) | Mode(TOTAL UnitSales) | Mode({1} TOTAL UnitSales) |
|---|---|---|---|---|
| **Totals** | **4** | **Astrida** | **4** | **4** |
| Astrida | 10 | Astrida | 4 | 4 |
| Betacab | - | Betacab | 4 | 4 |
| Canutility | - | Canutility | 4 | 4 |

Volg in dit voorbeeld het volgende:

- In de eerste meting (`Mode(UnitSales)`) is de meest voorkomende waarde voor de klant `Astrida` `10`. De cel `Totals` retourneert `4`, wat de meest voorkomende waarde is voor `UnitSales` in de hele gegevensverzameling. Er zijn geen exemplaren van meest voorkomende waarden voor `Betacab` en `Canutility`, dus de resultaten retourneren `NULL` voor die klanten.
- In de tweede meting (`Mode(Customer)`) retourneren de resultaten elke klantnaam. De cel `Totals` retourneert `Astrida`, omdat dit de meest voorkomende klantwaarde is in de hele gegevensverzameling.
- De derde meting (`Mode(TOTAL UnitSales)`) retourneert de waarde 4. Als er geen `Customer` is geselecteerd, dan retourneren alle cellen in deze kolom de waarde 4. Als u een klant selecteert, zoals `Betacab`, dan retourneert deze kolom `NULL`, omdat er geen algemeen voorkomende waarde is voor `Betacab`.
- De vierde meting (`Mode({1} TOTAL UnitSales)`) retourneert `4` ongeacht eventuele selecties in het veld `Customer` omdat de set-analyseuitdrukking `{1}` de set records definieert die moet worden geëvalueerd als `ALL`. Als u bijvoorbeeld `Betacab` selecteert, blijft deze kolom de waarde `4` retourneren.

## Voorbeeld - Mode-scenario om afwijkingen in verkoopgegevens te bepalen

Diagramuitdrukking

### Overzicht

Een gegevensverzameling bevat verkoopcijfers per maand. Het bedrijf wil de meest voorkomende verkoopwaarde voor elk product en voor de hele gegevensverzameling zien. Het wil ook de verkoop per product vergelijken met de volledige gegevensverzameling om eventuele atypische afwijkingen in de gegevens te identificeren.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
    - `Date`
    - `Product`
    - `Category`
    - `Sales`

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `Category`
- `Product`

Maak de volgende metingen:

- `=Mode(Sales)`, om de meest voorkomende waarde voor elk product te berekenen.
- `=Mode(TOTAL Sales)`, om de meest voorkomende waarde voor de gehele gegevensverzameling te berekenen.
- `If(Mode(Sales) = Mode(TOTAL Sales), 'Typical', 'Atypical')`, om te berekenen of de meest voorkomende waarde voor elk product typisch is in vergelijking met de resultaten van de hele gegevensverzameling.

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 |

De uitvoer van de functie `Mode` geeft de meest voorkomende waarde van de verkoop per product, `Mode (Sales)`, en binnen alle producten, `Mode(TOTAL Sales)`. Deze twee waarden worden vervolgens vergeleken om te bepalen welke verkoopresultaten typisch of atypisch zijn.

**Zie ook:**

p *Avg - diagramfunctie (page 464)*
p *Median - diagramfunctie (page 505)*

## Only

**Only()** retourneert een waarde als er één, en niet meer dan één, mogelijk resultaat is voor de geaggregeerde gegevens. Als records slechts één waarde bevatten, wordt die waarde geretourneerd, anders wordt NULL geretourneerd. Gebruik de **group by**-clausule om meerdere records te evalueren. De functie **Only()** kan numerieke waarden en tekstwaarden retourneren.

**Syntaxis:**

```
Only ( expr )
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| expr<br>Expression | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens ten minste de velden die in de resultatenkolom staan aan een werkblad in onze app toe om het resultaat te bekijken.

Als u dezelfde weergave wilt als in de onderstaande resultatenkolom, schakelt u in het eigenschappenvenster onder Sorteren over van Automatisch naar Aangepast en schakelt u numeriek en alfabetisch sorteren uit.

```
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;
```

Resultaattabel

| Klant | MyUniqIDCheck |
|---|---|
| Astrida | 1<br><br>omdat alleen klant Astrida volledige records heeft die CustomerID bevatten. |

## Only - diagramfunctie

**Only()** retourneert een waarde als er één, en niet meer dan één, mogelijk resultaat is voor de geaggregeerde gegevens. Als u bijvoorbeeld zoekt naar het enige product waarvan de prijs per eenheid =9 en er zijn meerdere producten met een prijs per eenheid van 9, dan wordt het resultaat NULL geretourneerd.

**Syntaxis:**

```
Only([{SetExpression}] [TOTAL [<fld {,fld}>]] expr)
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **expr** | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| **SetExpression** | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| **TOTAL** | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br>Wanneer u **TOTAL [<fld {,fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

*Gebruik Only() als u een NULL-resultaat wilt als er meerdere mogelijke waarden zijn in de steekproefgegevens.*

Voorbeeld: diagramuitdrukking

| Voorbeeld | Resultaat |
|---|---|
| `Only(Category)` | Toont de `Category` zolang deze uniek is binnen de context. |

## Voorbeeld - Basisprincipes voor Only

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Customer`
  - `Product`
  - `UnitSales`
  - `UnitPrice`

### Load-script

```
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 '|');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel met de volgende metingen:

- `=Only({<UnitPrice={9}>} Product)`, om de enige `Product` te berekenen die een `UnitPrice` van `9` heeft.
- `=Only({<Product={DD}>} Customer)`, om de enige `Customer` te berekenen die een `Product` genaamd `DD` verkoopt.
- `=Only({<UnitPrice={20}>} UnitSales)`, om het aantal `UnitSales` te berekenen waarbij `UnitPrice 20` is.
- `=Only({<UnitPrice={15}>} UnitSales)`, om de waarden van `UnitSales` te berekenen waarbij de `UnitPrice = 15` is.

Results table

| Only({<UnitPrice={9}>} Product) | Only({<Product={DD}>} Customer) | Only({<UnitPrice={20}>} UnitSales) | Only({<UnitPrice={15}>} UnitSales) |
|---|---|---|---|
| BB | Betacab | 2 | - |

Volg in dit voorbeeld het volgende:

- De eerste kolom retourneert de waarde `BB` omdat dit het enige `Product` is dat een `UnitPrice` van `9` heeft.
- De tweede kolom retourneert `Betacab`, omdat dit de enige `Customer` is die een `Product` verkoopt met de naam `DD`.
- De derde kolom retourneert `2` omdat dit de enige waarde van `UnitSales` is waarbij `UnitPrice = 20` is.
- De vierde kolom retourneert `NULL` omdat er geen enkel resultaat is waarbij `UnitPrice = 15` is. In deze gegevensverzameling zijn er twee waarden van `UnitSales` waarbij de `UnitPrice =15` is.

## Voorbeeld 5 – Scenario met Only

Diagramuitdrukking

### Overzicht

Een gegevensverzameling bevat productverkopen per periode. Het bedrijf wil bepalen of een product in één winkel of in meerdere winkels wordt verkocht.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Date`
  - `RecordID`
  - `StoreProduct`
  - `Sales`

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `StoreProduct`

Maak de volgende metingen:

- `=Only(Store)`, om de unieke winkel te berekenen waar het product wordt verkocht.
- `=If(IsNull(Only(Store)), 'Multiple Stores', 'Single Store')`, om te berekenen of het product in één winkel of in meerdere winkels wordt verkocht.
- `=Sum(Sales)`, om de totale productverkoop te berekenen.

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 |

De resultaten geven aan dat `Apples` en `Bananas` slechts in één winkel worden verkocht, terwijl `Carrots` in meerdere winkels wordt verkocht.

### Voorbeeld - Geavanceerd scenario om een tabel met records voor één maand te maken

Diagramuitdrukking

#### Overzicht

Dit voorbeeld gebruikt de functie `Only` om een regel-voor-regel tabel met records voor een enkele maand weer te geven. Om dit te bereiken, wordt set-analyse gebruikt om het bereik van ed gewenste recordverzameling te bepalen. Voor set-analyse is ten minste één meting nodig, maar omdat we de recordverzameling niet echt hoeven te aggregeren, kunnen we de aggregatiefunctie `Only` effectief gebruiken om het bereik van de set-analyse te hosten.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Een voorafgaande LOAD-instructie die het veld `MonthYear` afleidt van het veld `Date` door de dag van de maand in te korten met behulp van de functie `MonthStart` en een datumweergave van de 1e van elke maand te retourneren. Dit resultaat wordt doorgegeven aan de functie `Date` om het weergaveformaat in te stellen op `MMM YY`, bijvoorbeeld `Jan 24`.
- De volgende velden in de gegevenstabel:
  - `Date`
  - `RecordID`
  - `StoreProduct`
  - `Sales`
  - `Store`

#### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `MonthYear`
- `RecordID`
- `Date`
- `StoreProduct`
- `Store`

Maak de volgende metingen:

- `=Only({$<MonthYear={'$(=MaxString(MonthYear))'}>} Sales)`, om de waarde `Sales` te evalueren en alleen die waarden te retourneren waarbij `MonthYear` de laatste (meest recente) `MonthYear` is die beschikbaar is, gebaseerd op de huidige selecties in de tabel. De uitbreiding met dollarteken ($) voert een inline berekening uit met de functie `MaxString`, `$(=MaxString(MonthYear))`, die de tekstweergave van de laatste `MonthYear` waarde teruggeeft.

*Pogingen om `MonthYear` te filteren op basis van de numerieke weergave, bijvoorbeeld met behulp van de functie `Max , Max(MonthYear)`, zouden geen resultaat opleveren. Set-analyse past filterselecties toe die alleen gebaseerd zijn op de tekstweergave van een dubbel gegevenstype; een gegevenstype met een numerieke en tekstweergave zoals een `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 |

De uitvoer van de functie `Only` produceert een tabel met records die standaard de verkoopwaarde weergeeft voor de laatste, of meest recente, `MonthYear` die beschikbaar is in de gegevens. Als u in de kolom `MonthYear` een eerdere maand selecteert, zoals `Feb 24`, verschijnt de laatste verkoopwaarde voor die maand.

## Sum

**Sum()** berekent het totaal van de waarden die zijn geaggregeerd in de uitdrukking, zoals gedefinieerd door een **group by**-clausule.

**Syntaxis:**

```
sum ( [ distinct] expr)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| distinct | Als het woord **distinct** voor de uitdrukking staat, worden alle duplicaten genegeerd. |
| expr Expression | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens ten minste de velden die in de resultatenkolom staan aan een werkblad in onze app toe om het resultaat te bekijken.

Als u dezelfde weergave wilt als in de onderstaande resultatenkolom, schakelt u in het eigenschappenvenster onder Sorteren over van Automatisch naar Aangepast en schakelt u numeriek en alfabetisch sorteren uit.

```
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;
```

Resultaattabel

| Customer | MySum |
|---|---|
| Astrida | 39 |
| Betacab | 9 |
| Canutility | 8 |

## Sum - diagramfunctie

**Sum()** berekent het totaal van de waarden die worden geleverd door de uitdrukking of het veld voor de geaggregeerde gegevens.

**Syntaxis:**

```
Sum([{SetExpression}] [DISTINCT] [TOTAL [<fld {,fld}>]] expr)
```

**Retourgegevenstypen:** numeriek

Argumenten

| Argument | Beschrijving |
|---|---|
| **expr** | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| **SetExpression** | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| **DISTINCT** | Als het woord **DISTINCT** voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd.<br>*Hoewel de kwalificatie DISTINCT wordt ondersteund, moet u heel zorgvuldig zijn bij het gebruiken hiervan omdat de lezer ten onrechte het idee kan krijgen dat een totale waarde wordt weergegeven als sommige gegevens zijn weggelaten.* |
| **TOTAL** | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `Sum(UnitSales)` | Retourneert het totaal van de waarden in `UnitSales`. |
| `Sum(UnitSales*UnitPrice)` | Retourneert het totaal van `UnitPrice` vermenigvuldigd met `UnitSales` geaggregeerd. |
| `Sum (TOTAL UnitSales*UnitPrice)` | Retourneert de totale som voor alle rijen in de tabel alsmede het totaal, omdat de kwalificatie `TOTAL` betekent dat de som is berekend, ongeacht de diagramdimensies. |
| `Sum({1} TOTAL UnitSales*UnitPrice)` | De set-analyseuitdrukking `{1}` definieert de set records die moeten worden geëvalueerd als `ALL`, ongeacht welke selectie wordt gemaakt. |

## Voorbeeld - Basisprincipes voor Sum

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Customer`
  - `Product`
  - `UnitSales`
  - `UnitPrice`

### Load-script

```
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 '|');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- Customer
- Product

Maak de volgende metingen:

- =Sum(UnitSales), om de som van UnitSales te berekenen.
- =Sum(UnitSales*UnitPrice) om het geaggregeerde total van UnitSales vermenigvuldigd met UnitPrice te berekenen.
- =Sum(TOTAL UnitSales*UnitPrice), om de totale som voor alle rijen in de tabel alsmede het totaal te berekenen, omdat de kwalificatie TOTAL betekent dat de som is berekend, ongeacht de diagramdimensies.
- =Sum({1} TOTAL UnitSales*UnitPrice), om het totaal te berekenen, ongeacht de uitgevoerde selectie, omdat de set- analyseuitdrukking {1} de set records definieert die moeten worden geëvalueerd als ALL, ongeacht welke selectie wordt gemaakt.

Results table

| Customer | Product | Sum (UnitSales) | Sum (UnitSales*UnitPrice) | Sum(TOTAL UnitSales*UnitPrice) | Sum({1} TOTAL UnitSales*UnitPrice) |
|---|---|---|---|---|---|
| **Totals** | - | **38** | **505** | **505** | **505** |
| 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 |

Volg in dit voorbeeld het volgende:

- Voor de eerste meting, Sum(UnitSales, retourneert de cel Totals38, wat het totaal is van de waarden in UnitSales binnen alle klanten en producten. De resultaten voor elke combinatie van klant en product worden samengevoegd tot de totale som van UnitSales voor elke combinatie. Bijvoorbeeld, de som van alle UnitSales van product AA van klant Astrida is 14.
- Voor de tweede meting, Sum(UnitSales*UnitPrice), retourneert de cel Totals 505 . Dit is het geaggregeerde totaal van UnitSales vermenigvuldigd met UnitPrice voor elke klant- en productcombinatie.
- Voor de derde meting, Sum(TOTAL UnitSales*UnitPrice), is de geretourneerde waarde in alle rijen inclusief Totals 505 , omdat de kwalificatie TOTAL de diagramdimensies negeert. Als er echter een selectie wordt gemaakt op een dimensie zoals Customer, dan zullen de waarden veranderen. Als u bijvoorbeeld Betacab selecteert, zal de resulterende tabel slechts drie rijen hebben, één voor elk van de Betacab records, maar de kolom zal dezelfde totale waarde voor alle rijen retourneren (90).

- De vierde meting, `Sum({1} TOTAL UnitSales*UnitPrice)`, retourneert de waarde `505` ongeacht eventuele selecties omdat de set-analyseuitdrukking `{1}` de set records definieert die moet worden geëvalueerd als `ALL`. Als u een klant selecteert, bijvoorbeeld `Canutility`, is de geretourneerde waarde nog steeds `505`.

## Voorbeeld - Sum-scenario waarin verkoopgegevens worden geaggregeerd

Diagramuitdrukking

### Overzicht

Een gegevensverzameling bevat verkoopcijfers voor winkels per regio. Het bedrijf wil de verkopen aggregeren.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Month`
  - `SalesProduct`
  - `Region`
  - `Sales`

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe draaotabel en voeg deze velden toe als dimensies:

- `Month`
- `Region`

Voeg dit veld toe als een kolomdimensie:

- `SalesProduct`

Maak de volgende meting:

- `=Sum(Sales)`, om de totale verkoop voor elk product te berekenen.

Om de draaitabel uit te breiden zodat alle rijen zichtbaar zijn, selecteert u in het eigenschappenvenster onder **Uiterlijk Presentatie**. Onder **Rijen** selecteert u **Volledig uitgevouwen**.

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 |

De resultaten tonen de verkoop voor elke `SalesProduct` per `Month` en `Region`, met een geaggregeerd totaal voor elke `Month`.

## Teller-aggregatiefuncties

Teller-aggregatiefuncties retourneren verschillende typen tellers van een uitdrukking via een aantal records in een load-script voor gegevens of een aantal waarden in een diagramdimensie.

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

### Teller-aggregatiefuncties in het script voor het laden van gegevens

**Count**

**Count()** retourneert het aantal waarden dat is geaggregeerd in de uitdrukking, zoals gedefinieerd door een **group by**-clausule.

```
Count ([distinct ] expression | * )
```

**MissingCount**

**MissingCount()** retourneert het aantal ontbrekende waarden dat is geaggregeerd in de uitdrukking, zoals gedefinieerd door een **group by**-clausule.

```
MissingCount ([ distinct ] expression)
```

### NullCount

**NullCount()** retourneert het aantal NULL-waarden dat is geaggregeerd in de uitdrukking, zoals gedefinieerd door een **group by**-clausule.

```
NullCount ([ distinct ] expression)
```

### NumericCount

**NumericCount()** retourneert het aantal numerieke waarden dat is gevonden in de uitdrukking, zoals gedefinieerd door een **group by**-clausule.

```
NumericCount ([ distinct ] expression)
```

### TextCount

**TextCount()** retourneert het aantal niet-numerieke veldwaarden dat is geaggregeerd in de uitdrukking, zoals gedefinieerd door een group by-clausule.group by

```
TextCount ([ distinct ] expression)
```

## Teller-aggregatiefuncties in diagramuitdrukkingen

De volgende teller-aggregatiefuncties kunnen worden gebruikt in diagrammen.

### Count

**Count()** wordt gebruikt om het aantal waarden, tekst en numeriek, in elke diagramdimensie te aggregeren.

```
Count - diagramfunctie({[SetExpression] [DISTINCT] [TOTAL [<fld {,fld}>]]}
expr)
```

### MissingCount

**MissingCount()** wordt gebruikt om het aantal ontbrekende waarde in elke diagramdimensie te aggregeren. Ontbrekende waarden zijn alle niet-numerieke waarden.

```
MissingCount - diagramfunctie({[SetExpression] [DISTINCT] [TOTAL [<fld
{,fld}>]] expr)
```

### NullCount

**NullCount()** wordt gebruikt om het aantal null-waarden in elke diagramdimensie te aggregeren.

```
NullCount - diagramfunctie({[SetExpression][DISTINCT] [TOTAL [<fld {,fld}>]]}
expr)
```

### NumericCount

**NumericCount()** wordt gebruikt om het aantal numerieke waarden in elke diagramdimensie te aggregeren.

```
NumericCount - diagramfunctie({[SetExpression] [DISTINCT] [TOTAL [<fld
{,fld}>]]} expr)
```

### TextCount

**TextCount()** wordt gebruikt om het aantal veldwaarden die niet-numeriek zijn in elke diagramdimensie te aggregeren.

```
TextCount - diagramfunctie({[SetExpression] [DISTINCT] [TOTAL [<fld
{,fld}>]]} expr)
```

## Count

**Count()** retourneert het aantal waarden dat is geaggregeerd in de uitdrukking, zoals gedefinieerd door een **group by**-clausule.

**Syntaxis:**

```
Count( [distinct ] expr)
```

**Retourgegevenstypen:** geheel getal

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| distinct | Als het woord **distinct** voor de uitdrukking staat, worden alle duplicaten genegeerd. |

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens ten minste de velden die in de resultatenkolom staan aan een werkblad in onze app toe om het resultaat te bekijken.

Als u dezelfde weergave wilt als in de onderstaande resultatenkolom, schakelt u in het eigenschappenvenster onder Sorteren over van Automatisch naar Aangepast en schakelt u numeriek en alfabetisch sorteren uit.

Scriptvoorbeelden

| Voorbeeld | Resultaat |
|---|---|
| Temp:<br>LOAD * inline [<br>Customer\|Product\|OrderNumber\|UnitSales\|UnitPrice<br>Astrida\|AA\|1\|4\|16<br>Astrida\|AA\|7\|10\|15<br>Astrida\|BB\|4\|9\|9<br>Betacab\|CC\|6\|5\|10<br>Betacab\|AA\|5\|2\|20<br>Betacab\|BB\|1\|25\| 25<br>Canutility\|AA\|3\|8\|15<br>Canutility\|CC\|\|\|19<br>Divadip\|CC\|2\|4\|16<br>Divadip\|DD\|3\|1\|25<br>] (delimiter is '\|');<br><br>Count1:<br><br>LOAD Customer,Count(OrderNumber) as OrdersByCustomer Resident Temp Group By Customer; | Customer OrdersByCustomer<br><br>Astrida 3<br><br>Betacab 3<br><br>Canutility 2<br><br>Divadip 2<br>Zolang de dimensie Customer wordt opgenomen in de tabel op het werkblad, anders is 3, 2 het resultaat voor OrdersByCustomer. |
| Stel dat de tabel **Temp** is geladen zoals in het vorige voorbeeld:<br><br>LOAD Count(OrderNumber) as TotalOrderNumber Resident Temp; | TotalOrderNumber<br><br>10 |
| Stel dat de tabel **Temp** is geladen zoals in het eerste voorbeeld:<br><br>LOAD Count(distinct OrderNumber) as TotalOrderNumber Resident Temp; | TotalOrderNumber<br><br>8<br>Omdat er twee waarden van OrderNumber zijn die dezelfde waarde, 1, hebben en één null-waarde. |

## Count - diagramfunctie

**Count()** wordt gebruikt om het aantal waarden, tekst en numeriek, in elke diagramdimensie te aggregeren.

**Syntaxis:**

```
Count({[SetExpression] [DISTINCT] [TOTAL [<fld {,fld}>]]} expr)
```

**Retourgegevenstypen:** geheel getal

Argumenten

| Argument | Beschrijving |
|---|---|
| **expr** | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |

| Argument | Beschrijving |
|---|---|
| **SetExpression** | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| **DISTINCT** | Als het woord **DISTINCT** voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd. |
| **TOTAL** | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `Count(OrderNumber)` | Retourneert het aantal rijen dat een waarde zou kunnen hebben voor `OrderNumber`. Alle records, zelfs lege, worden geteld.<br>*"0" telt als waarde en niet als lege cel. Alleen als een meting aggregeert tot 0 voor een dimensie, wordt die dimensie niet opgenomen in diagrammen.* |
| `Count(Customer)` | Evalueert het aantal exemplaren in alle rijen. |
| `Count(DISTINCT [Customer])` | De kwalificatie `DISTINCT` beperkt de `Count` tot het evalueren van alleen unieke exemplaren. |
| `Count (OrderNumber)/Count({1} TOTAL OrderNumber)` | De uitdrukking retourneert het aantal orders van de geselecteerde klant als percentage van de orders van alle klanten. |
| `Count(TOTAL <Product> OrderNumber)` | Retourneert het aantal orders is dat is geplaatst voor producten voor alleen de geselecteerde klanten. Lege cellen worden ook geteld. |

## Voorbeeld - Basisprincipes voor Count

### Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
    - `Customer`
    - `Product`
    - `OrderNumber`
    - `UnitSales`
    - `UnitPrice`

### Load-script

```
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 '|');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Customer`

Maak de volgende metingen:

- `=Count(OrderNumber)`, om het aantal orders te tellen.
- `=Count(Customer)`, om het aantal klantrecords te berekenen.
- `=Count(DISTINCT Customer)`, om het aantal unieke klanten te tellen.
- `=Count(OrderNumber)/Count({1} TOTAL OrderNumber)`, om de verhouding van de geselecteerde klantorders tot het totaal van de klantorders te tellen.
- `=Count(TOTAL <Product> OrderNumber)`, om het aantal orders te tellen dat is geplaatst voor producten voor alleen de geselecteerde klanten (lege cellen worden meegeteld).

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 |

Als we naar de resultaten kijken, zien we het volgende:

- Eerste meting: er zijn `10` orderrecords in de gegevensverzameling.
- Tweede meting: er zijn `10` klantrecords in de gegevensverzameling.
- Derde meting: er zijn `4` verschillende (unieke) klanten in de gegevensverzameling.
- Vierde meting: retourneert `1` omdat er geen selectie is toegepast. Als u de klant `Canutility` selecteert, verandert de waarde in `0.2` omdat er `2` orderrecords voor deze klant zijn op `10` totale records in de gegevensverzameling.
- Vijfde meting: retourneert `10` omdat er geen selectie is toegepast. Als u de klanten `Astrida` en `Canutility` selecteert, verandert de waarde in `5` omdat dit het aantal orders is dat is geplaatst voor producten voor alleen deze klanten (lege cellen worden meegeteld).

### Voorbeeld - Count gebruiken om het aantal verkopen per product te vinden

Diagramuitdrukking

**Overzicht**

Een gegevensverzameling bevat gegevens voor winkelverkopen. U wilt het aantal verkopen per product vinden.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Product`
  - `Sales`
  - `Region`
  - `Status`

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Product`

Maak de volgende metingen:

- `=Count(Sales)`, om het aantal verkopen per product te berekenen.
- `=Count({<Status={'Completed'}>} Sales)`, om het aantal verkopen met voltooide orders te berekenen.
- `=Count({<Status={'Pending'}>} Sales)`, om het aantal verkopen met orders in behandeling te berekenen.
- `=Count({<Region={'North'}>} Sales)`, om het aantal orders in de regio `North` te berekenen.

Results table

| Product | Count (Sales) | Count({<Status={'Completed'}>} Sales) | Count({<Status={'Pending'}>} Sales) | Count({<Region={'North'}>} Sales) |
|---|---|---|---|---|
| **Totals** | **9** | **6** | **3** | **3** |
| Apple | 3 | 3 | 0 | 3 |
| Banana | 3 | 2 | 1 | 0 |
| Carrot | 3 | 1 | 2 | 0 |

De uitvoer van de functie `Count` retourneert een uitsplitsing van het totale aantal verkooporders per product, evenals het aantal orders dat is voltooid en het aantal dat nog in behandeling is. De laatste meting geeft een telling van het aantal verkooprecords voor de regio `North`.

## MissingCount

**MissingCount()** retourneert het aantal ontbrekende waarden dat is geaggregeerd in de uitdrukking, zoals gedefinieerd door een **group by**-clausule.

**Syntaxis:**

```
MissingCount ( [ distinct ] expr)
```

**Retourgegevenstypen:** geheel getal

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| expr Expression | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| distinct | Als het woord **distinct** voor de uitdrukking staat, worden alle duplicaten genegeerd. |

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens ten minste de velden die in de resultatenkolom staan aan een werkblad in onze app toe om het resultaat te bekijken.

Als u dezelfde weergave wilt als in de onderstaande resultatenkolom, schakelt u in het eigenschappenvenster onder Sorteren over van Automatisch naar Aangepast en schakelt u numeriek en alfabetisch sorteren uit.

Scriptvoorbeelden

| Voorbeeld | Resultaat |
|---|---|
| `Temp:`<br>`LOAD * inline [`<br>`Customer\|Product\|OrderNumber\|UnitSales\|UnitPrice`<br>`Astrida\|AA\|1\|4\|16`<br>`Astrida\|AA\|7\|10\|15`<br>`Astrida\|BB\|4\|9\|9`<br>`Betacab\|CC\|6\|5\|10`<br>`Betacab\|AA\|5\|2\|20`<br>`Betacab\|BB\|\|\| 25`<br>`Canutility\|AA\|\|\|15`<br>`Canutility\|CC\| \|\|19`<br>`Divadip\|CC\|2\|4\|16`<br>`Divadip\|DD\|3\|1\|25`<br>`] (delimiter is '\|');`<br>`MissCount1:`<br><br>`LOAD Customer,MissingCount(OrderNumber) as MissingOrdersByCustomer Resident Temp Group By Customer;`<br><br>`Load MissingCount(OrderNumber) as TotalMissingCount Resident Temp;` | `Customer MissingOrdersByCustomer`<br><br>`Astrida 0`<br><br>`Betacab 1`<br><br>`Canutility 2`<br><br>`Divadip 0`<br><br>De tweede opdracht geeft als resultaat:<br>`TotalMissingCount`<br><br>`3`<br>in een tabel met die dimensie. |
| Stel dat de tabel **Temp** is geladen zoals in het vorige voorbeeld:<br>`LOAD MissingCount(distinct OrderNumber) as TotalMissingCountDistinct Resident Temp;` | `TotalMissingCountDistinct`<br><br>`1`<br>omdat er slechts één OrderNumber is, ontbreekt er één waarde. |

## MissingCount - diagramfunctie

**MissingCount()** wordt gebruikt om het aantal ontbrekende waarde in elke diagramdimensie te aggregeren. Ontbrekende waarden zijn alle niet-numerieke waarden.

**Syntaxis:**

```
MissingCount({[SetExpression] [DISTINCT] [TOTAL [<fld {,fld}>]]} expr)
```

**Retourgegevenstypen:** geheel getal

Argumenten

| Argument | Beschrijving |
|---|---|
| **expr** | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| **SetExpression** | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |

| Argument | Beschrijving |
|---|---|
| DISTINCT | Als het woord **DISTINCT** voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd. |
| TOTAL | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `MissingCount(OrderNumber)` | Retourneert het aantal ontbrekende waarden waarbij `OrderNumber` velden leeg zijn.<br>*"0" telt als waarde en niet als lege cel. Alleen als een meting aggregeert tot 0 voor een dimensie, wordt die dimensie niet opgenomen in diagrammen.* |
| `MissingCount (OrderNumber)/MissingCount ({1} TOTAL OrderNumber)` | De uitdrukking retourneert het aantal onvolledige orders van de geselecteerde klant retourneert als fractie van de onvolledige orders van alle klanten. |

## Voorbeeld - Basisprincipes voor MissingCount

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Customer`
  - `Product`
  - `OrderNumber`
  - `UnitSales`
  - `UnitPrice`

### Load-script

```
Example:
LOAD * inline [
Customer, Product, OrderNumber, UnitSales, 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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Customer`

Maak de volgende metingen:

- `=MissingCount(OrderNumber)`, om het aantal orders zonder `OrderNumber` te berekenen.
- `=MissingCount(OrderNumber)/MissingCount ({1} Total [OrderNumber])`, om het aantal onvolledige orders van de geselecteerde klant te retourneren als fractie van de onvolledige orders van alle klanten. Om deze waarde als percentage weer te geven, selecteert u onder **GetalnotatieGetal > Eenvoudige opmaak > 12,34%**.

Results table

| Customer | MissingCount (OrderNumber) | MissingCount(OrderNumber)/MissingCount ({1} Total [OrderNumber]) |
|---|---|---|
| **Totals** | **4** | **100.00%** |
| Astrida | 0 | 0.00% |
| Betacab | 1 | 25.00% |
| Canutility | 2 | 50.00% |
| Divadip | 1 | 25.00% |

In de eerste meting retourneert de `MissingCount` functie `4` omdat `4` van de `11` records in het veld `OrderNumber` leeg zijn.

*"0" telt als waarde en niet als lege cel. Alleen als een meting aggregeert tot 0 voor een dimensie, wordt die dimensie niet opgenomen in diagrammen.*

In de tweede meting retourneert de uitvoer van de functie `MissingCount` het aantal onvolledige orders van de geselecteerde klant als fractie van de onvolledige orders van alle klanten. Als u bijvoorbeeld `Canutility` uit de dimensie `Customer` selecteert, retourneert de meting `50%` omdat de helft van alle onvolledige orders (`2` van `4`) bij deze klant zijn.

### Voorbeeld - MissingCount gebruiken om ontbrekende productgegevens te identificeren

Diagramuitdrukking

#### Overzicht

Een gegevensverzameling bevat maandelijkse KPI-gegevens (Key Product Indicator). Een productmanager wil uitzoeken welke KPI's ontbrekende gegevens hebben.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
    - `Month`
    - `KPI`
    - `Value`

#### Load-script

```
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,
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `KPI`

Maak de volgende meting:

- `=MissingCount(Value)`, om het aantal ontbrekende waarden te berekenen.

Results table

| KPI | MissingCount(Value) |
|---|---|
| **Totals** | **5** |
| Customer Count | 2 |
| Expenses | 1 |
| Revenue | 2 |

De resultaten laten zien dat `5` records in de gegevensverzameling ontbrekende gegevens hebben, evenals de KPI-categorieën waarop deze records betrekking hebben.

## NullCount

**NullCount()** retourneert het aantal NULL-waarden dat is geaggregeerd in de uitdrukking, zoals gedefinieerd door een **group by**-clausule.

**Syntaxis:**

```
NullCount ( [ distinct ] expr)
```

**Retourgegevenstypen:** geheel getal

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| expr Expression | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| distinct | Als het woord **distinct** voor de uitdrukking staat, worden alle duplicaten genegeerd. |

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens ten minste de velden die in de resultatenkolom staan aan een werkblad in onze app toe om het resultaat te bekijken.

Als u dezelfde weergave wilt als in de onderstaande resultatenkolom, schakelt u in het eigenschappenvenster onder Sorteren over van Automatisch naar Aangepast en schakelt u numeriek en alfabetisch sorteren uit.

Scriptvoorbeelden

| Voorbeeld | Resultaat |
|---|---|
| Set NULLINTERPRET = NULL;<br>Temp:<br>LOAD * inline [<br>Customer\|Product\|OrderNumber\|UnitSales\|CustomerID<br>Astrida\|AA\|1\|10\|1<br>Astrida\|AA\|7\|18\|1<br>Astrida\|BB\|4\|9\|1<br>Astrida\|CC\|6\|2\|1<br>Betacab\|AA\|5\|4\|2<br>Betacab\|BB\|2\|5\|2<br>Betacab\|DD\|\|\|<br>Canutility\|AA\|3\|8\|<br>Canutility\|CC\|NULL\|\|<br>] (delimiter is '\|');<br>Set NULLINTERPRET=;<br>NullCount1:<br><br>LOAD Customer,NullCount(OrderNumber) as NullOrdersByCustomer Resident Temp Group By Customer;<br><br><br>LOAD NullCount(OrderNumber) as TotalNullCount Resident Temp; | Customer NullOrdersByCustomer<br><br>Astrida 0<br><br>Betacab 0<br><br>Canutility 1<br><br>De tweede opdracht geeft als resultaat:<br>TotalNullCount<br><br>1<br>in een tabel met die dimensie, omdat slechts één record een null-waarde bevat. |

## NullCount - diagramfunctie

**NullCount()** wordt gebruikt om het aantal null-waarden in elke diagramdimensie te aggregeren.

**Syntaxis:**

```
NullCount({[SetExpression][DISTINCT] [TOTAL [<fld {,fld}>]]} expr)
```

**Retourgegevenstypen:** geheel getal

Argumenten

| Argument | Beschrijving |
|---|---|
| **expr** | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| **SetExpression** | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| **DISTINCT** | Als het woord **DISTINCT** voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd. |

| Argument | Beschrijving |
|---|---|
| TOTAL | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `NullCount(OrderNumber)` | Retourneert het aantal null-waarden in het veld `OrderNumber`. |
| `NullCount (OrderNumber)/NullCount ({1} Total OrderNumber)` | De uitdrukking retourneert het aantal orders met `NULL` als het `OrderNumber` van de geselecteerde klant als fractie van alle orders van klanten met `NULL` als het `OrderNumber`. |

## Voorbeeld - Basisprincipes voor NullCount

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De `NullInterpret` variabele die waarde afhandelt, is ingesteld op `NULL`. De variabele zal alle records met de tekst `NULL` als nullwaarden interpreteren.
- De volgende velden in de gegevenstabel:
  - `Customer`
  - `Product`
  - `OrderNumber`

### Load-script

```
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 '|');
```

*Om de variabele `NullInterpret` weer op de standaardwaarde in te stellen, neemt u de volgende variabele aan het einde van het load-script op: `Set NULLINTERPRET =;`*

**Resultaten**

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Customer`

Maak de volgende metingen:

- `=NullCount(OrderNumber)`, om het aantal orders met `NULL` als `OrderNumber` te berekenen.
- `=NullCount (OrderNumber)/NullCount ({1} Total OrderNumber)`, om het aantal orders (van de geselecteerde klant) met `NULL` als het `OrderNumber` te berekenen als fractie van alle orders van klanten met `NULL` als het `OrderNumber`. Om deze waarde als percentage weer te geven, selecteert u onder **GetalnotatieGetal > Eenvoudige opmaak > 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% |

De eerste meting retourneert `3` omdat er `3` totale `OrderNumber` records zijn met een `NULL` waarde (`1` record met de klant `Betacab` en `2` records met `Canutility`).

De tweede meting retourneert het aantal `NULL` orders als fractie van `NULL` orders van alle klanten. Bijvoorbeeld, `66.67%` van alle ordernummers met een waarde `NULL` worden toegewezen aan de klant `Canutility`.

## Voorbeeld - NullCount gebruiken om nullwaarden in KPI-gegevens te identificeren

Diagramuitdrukking

**Overzicht**

Een gegevensverzameling bevat maandelijkse KPI-gegevens (Key Product Indicator). Een productmanager wil uitzoeken welke KPI's nullwaarden in de gegevens hebben.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De `NullInterpret` variabele die waarde afhandelt, is ingesteld op `NULL`. De variabele zal alle records met de tekst `NULL` als nullwaarden interpreteren.
- De volgende velden in de gegevenstabel:
  - `Month`
  - `KPI`
  - `Value`

### Load-script

```
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
];
```

*Om de variabele `NullInterpret` weer op de standaardwaarde in te stellen, neemt u de volgende variabele aan het einde van het load-script op: `Set NULLINTERPRET =;`*

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe draaitabel en voeg dit veld toe als een rijdimensie:

- `KPI`

Maak de volgende meting:

- `=NullCount(Value)`, om het aantal ontbrekende waarden te berekenen.

Results table

| KPI | NullCount(Value) |
|---|---|
| Customer Count | 2 |
| Expenses | 1 |
| Revenue | 2 |

De resultaten laten zien dat 5 records in de gegevensverzameling NULL-waarden en KPI-categorieën hebben waarop deze records betrekking hebben.

## NumericCount

**NumericCount()** retourneert het aantal numerieke waarden dat is gevonden in de uitdrukking, zoals gedefinieerd door een **group by**-clausule.

**Syntaxis:**

```
NumericCount ( [ distinct ] expr)
```

**Retourgegevenstypen:** geheel getal

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| expr Expression | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| distinct | Als het woord **distinct** voor de uitdrukking staat, worden alle duplicaten genegeerd. |

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens ten minste de velden die in de resultatenkolom staan aan een werkblad in onze app toe om het resultaat te bekijken.

Als u dezelfde weergave wilt als in de onderstaande resultatenkolom, schakelt u in het eigenschappenvenster onder Sorteren over van Automatisch naar Aangepast en schakelt u numeriek en alfabetisch sorteren uit.

Scriptvoorbeeld

| Voorbeeld | Resultaat |
|---|---|
| `LOAD NumericCount(OrderNumber) as TotalNumericCount Resident Temp;` | De tweede opdracht geeft als resultaat:<br>TotalNumericCount<br>7<br>in een tabel met die dimensie. |

| Voorbeeld | Resultaat |
| --- | --- |
| Stel dat de tabel **Temp** is geladen zoals in het vorige voorbeeld:<br>`LOAD NumericCount(distinct OrderNumber) as TotalNumericCountDistinct Resident Temp;` | TotalNumericCountDistinct<br>6<br>6omdat er één is die een andere dupliceert, dus is het resultaat 6 niet-duplicaten.OrderNumber |

**Voorbeeld:**

```
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(OrderNumber) as NumericCountByCustomer Resident Temp Group By
Customer;
```

Resultaattabel

| Customer | NumericCountByCustomer |
| --- | --- |
| Astrida | 3 |
| Betacab | 2 |
| Canutility | 0 |
| Divadip: | 2 |

## NumericCount - diagramfunctie

**NumericCount()** wordt gebruikt om het aantal numerieke waarden in elke diagramdimensie te aggregeren.

**Syntaxis:**

```
NumericCount({[SetExpression] [DISTINCT] [TOTAL [<fld {,fld}>]]} expr)
```

**Retourgegevenstypen:** geheel getal

Argumenten

| Argument | Beschrijving |
|---|---|
| **expr** | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| **SetExpression** | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| **DISTINCT** | Als het woord **DISTINCT** voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd. |
| **TOTAL** | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `NumericCount (OrderNumber)` | Retourneert als resultaat het aantal rijen in het veld `OrderNumber` met numerieke waarden.<br><br>*"0" telt als waarde en niet als lege cel. Alleen als een meting aggregeert tot 0 voor een dimensie, wordt die dimensie niet opgenomen in diagrammen. Lege cellen worden geëvalueerd als niet-numerieke waarden en worden niet geteld door `NumericCount`.* |
| `NumericCount (Product)` | Retourneert het aantal productnamen die numerieke waarden zijn. Normaal gesproken zou u hiermee kunnen controleren of er geen tekstvelden zijn met numerieke inhoud. |

| Voorbeeld | Resultaat |
| --- | --- |
| `NumericCount (DISTINCT OrderNumber)/Count (DISTINCT OrderNumber)` | Hiermee wordt het aantal verschillende numerieke ordernummers geteld en gedeeld door het aantal numerieke en niet-numerieke ordernummers. Dit wordt 1 als alle veldwaarden numeriek zijn. |

### Voorbeeld - Basisprincipes voor NumericCount

Diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Customer`
  - `Product`
  - `OrderNumber`

#### Load-script

```
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 '|');
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `OrderNumber`

Maak de volgende metingen:

- `=NumericCount(OrderNumber)`, om het aantal orders met numerieke waarden te berekenen.
- `=NumericCount(Product)`, om het aantal productnamen die volledig numerieke waarden zijn te retourneren.
- `=NumericCount(DISTINCT OrderNumber)/Count(DISTINCT OrderNumber)`, om het totale aantal ordernummers met verschillende numerieke waarden te berekenen en dit bedrag te delen door het totale aantal ordernummers met verschillende numerieke en niet-numerieke waarden. Dit wordt `1` als alle veldwaarden numeriek zijn.

Results table

| OrderNumber | NumericCount (OrderNumber) | NumericCount (Product) | NumericCount (DISTINCT OrderNumber)/Count(DISTINCT OrderNumber) |
|---|---|---|---|
| **Totals** | **7** | **0** | **0.875** |
| 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 |

De uitvoer van de eerste meting retourneert `7` omdat drie van de tien records in `OrderNumber` leeg zijn en de overige numeriek zijn.

De tweede meting retourneert `0` omdat er geen numerieke waarden zijn; alle productnamen zijn tekst. Normaal gesproken zou dit type metinge gebruiken om te controleren of er geen tekstvelden zijn met numerieke inhoud.

De derde meting berekent het aantal verschillende numerieke ordernummers geteld en gedeeld door het aantal numerieke en niet-numerieke ordernummers. Dit wordt `1` als alle veldwaarden numeriek zijn. Normaal gesproken zou u hiermee kunnen controleren of alle veldwaarden numeriek zijn. In dit voorbeeld zijn er `7` unieke numerieke waarden voor `OrderNumber` van `8` unieke numerieke en niet-numerieke waarden, zodat de uitdrukking `0.875` retourneert.

## Voorbeeld 5 – Scenario met NumericCount

Diagramuitdrukking

### Overzicht

Een gegevensverzameling bevat maandelijkse verkoopgegevens van verschillende winkels. De winkelmanager wil weten welke winkels en producten verkoopwaarden hebben die niet numeriek zijn. Dit geeft aan welke verkooprecords ongeldige waarden hebben, bijvoorbeeld omdat het record leeg was of niet-numerieke gegevens bevat.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
    - `Store`
    - `StoreProduct`
    - `Sales`

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe draaitabel en voeg dit veld toe als een rijdimensie:

- `Store`

Voeg dit veld toe als een kolomdimensie:

- `StoreProduct`

Maak de volgende meting:

- `=NumericCount(Sales)`, om de winkelverkopen met numerieke gegevens te berekenen.

Results table

| Store | StoreProduct | | |
|---|---|---|---|
| | Widget A | Widget B | Widget C |
| A | 1 | 1 | 1 |
| B | 0 | 0 | 1 |
| C | 1 | 0 | 1 |

De resultaten laten zien dat er `3` records zijn waarvan de numerieke telling `0` is. Dit zijn de verkooprecords die moeten worden onderzocht omdat ze niet-numerieke waarden bevatten.

## TextCount

**TextCount()** retourneert het aantal niet-numerieke veldwaarden dat is geaggregeerd in de uitdrukking, zoals gedefinieerd door een group by-clausule.group by

**Syntaxis:**

```
TextCount ( [ distinct ] expr)
```

**Retourgegevenstypen:** geheel getal

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| expr Expression | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| distinct | Als het woord **distinct** voor de uitdrukking staat, worden alle duplicaten genegeerd. |

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens ten minste de velden die in de resultatenkolom staan aan een werkblad in onze app toe om het resultaat te bekijken.

Als u dezelfde weergave wilt als in de onderstaande resultatenkolom, schakelt u in het eigenschappenvenster onder Sorteren over van Automatisch naar Aangepast en schakelt u numeriek en alfabetisch sorteren uit.

**Voorbeeld:**

```
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;
```

Resultaattabel

| Customer | ProductTextCount |
|---|---|
| Astrida | 3 |
| Betacab | 3 |
| Canutility | 2 |
| Divadip: | 2 |

**Voorbeeld:**

```
LOAD Customer,TextCount(OrderNumber) as OrderNumberTextCount Resident Temp Group By Customer;
```

Resultaattabel

| Customer | OrderNumberTextCount |
|---|---|
| Astrida | 0 |
| Betacab | 1 |
| Canutility | 2 |
| Divadip: | 0 |

## TextCount - diagramfunctie

**TextCount()** wordt gebruikt om het aantal veldwaarden die niet-numeriek zijn in elke diagramdimensie te aggregeren.

**Syntaxis:**

```
TextCount({[SetExpression] [DISTINCT] [TOTAL [<fld {,fld}>]]} expr)
```

**Retourgegevenstypen:** geheel getal

Argumenten

| Argument | Beschrijving |
|---|---|
| **expr** | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |

| Argument | Beschrijving |
|---|---|
| **SetExpression** | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| **DISTINCT** | Als het woord **DISTINCT** voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd. |
| **TOTAL** | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br>Wanneer u **TOTAL [<fld {,fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `TextCount (Product)` | Retourneert het aantal productnamen die tekstwaarden hebben. |
| `TextCount (OrderNumber)` | Retourneert het aantal rijen in het veld `OrderNumber` met tekstwaarden.<br>*"0" telt als waarde en niet als lege cel. Alleen als een meting aggregeert tot 0 voor een dimensie, wordt die dimensie niet opgenomen in diagrammen. Lege cellen worden geëvalueerd als niet-tekst en worden niet geteld door `TextCount`.* |
| `TextCount (DISTINCT Product)/Count (Product)` | Hiermee wordt het aantal distinctieve tekstwaarden van `Product` opgeteld en gedeeld door het totale aantal waarden in `Product`. |

## Voorbeeld - Basisprincipes voor TextCount

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:

- Customer
- Product
- OrderNumber

### Load-script

```
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 '|');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg de volgende metingen toe:

- `=TextCount(Product)`, om het aantal productnamen die tekstwaarden hebben te retourneren.
- `=TextCount(OrderNumber)`, om het aantal ordernummers te berekenen die tekstwaarden zijn.
- `=TextCount(DISTINCT Product)/Count(Product)`, om het aantal distinctieve tekstwaarden van `Product` op te tellen en te delen door het totale aantal waarden in `Product`.

Results table

| TextCount(Product) | TextCount(OrderNumber) | TextCount(DISTINCT Product)/Count(Product) |
|---|---|---|
| 10 | 3 | 0.4 |

De eerste meting retourneert `10` omdat alle tien records in `Product` tekst zijn.

De tweede meting retourneert `3` omdat lege cellen als tekst worden behandeld en geteld. Normaal gesproken zou u dit type metingen kunnen gebruiken om te controleren of er geen numerieke velden zijn die tekstwaarden bevatten of leeg zijn.

De derde meting retourneert `0.4`. Dit resultaat wordt berekend door het aantal distinctieve tekstwaarden van `Product` (`4`) op te tellen en te delen door het totale aantal waarden in `Product` (`10`).

## Voorbeeld 5 – Scenario met TextCount

Diagramuitdrukking

### Overzicht

Een gegevensverzameling bevat responsgegevens uit enquêtes met zowel klachten als algemene feedback per product. U wilt alleen reacties met tekstwaarden bekijken.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `CustomerID`
  - `ProductPurchased`
  - `Complaint`
  - `Feedback`

### Load-script

```
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"
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `ProductPurchased`

Maak de volgende metingen:

- `=TextCount(ProductPurchased)`, om het aantal productnamen met niet-numerieke waarden te berekenen.
- `=TextCount(Complaint)`, om het aantal klachten met niet-numerieke waarden te berekenen.

- `=TextCount(Feedback)`, om het aantal feedbackwaarden met niet-numerieke waarden te berekenen.

Results table

| ProductPurchased | TextCount (ProductPurchased) | TextCount (Complaint) | TextCount (Feedback) |
|---|---|---|---|
| Laptop | 3 | 2 | 2 |
| Mobile | 4 | 2 | 3 |
| Tablet | 2 | 2 | 1 |

De resultaten laten zien dat er voor het product `Laptop` 3 records zijn, waarvan `2` met klachten en `2` met feedback. Ook voor het product `Mobile` zijn er `4` records, inclusief `2` klachten en `3` reacties met feedback. Numerieke waarden of lege waarden worden niet meegeteld omdat verwacht wordt dat de velden `ProductPurchased`, `Complaint` en `Feedback` tekstwaarden zijn.

## Financiële aggregatiefuncties

In dit hoofdstuk worden aggregatiefuncties voor financiële activiteiten beschreven met betrekking tot betalingen en cashflow.

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

### Financiële aggregatiefuncties in het script voor het laden van gegevens

#### IRR

**IRR()** retourneert het geaggregeerde interne opbrengstpercentage voor een reeks cashflows die wordt vertegenwoordigd door de getallen in een uitdrukking, herhaald voor een aantal records dat is vastgelegd door een group by-clausule.

```
IRR (expression)
```

#### XIRR

**XIRR()** retourneert het geaggregeerde interne opbrengstpercentage (jaarlijks) voor een schema van cashflows (niet noodzakelijk periodiek) die worden gerepresenteerd door getallenparen in **pmt** en **date**, die worden herhaald over een aantal records zoals vastgelegd door een group by-clausule. De kortingen voor alle betalingen zijn gebaseerd op een jaar van 365 dagen.

```
XIRR (valueexpression, dateexpression )
```

#### NPV

De **NPV()**-scriptfunctie neemt een kortingstarief en meerdere waarden op volgorde van periode. Ontvangsten, (inkomsten) zijn positief en uitgaven (toekomstige betalingen) worden voor deze berekeningen als negatieve waarden beschouwd. Deze vinden plaats aan het einde van elke periode.

```
NPV (rate, expression)
```

### XNPV

**XNPV()** retourneert de geaggregeerde huidige nettowaarde voor een schema van cashflows (niet noodzakelijk periodiek) die worden gerepresenteerd door getallenparen in **pmt** en **date**. De kortingen voor alle betalingen zijn gebaseerd op een jaar van 365 dagen.

```
XNPV (rate, valueexpression, dateexpression)
```

## Financiële aggregatiefuncties in diagramuitdrukkingen

Deze financiële aggregatiefuncties kunnen worden gebruikt in diagrammen.

### IRR

**IRR()** retourneert het geaggregeerde interne opbrengstpercentage voor een reeks cashflows die wordt vertegenwoordigd door de getallen in een uitdrukking die wordt geleverd door **value**, herhaald voor de diagramdimensies.

```
IRR - diagramfunctie[TOTAL [<fld {,fld}>]] value)
```

### NPV

**NPV()** retourneert de geaggregeerde huidige nettowaarde van een investering gebaseerd op een **discount_rate** per periode en een reeks toekomstige betalingen (negatieve waarden) en inkomsten (positieve waarden) vertegenwoordigd door de getallen in **value**, herhaald op de dimensies van het diagram. We gaan er vanuit dat de betalingen en inkomsten telkens aan het eind van elke periode worden verrekend.

```
NPV - diagramfunctie([TOTAL [<fld {,fld}>]] discount_rate, value)
```

### XIRR

**XIRR()** retourneert het geaggregeerde interne opbrengstpercentage (jaarlijks) voor een schema van cashflows (niet noodzakelijk periodiek) dat wordt vertegenwoordigd door getallenparen in de uitdrukkingen **pmt** en **date**, herhaald op de dimensies van het diagram. De kortingen voor alle betalingen zijn gebaseerd op een jaar van 365 dagen.

```
XIRR - diagramfunctie([TOTAL [<fld {,fld}>]] pmt, date)
```

### XNPV

**XNPV()** retourneert de geaggregeerde interne opbrengstwaarde voor een schema van cashflows (niet noodzakelijk periodiek) dat wordt vertegenwoordigd door getallenparen in de uitdrukkingen **pmt** en **date**, herhaald op de dimensies van het diagram. De kortingen voor alle betalingen zijn gebaseerd op een jaar van 365 dagen.

```
XNPV - diagramfunctie([TOTAL [<fld{,fld}>]] discount_rate, pmt, date)
```

## IRR

**IRR()** retourneert het geaggregeerde interne opbrengstpercentage voor een reeks cashflows die wordt vertegenwoordigd door de getallen in een uitdrukking, herhaald voor een aantal records dat is vastgelegd door een group by-clausule.

Deze cashflows hoeven niet even groot te zijn zoals voor een annuïteit. De cashflows moeten echter met regelmatige intervallen plaatsvinden, zoals maandelijks of jaarlijks. De interne rentabiliteit is het rentetarief dat wordt ontvangen voor een investering die bestaat uit betalingen (negatieve waarden) en inkomsten (positieve waarden) die zich in regelmatige perioden voordoen. De functie heeft ten minste een positieve en een negatieve waarde nodig om te worden berekend.

Deze functie gebruikt een vereenvoudigde versie van de Newton-methode voor het berekenen van de interne rentabiliteit (IRR).

**Syntaxis:**

```
IRR(value)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden worden genegeerd.

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

**Voorbeelden en resultaten:**

Voorbeelden en resultaten

| Voorbeeld | Jaar | IRR2013 |
|---|---|---|
| Cashflow:<br>LOAD 2013 as Year, * inline [<br>Date\|Discount\|Payments<br>2013-01-01\|0.1\|-10000<br>2013-03-01\|0.1\|3000<br>2013-10-30\|0.1\|4200<br>2014-02-01\|0.2\|6800<br>] (delimiter is '\|');<br><br>Cashflow1:<br><br>LOAD Year,IRR(Payments) as IRR2013 Resident Cashflow Group By Year; | 2013 | 0.1634 |

## IRR - diagramfunctie

**IRR()** retourneert het geaggregeerde interne opbrengstpercentage voor een reeks cashflows die wordt vertegenwoordigd door de getallen in een uitdrukking die wordt geleverd door **value**, herhaald voor de diagramdimensies.

Deze cashflows hoeven niet even groot te zijn zoals voor een annuïteit. De cashflows moeten echter met regelmatige intervallen plaatsvinden, zoals maandelijks of jaarlijks. De interne rentabiliteit is het rentetarief dat wordt ontvangen voor een investering die bestaat uit betalingen (negatieve waarden) en inkomsten (positieve waarden) die zich in regelmatige perioden voordoen. De functie heeft ten minste een positieve en een negatieve waarde nodig om te worden berekend.

Deze functie gebruikt een vereenvoudigde versie van de Newton-methode voor het berekenen van de interne rentabiliteit (IRR).

**Syntaxis:**

```
IRR([TOTAL [<fld {,fld}>]] value)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| TOTAL | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br><br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

**Beperkingen:**

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Tekstwaarden, NULL-waarden en ontbrekende waarden worden genegeerd.

**Voorbeelden en resultaten:**

Voorbeelden en resultaten

| Voorbeeld | Resultaat |
|---|---|
| `IRR (Payments)` | 0.1634<br>De betalingen worden geacht periodiek van aard te zijn, bijvoorbeeld maandelijks.<br>*Het veld Datum wordt gebruikt in het XIRR-voorbeeld waarbij betalingen niet-periodiek kunnen zijn, mits u de datums opgeeft waarop de betalingen werden uitgevoerd.* |

Gegevens die worden gebruikt in voorbeelden:

```
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 '|');
```

**Zie ook:**

p *XIRR - diagramfunctie (page 443)*
p *Aggr - diagramfunctie (page 616)*

## NPV

De **NPV()**-scriptfunctie neemt een kortingstarief en meerdere waarden op volgorde van periode. Ontvangsten, (inkomsten) zijn positief en uitgaven (toekomstige betalingen) worden voor deze berekeningen als negatieve waarden beschouwd. Deze vinden plaats aan het einde van elke periode.

De huidige nettowaarde of Net Present Value/ NPV, wordt gebruikt om de huidige totale waarde van een toekomstige stream van cashflows te berekenen. Om de NPV te berekenen, moeten we de toekomstige cashflows per periode berekenen en het correcte kortingstatief bepalen. De **NPV()**-scriptfunctie neemt een kortingstarief en meerdere waarden op volgorde van periode. Ontvangsten, (inkomsten) zijn positief en uitgaven (toekomstige betalingen) worden voor deze berekeningen als negatieve waarden beschouwd. Deze vinden plaats aan het einde van elke periode.

**Syntaxis:**

```
NPV(discount_rate, value)
```

**Retourgegevenstypen:** numeriek. Het resultaat wordt standaard geformatteerd als een valuta.

De formule om de huidige nettowaarde te berekenen is:

$$NPV = \sum_{t=1}^{n} \frac{R_t}{(1+i)^t}$$

waarbij:

- $R_t$ = Netto-ontvangsten/-uitgaven in cash gedurende één periode `t`
- `i` = Kortingspercentage dat of opbrengst die zou kunnen worden verdiend uit alternatieve investeringen
- `t` = Aantal timerperioden

Argumenten

| Argument | Beschrijving |
|---|---|
| discount_rate | **discount_rate** is het percentagetarief van de toegepaste korting.<br><br>Een waarde van 0,1 zou een kortingspercentage van 10% aangeven. |
| value | Dit veld bevat waarden voor meerdere perioden op volgorde van periode. De eerste waarde wordt aangenomen als de cashflow aan het eind van periode 1, enzovoort. |

**Beperkingen:**

De `NPV()`-functie heeft de volgende beperkingen:

- Tekstwaarden, NULL-waarden en ontbrekende waarden worden genegeerd.
- Cashflowwaarden moeten op volgorde van oplopende periode staan.

## Wanneer gebruiken

`NPV()` is een financiële functie die wordt gebruikt om de winstgevendheid van een project te controleren en andere metingen af te leiden. Deze functie is handig wanneer cashflows beschikbaar zijn in de vorm van onbewerkte gegevens.

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – Enkele betaling (script)

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling van één project en de daarbij behorende cashflow voor één periode die in de tabel `CashFlow` wordt geladen.
- Een resident-load uit de tabel `CashFlow` die wordt gebruikt om het NPV-veld te berekenen voor het project in de tabel `NPV`.
- Een vastgesteld kortingspercentage van 10%, dat wordt gebruikt in de berekening van de NPV.
- De opdracht `Group By` die wordt gebruikt om alle betalingen voor het project te groeperen.

### Load-script

```
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;
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `PrjId`
- `NPV`

Resultatentabel

| PrjId | NPV |
|---|---|
| 1 | $909.09 |

Om aan het eind van één periode een enkele betaling van $1000 te kunnen ontvangen met een kortingspercentage van 10% per periode, is de NPV gelijk aan $1000 gedeeld door (1 + kortingspercentage). De effectieve NPV is gelijk aan $909,09

### Voorbeeld 2 – Meerdere betalingen (script)

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset van één project en de daarbij behorende cashflow voor meerdere perioden die in de tabel `CashFlow` wordt geladen.
- Een resident-load uit de tabel `CashFlow` die wordt gebruikt om het NPV-veld te berekenen voor het project in de tabel `NPV`.
- Een vastgesteld kortingspercentage van 10% (0,1), dat wordt gebruikt in de berekening van de NPV.
- De opdracht `Group By` die wordt gebruikt om alle betalingen voor het project te groeperen.

#### Load-script

```
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;
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- PrjId
- NPV

Resultatentabel

| PrjId | NPV |
|---|---|
| 1 | $1735.54 |

Om aan het eind van twee perioden betalingen van $1000 te kunnen ontvangen met een kortingspercentage van 10% per periode, is de effectieve NPV gelijk aan $1735,54.

### Voorbeeld 3 – Meerdere betalingen (script)

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Kortingstarieven voor twee projecten die in de tabel `Project` worden geladen.
- Cashflows voor meerdere perioden per project op project-id en periode-id. Deze periode-id kan worden gebruikt om de records te sorteren in het geval dat de gegevens niet zijn gesorteerd.
- De combinatie van `NoConcatenate`, resident-loads en de `Left Join`-functie om een tijdelijke tabel aan te maken, `tmpNPV`. De tabel combineert de records uit de tabellen `Project` en `CashFlow` in één tabel. In deze tabel worden de kortingstarieven voor iedere periode herhaald.
- Een resident-load uit de tabel `tmpNPV` die wordt gebruikt om het NPV-veld te berekenen voor elk project in de tabel `NPV`.
- Het kortingspercentage voor één waarde dat aan elk project is gekoppeld. Dit wordt opgehaald met de `only()`-functie en wordt gebruikt in de NPV-berekening per project.
- De `Group By`-opdracht, die wordt gebruikt om alle betalingen per project te groeperen op project-id.

Om te vermijden dat synthetische of overbodige gegevens in het gegevensmodel worden geladen, wordt de tabel `tmpNPV` aan het einde van het script verwijderd.

#### Load-script

```
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;
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- PrjId
- NPV

Resultatentabel

| PrjId | NPV |
|---|---|
| 1 | $2486.85 |
| 2 | $2042.12 |

Project-id 1 verwacht dat betalingen van $1000 aan het einde van drie perioden worden ontvangen met een kortingspercentage van 10% per periode. Daarom is de effectieve NPV gelijk aan $2486,85.

Project-id 2 verwacht twee betalingen van $500 en nog eens betalingen van $1000 in vier perioden met een kortingspercentage van 15%. Daarom is de effectieve NPV gelijk aan $2042.12.

## Voorbeeld 4 – Voorbeeld van winstgevendheid van project (script)

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Kortingstarieven en aanvankelijke investeringen (periode 0) voor twee projecten die in de tabel `Project` worden geladen.
- Cashflows voor meerdere perioden per project op project-id en periode-id. Deze periode-id kan worden gebruikt om de records te sorteren in het geval dat de gegevens niet zijn gesorteerd.
- De combinatie van `NoConcatenate`, resident-loads en de `Left Join`-functie om een tijdelijke tabel aan te maken, `tmpNPV`. De tabel combineert de records uit de tabellen `Project` en `CashFlow` in één tabel. In deze tabel worden de kortingstarieven voor iedere periode herhaald.
- Het kortingspercentage voor de enkele waarden dat bij elk project hoort, dat wordt opgehaald met de `only()`-functie en wordt gebruikt in de NPV-berekening per project.
- Een resident-load uit de tabel tmpNPV wordt gebruikt om het NPV-veld te berekenen voor elk project in de tabel NPV.
- Een extra veld dat de NPV verdeelt op basis van de aanvankelijke investering van elk project, wordt gemaakt om de winstgevendheidsindex van het project te berekenen.
- Een Group By-opdracht, die op project-id groepeert, wordt gebruikt om alle betalingen voor elk project te groeperen.

Om te vermijden dat synthetische of overbodige gegevens in het gegevensmodel worden geladen, wordt de tabel `tmpNPV` aan het einde van het script verwijderd.

### Load-script

```
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;
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- PrjId
- NPV

Maak de volgende meting:

```
=only(Profitability_Index)
```

Resultatentabel

| PrjId | NPV | =only(Profitability_Index) |
|---|---|---|
| 1 | $87039.82 | 0.87 |
| 2 | $123513.71 | 1.24 |

Project-id 1 heeft een effectieve NPV van $87039,82 en een aanvankelijke investering van $100.000. Daarom is de winstgevenheidsindex gelijk aan 0,87. Omdat het minder is dan 1, is het project niet winstgevend.

Project-id 2 heeft een effectieve NPV van $123513,71 en een aanvankelijke investering van $100.000. Daarom is de winstgevenheidsindex gelijk aan 1,24. Omdat het groter is dan 1, is het project winstgevend.

## NPV - diagramfunctie

**NPV()** retourneert de geaggregeerde huidige nettowaarde van een investering gebaseerd op een **discount_rate** per periode en een reeks toekomstige betalingen (negatieve waarden) en inkomsten (positieve waarden) vertegenwoordigd door de getallen in **value**, herhaald op de dimensies van het diagram. We gaan er vanuit dat de betalingen en inkomsten telkens aan het eind van elke periode worden verrekend.

**Syntaxis:**

```
NPV([TOTAL [<fld {,fld}>]] discount_rate, value)
```

**Retourgegevenstypen:** numeriek Het resultaat wordt standaard geformatteerd als een valuta.

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| discount_rate | **discount_rate** is het percentagetarief van de toegepaste korting. |
| value | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| TOTAL | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br><br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.<br><br>Na de kwalificatie **TOTAL** kan een lijst met een of meer veldnamen tussen punthaken worden opgegeven. Deze veldnamen moeten een subset zijn van de dimensievariabelen van het diagram. In dit geval wordt bij de berekening geen rekening gehouden met de dimensievariabelen van het diagram met uitzondering van de aangegeven dimensievariabelen. Er wordt één waarde geretourneerd voor elke combinatie van veldwaarden in de opgegeven dimensievelden. Ook velden die momenteel niet een dimensie zijn in een diagram, kunnen worden opgenomen in de lijst. Dit kan nuttig zijn bij groepsdimensies waarbij de dimensievelden niet vast zijn. Als alle variabelen in de groep worden aangegeven, werkt de functie als het drill-downniveau verandert. |

**Beperkingen:**

**discount_rate** en**value** mogen geen aggregatiefuncties bevatten, tenzij deze interne aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Tekstwaarden, NULL-waarden en ontbrekende waarden worden genegeerd.

**Voorbeelden en resultaten:**

Voorbeelden en resultaten

| Voorbeeld | Resultaat |
|---|---|
| `NPV(Discount, Payments)` | -$540.12 |

Gegevens die worden gebruikt in voorbeelden:

```
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 '|');
```

**Zie ook:**

p *XNPV - diagramfunctie (page 453)*
p *Aggr - diagramfunctie (page 616)*

## XIRR

**XIRR()** retourneert het geaggregeerde interne opbrengstpercentage (jaarlijks) voor een schema van cashflows (niet noodzakelijk periodiek) die worden gerepresenteerd door getallenparen in **pmt** en **date**, die worden herhaald over een aantal records zoals vastgelegd door een group by-clausule. De kortingen voor alle betalingen zijn gebaseerd op een jaar van 365 dagen.

De XIRR-functionaliteit van Qlik (de functies **XIRR()** en **RangeXIRR()**) gebruiken de volgende vergelijking, als oplossing voor de `Rate`-waarde, om de juiste XIRR-waarde te bepalen:

```
XNPV(Rate, pmt, date) = 0
```

De vergelijking wordt opgelost met een vereenvoudigde versie van de Newton-methode.

**Syntaxis:**

```
XIRR(pmt, date )
```

**Retourgegevenstypen:** numeriek

Argumenten

| Argument | Beschrijving |
|---|---|
| pmt | Betalingen. De uitdrukking of het veld met de cashflows die horen bij het betalingenschema dat is gegeven in **date**. |
| date | De uitdrukking of het veld met het datumschema dat hoort bij de cashflowbetalingen die zijn gegeven in **pmt**. |

Als u werkt met deze functie, gelden de volgende beperkingen:

- Als een of beide delen van een gegevenspaar tekstwaarden NULL-waarden of ontbrekende waarden bevat, wordt het hele gegevenspaar genegeerd.

- Voor deze functie is ten minste één geldige negatieve en één geldige positieve betaling vereist (met bijbehorende geldige datums). Als deze betalingen niet zijn opgegeven, wordt er een NULL-waarde geretourneerd.

Deze onderwerpen kunnen u helpen bij het werken met deze functie:

- *XNPV (page 447)*: gebruik deze functie om de geaggregeerde huidige nettowaarde te berekenen voor een schema met cashflows.
- *RangeXIRR (page 1638)*: **RangeXIRR()** is de bijbehorende bereikfunctie voor de **XIRR()**-functie.

*Binnen verschillende versies van Qlik Sense met clientbeheer bestaan er variaties op het onderliggende algoritme dat door deze functie wordt gebruikt. Voor meer informatie over recente updates van het algoritme, raadpleegt u het ondersteuningsartikel* XIRR-functie herstellen en bijwerken.

## Voorbeeld

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Transactiegegevens voor een reeks cashflows.
- Het gebruik van de **XIRR()** functie om het interne jaarlijkse opbrengstpercentage voor deze cashflows te berekenen.

### Load-script

```
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;
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- Year
- XIRR2013

Resultatentabel

| Jaar | XIRR2013 |
|---|---|
| 2013 | 0.5385 |

### De XIRR-retourwaarde interpreteren

De XIRR-functionaliteit wordt meestal gebruikt om een investering te analyseren met een uitgaande (negatieve) betaling in het begin en later een aantal kleinere binnenkomende (positieve) betalingen. Hieronder ziet u een vereenvoudigd voorbeeld met één negatieve en één positieve betaling:

```
Cashflow:
LOAD * inline [
Date|Payments
2023-01-01|-100
2024-01-01|110
] (delimiter is '|');
```

We voeren eerst een betaling van 100 uit en precies één jaar later ontvangen we 110 terug. Dit vertegenwoordigt een opbrengstpercentage van 10% per jaar. `XIRR(Payments, Date)` retourneert een waarde van 0.1.

De retourwaarde van de XIRR-functionaliteit kan positief of negatief zijn. In het geval van een investering geeft een negatief resultaat aan dat er sprake is van verlies. De hoeveelheid winst of verlies kan worden berekend door een sum-aggregatie uit te voeren voor de betalingsvelden.

In het bovenstaande voorbeeld lenen we één jaar geld uit. Het opbrengstpercentage kan als rente worden beschouwd. U kunt de XIRR-functionaliteit ook gebruiken als u zich aan de andere kant van de transactie bevindt (bijvoorbeeld als u de lener in plaats van de geldschieter bent).

Neem het volgende voorbeeld:

```
Cashflow:
LOAD * inline [
Date|Payments
2023-01-01|100
2024-01-01|-110
] (delimiter is '|');
```

dit voorbeeld is hetzelfde als hierboven, maar dan omgedraaid. Hier lenen we 100 voor één jaar en we betalen het terug met 10% rente. In dit voorbeeld geeft de XIRR-functionaliteit 0.1 (10%), dezelfde waarde als in het eerste voorbeeld.

In het eerste voorbeeld ontvingen we een winst van 10 en in het tweede voorbeeld hadden we een verlies van 10, maar de retourwaarde van de XIRR-functionaliteit is in beide voorbeelden positief. Dit komt doordat de XIRR-functionaliteit de verborgen rente van de transactie berekend, ongeacht aan welke kant van de transactie u zich bevindt.

### Beperkingen met meerdere uitkomsten

DE XIRR-functionaliteit van Qlik wordt gedefinieerd door de volgende vergelijking waarin de `Rate`-waarde wordt opgelost:

```
XNPV(Rate, pmt, date) = 0
```

bij deze vergelijking is er meer dan één uitkomst mogelijk. Dit wordt ook wel het 'meervoudig IRR-probleem' genoemd en dit wordt veroorzaakt door een ongewone cashflowstroom (ook wel een niet-conventionele cashflow genoemd). Het volgende load-script is hier een voorbeeld van:

```
Cashflow:
LOAD * inline [
Date|Payments
2021-01-01|-200
2022-01-01|500
2023-01-01|-250
] (delimiter is '|');
```

In dit voorbeeld is er een negatieve uitkomt en een positieve uitkomst (`Rate` = -0.3 en `Rate` = 0.8). **XIRR()** geeft 0.8.

Als de XIRR-functionaliteit van Qlik zoekt naar een uitkomst, is het startpunt `Rate` = 0 en wordt het tarief in stappen verhoogd totdat de uitkomst is gevonden. Als er meer dan één positieve uitkomst mogelijk is, wordt de eerste uitkomst gegeven. Als er geen positieve uitkomst wordt gevonden, wordt `Rate` teruggezet naar nul en wordt er gezocht naar een negatieve uitkomst.

Bij een 'normale' cashflowstream is er gegarandeerd maar één uitkomst mogelijk. Een 'normale' cashflowstream betekent dat alle betalingen met hetzelfde teken (positief of negatief) zich in een onafgebroken groep bevinden.

---

**Zie ook:**

p *XNPV (page 447)*
p *RangeXIRR (page 1638)*
≤ XIRR-functie herstellen en bijwerken

## XIRR - diagramfunctie

**XIRR()** retourneert het geaggregeerde interne opbrengstpercentage (jaarlijks) voor een schema van cashflows (niet noodzakelijk periodiek) dat wordt vertegenwoordigd door getallenparen in de uitdrukkingen **pmt** en **date**, herhaald op de dimensies van het diagram. De kortingen voor alle betalingen zijn gebaseerd op een jaar van 365 dagen.

De XIRR-functionaliteit van Qlik (de functies **XIRR()** en **RangeXIRR()**) gebruiken de volgende vergelijking, als oplossing voor de `Rate`-waarde, om de juiste XIRR-waarde te bepalen:

```
XNPV(Rate, pmt, date) = 0
```

De vergelijking wordt opgelost met een vereenvoudigde versie van de Newton-methode.

**Syntaxis:**

```
XIRR([TOTAL [<fld {,fld}>]] pmt, date)
```

**Retourgegevenstypen:** numeriek

Argumenten

| Argument | Beschrijving |
|---|---|
| pmt | Betalingen. De uitdrukking of het veld met de cashflows die horen bij het betalingenschema dat is gegeven in **date**. |
| date | De uitdrukking of het veld met het datumschema dat hoort bij de cashflowbetalingen die zijn gegeven in **pmt**. |
| TOTAL | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br><br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

Als u werkt met deze functie, gelden de volgende beperkingen:

- **pmt** en**date** mogen geen aggregatiefuncties bevatten, tenzij deze interne aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.
- Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.
- Voor deze functie is ten minste één geldige negatieve en één geldige positieve betaling vereist (met bijbehorende geldige datums). Als deze betalingen niet zijn opgegeven, wordt er een NULL-waarde geretourneerd.

Deze onderwerpen kunnen u helpen bij het werken met deze functie:

- *XNPV - diagramfunctie (page 453)*: gebruik deze functie om de geaggregeerde huidige nettowaarde te berekenen voor een schema met cashflows.
- *RangeXIRR (page 1638)*: **RangeXIRR()** is de bijbehorende bereikfunctie voor de **XIRR()**-functie.

*Binnen verschillende versies van Qlik Sense met clientbeheer bestaan er variaties op het onderliggende algoritme dat door deze functie wordt gebruikt. Voor meer informatie over recente updates van het algoritme, raadpleegt u het ondersteuningsartikel* XIRR-functie herstellen en bijwerken.

## Voorbeeld

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die cashflowtransacties bevat.
- Gegevens die zijn opgeslagen in een tabel met de naam `Cashflow`.

### Load-script

```
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 '|');
```

### Resultaten

### Doe het volgende:

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg de volgende berekening als een meting toe:

```
=XIRR(Payments, Date)
```

Resultatentabel

| =XIRR(Payments, Date) |
| --- |
| 0.5385 |

## De XIRR-retourwaarde interpreteren

De XIRR-functionaliteit wordt meestal gebruikt om een investering te analyseren met een uitgaande (negatieve) betaling in het begin en later een aantal kleinere binnenkomende (positieve) betalingen. Hieronder ziet u een vereenvoudigd voorbeeld met één negatieve en één positieve betaling:

```
Cashflow:
LOAD * inline [
Date|Payments
2023-01-01|-100
2024-01-01|110
] (delimiter is '|');
```

We voeren eerst een betaling van 100 uit en precies één jaar later ontvangen we 110 terug. Dit vertegenwoordigt een opbrengstpercentage van 10% per jaar. `XIRR(Payments, Date)` retourneert een waarde van 0.1.

De retourwaarde van de XIRR-functionaliteit kan positief of negatief zijn. In het geval van een investering geeft een negatief resultaat aan dat er sprake is van verlies. De hoeveelheid winst of verlies kan worden berekend door een sum-aggregatie uit te voeren voor de betalingsvelden.

In het bovenstaande voorbeeld lenen we één jaar geld uit. Het opbrengstpercentage kan als rente worden beschouwd. U kunt de XIRR-functionaliteit ook gebruiken als u zich aan de andere kant van de transactie bevindt (bijvoorbeeld als u de lener in plaats van de geldschieter bent).

Neem het volgende voorbeeld:

```
Cashflow:
LOAD * inline [
Date|Payments
2023-01-01|100
2024-01-01|-110
] (delimiter is '|');
```

dit voorbeeld is hetzelfde als hierboven, maar dan omgedraaid. Hier lenen we 100 voor één jaar en we betalen het terug met 10% rente. In dit voorbeeld geeft de XIRR-functionaliteit 0.1 (10%), dezelfde waarde als in het eerste voorbeeld.

In het eerste voorbeeld ontvingen we een winst van 10 en in het tweede voorbeeld hadden we een verlies van 10, maar de retourwaarde van de XIRR-functionaliteit is in beide voorbeelden positief. Dit komt doordat de XIRR-functionaliteit de verborgen rente van de transactie berekend, ongeacht aan welke kant van de transactie u zich bevindt.

### Beperkingen met meerdere uitkomsten

DE XIRR-functionaliteit van Qlik wordt gedefinieerd door de volgende vergelijking waarin de `Rate`-waarde wordt opgelost:

```
XNPV(Rate, pmt, date) = 0
```

bij deze vergelijking is er meer dan één uitkomst mogelijk. Dit wordt ook wel het 'meervoudig IRR-probleem' genoemd en dit wordt veroorzaakt door een ongewone cashflowstroom (ook wel een niet-conventionele cashflow genoemd). Het volgende load-script is hier een voorbeeld van:

```
Cashflow:
LOAD * inline [
Date|Payments
2021-01-01|-200
2022-01-01|500
2023-01-01|-250
] (delimiter is '|');
```

In dit voorbeeld is er een negatieve uitkomt en een positieve uitkomst (`Rate` = -0.3 en `Rate` = 0.8). **XIRR()** geeft 0.8.

Als de XIRR-functionaliteit van Qlik zoekt naar een uitkomst, is het startpunt `Rate` = 0 en wordt het tarief in stappen verhoogd totdat de uitkomst is gevonden. Als er meer dan één positieve uitkomst mogelijk is, wordt de eerste uitkomst gegeven. Als er geen positieve uitkomst wordt gevonden, wordt `Rate` teruggezet naar nul en wordt er gezocht naar een negatieve uitkomst.

Bij een 'normale' cashflowstream is er gegarandeerd maar één uitkomst mogelijk. Een 'normale' cashflowstream betekent dat alle betalingen met hetzelfde teken (positief of negatief) zich in een onafgebroken groep bevinden.

**Zie ook:**

p *IRR - diagramfunctie (page 430)*
p *Aggr - diagramfunctie (page 616)*
≤ XIRR-functie herstellen en bijwerken

## XNPV

**XNPV()** retourneert de geaggregeerde huidige nettowaarde voor een schema van cashflows (niet noodzakelijk periodiek) die worden gerepresenteerd door getallenparen in **pmt** en **date**. De kortingen voor alle betalingen zijn gebaseerd op een jaar van 365 dagen.

**Syntaxis:**

```
XNPV(discount_rate, pmt, date)
```

**Retourgegevenstypen:** numeriek

*Het resultaat wordt standaard geformatteerd als een valuta.*

De formule om de XNPV te berekenen wordt hieronder getoond:

*XNPV aggregatieformule*

$$XNPV = \sum_{i=1}^{n} \frac{P_i}{(1+rate)^{(di-d1)/365}}$$

waarbij:

- $P_i$ = Netto-ontvangsten/-uitgaven in cash gedurende één periode `i`
- $d_1$ = de eerste betaaldatum
- $d_i$ = de `i`e betaaldatum
- `rate` = kortingspercentage

De huidige nettowaarde of Net Present Value/ NPV, wordt gebruikt om de huidige totale waarde van een toekomstige stream van cashflows met een kortingspecentage te berekenen. Om de XNPV te berekenen moeten we de toekomstige cashflows met bijbehorende datums inschatten. Hierna passen we voor elke betaling het samengestelde kortingspercentage op basis van de betalingsdatum toe.

Het uitvoeren van de XNPV-aggregatie voor een reeks betalingen is vergelijkbaar met het uitvoeren van een sum-aggregatie voor die betalingen. Het verschil is dat elk bedrag wordt aangepast (of dat korting wordt toegepast) op basis van het gekozen kortingspercentage (vergelijkbaar met het rentepercentage) en hoe ver in de toekomst de betaling wordt uitgevoerd. Als u XNPV uitvoert met de parameter **discount_rate** ingesteld op nul, wordt de XNPV gelijkgesteld aan de sum-bewerking (de betalingen worden niet aangepast voordat ze bij elkaar worden opgeteld). Over het algemeen geldt dat hoe dichter de waarde van **discount_rate** bij nul ligt, hoe meer het XNPV-resultaat overeenkomt met dat van een sum-aggregatie.

Argumenten

| Argument | Beschrijving |
|---|---|
| discount_rate | **discount_rate** is het jaartarief waarmee korting op betalingen wordt toegepast.<br><br>Een waarde van 0,1 zou een kortingspercentage van 10% aangeven. |
| pmt | Betalingen. De uitdrukking of het veld met de cashflows die horen bij het betalingenschema dat is gegeven in **date**. Positieve waarden worden als ontvangsten beschouwd en negatieve waarden als uitgaven.<br><br>*XNPV() past geen korting toe op de initiële cashflow omdat dit altijd gebeurt op de begindatum. De kortingen voor daaropvolgende betalingen zijn gebaseerd op een jaar van 365 dagen. Dit verschilt van de NPV(), waarbij ook korting op de eerste betaling wordt toegepast.* |
| date | De uitdrukking of het veld met het datumschema dat hoort bij de cashflowbetalingen die zijn gegeven in **pmt**. De eerste waarde wordt gebruikt als begindatum voor het berekenen van de tijdverschuivingen voor toekomstige cashflows. |

Als u werkt met deze functie, gelden de volgende beperkingen:

- Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

## Wanneer gebruiken

- `XNPV()` wordt gebruikt in financiële modellen voor het berekenen van de huidige nettowaarde (NPV) van een investeringskans.
- Vanwege de grotere nauwkeurigheid wordt voor alle typen financiële modellen de voorkeur gegeven aan XNPV boven NPV.

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – Enkele betaling (script)

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling van één project en de daarbij behorende cashflow voor één jaar in de tabel `CashFlow`. De begindatum voor de berekening is ingesteld op 1 juli 2022, met een nettocashflow van 0. Na één jaar, treedt een cashflow van $1000 op.
- Een resident-load uit de tabel `CashFlow` die wordt gebruikt om het `XNPV`-veld te berekenen voor het project in de tabel `XNPV`.
- Een vastgesteld kortingspercentage van 10% (0,1) wordt gebruikt in de berekening van de XNPV.
- De opdracht `Group By` wordt gebruikt om alle betalingen voor het project te groeperen.

### Load-script

```
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;
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- PrjId
- XNPV

Resultatentabel

| PrjId | XNPV |
|---|---|
| 1 | $909.09 |

Volgens formule, is de XNPV-waarde voor de eerste record 0 en voor de tweede is de XNPV-waarde $909,09. Daarmee is de totale XNPV $909,09.

### Voorbeeld 2 – Meerdere betalingen (script)

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset van één project en de daarbij behorende cashflow voor één jaar in de tabel CashFlow.
- Een resident-load uit de tabel CashFlow die wordt gebruikt om het XNPV-veld te berekenen voor het project in de tabel XNPV.
- Een vastgesteld kortingspercentage van 10% (0,1) wordt gebruikt in de berekening van de XNPV.
- De opdracht Group By wordt gebruikt om alle betalingen voor het project te groeperen.

#### Load-script

```
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;
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- PrjId
- XNPV

Resultatentabel

| PrjId | XNPV |
|---|---|
| 1 | $1322.21 |

In dit voorbeeld wordt er aan het eind van het eerste jaar een betaling van $1000 ontvangen en een betaling van $500 aan het eind van het tweede jaar. Met een kortingspercentage van 10% per periode is de effectieve XNPV gelijk aan $1322,21.

Let op: alleen de eerste rij gegevens mag verwijzen naar de basisgegevens voor berekeningen. Voor de rest van de rijen is de volgorde niet belangrijk omdat de datumparameter wordt gebruikt om de verlopen periode te berekenen.

### Voorbeeld 3 – Meerdere betalingen en onregelmatige cashflows (script)

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Kortingstarieven voor twee projecten in de tabel Project.
- Cashflows voor meerdere perioden per project op project-id en datums. Het Dates-veld wordt gebruikt om de duur te berekenen dat het kortingspercentage wordt toegepast op de cashflow. Met uitzondering van de eerste record (aanvankelijke cashflow en datum), is de volgorde van de records niet belangrijk en deze veranderen zou geen invloed moeten hebben op de berekeningen.
- Door een combinatie van NoConcatenate, resident-loads en de functie Left Join te gebruiken, wordt een tijdelijke tabel, tmpNPV, aangemaakt die de records van de tabellen Project en CashFlow in één tabel combineert. In deze tabel worden de kortingstarieven voor iedere cashflow herhaald.
- Een resident-load uit de tabel tmpNPV die wordt gebruikt om het XNPV-veld te berekenen voor elk project in de tabel XNPV.
- Het kortingspercentage voor de enkele waarde dat bij elk project hoort, wordt opgehaald met de only() -functie en wordt gebruikt in de XNPV-berekening per project.
- Een Group By-opdracht, gegroepeerd op project-id, wordt gebruikt om alle betalingen en bijbehorende datums voor elk project te groeperen.
- Om te vermijden dat synthetische of overbodige gegevens in het gegevensmodel worden geladen, wordt de tabel tmpXNPV aan het einde van het script verwijderd.

### Load-script

```
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;
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- PrjId
- XNPV

Resultatentabel

| PrjId | XNPV |
|---|---|
| 1 | $1735.54 |
| 2 | $278.36 |

Project-id 1 heeft een aanvankelijke cashflow van $0 op 1 juli 2021. Er zijn twee betalingen van $1000 die aan het eind van twee achtereenvolgende jaren moeten worden ontvangen met een kortingspercentage van 10% per periode. Daarom is de effectieve NPV gelijk aan $1735,54.

Project-id 2 heeft een aanvankelijke uitgave van $1000 (vandaar het negatieve teken) op 1 juli 2020. Na twee jaar wordt een betaling van $500 verwacht. Na drie jaar wordt nog een betaling van $500 verwacht. Als laatste wordt op 1 juli 2024 een betaling van $1000 verwacht. Met het kortingspercentage van 15% is de effectieve XNPV gelijk aan $278,36.

**Zie ook:**

p *Drop table (page 158)*
p *group by (page 170)*
p *Join (page 75)*
p *Max (page 369)*
p *NoConcatenate (page 93)*
p *NPV - diagramfunctie (page 438)*
p *Only (page 386)*

## XNPV - diagramfunctie

**XNPV()** retourneert de geaggregeerde interne opbrengstwaarde voor een schema van cashflows (niet noodzakelijk periodiek) dat wordt vertegenwoordigd door getallenparen in de uitdrukkingen **pmt** en **date**, herhaald op de dimensies van het diagram. De kortingen voor alle betalingen zijn gebaseerd op een jaar van 365 dagen.

**Syntaxis:**

```
XNPV([TOTAL [<fld{,fld}>]] discount_rate, pmt, date)
```

**Retourgegevenstypen:** numeriek

> *Het resultaat wordt standaard geformatteerd als een valuta.*

De formule om de XNPV te berekenen wordt hieronder getoond:

*XNPV aggregatieformule*

$$XNPV = \sum_{i=1}^{n} \frac{P_i}{(1+rate)^{(di-d1)/365}}$$

waarbij:

- $P_i$ = Netto-ontvangsten/-uitgaven in cash gedurende één periode `i`
- $d_1$ = de eerste betaaldatum
- $d_i$ = de $i^e$ betaaldatum
- `rate` = kortingspercentage

De huidige nettowaarde of Net Present Value/ NPV, wordt gebruikt om de huidige totale waarde van een toekomstige stream van cashflows met een kortingspecentage te berekenen. Om de XNPV te berekenen moeten we de toekomstige cashflows met bijbehorende datums inschatten. Hierna passen we voor elke betaling het samengestelde kortingspercentage op basis van de betalingsdatum toe.

Het uitvoeren van de XNPV-aggregatie voor een reeks betalingen is vergelijkbaar met het uitvoeren van een sum-aggregatie voor die betalingen. Het verschil is dat elk bedrag wordt aangepast (of dat korting wordt toegepast) op basis van het gekozen kortingspercentage (vergelijkbaar met het rentepercentage) en hoe ver in de toekomst de betaling wordt uitgevoerd. Als u XNPV uitvoert met de parameter **discount_rate** ingesteld op nul, wordt de XNPV gelijkgesteld aan de sum-bewerking (de betalingen worden niet aangepast voordat ze bij elkaar worden opgeteld). Over het algemeen geldt dat hoe dichter de waarde van **discount_rate** bij nul ligt, hoe meer het XNPV-resultaat overeenkomt met dat van een sum-aggregatie.

Argumenten

| Argument | Beschrijving |
|---|---|
| discount_rate | **discount_rate** is het jaartarief waarmee korting op betalingen wordt toegepast.<br><br>Een waarde van 0,1 zou een kortingspercentage van 10% aangeven. |
| pmt | Betalingen. De uitdrukking of het veld met de cashflows die horen bij het betalingenschema dat is gegeven in **date**. Positieve waarden worden als ontvangsten beschouwd en negatieve waarden als uitgaven.<br><br>*XNPV() past geen korting toe op de initiële cashflow omdat dit altijd gebeurt op de begindatum. De kortingen voor daaropvolgende betalingen zijn gebaseerd op een jaar van 365 dagen. Dit verschilt van de **NPV()**, waarbij ook korting op de eerste betaling wordt toegepast.* |
| date | De uitdrukking of het veld met het datumschema dat hoort bij de cashflowbetalingen die zijn gegeven in **pmt**. De eerste waarde wordt gebruikt als begindatum voor het berekenen van de tijdverschuivingen voor toekomstige cashflows. |
| TOTAL | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br><br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

Als u werkt met deze functie, gelden de volgende beperkingen:

- **discount_rate**, **pmt** en **date** mogen geen aggregatiefuncties bevatten, tenzij deze interne aggregaties de kwalificaties **TOTAL** of **ALL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

- Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

## Wanneer gebruiken

- `XNPV()` wordt gebruikt in financiële modellen voor het berekenen van de huidige nettowaarde (NPV) van een investeringskans.
- Vanwege de grotere nauwkeurigheid wordt voor alle typen financiële modellen de voorkeur gegeven aan XNPV boven NPV.

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld

### Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die cashflowtransacties bevat.
- Gegevens die zijn opgeslagen in een tabel met de naam `Cashflow`.

#### Load-script

```
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 '|');
```

**Resultaten**

**Doe het volgende:**

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg de volgende berekening als een meting toe:

```
=XNPV(0.09, Payments, Date)
```

Resultatentabel

| =XNPV(0.09, Payments, Date) |
| --- |
| $3062.49 |

**Zie ook:**

p *NPV - diagramfunctie (page 438)*
p *Aggr - diagramfunctie (page 616)*

## Statistische aggregatiefuncties

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

### Statistische aggregatiefuncties in het script voor het laden van gegevens

De volgende statistische aggregatiefuncties kunnen worden gebruikt in scripts.

**Avg**
**Avg()** retourneert de gemiddelde waarde van de geaggregeerde gegevens in de uitdrukking voor een aantal records, zoals gedefinieerd door een **group by**-clausule.

```
Avg ([distinct] expression)
```

**Correl**
**Correl()** retourneert de geaggregeerde correlatiecoëfficiënt voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals vastgelegd door een **group by**-clausule.

```
Correl (x-expression, y-expression)
```

**Fractile**
**Fractile()** retourneert de waarde die overeenkomt met de inclusieve fractiel (kwantiel) van de geaggregeerde gegevens in de uitdrukking voor een aantal records, zoals gedefinieerd door een **group by**-clausule.

```
Fractile (expression, fractile)
```

### FractileExc

**FractileExc()** retourneert de waarde die overeenkomt met de exclusieve fractiel (kwantiel) van de geaggregeerde gegevens in de uitdrukking voor een aantal records, zoals gedefinieerd door een **group by**-clausule.

```
FractileExc (expression, fractile)
```

### Kurtosis

**Kurtosis()** retourneert de kurtosis van de gegevens in de uitdrukking voor een aantal records, zoals gedefinieerd door een **group by**-clausule.

```
Kurtosis ([distinct ] expression )
```

### LINEST_B

**LINEST_B()** retourneert geaggregeerde b-waarde (y-intercept) van een lineaire regressie, die wordt gedefinieerd door de vergelijking y=mx+b voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals vastgelegd door een **group by**-clausule.

```
LINEST_B (y-expression, x-expression [, y0 [, x0 ]])
```

### LINEST_df

**LINEST_DF()** retourneert geaggregeerde vrijheidsgraden van een lineaire regressie, die wordt gedefinieerd door de vergelijking y=mx+b voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals vastgelegd door een **group by**-clausule.

```
LINEST_DF (y-expression, x-expression [, y0 [, x0 ]])
```

### LINEST_f

Deze scriptfunctie retourneert de geaggregeerde F-statistiek ($r^2/(1-r^2)$) van een lineaire regressie, die wordt gedefinieerd door de vergelijking y=mx+b voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression die worden herhaald over een aantal records zoals vastgelegd door een **group by**-clausule.

```
LINEST_F (y-expression, x-expression [, y0 [, x0 ]])
```

### LINEST_m

**LINEST_M()** retourneert geaggregeerde m-waarde (helling) van een lineaire regressie, die wordt gedefinieerd door de vergelijking y=mx+b voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals vastgelegd door een **group by**-clausule.

```
LINEST_M (y-expression, x-expression [, y0 [, x0 ]])
```

### LINEST_r2

retourneert de geaggregeerde -waarde (determinatiecoëfficiënt) van een lineaire regressie, die wordt gedefinieerd door de vergelijking voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in en , die worden herhaald over aantal records zoals vastgelegd door een -

$r^2$y=mx+bx-expressiony-expressiongroup by

```
LINEST_R2 (y-expression, x-expression [, y0 [, x0 ]])
```

### LINEST_seb

**LINEST_SEB()** retourneert de geaggregeerde standaardfout van de b-waarde van een lineaire regressie, die wordt gedefinieerd door de vergelijking y=mx+b voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals vastgelegd door een **group by**-clausule.

```
LINEST_SEB (y-expression, x-expression [, y0 [, x0 ]])
```

### LINEST_sem

**LINEST_SEM()** retourneert de geaggregeerde standaardfout van de m-waarde van een lineaire regressie, die wordt gedefinieerd door de vergelijking y=mx+b voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals vastgelegd door een **group by**-clausule.

```
LINEST_SEM (y-expression, x-expression [, y0 [, x0 ]])
```

### LINEST_sey

**LINEST_SEY()** retourneert de geaggregeerde standaardfout van de geschatte y-waarde van een lineaire regressie, die wordt gedefinieerd door de vergelijking y=mx+b voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals vastgelegd door een **group by**-clausule.

```
LINEST_SEY (y-expression, x-expression [, y0 [, x0 ]])
```

### LINEST_ssreg

retourneert de geaggregeerde regressie voor de som van de kwadraten van een lineaire regressie, die wordt gedefinieerd door de vergelijking voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in en , herhaald over een aantal records zoals vastgelegd door een -clausule.y=mx+bx-expressiony-expressiongroup by

```
LINEST_SSREG (y-expression, x-expression [, y0 [, x0 ]])
```

### Linest_ssresid

**LINEST_SSRESID()** retourneert de geaggregeerde regressie voor de som van de kwadraten van een lineaire regressie, die wordt gedefinieerd door de vergelijking y=mx+b voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals vastgelegd door een **group by**-clausule.

```
LINEST_SSRESID (y-expression, x-expression [, y0 [, x0 ]])
```

### Median

**Median()** retourneert de geaggregeerde mediaan van de gegevens in de uitdrukking voor een aantal records, zoals gedefinieerd door een **group by**-clausule.

```
Median (expression)
```

### Skew

**Skew()** retourneert de scheefheid van de uitdrukking voor een aantal records zoals gedefinieerd door een **group by**-clausule.

```
Skew ([ distinct] expression)
```

### Stdev

**Stdev()** retourneert de standaarddeviatie van de waarden in de uitdrukking voor een aantal records, zoals gedefinieerd door een **group by**-clausule.

```
Stdev ([distinct] expression)
```

### Sterr

**Sterr()** retourneert de geaggregeerde standaardfout (stdev/sqrt(n)) voor een reeks waarden die wordt vertegenwoordigd door de uitdrukking, herhaald over een aantal records zoals gedefinieerd door een **group by**-clausule.

```
Sterr ([distinct] expression)
```

### STEYX

**STEYX()** retourneert de geaggregeerde standaardfout van de voorspelde y-waarde voor elke x-waarde in de regressie voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals gedefinieerd door een **group by**-clausule.

```
STEYX (y-expression, x-expression)
```

## Statistische aggregatiefuncties in diagramuitdrukkingen

De volgende statistische aggregatiefuncties kunnen worden gebruikt in diagrammen.

### Avg

**Avg()** retourneert het geaggregeerde gemiddelde van uitdrukking of veld geïtereerd op de diagramdimensies.

```
Avg - diagramfunctie({[SetExpression] [DISTINCT] [TOTAL [<fld{, fld}>]]}
expr)
```

### Correl

**Correl()** retourneert de geaggregeerde correlatiecoëfficiënt voor twee gegevensverzamelingen. De correlatiefunctie is een maat voor de relatie tussen de gegevensverzamelingen en is geaggregeerd voor (x,y)-waardeparen, herhaald op de dimensies van het diagram.

```
Correl - diagramfunctie({[SetExpression] [TOTAL [<fld {, fld}>]]} value1,
value2 )
```

### Fractile

**Fractile()** retourneert de waarde die correspondeert met de inclusieve fractiel (kwantiel) van de geaggregeerde gegevens in het bereik van de uitdrukking, geïtereerd op de diagramdimensies.

```
Fractile - diagramfunctie({[SetExpression] [TOTAL [<fld {, fld}>]]} expr,
fraction)
```

FractileExc

**FractileExc()** retourneert de waarde die correspondeert met de exclusieve fractiel (kwantiel) van de geaggregeerde gegevens in het bereik van de uitdrukking, geïtereerd op de diagramdimensies.

```
FractileExc - diagramfunctie({[SetExpression] [TOTAL [<fld {, fld}>]]} expr, fraction)
```

Kurtosis

**Kurtosis()** retourneert de piekvorm van het bereik van geaggregeerde waarden in de uitdrukking of het veld, geïtereerd op de diagramdimensies.

```
Kurtosis - diagramfunctie({[SetExpression] [DISTINCT] [TOTAL [<fld{, fld}>]]} expr)
```

LINEST_b

**LINEST_B()** retourneert de geaggregeerde b-waarde (y-intercept) van een lineaire regressie, gedefinieerd door de vergelijking y=mx+b voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x_value** en **y_value**, geïtereerd op de diagramdimensies.

```
LINEST_R2 - diagramfunctie({[SetExpression] [TOTAL [<fld{ ,fld}>]] }y_value, x_value[, y0_const[, x0_const]])
```

LINEST_df

**LINEST_DF()** retourneert de geaggregeerde vrijheidsgraden van een lineaire regressie, gedefinieerd door de vergelijking y=mx+b voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x_value** en **y_value**, geïtereerd op de diagramdimensies.

```
LINEST_DF - diagramfunctie({[SetExpression] [TOTAL [<fld{, fld}>]]} y_value, x_value [, y0_const [, x0_const]])
```

LINEST_f

**LINEST_F()** retourneert de geaggregeerde F-statistiek (r2/(1-r2)) van een lineaire regressie, gedefinieerd door de vergelijking y=mx+b voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x_value** en **y_value**, geïtereerd op de diagramdimensies.

```
LINEST_F - diagramfunctie({[SetExpression] [TOTAL[<fld{, fld}>]]} y_value, x_value [, y0_const [, x0_const]])
```

LINEST_m

**LINEST_M()** retourneert de geaggregeerde m-waarde (helling) van een lineaire regressie, gedefinieerd door de vergelijking  y=mx+b voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x_value** en **y_value**, geïtereerd op de diagramdimensies.

```
LINEST_M - diagramfunctie({[SetExpression] [TOTAL[<fld{, fld}>]]} y_value, x_value [, y0_const [, x0_const]])
```

LINEST_r2

**LINEST_R2()** retourneert de geaggregeerde r2-waarde (determinatiecoëfficient) van een lineaire regressie, gedefinieerd door de vergelijking y=mx+b voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x_value** en **y_value**, geïtereerd op de diagramdimensies.

```
LINEST_R2 - diagramfunctie({[SetExpression] [TOTAL [<fld{ ,fld}>]] }y_value,
x_value[, y0_const[, x0_const]])
```

LINEST_seb

**LINEST_SEB()** retourneert de geaggregeerde standaardfout van de b-waarde van een lineaire regressie, gedefinieerd door de vergelijking y=mx+b voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x_value** en **y_value**, geïtereerd op de diagramdimensies.

```
LINEST_SEB - diagramfunctie({[SetExpression] [TOTAL [<fld{ ,fld}>]] }y_value,
x_value[, y0_const[, x0_const]])
```

LINEST_sem

**LINEST_SEM()** retourneert de geaggregeerde standaardfout van de m-waarde van een lineaire regressie, gedefinieerd door de vergelijking y=mx+b voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x_value** en **y_value**, geïtereerd op de diagramdimensies.

```
LINEST_SEM - diagramfunctie([{set_expression}][ distinct ] [total [<fld
{,fld}>] ] y-expression, x-expression [, y0 [, x0 ]] )
```

LINEST_sey

**LINEST_SEY()** retourneert de geaggregeerde standaardfout van de geschatte y-waarde van een lineaire regressie, gedefinieerd door de vergelijking y=mx+b voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x_value** en **y_value**, geïtereerd op de diagramdimensies.

```
LINEST_SEY - diagramfunctie({[SetExpression] [TOTAL [<fld{ ,fld}>]] }y_value,
x_value[, y0_const[, x0_const]])
```

LINEST_ssreg

**LINEST_SSREG()** retourneert de geaggregeerde regressie voor de som van de kwadraten van een lineaire regressie die wordt gedefinieerd door de vergelijking y=mx+b voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x_value** en **y_value**, geïtereerd op de diagramdimensies.

```
LINEST_SSREG - diagramfunctie({[SetExpression] [TOTAL [<fld{ ,fld}>]] }y_
value, x_value[, y0_const[, x0_const]])
```

LINEST_ssresid

**LINEST_SSRESID()** retourneert de geaggregeerde residu-som van het kwadraat van een lineaire regressie, gedefinieerd door de vergelijking y=mx+b voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x_value** en **y_value**, geïtereerd op de diagramdimensies.

```
LINEST_SSRESID - diagramfunctie({[SetExpression] [TOTAL [<fld{ ,fld}>]] }y_value, x_value[, y0_const[, x0_const]])
```

Median

**Median()** retourneert de mediaan van het bereik van waarden dat is geaggregeerd in de uitdrukking, geïtereerd op de diagramdimensies.

```
Median - diagramfunctie({[SetExpression] [TOTAL [<fld{, fld}>]]} expr)
```

MutualInfo

**MutualInfo** berekent de wederzijdse informatie tussen twee velden of tussen geaggregeerde waarden in **Aggr()**.

```
MutualInfo - diagramfunctie{[SetExpression] [DISTINCT] [TOTAL target, driver [, datatype [, breakdownbyvalue [, samplesize ]]])
```

Skew

**Skew()** retourneert het geaggregeerde scheefheid van de uitdrukking of het veld geïtereerd op de diagramdimensies.

```
Skew - diagramfunctie{[SetExpression] [DISTINCT] [TOTAL [<fld{ ,fld}>]]} expr)
```

Stdev

**Stdev()** retourneert de standaarddeviatie van het bereik van geaggregeerde waarden in de uitdrukking of het veld, geïtereerd op de diagramdimensies.

```
Stdev - diagramfunctie({[SetExpression] [DISTINCT] [TOTAL [<fld{, fld}>]]} expr)
```

Sterr

**Sterr()** retourneert de waarde van de standaardfout van het gemiddelde, (stdev/sqrt(n)), voor de serie geaggregeerde waarden in de uitdrukking, geïtereerd op de diagramdimensies.

```
Sterr - diagramfunctie({[SetExpression] [DISTINCT] [TOTAL[<fld{, fld}>]]} expr)
```

STEYX

**STEYX()** retourneert de geaggregeerde standaardfout bij het voorspellen van y-waarden voor elke x-waarde in een lineaire regressie die wordt gedefinieerd door een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **y_value** en **x_value**.

```
STEYX - diagramfunctie{[SetExpression] [TOTAL [<fld{, fld}>]]} y_value, x_value)
```

## Avg

**Avg()** retourneert de gemiddelde waarde van de geaggregeerde gegevens in de uitdrukking voor een aantal records, zoals gedefinieerd door een **group by**-clausule.

**Syntaxis:**

```
Avg([DISTINCT] expr)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| DISTINCT | Als het woord **distinct** voor de uitdrukking staat, worden alle duplicaten genegeerd. |

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

Resulterende gegevens

| Voorbeeld | Resultaat |
|---|---|
| `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 '\|');`<br><br>`Avg1:`<br><br>`LOAD Customer, Avg(Sales) as MyAverageSalesByCustomer`<br>`Resident Temp Group By Customer;` | `Customer MyAverageSalesByCustomer`<br><br>`Astrida 48.916667`<br><br>`Betacab 44.916667`<br><br>`Canutility 56.916667`<br><br>`Divadip 63.083333`<br>Dit kan worden gecontroleerd in het werkblad door een tabel te maken die de meting bevat.<br>`Sum(Sales)/12` |

| Voorbeeld | Resultaat |
|---|---|
| Stel dat de tabel **Temp** is geladen zoals in het vorige voorbeeld:<br>`LOAD Customer,Avg(DISTINCT Sales) as MyAvgSalesDistinct Resident Temp Group By Customer;` | `Customer MyAverageSalesByCustomer`<br><br>`Astrida 43.1`<br><br>`Betacab 43.909091`<br><br>`Canutility 55.909091`<br><br>`Divadip 61`<br>Alleen de distinctieve waarden worden geteld. Deel het totaal door het aantal niet-dubbele waarden. |

## Avg - diagramfunctie

**Avg()** retourneert het geaggregeerde gemiddelde van uitdrukking of veld geïtereerd op de diagramdimensies.

**Syntaxis:**

```
Avg([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] expr)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| SetExpression | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| DISTINCT | Als het woord **DISTINCT** voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd. |
| TOTAL | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br><br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

**Beperkingen:**

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

**Voorbeelden en resultaten:**

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 |

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `Avg(Sales)` | Voor een tabel met de dimensie `Customer` en de meting `Avg([Sales])`, is 2566 het resultaat als **Totalen** worden weergegeven. |
| `Avg([TOTAL (Sales))` | 53,458333 voor alle waarden van `Customer`, omdat de kwalificatie TOTAL betekent dat dimensies buiten beschouwing worden gelaten. |
| `Avg (DISTINCT (Sales))` | 51,862069 voor het totaal, omdat het gebruik van de kwalificatie Distinct betekent dat alleen unieke waarden in `Sales` worden geëvalueerd voor elke `Customer`. |

Gegevens die worden gebruikt in voorbeelden:

```
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 '|');
```

**Zie ook:**

p *Aggr - diagramfunctie (page 616)*

## Correl

**Correl()** retourneert de geaggregeerde correlatiecoëfficiënt voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals vastgelegd door een **group by**-clausule.

**Syntaxis:**

```
Correl(value1, value2)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value1, value2 | De uitdrukkingen of velden die de twee voorbeeldsets bevatten waarvoor de correlatiecoëfficiënt moet worden gemeten. |

**Beperkingen:**

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

Resulterende gegevens

| Voorbeeld | Resultaat |
|---|---|
| Salary:<br><br>Load *, 1 as Grp;<br><br>LOAD * inline [<br><br>"Employee name"\|Gender\|Age\|Salary<br><br>Aiden Charles\|Male\|20\|25000<br><br>Brenda Davies\|Male\|25\|32000<br><br>Charlotte Edberg\|Female\|45\|56000<br><br>Daroush Ferrara\|Male\|31\|29000<br><br>Eunice Goldblum\|Female\|31\|32000<br><br>Freddy Halvorsen\|Male\|25\|26000<br><br>Gauri Indu\|Female\|36\|46000<br><br>Harry Jones\|Male\|38\|40000<br><br>Ian Underwood\|Male\|40\|45000<br><br>Jackie Kingsley\|Female\|23\|28000<br><br>] (delimiter is '\|');<br><br>Correl1:<br>LOAD Grp,<br>Correl(Age,Salary) as Correl_Salary Resident Salary Group By Grp; | In een tabel met de dimensie `Correl_Salary`, wordt het resultaat van de berekening Correl() in het load-script voor gegevens weergegeven: 0,9270611 |

## Correl - diagramfunctie

**Correl()** retourneert de geaggregeerde correlatiecoëfficiënt voor twee gegevensverzamelingen. De correlatiefunctie is een maat voor de relatie tussen de gegevensverzamelingen en is geaggregeerd voor (x,y)-waardeparen, herhaald op de dimensies van het diagram.

**Syntaxis:**

```
Correl([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] value1, value2 )
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value1, value2 | De uitdrukkingen of velden die de twee voorbeeldsets bevatten waarvoor de correlatiecoëfficiënt moet worden gemeten. |
| SetExpression | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| DISTINCT | Als het woord **DISTINCT** voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd. |
| TOTAL | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br><br>Wanneer u **TOTAL [<fld {,fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

**Beperkingen:**

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

**Voorbeelden en resultaten:**

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `Correl(Age, Salary)` | Voor een tabel die de dimensie `Employee name` en de meting `Correl(Age, Salary)` bevat, is 0.9270611 het resultaat. Het resultaat wordt alleen weergegeven voor de cel met totalen. |

| Voorbeeld | Resultaat |
|---|---|
| `Correl (TOTAL Age, Salary))` | 0.927. Dit en de volgende resultaten worden weergegeven met drie cijfers achter het decimaalteken omwille van de leesbaarheid.<br><br>Als u een filtervak maakt met de dimensie Gender en van hieruit selecties uitvoert, ziet u het resultaat 0.951 als Female is geselecteerd en 0.939 als Male is geselecteerd. Dit komt doordat bij de selectie alle resultaten die niet tot de andere waarde van Gender behoren worden uitgesloten. |
| `Correl({1} TOTAL Age, Salary))` | 0.927. Onafhankelijk van selecties. Dit komt doordat in de set-uitdrukking {1} alle selecties en dimensies buiten beschouwing worden gelaten.{1} |
| `Correl (TOTAL <Gender> Age, Salary))` | 0,927 in de cel met totalen, 0,939 voor alle waarden van Male en 0,951 voor alle waarden van Female. Dit komt overeen met de resultaten van het uitvoeren van de selecties in een filtervak op basis van Gender. |

Gegevens die worden gebruikt in voorbeelden:

```
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 '|');
```

**Zie ook:**

p *Aggr - diagramfunctie (page 616)*

p *Avg - diagramfunctie (page 464)*
p *RangeCorrel (page 1605)*

## Fractile

**Fractile()** retourneert de waarde die overeenkomt met de inclusieve fractiel (kwantiel) van de geaggregeerde gegevens in de uitdrukking voor een aantal records, zoals gedefinieerd door een **group by**-clausule.

*U kunt FractileExc (page 474) gebruiken om de exclusieve fractiel te berekenen.*

**Syntaxis:**

```
Fractile(expr, fraction)
```

**Retourgegevenstypen:** numeriek

De functie retourneert de waarde die overeenkomt met de rangorde zoals gedefinieerd door `rank = fraction * (N-1) + 1` waarbij `N` het aantal waarden is in `expr`. Als `rank` een niet-geheel getal is, wordt een interpolatie uitgevoerd tussen de twee dichtstbijzijnde waarden.

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| expr | De uitdrukking of het veld met de te gebruiken gegevens bij het berekenen van de fractiel. |
| fraction | Een getal tussen 0 en 1 die correspondeert met de fractiel (kwantiel uitgedrukt als een fractie) die moet worden berekend. |

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

Resulterende gegevens

| Voorbeeld | Resultaat |
|---|---|
| Table1:<br>Crosstable (Type, Value)<br>Load recno() as ID, * inline [<br>Observation\|Comparison<br>35\|2<br>40\|27<br>12\|38<br>15\|31<br>21\|1<br>14\|19<br>46\|1<br>10\|34<br>28\|3<br>48\|1<br>16\|2<br>30\|3<br>32\|2<br>48\|1<br>31\|2<br>22\|1<br>12\|3<br>39\|29<br>19\|37<br>25\|2 ] (delimiter is '\|');<br><br>Fractile1:<br>LOAD Type,<br>Fractile(Value,0.75) as<br>MyFractile Resident Table1 Group<br>By Type; | In een tabel met de dimensies `Type` en `MyFractile` zijn de resultaten van de berekeningen Fractile() in het load-script voor gegevens:<br>`Type MyFractile`<br><br>`Comparison 27.5`<br><br>`Observation 36` |

## Fractile - diagramfunctie

**Fractile()** retourneert de waarde die correspondeert met de inclusieve fractiel (kwantiel) van de geaggregeerde gegevens in het bereik van de uitdrukking, geïtereerd op de diagramdimensies.

*U kunt FractileExc - diagramfunctie (page 475) gebruiken om de exclusieve fractiel te berekenen.*

**Syntaxis:**

```
Fractile([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] expr, fraction)
```

**Retourgegevenstypen:** numeriek

De functie retourneert de waarde die overeenkomt met de rangorde zoals gedefinieerd door `rank = fraction * (N-1) + 1` waarbij `N` het aantal waarden is in `expr`. Als `rank` een niet-geheel getal is, wordt een interpolatie uitgevoerd tussen de twee dichtstbijzijnde waarden.

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| expr | De uitdrukking of het veld met de te gebruiken gegevens bij het berekenen van de fractiel. |
| fraction | Een getal tussen 0 en 1 die correspondeert met de fractiel (kwantiel uitgedrukt als een fractie) die moet worden berekend. |
| SetExpression | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| DISTINCT | Als het woord **DISTINCT** voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd. |
| TOTAL | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br><br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

**Beperkingen:**

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

**Voorbeelden en resultaten:**

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 |

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `Fractile (Sales, 0.75)` | Voor een tabel met de dimensie `Customer` en de meting `Fractile([Sales])`, is 71.75 het resultaat als **Totalen** worden weergegeven. Dit is het punt in de verdeling van waarden van `Sales` waar 75% van de waarden onder vallen. |
| `Fractile (TOTAL Sales, 0.75))` | 71.75 voor alle waarden van `Customer`, omdat de kwalificatie TOTAL betekent dat dimensies buiten beschouwing worden gelaten. |
| `Fractile (DISTINCT Sales, 0.75)` | 70 voor het totaal, omdat het gebruik van de kwalificatie DISTINCT betekent dat alleen unieke waarden in `Sales` worden geëvalueerd voor elke `Customer`. |

Gegevens die worden gebruikt in voorbeelden:

```
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 '|');
```

**Zie ook:**

p *Aggr - diagramfunctie (page 616)*

## FractileExc

**FractileExc()** retourneert de waarde die overeenkomt met de exclusieve fractiel (kwantiel) van de geaggregeerde gegevens in de uitdrukking voor een aantal records, zoals gedefinieerd door een **group by**-clausule.

*U kunt Fractile (page 470) gebruiken om de inclusieve fractiel te berekenen.*

**Syntaxis:**

```
FractileExc(expr, fraction)
```

**Retourgegevenstypen:** numeriek

De functie retourneert de waarde die overeenkomt met de rangorde zoals gedefinieerd door `rank = fraction * (N+1)` waarbij `N` het aantal waarden is in `expr`. Als `rank` een niet-geheel getal is, wordt een interpolatie uitgevoerd tussen de twee dichtstbijzijnde waarden.

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| expr | De uitdrukking of het veld met de te gebruiken gegevens bij het berekenen van de fractiel. |
| fraction | Een getal tussen 0 en 1 die correspondeert met de fractiel (kwantiel uitgedrukt als een fractie) die moet worden berekend. |

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

Resulterende gegevens

| Voorbeeld | Resultaat |
|---|---|
| Table1:<br>Crosstable (Type, Value)<br>Load recno() as ID, * inline [<br>Observation\|Comparison<br>35\|2<br>40\|27<br>12\|38<br>15\|31<br>21\|1<br>14\|19<br>46\|1<br>10\|34<br>28\|3<br>48\|1<br>16\|2<br>30\|3<br>32\|2<br>48\|1<br>31\|2<br>22\|1<br>12\|3<br>39\|29<br>19\|37<br>25\|2 ] (delimiter is '\|');<br><br>Fractile1:<br>LOAD Type,<br>FractileExc(Value,0.75) as<br>MyFractile Resident Table1 Group<br>By Type; | In een tabel met de dimensies `Type` en `MyFractile` zijn de resultaten van de berekeningen FractileExc() in het load-script voor gegevens:<br><br>`Type MyFractile`<br><br>`Comparison 28.5`<br><br>`Observation 38` |

## FractileExc - diagramfunctie

**FractileExc()** retourneert de waarde die correspondeert met de exclusieve fractiel (kwantiel) van de geaggregeerde gegevens in het bereik van de uitdrukking, geïtereerd op de diagramdimensies.

*U kunt Fractile - diagramfunctie (page 471) gebruiken om de inclusieve fractiel te berekenen.*

**Syntaxis:**

```
FractileExc([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] expr, fraction)
```

**Retourgegevenstypen:** numeriek

De functie retourneert de waarde die overeenkomt met de rangorde zoals gedefinieerd door `rank = fraction * (N+1)` waarbij `N` het aantal waarden is in `expr`. Als `rank` een niet-geheel getal is, wordt een interpolatie uitgevoerd tussen de twee dichtstbijzijnde waarden.

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| expr | De uitdrukking of het veld met de te gebruiken gegevens bij het berekenen van de fractiel. |
| fraction | Een getal tussen 0 en 1 die correspondeert met de fractiel (kwantiel uitgedrukt als een fractie) die moet worden berekend. |
| SetExpression | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| DISTINCT | Als het woord **DISTINCT** voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd. |
| TOTAL | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br><br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

**Beperkingen:**

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

**Voorbeelden en resultaten:**

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 |

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `FractileExc (Sales, 0.75)` | Voor een tabel met de dimensie `Customer` en de meting `FractileExc([Sales])`, is 75.25 het resultaat als **Totalen** worden weergegeven. Dit is het punt in de verdeling van waarden van `Sales` waar 75% van de waarden onder vallen. |
| `FractileExc (TOTAL Sales, 0.75))` | 75,25 voor alle waarden van `Customer`, omdat de kwalificatie TOTAL betekent dat dimensies buiten beschouwing worden gelaten. |
| `FractileExc (DISTINCT Sales, 0.75)` | 73,50 voor het totaal, omdat het gebruik van de kwalificatie DISTINCT betekent dat alleen unieke waarden in `Sales` worden geëvalueerd voor elke `Customer`. |

Gegevens die worden gebruikt in voorbeelden:

```
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 '|');
```

**Zie ook:**

p *Aggr - diagramfunctie (page 616)*

## Kurtosis

**Kurtosis()** retourneert de kurtosis van de gegevens in de uitdrukking voor een aantal records, zoals gedefinieerd door een **group by**-clausule.

**Syntaxis:**

```
Kurtosis([distinct ] expr )
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| distinct | Als het woord **distinct** voor de uitdrukking staat, worden alle duplicaten genegeerd. |

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

Resulterende gegevens

| Voorbeeld | Resultaat |
|---|---|
| Table1:<br>Crosstable (Type, Value)<br>Load recno() as ID, * inline [<br>Observation\|Comparison<br>35\|2<br>40\|27<br>12\|38<br>15\|31<br>21\|1<br>14\|19<br>46\|1<br>10\|34<br>28\|3<br>48\|1<br>16\|2<br>30\|3<br>32\|2<br>48\|1<br>31\|2<br>22\|1<br>12\|3<br>39\|29<br>19\|37<br>25\|2 ] (delimiter is '\|');<br><br>Kurtosis1:<br>LOAD Type,<br>Kurtosis(Value) as MyKurtosis1,<br>Kurtosis(DISTINCT Value) as<br>MyKurtosis2<br>Resident Table1 Group By Type; | In een tabel met de dimensies Type, MyKurtosis1 en MyKurtosis2 zijn de resultaten van de berekeningen Kurtosis()in het load-script voor gegevens:<br><br>Type MyKurtosis1 MyKurtosis2<br><br>Comparison -1.1612957 -1.4982366<br><br>Observation -1.1148768 -0.93540144 |

## Kurtosis - diagramfunctie

**Kurtosis()** retourneert de piekvorm van het bereik van geaggregeerde waarden in de uitdrukking of het veld, geïtereerd op de diagramdimensies.

**Syntaxis:**

```
Kurtosis([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] expr)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |

| Argument | Beschrijving |
|---|---|
| SetExpression | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| DISTINCT | Als het woord **DISTINCT** voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd. |
| TOTAL | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br><br>Wanneer u **TOTAL [<fld {,fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

**Beperkingen:**

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

**Voorbeelden en resultaten:**

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 |

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `Kurtosis(Value)` | Voor een tabel die de dimensie `Type` en de meting `Kurtosis(Value)` bevat, is 1.252 het resultaat als **Totalen** worden weergegeven voor de tabel en de cijfernotatie is ingesteld op 3 significante posities. Voor `Comparison` is het 1.161 en voor `Observation` is het 1.115. |
| `Kurtosis(TOTAL Value))` | 1.252 voor alle waarden van `Type`, omdat de kwalificatie TOTAL betekent dat dimensies buiten beschouwing worden gelaten. |

Gegevens die worden gebruikt in voorbeelden:

```
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 '|');
```

**Zie ook:**

p *Avg - diagramfunctie (page 464)*

## LINEST_B

**LINEST_B()** retourneert geaggregeerde b-waarde (y-intercept) van een lineaire regressie, die wordt gedefinieerd door de vergelijking y=mx+b voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals vastgelegd door een **group by**-clausule.

**Syntaxis:**

```
LINEST_B (y_value, x_value[, y0 [, x0 ]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| y_value | De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten. |

| Argument | Beschrijving |
|---|---|
| x_value | De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten. |
| y(0), x(0) | Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.<br><br>Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende. |

**Beperkingen:**

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

**Zie ook:**

p *Voorbeelden van het gebruik van linest-functies (page 526)*

## LINEST_B - diagramfunctie

**LINEST_B()** retourneert de geaggregeerde b-waarde (y-intercept) van een lineaire regressie, gedefinieerd door de vergelijking y=mx+b voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x_value** en **y_value**, geïtereerd op de diagramdimensies.

**Syntaxis:**

```
LINEST_B([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value, x_value
[, y0_const [ , x0_const]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| y_value | De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten. |
| x_value | De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten. |

| Argument | Beschrijving |
|---|---|
| y0_const, x0_const | Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.<br><br>*Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende.* |
| SetExpression | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| DISTINCT | Als het woord **DISTINCT** voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd. |
| TOTAL | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br><br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

**Beperkingen:**

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

**Zie ook:**

p *Voorbeelden van het gebruik van linest-functies (page 526)*
p *Avg - diagramfunctie (page 464)*

## LINEST_DF

**LINEST_DF()** retourneert geaggregeerde vrijheidsgraden van een lineaire regressie, die wordt gedefinieerd door de vergelijking y=mx+b voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals vastgelegd door een **group by**-clausule.

**Syntaxis:**

```
LINEST_DF (y_value, x_value[, y0 [, x0 ]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| y_value | De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten. |
| x_value | De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten. |
| y(0), x(0) | Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.<br><br>Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende. |

**Beperkingen:**

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

**Zie ook:**

p *Voorbeelden van het gebruik van linest-functies (page 526)*

## LINEST_DF - diagramfunctie

**LINEST_DF()** retourneert de geaggregeerde vrijheidsgraden van een lineaire regressie, gedefinieerd door de vergelijking y=mx+b voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x_value** en **y_value**, geïtereerd op de diagramdimensies.

**Syntaxis:**

```
LINEST_DF([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value, x_
value [, y0_const [, x0_const]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| y_value | De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten. |
| x_value | De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten. |
| y0, x0 | Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.<br><br>*Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende.* |
| SetExpression | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| DISTINCT | Als het woord **DISTINCT** voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd. |
| TOTAL | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br><br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

**Beperkingen:**

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

**Zie ook:**

p *Voorbeelden van het gebruik van linest-functies (page 526)*
p *Avg - diagramfunctie (page 464)*

## LINEST_F

Deze scriptfunctie retourneert de geaggregeerde F-statistiek ($r^2/(1-r^2)$) van een lineaire regressie, die wordt gedefinieerd door de vergelijking y=mx+b voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression die worden herhaald over een aantal records zoals vastgelegd door een **group by**-clausule.

**Syntaxis:**

```
LINEST_F (y_value, x_value[, y0 [, x0 ]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| y_value | De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten. |
| x_value | De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten. |
| y(0), x(0) | Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.<br><br>Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende. |

**Beperkingen:**

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

**Zie ook:**

p *Voorbeelden van het gebruik van linest-functies (page 526)*

## LINEST_F - diagramfunctie

**LINEST_F()** retourneert de geaggregeerde F-statistiek (r2/(1-r2)) van een lineaire regressie, gedefinieerd door de vergelijking y=mx+b voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x_value** en **y_value**, geïtereerd op de diagramdimensies.

**Syntaxis:**

```
LINEST_F([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value, x_value
[, y0_const [, x0_const]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| y_value | De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten. |
| x_value | De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten. |
| y0, x0 | Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.<br>*Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende.* |
| SetExpression | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| DISTINCT | Als het woord **DISTINCT** voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd. |
| TOTAL | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

**Beperkingen:**

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

---

**Zie ook:**

p *Voorbeelden van het gebruik van linest-functies (page 526)*
p *Avg - diagramfunctie (page 464)*

## LINEST_M

**LINEST_M()** retourneert geaggregeerde m-waarde (helling) van een lineaire regressie, die wordt gedefinieerd door de vergelijking y=mx+b voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals vastgelegd door een **group by**-clausule.

**Syntaxis:**

```
LINEST_M (y_value, x_value[, y0 [, x0 ]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| y_value | De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten. |
| x_value | De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten. |
| y(0), x(0) | Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.<br><br>Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende. |

**Beperkingen:**

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

**Zie ook:**

p *Voorbeelden van het gebruik van linest-functies (page 526)*

## LINEST_M - diagramfunctie

**LINEST_M()** retourneert de geaggregeerde m-waarde (helling) van een lineaire regressie, gedefinieerd door de vergelijking y=mx+b voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x_value** en **y_value**, geïtereerd op de diagramdimensies.

**Syntaxis:**

```
LINEST_M([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value, x_value [, y0_const [, x0_const]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| y_value | De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten. |
| x_value | De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten. |
| y0, x0 | Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.<br>*Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende.* |
| SetExpression | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| DISTINCT | Als het woord **DISTINCT** voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd. |

| Argument | Beschrijving |
|---|---|
| TOTAL | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br><br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

**Beperkingen:**

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

**Zie ook:**

p *Voorbeelden van het gebruik van linest-functies (page 526)*
p *Avg - diagramfunctie (page 464)*

## LINEST_R2

retourneert de geaggregeerde -waarde (determinatiecoëfficiënt) van een lineaire regressie, die wordt gedefinieerd door de vergelijking voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in en , die worden herhaald over aantal records zoals vastgelegd door een -clausule.$r^2$y=mx+bx-expressiony-expressiongroup by

**Syntaxis:**

```
LINEST_R2 (y_value, x_value[, y0 [, x0 ]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| y_value | De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten. |
| x_value | De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten. |

| Argument | Beschrijving |
|---|---|
| y(0), x(0) | Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.<br><br>Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende. |

**Beperkingen:**

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

**Zie ook:**

p *Voorbeelden van het gebruik van linest-functies (page 526)*

## LINEST_R2 - diagramfunctie

**LINEST_R2()** retourneert de geaggregeerde r2-waarde (determinatiecoëfficient) van een lineaire regressie, gedefinieerd door de vergelijking y=mx+b voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x_value** en **y_value**, geïtereerd op de diagramdimensies.

**Syntaxis:**

```
LINEST_R2([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value, x_value[, y0_const[, x0_const]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| y_value | De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten. |
| x_value | De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten. |

| Argument | Beschrijving |
|---|---|
| y0, x0 | Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.<br><br>*Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende.* |
| SetExpression | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| DISTINCT | Als het woord **DISTINCT** voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd. |
| TOTAL | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br><br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

**Beperkingen:**

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

**Zie ook:**

p *Voorbeelden van het gebruik van linest-functies (page 526)*
p *Avg - diagramfunctie (page 464)*

## LINEST_SEB

**LINEST_SEB()** retourneert de geaggregeerde standaardfout van de b-waarde van een lineaire regressie, die wordt gedefinieerd door de vergelijking y=mx+b voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals vastgelegd door een **group by**-clausule.

**Syntaxis:**

```
LINEST_SEB (y_value, x_value[, y0 [, x0 ]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| y_value | De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten. |
| x_value | De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten. |
| y(0), x(0) | Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.<br><br>Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende. |

**Beperkingen:**

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

**Zie ook:**

p *Voorbeelden van het gebruik van linest-functies (page 526)*

## LINEST_SEB - diagramfunctie

**LINEST_SEB()** retourneert de geaggregeerde standaardfout van de b-waarde van een lineaire regressie, gedefinieerd door de vergelijking y=mx+b voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x_value** en **y_value**, geïtereerd op de diagramdimensies.

**Syntaxis:**

```
LINEST_SEB([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value, x_
value[, y0_const[, x0_const]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| y_value | De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten. |
| x_value | De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten. |
| y0, x0 | Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.<br><br>*Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende.* |
| SetExpression | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| DISTINCT | Als het woord **DISTINCT** voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd. |
| TOTAL | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br><br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

**Beperkingen:**

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

**Zie ook:**

p *Voorbeelden van het gebruik van linest-functies (page 526)*
p *Avg - diagramfunctie (page 464)*

## LINEST_SEM

**LINEST_SEM()** retourneert de geaggregeerde standaardfout van de m-waarde van een lineaire regressie, die wordt gedefinieerd door de vergelijking y=mx+b voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals vastgelegd door een **group by**-clausule.

**Syntaxis:**

```
LINEST_SEM (y_value, x_value[, y0 [, x0 ]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

| Argument | Beschrijving |
|---|---|
| y_value | De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten. |
| x_value | De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten. |
| y(0), x(0) | Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.<br><br>Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende. |

**Beperkingen:**

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

**Zie ook:**

p *Voorbeelden van het gebruik van linest-functies (page 526)*

## LINEST_SEM - diagramfunctie

**LINEST_SEM()** retourneert de geaggregeerde standaardfout van de m-waarde van een lineaire regressie, gedefinieerd door de vergelijking y=mx+b voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x_value** en **y_value**, geïtereerd op de diagramdimensies.

**Syntaxis:**

```
LINEST_SEM([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value, x_value[, y0_const[, x0_const]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| y_value | De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten. |
| x_value | De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten. |
| y0, x0 | Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.<br>*Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende.* |
| SetExpression | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| DISTINCT | Als het woord **DISTINCT** voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd. |
| TOTAL | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

**Beperkingen:**

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

**Zie ook:**

p *Voorbeelden van het gebruik van linest-functies (page 526)*
p *Avg - diagramfunctie (page 464)*

## LINEST_SEY

**LINEST_SEY()** retourneert de geaggregeerde standaardfout van de geschatte y-waarde van een lineaire regressie, die wordt gedefinieerd door de vergelijking y=mx+b voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals vastgelegd door een **group by**-clausule.

**Syntaxis:**

```
LINEST_SEY (y_value, x_value[, y0 [, x0 ]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

| Argument | Beschrijving |
|---|---|
| y_value | De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten. |
| x_value | De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten. |
| y(0), x(0) | Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.<br><br>Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende. |

**Beperkingen:**

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

**Zie ook:**

p *Voorbeelden van het gebruik van linest-functies (page 526)*

## LINEST_SEY - diagramfunctie

**LINEST_SEY()** retourneert de geaggregeerde standaardfout van de geschatte y-waarde van een lineaire regressie, gedefinieerd door de vergelijking y=mx+b voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x_value** en **y_value**, geïtereerd op de diagramdimensies.

**Syntaxis:**

```
LINEST_SEY([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value, x_value[, y0_const[, x0_const]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| y_value | De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten. |
| x_value | De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten. |
| y0, x0 | Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.<br>*Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende.* |
| SetExpression | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| DISTINCT | Als het woord **DISTINCT** voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd. |

| Argument | Beschrijving |
|---|---|
| TOTAL | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br><br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

**Beperkingen:**

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

**Zie ook:**

p *Voorbeelden van het gebruik van linest-functies (page 526)*
p *Avg - diagramfunctie (page 464)*

## LINEST_SSREG

retourneert de geaggregeerde regressie voor de som van de kwadraten van een lineaire regressie, die wordt gedefinieerd door de vergelijking voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in en , herhaald over een aantal records zoals vastgelegd door een -clausule.y=mx+bx-expressiony-expressiongroup by

**Syntaxis:**

```
LINEST_SSREG (y_value, x_value[, y0 [, x0 ]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| y_value | De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten. |
| x_value | De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten. |

| Argument | Beschrijving |
|---|---|
| y(0), x(0) | Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.<br><br>Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende. |

**Beperkingen:**

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

**Zie ook:**

p *Voorbeelden van het gebruik van linest-functies (page 526)*

## LINEST_SSREG - diagramfunctie

**LINEST_SSREG()** retourneert de geaggregeerde regressie voor de som van de kwadraten van een lineaire regressie die wordt gedefinieerd door de vergelijking y=mx+b voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x_value** en **y_value**, geïtereerd op de diagramdimensies.

**Syntaxis:**

```
LINEST_SSREG([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value, x_value[, y0_const[, x0_const]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| y_value | De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten. |
| x_value | De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten. |

| Argument | Beschrijving |
|---|---|
| y0, x0 | Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.<br><br>*Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende.* |
| SetExpression | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| DISTINCT | Als het woord **DISTINCT** voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd. |
| TOTAL | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br><br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

**Beperkingen:**

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

**Zie ook:**

p *Voorbeelden van het gebruik van linest-functies (page 526)*
p *Avg - diagramfunctie (page 464)*

## LINEST_SSRESID

**LINEST_SSRESID()** retourneert de geaggregeerde regressie voor de som van de kwadraten van een lineaire regressie, die wordt gedefinieerd door de vergelijking y=mx+b voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals vastgelegd door een **group by**-clausule.

**Syntaxis:**

```
LINEST_SSRESID (y_value, x_value[, y0 [, x0 ]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| y_value | De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten. |
| x_value | De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten. |
| y(0), x(0) | Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.<br><br>Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende. |

**Beperkingen:**

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

**Zie ook:**

p *Voorbeelden van het gebruik van linest-functies (page 526)*

## LINEST_SSRESID - diagramfunctie

**LINEST_SSRESID()** retourneert de geaggregeerde residu-som van het kwadraat van een lineaire regressie, gedefinieerd door de vergelijking y=mx+b voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x_value** en **y_value**, geïtereerd op de diagramdimensies.

**Syntaxis:**

```
LINEST_SSRESID([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value,
x_value[, y0_const[, x0_const]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| y_value | De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten. |
| x_value | De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten. |
| y0, x0 | Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.<br><br>*Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende.* |
| SetExpression | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| DISTINCT | Als het woord **DISTINCT** voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd. |
| TOTAL | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br><br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.

**Beperkingen:**

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

**Zie ook:**

p *Voorbeelden van het gebruik van linest-functies (page 526)*
p *Avg - diagramfunctie (page 464)*

## Median

**Median()** retourneert de geaggregeerde mediaan van de gegevens in de uitdrukking voor een aantal records, zoals gedefinieerd door een **group by**-clausule.

**Syntaxis:**

```
Median (expr)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |

### Voorbeeld: script met mediaan

Voorbeeld - scriptuitdrukking

**Load-script**

Laad de volgende inline-gegevens- en scriptuitdrukking in de editor voor het laden van gegevens voor dit voorbeeld.

```
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;
```

**Een visualisatie maken**

Maak een tabelvisualisatie in een Qlik Sense-werkblad met **Letter** en **MyMedian** als dimensies.

**Resultaat**

| Letter | MyMedian |
|---|---|
| A | 3.5 |
| B | 8 |

**Uitleg**

De mediaan wordt beschouwd als het "middelste" getal als de getallen gesorteerd zijn in volgorde van kleinste naar grootste. Als de gegevensverzameling een even aantal waarden heeft, retourneert de functie het gemiddelde van de twee middelste waarden. In dit voorbeeld wordt de mediaan berekend voor elke set van waarden van **A** en **B**, die respectievelijk 3.5 en 8 zijn.

## Median - diagramfunctie

**Median()** retourneert de mediaan van het bereik van waarden dat is geaggregeerd in de uitdrukking, geïtereerd op de diagramdimensies.

**Syntaxis:**

```
Median([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] expr)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| SetExpression | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| DISTINCT | Als het woord **DISTINCT** voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd. |
| TOTAL | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br><br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

**Beperkingen:**

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

## Voorbeeld: diagramuitdrukking met mediaan

Voorbeeld - diagramuitdrukking

### Load-script

Laad de volgende gegevens als een inline-load in de editor voor laden van gegevens om de onderstaande voorbeelden van diagramuitdrukkingen te maken.

```
Load RecNo() as RowNo, Letter, Number Inline
[Letter, Number
A,1
A,3
A,4
A,9
B,2
B,8
B,9];
```

### Een visualisatie maken

Maak een tabelvisualisatie in een Qlik Sense-werkblad met **Letter** als dimensie.

### Diagramuitdrukking

Voeg de volgende uitdrukking als meting toe aan de tabel:

```
Median(Number)
```

### Resultaat

| Letter | Median(Number) |
|---|---|
| Totals | 4 |
| A | 3.5 |
| B | 8 |

### Uitleg

De mediaan wordt beschouwd als het "middelste" getal als de getallen gesorteerd zijn in volgorde van kleinste naar grootste. Als de gegevensverzameling een even aantal waarden heeft, retourneert de functie het gemiddelde van de twee middelste waarden. In dit voorbeeld wordt de mediaan berekend voor elke set van waarden van **A** en **B**, die respectievelijk 3.5 en 8 zijn.

De mediaan voor **Totalen** wordt berekend vanaf alle waarden en is gelijk aan 4.

**Zie ook:**

p *Avg - diagramfunctie (page 464)*

## MutualInfo - diagramfunctie

**MutualInfo** berekent de wederzijdse informatie tussen twee velden of tussen geaggregeerde waarden in **Aggr()**.

**MutualInfo** retourneert de geaggregeerde wederzijde informatie voor twee gegevensverzamelingen. Hierdoor is een analyse van de sleutelfactor tussen een veld een mogelijke driver mogelijk. Wederzijde informatie is een maat voor de relatie tussen de gegevensverzamelingen en is geaggregeerd voor (x,y)-waardeparen, herhaald op de dimensies van het diagram. Wederzijdse informatie wordt gemeten tussen 0 en 1 en kan worden weergegeven als een percentielwaarde. **MutualInfo** wordt gedefinieerd door selecties of door een set-uitdrukking.

**MutualInfo** maakt verschillende soorten van MI-analyse mogelijk:

- Paarsgewijze MI: bereken de MI tussen een driverveld en een doelveld.
- Driveranalyse op waarde: de MI wordt berekend tussen individuele veldwaarden in de driver- en veldwaarden.
- Functieselectie: gebruik **MutualInfo** in een rasterdiagram zodat een matrix ontstaat waarin alle velden met elkaar worden vergeleken op basis van de MI.

**MutualInfo** geeft niet noodzakelijk een causaliteit aan tussen velden die wederzijde informatie bevatten. Twee velden kunnen wederzijdse informatie bevatten, maar zijn mogelijk geen gelijkwaardige drivers. Bij het vergelijken van de verkoop van ijs en de buitentemperatuur toont **MutualInfo** bijvoorbeeld dat er wederzijdse informatie aanwezig is. Het geeft niet aan of de buitentemperatuur invloed heeft op de verkoop van ijs, wat aannemelijk is, of dat de verkoop van ijs invloed heeft op de buitentemperatuur, wat onwaarschijnlijk is.

Bij het berekenen van wederzijde informatie hebben koppelingen invloed op de overeenkomst tussen de frequentie van waarden van velden die afkomstig zijn uit verschillende tabellen.

De geretourneerde waarden voor dezelfde velden of selecties kunnen licht van elkaar afwijken. Dit komt doordat de oproep van **MutualInfo** wordt uitgevoerd voor willekeurig geselecteerde voorbeeldgegevens en de inherente willekeur van het **MutualInfo**-algoritme.

**MutualInfo** kan worden toegepast op de functie **Aggr()**.

**Syntaxis:**

```
MutualInfo({SetExpression}] [DISTINCT] [TOTAL] field1, field2 , datatype [,
breakdownbyvalue [, samplesize ]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| field1, field2 | De uitdrukkingen of velden die de twee voorbeeldsets bevatten waarvoor de wederzijde informatie moet worden gemeten. |
| datatype | De gegevenstypen in het doel en de driver,<br>1 of 'dd' voor discreet:discreet<br>2 of 'cc' voor doorlopend:doorlopend<br>3 of 'cd' voor doorlopend:discreet<br>4 of 'dc' voor discreet:doorlopend<br>Gegevenstypen zijn niet hoofdlettergevoelig. |
| breakdownbyvalue | Een statische waarde die overeenkomt met de waarde in de driver. Indien opgegeven, wordt met de berekening de MI-bijdrage voor die waarde berekend. U kunt **ValueList()** of **ValueLoop()** gebruiken. Als **Null()** is toegevoegd, wordt met de berekening de totale MI berekend voor alle waarden in de driver.<br>Uitsplitsing naar waarde vereist dat de driver discrete gegevens bevat. |
| samplesize | Het aantal waarden van het doel en de driver in de steekproef. De steekproef vindt willekeurig plaats. **MutualInfo** vereist een minimale steekproefgrootte van 80. Standaard bemonstert **MutualInfo** tot 10.000 gegevensparen, aangezien **MutualInfo** zwaar en tijdrovend kan zijn. U kunt een groter aantal gegevensparen in de steekproef specificeren. Als er bij **MutualInfo** een time-out optreedt, maak dan de grootte van de steekproef kleiner. |
| SetExpression | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| DISTINCT | Als het woord **DISTINCT** voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd. |

| Argument | Beschrijving |
|---|---|
| TOTAL | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br><br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

**Beperkingen:**

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `mutualinfo (Age, Salary, 1)` | Voor een tabel die de dimensie `Employee name` en de meting `mutualinfo(Age, Salary, 1)` bevat, is 0.99820986 het resultaat. Het resultaat wordt alleen weergegeven voor de cel met totalen. |
| `mutualinfo (TOTAL Age, Salary, 1, null(), 81)` | Als u een filtervak maakt met de dimensie Gender en van hieruit selecties uitvoert, ziet u het resultaat 0.99805677 als Female is geselecteerd en 0.99847373 als Male is geselecteerd. Dit komt doordat bij de selectie alle resultaten die niet tot de andere waarde van Gender behoren worden uitgesloten. |
| `mutualinfo (TOTAL Age, Gender, 1, ValueLoop (25,35))` | 0.68196996. Als u een willekeurige waarde in Gender kiest, verandert deze in 0. |
| `mutualinfo ({1} TOTAL Age, Salary, 1, null())` | 0.99820986. Dit vindt onafhankelijk van selecties plaats. De set-uitdrukking {1} negeert alle selecties en dimensies. |

Gegevens die worden gebruikt in voorbeelden:

```
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 Baumel|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 Straubaum|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 Wesenlund|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()** retourneert de scheefheid van de uitdrukking voor een aantal records zoals gedefinieerd door een **group by**-clausule.

**Syntaxis:**

```
Skew([ distinct] expr)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| DISTINCT | Als het woord **distinct** voor de uitdrukking staat, worden alle duplicaten genegeerd. |

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Bouw vervolgens een strakke tabel met `Type` en *MySkew* als dimensies.

Resulterende gegevens

| Voorbeeld | Resultaat |
|---|---|
| Table1:<br>Crosstable (Type, Value)<br>Load recno() as ID, * inline [<br>Observation\|Comparison<br>35\|2<br>40\|27<br>12\|38<br>15\|31<br>21\|1<br>14\|19<br>46\|1<br>10\|34<br>28\|3<br>48\|1<br>16\|2<br>30\|3<br>32\|2<br>48\|1<br>31\|2<br>22\|1<br>12\|3<br>39\|29<br>19\|37<br>25\|2 ] (delimiter is '\|');<br><br>Skew1:<br>LOAD Type,<br>Skew(Value) as MySkew<br>Resident Table1 Group By Type; | De resultaten van de berekening Skew() zijn:<br>• Type is MySkew<br>• Comparison is 0.86414768<br>• Observation is 0.32625351 |

## Skew - diagramfunctie

**Skew()** retourneert het geaggregeerde scheefheid van de uitdrukking of het veld geïtereerd op de diagramdimensies.

**Syntaxis:**

```
Skew([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] expr)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |

| Argument | Beschrijving |
|---|---|
| SetExpression | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| DISTINCT | Als het woord **DISTINCT** voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd. |
| TOTAL | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br><br>Wanneer u **TOTAL [<fld {,fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

**Beperkingen:**

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Bouw vervolgens een strakke tabel met `Type` als dimensie en `Skew(Value)` als meting.

`Totals` moet zijn ingeschakeld in de eigenschappen van de tabel.

| Voorbeeld | Resultaat |
|---|---|
| Table1:<br>Crosstable (Type, Value)<br>Load recno() as ID, * inline [<br>Observation\|Comparison<br>35\|2<br>40\|27<br>12\|38<br>15\|31<br>21\|1<br>14\|19<br>46\|1<br>10\|34<br>28\|3<br>48\|1<br>16\|2<br>30\|3<br>32\|2<br>48\|1<br>31\|2<br>22\|1<br>12\|3<br>39\|29<br>19\|37<br>25\|2 ] (delimiter is '\|'); | De resultaten van de berekening Skew(Value) zijn:<br>• Totalis 0.23522195<br>• Comparison is 0.86414768<br>• Observation is 0.32625351 |

**Zie ook:**

p *Avg - diagramfunctie (page 464)*

## Stdev

**Stdev()** retourneert de standaarddeviatie van de waarden in de uitdrukking voor een aantal records, zoals gedefinieerd door een **group by**-clausule.

**Syntaxis:**

```
Stdev([distinct] expr)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| distinct | Als het woord **distinct** voor de uitdrukking staat, worden alle duplicaten genegeerd. |

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Bouw vervolgens een strakke tabel met `Type` en `MyStdev` als dimensies.

Resulterende gegevens

| Voorbeeld | Resultaat |
|---|---|
| `Table1:`<br>`Crosstable (Type, Value)`<br>`Load recno() as ID, * inline [`<br>`Observation\|Comparison`<br>`35\|2`<br>`40\|27`<br>`12\|38`<br>`15\|31`<br>`21\|1`<br>`14\|19`<br>`46\|1`<br>`10\|34`<br>`28\|3`<br>`48\|1`<br>`16\|2`<br>`30\|3`<br>`32\|2`<br>`48\|1`<br>`31\|2`<br>`22\|1`<br>`12\|3`<br>`39\|29`<br>`19\|37`<br>`25\|2 ] (delimiter is '\|');`<br><br>`Stdev1:`<br>`LOAD Type,`<br>`Stdev(Value) as MyStdev`<br>`Resident Table1 Group By Type;` | De resultaten van de berekening Stdev() zijn:<br>• `Type` is `MyStdev`<br>• `Comparison` is 14.61245<br>• `Observation` is 12.507997 |

## Stdev - diagramfunctie

**Stdev()** retourneert de standaarddeviatie van het bereik van geaggregeerde waarden in de uitdrukking of het veld, geïtereerd op de diagramdimensies.

**Syntaxis:**

```
Stdev([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] expr)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| SetExpression | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| DISTINCT | Als het woord **DISTINCT** voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd. |
| TOTAL | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br><br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

**Beperkingen:**

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Bouw vervolgens een strakke tabel met `Type` als dimensie en `Stdev(Value)` als meting.

`Totals` moet zijn ingeschakeld in de eigenschappen van de tabel.

| Voorbeeld | Resultaat |
|---|---|
| `Stdev(Value)`<br>`Table1:`<br>`Crosstable (Type, Value)`<br>`Load recno() as ID, * inline [`<br>`Observation\|Comparison`<br>`35\|2`<br>`40\|27`<br>`12\|38`<br>`15\|31`<br>`21\|1`<br>`14\|19`<br>`46\|1`<br>`10\|34`<br>`28\|3`<br>`48\|1`<br>`16\|2`<br>`30\|3`<br>`32\|2`<br>`48\|1`<br>`31\|2`<br>`22\|1`<br>`12\|3`<br>`39\|29`<br>`19\|37`<br>`25\|2 ] (delimiter is '\|');` | De resultaten van de berekening Stdev(Value) zijn:<br>• `Total` is 15.47529<br>• `Comparison` is 14.61245<br>• `Observation` is 12.507997 |

**Zie ook:**

p *Avg - diagramfunctie (page 464)*
p *STEYX - diagramfunctie (page 524)*

## Sterr

**Sterr()** retourneert de geaggregeerde standaardfout (stdev/sqrt(n)) voor een reeks waarden die wordt vertegenwoordigd door de uitdrukking, herhaald over een aantal records zoals gedefinieerd door een **group by**-clausule.

**Syntaxis:**

```
Sterr ([distinct] expr)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| distinct | Als het woord **distinct** voor de uitdrukking staat, worden alle duplicaten genegeerd. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden worden genegeerd.

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

Resulterende gegevens

| Voorbeeld | Resultaat |
|---|---|
| `Table1:`<br>`Crosstable (Type, Value)`<br>`Load recno() as ID, * inline [`<br>`Observation\|Comparison`<br>`35\|2`<br>`40\|27`<br>`12\|38`<br>`15\|31`<br>`21\|1`<br>`14\|19`<br>`46\|1`<br>`10\|34`<br>`28\|3`<br>`48\|1`<br>`16\|2`<br>`30\|3`<br>`32\|2`<br>`48\|1`<br>`31\|2`<br>`22\|1`<br>`12\|3`<br>`39\|29`<br>`19\|37`<br>`25\|2 ] (delimiter is '\|');`<br><br>`Sterr1:`<br>`LOAD Type,`<br>`Sterr(Value) as MySterr`<br>`Resident Table1 Group By Type;` | In een tabel met de dimensies `Type` en `MySterr` zijn de resultaten van de berekening Sterr() in het load-script voor gegevens:<br><br>`Type MySterr`<br><br>`Comparison 3.2674431`<br><br>`Observation 2.7968733` |

## Sterr - diagramfunctie

**Sterr()** retourneert de waarde van de standaardfout van het gemiddelde, (stdev/sqrt(n)), voor de serie geaggregeerde waarden in de uitdrukking, geïtereerd op de diagramdimensies.

**Syntaxis:**

```
Sterr([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] expr)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| SetExpression | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| DISTINCT | Als het woord **DISTINCT** voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd. |
| TOTAL | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br><br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

**Beperkingen:**

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Tekstwaarden, NULL-waarden en ontbrekende waarden worden genegeerd.

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Bouw vervolgens een strakke tabel met `Type` als dimensie en `Sterr(Value)` als meting.

`Totals` moet zijn ingeschakeld in de eigenschappen van de tabel.

| Voorbeeld | Resultaat |
|---|---|
| `Table1:`<br>`Crosstable (Type, Value)`<br>`Load recno() as ID, * inline [`<br>`Observation\|Comparison`<br>`35\|2`<br>`40\|27`<br>`12\|38`<br>`15\|31`<br>`21\|1`<br>`14\|19`<br>`46\|1`<br>`10\|34`<br>`28\|3`<br>`48\|1`<br>`16\|2`<br>`30\|3`<br>`32\|2`<br>`48\|1`<br>`31\|2`<br>`22\|1`<br>`12\|3`<br>`39\|29`<br>`19\|37`<br>`25\|2 ] (delimiter is '\|');` | De resultaten van de berekening Sterr(Value) zijn:<br>• `Total`is 2.4468583<br>• `Comparison` is 3.2674431<br>• `Observation` is 2.7968733 |

**Zie ook:**

p *Avg - diagramfunctie (page 464)*
p *STEYX - diagramfunctie (page 524)*

## STEYX

**STEYX()** retourneert de geaggregeerde standaardfout van de voorspelde y-waarde voor elke x-waarde in de regressie voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals gedefinieerd door een **group by**-clausule.

**Syntaxis:**

```
STEYX (y_value, x_value)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| y_value | De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten. |

| Argument | Beschrijving |
|---|---|
| x_value | De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten. |

**Beperkingen:**

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

Resulterende gegevens

| Voorbeeld | Resultaat |
| --- | --- |
| Trend:<br><br>Load *, 1 as Grp;<br><br>LOAD * inline [<br><br>Month\|KnownY\|KnownX<br><br>Jan\|2\|6<br><br>Feb\|3\|5<br><br>Mar\|9\|11<br><br>Apr\|6\|7<br><br>May\|8\|5<br><br>Jun\|7\|4<br><br>Jul\|5\|5<br><br>Aug\|10\|8<br><br>Sep\|9\|10<br><br>Oct\|12\|14<br><br>Nov\|15\|17<br><br>Dec\|14\|16<br><br>] (delimiter is '\|');<br><br>STEYX1:<br>LOAD Grp,<br>STEYX(KnownY, KnownX)<br>as MySTEYX<br>Resident Trend Group<br>By Grp; | In een tabel met de dimensie `MySTEYX` is het resultaat van de berekening STEYX() in het load-script voor gegevens 2.0714764. |

## STEYX - diagramfunctie

**STEYX()** retourneert de geaggregeerde standaardfout bij het voorspellen van y-waarden voor elke x-waarde in een lineaire regressie die wordt gedefinieerd door een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **y_value** en **x_value**.

**Syntaxis:**

```
STEYX([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value, x_value)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| y_value | De uitdrukking die of het veld dat de reeks van bekende y-waarden bevat die moet worden gemeten. |
| x_value | De uitdrukking die of het veld dat de reeks van bekende x-waarden bevat die moet worden gemeten. |
| SetExpression | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| DISTINCT | Als het woord **DISTINCT** voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd. |
| TOTAL | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br><br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

**Beperkingen:**

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Bouw vervolgens een strakke tabel met `KnownY` en `KnownX` als dimensies en `Steyx(KnownY,KnownX)` als meting.

`Totals` moet zijn ingeschakeld in de eigenschappen van de tabel.

| Voorbeeld | Resultaat |
| --- | --- |
| `Trend:`<br><br>`LOAD * inline [`<br><br>`Month\|KnownY\|KnownX`<br><br>`Jan\|2\|6`<br><br>`Feb\|3\|5`<br><br>`Mar\|9\|11`<br><br>`Apr\|6\|7`<br><br>`May\|8\|5`<br><br>`Jun\|7\|4`<br><br>`Jul\|5\|5`<br><br>`Aug\|10\|8`<br><br>`Sep\|9\|10`<br><br>`Oct\|12\|14`<br><br>`Nov\|15\|17`<br><br>`Dec\|14\|16`<br><br>`] (delimiter is '\|');` | Het resultaat van de berekening STEYX(KnownY,KnownX) is 2.071 (indien cijfernotatie is ingesteld op 3 cijfers achter het decimaalteken). |

**Zie ook:**

p *Avg - diagramfunctie (page 464)*
p *Sterr - diagramfunctie (page 520)*

## Voorbeelden van het gebruik van linest-functies

De linest-functies worden gebruikt om te zoeken naar waarden die verband houden met lineaire regressieanalyse. In dit gedeelte wordt beschreven hoe visualisaties kunnen worden gebouwd aan de hand van de voorbeeldgegevens, om de waarden te vinden van de linest-functies die beschikbaar zijn in Qlik Sense. De functies linest kunnen zowel worden gebruikt in het script voor het laden van gegevens als in diagramuitdrukkingen.

Raadpleeg de individuele onderwerpen over de linest-diagram- en scriptfunctie voor een beschrijving van syntaxis en argumenten.

### Gegevens- en scriptuitdrukkingen die in de voorbeelden worden gebruikt

Laad de volgende inline-gegevens- en scriptuitdrukkingen in de editor voor het laden van gegevens voor de onderstaande linest()-voorbeelden.

```
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;
```

### Voorbeeld 1: scriptuitdrukkingen met linest

Voorbeeld: scriptuitdrukkingen

**Een visualisatie maken vanaf berekeningen van het script voor het laden van gegevens**

Maak een tabelvisualisatie in een Qlik Sense-werkblad met de volgende velden als kolommen:

- Linest_B

- Linest_DF
- Linest_F
- Linest_M
- Linest_R2
- Linest_SEB
- Linest_SEM
- Linest_SEY
- Linest_SSREG
- Linest_SSRESID

**Resultaat**

De tabel die de resultaten van de linest-berekeningen bevat die zijn uitgevoerd in het script voor het laden van gegevens zouden er als volgt moeten uitzien:

Resultatentabel

| Linest_B | Linest_DF | Linest_F | Linest_M | Linest_R2 | Linest_SEB |
|---|---|---|---|---|---|
| -35.047 | 18 | 20.788 | 8.605 | 0.536 | 22.607 |

Resultatentabel

| Linest_SEM | Linest_SEY | Linest_SSREG | Linest_SSRESID |
|---|---|---|---|
| 1.887 | 48.666 | 49235.014 | 42631.186 |

## Voorbeeld 2: diagramuitdrukkingen met linest

**Voorbeeld: diagramuitdrukkingen**

Maak een tabelvisualisatie in een Qlik Sense-werkblad met de volgende velden als dimensies:

```
ValueList('Linest_b', 'Linest_df','Linest_f', 'Linest_m','Linest_r2','Linest_SEB','Linest_
SEM','Linest_SEY','Linest_SSREG','Linest_SSRESID')
```

Deze uitdrukking gebruikt de functie voor synthetische dimensies om labels voor de dimensies te maken met de namen van de linest-functies. U kunt het label wijzigen in **Linest functions** om ruimte te besparen.

Voeg de volgende uitdrukking als meting toe aan de tabel:

```
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) )
```

Deze uitdrukking geeft de waarde van het resultaat van iedere linest-functie weer naast de overeenkomstige naam in de synthetische dimensie. Het resultaat van`Linest_b(Y,X)` wordt weergegeven naast **linest_b**, enzovoort.

**Resultaat**

Resultatentabel

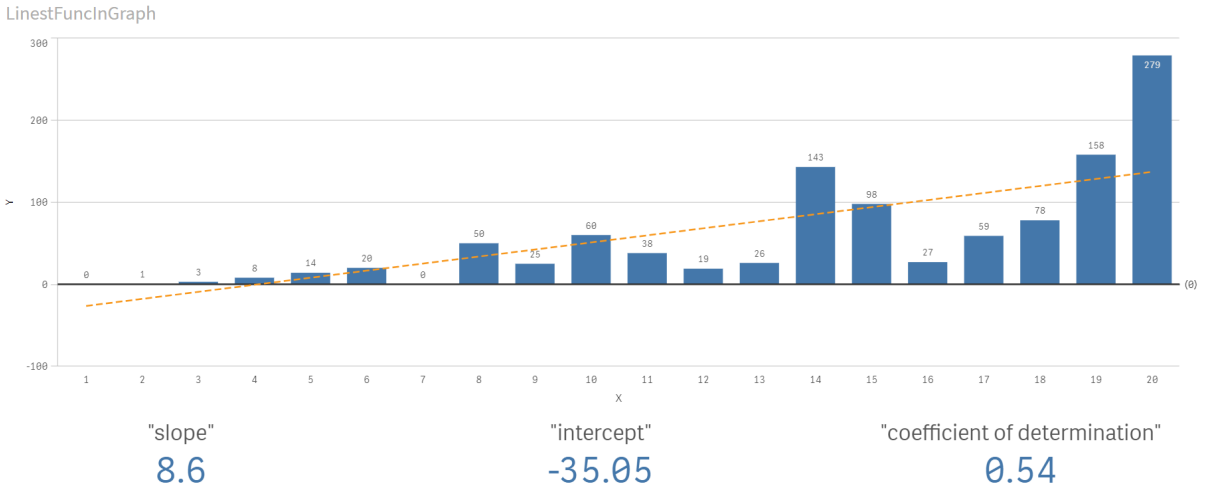
| Linest functions | Linest function results |
|---|---|
| Linest_b | -35.047 |
| Linest_df | 18 |
| Linest_f | 20.788 |
| Linest_m | 8.605 |
| Linest_r2 | 0.536 |
| Linest_SEB | 22.607 |
| Linest_SEM | 1.887 |
| Linest_SEY | 48.666 |
| Linest_SSREG | 49235.014 |
| Linest_SSRESID | 42631.186 |

### Voorbeeld 3: diagramuitdrukkingen met linest

Voorbeeld: diagramuitdrukkingen

1. Maak een visualisatie van een staafdiagram in een Qlik Sense-werkblad met **X** als een dimensie en **Y** als een meting.
2. Voeg een lineaire trendlijn toe aan de Y-meting.
3. Voeg een KPI-visualisatie toe aan het werkblad.
    1. Voeg *helling* toe als een label voor de KPI.
    2. Voeg `sum(Linest_M)` toe als een uitdrukking voor de KPI.
4. Voeg een tweede KPI-visualisatie toe aan het werkblad.
    1. Voeg *snijpunt* toe als een label voor de KPI.
    2. Voeg `Sum(Linest_B)` toe als een uitdrukking voor de KPI.
5. Voeg een derde KPI-visualisatie toe aan het werkblad.
    1. Voeg *Determinatiecoëfficiënt* toe als een label voor de KPI.
    2. Voeg `Sum(Linest_R2)` toe als een uitdrukking voor de KPI.

### Resultaat




| "slope" | "intercept" | "coefficient of determination" |
|---|---|---|
| 8.6 | -35.05 | 0.54 |

### Uitleg

De staafdiagram geeft het plotten van de X- en Y-gegevens weer. Relevante linest()-functies bieden waarden voor de vergelijking van de lineaire regressie waarop de trendlijn is gebaseerd, namelijk y = m * x + b. De vergelijking gebruikt de methode van de "minste vierkanten" om een rechte lijn (trendlijn) te berekenen door een matrix te retourneren die een lijn beschrijft die het beste overeenkomt met de gegevens.

De KPI's geven de resultaten weer van de linest()-functies **sum(Linest_M)** voor helling en **sum(Linest_B)** voor het Y-snijpunt, die variabelen zijn in de vergelijking van de lineaire regressie en de overeenkomstige geaggregeerde R2-waarde voor de determinatiecoëfficiënt.

## Statistische testfuncties

Statistische testfuncties kunnen zowel worden gebruikt in het load-script voor gegevens als in diagramuitdrukkingen, maar de syntaxis verschilt.

### Chi-2 testfuncties

Worden veel gebruikt in de studie van kwalitatieve variabelen. U kunt geobserveerde frequenties vergelijken in een eenzijdige frequentietabel met verwachte frequenties, of de verbinding tussen twee variabelen bestuderen in een contingentietabel.

### T-testfuncties

T-testfuncties worden gebruikt voor statistisch onderzoek van twee steekproefgemiddelden. Een t-test met twee steekproeven waarbij wordt onderzocht of twee steekproeven verschillend zijn, wordt veel gebruikt als twee normale verdelingen onbekende afwijkingen hebben en de steekproef voor het experiment klein is.

## Z-testfuncties

Een statistisch onderzoek van twee steekproefgemiddelden. Een z-test met twee steekproeven onderzoekt of twee steekproeven van elkaar verschillen. Dit wordt over het algemeen gebruikt als twee normale verdelingen onbekende varianties hebben en als een experiment een grote steekproefgrootte heeft.

## Chi2-testfuncties

Worden veel gebruikt in de studie van kwalitatieve variabelen. U kunt geobserveerde frequenties vergelijken in een eenzijdige frequentietabel met verwachte frequenties, of de verbinding tussen twee variabelen bestuderen in een contingentietabel.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.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

Chi2Test_chi2
retourneert de geaggregeerde chi-test voor een of twee series waarden.Chi2Test_chi2()

```
Chi2Test_chi2(col, row, actual_value[, expected_value])
```

Chi2Test_df
**Chi2Test_df()** retourneert de geaggregeerde df-waarde (vrijheidsgraden) van de $chi^2$-test voor een of twee series waarden.

```
Chi2Test_df(col, row, actual_value[, expected_value])
```

Chi2Test_p
**Chi2Test_p()** retourneert de geaggregeerde p-waarde (significantie) van de $chi^2$-test voor een of twee series waarden.

```
Chi2Test_p - diagramfunctie(col, row, actual_value[, expected_value])
```

**Zie ook:**

p *T-testfuncties (page 534)*
p *Z-testfuncties (page 571)*

## Chi2Test_chi2

retourneert de geaggregeerde chi-test voor een of twee series waarden.Chi2Test_chi2()

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

*Alle Qlik Sense chi$^2$-testfuncties hebben dezelfde argumenten.*

**Syntaxis:**

```
Chi2Test_chi2(col, row, actual_value[, expected_value])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| col, row | De opgegeven kolom en rij in de matrix met waarden die worden getest. |
| actual_value | De geobserveerde waarde van de gegevens in de opgegeven **col** en **row**. |
| expected_value | De verwachte waarde voor de verdeling in de opgegeven **col** en **row**. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeelden:**

```
Chi2Test_chi2( Grp, Grade, Count )
```

```
Chi2Test_chi2( Gender, Description, Observed, Expected )
```

**Zie ook:**

p *Voorbeelden van het gebruik van chi2-test-functies in grafieken (page 588)*
p *Voorbeelden van het gebruik van chi2-test-functies in het load-script voor gegevens (page 592)*

## Chi2Test_df

**Chi2Test_df()** retourneert de geaggregeerde df-waarde (vrijheidsgraden) van de chi$^2$-test voor een of twee series waarden.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

*Alle Qlik Sense chi$^2$-testfuncties hebben dezelfde argumenten.*

**Syntaxis:**

```
Chi2Test_df(col, row, actual_value[, expected_value])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| col, row | De opgegeven kolom en rij in de matrix met waarden die worden getest. |
| actual_value | De geobserveerde waarde van de gegevens in de opgegeven **col** en **row**. |
| expected_value | De verwachte waarde voor de verdeling in de opgegeven **col** en **row**. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeelden:**

```
Chi2Test_df( Grp, Grade, Count )
```

```
Chi2Test_df( Gender, Description, Observed, Expected )
```

**Zie ook:**

p *Voorbeelden van het gebruik van chi2-test-functies in grafieken (page 588)*
p *Voorbeelden van het gebruik van chi2-test-functies in het load-script voor gegevens (page 592)*

## Chi2Test_p - diagramfunctie

De test kan worden uitgevoerd op de waarden in **Chi2Test_p()** retourneert de geaggregeerde p-waarde (significantie) van de chi$^2$-test voor een of twee series waarden., waarbij wordt getest op variaties binnen de opgegeven - en -matrix, of de waarden in worden vergeleken met de corresponderende waarden in , indien deze zijn opgegeven.actual_ valuecolrowactual_valueexpected_value

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

*Alle Qlik Sense chi$^2$-testfuncties hebben dezelfde argumenten.*

**Syntaxis:**

```
Chi2Test_p(col, row, actual_value[, expected_value])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| col, row | De opgegeven kolom en rij in de matrix met waarden die worden getest. |
| actual_value | De geobserveerde waarde van de gegevens in de opgegeven **col** en **row**. |
| expected_value | De verwachte waarde voor de verdeling in de opgegeven **col** en **row**. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeelden:**

```
Chi2Test_p( Grp, Grade, Count )
Chi2Test_p( Gender, Description, Observed, Expected )
```

**Zie ook:**

p *Voorbeelden van het gebruik van chi2-test-functies in grafieken (page 588)*
p *Voorbeelden van het gebruik van chi2-test-functies in het load-script voor gegevens (page 592)*

## T-testfuncties

T-testfuncties worden gebruikt voor statistisch onderzoek van twee steekproefgemiddelden. Een t-test met twee steekproeven waarbij wordt onderzocht of twee steekproeven verschillend zijn, wordt veel gebruikt als twee normale verdelingen onbekende afwijkingen hebben en de steekproef voor het experiment klein is.

In de volgende gedeelten worden de statistische t-testfuncties gegroepeerd op basis van de steekproeftest voor cursisten die geldt voor elk type functie.

*Een typisch t-test-rapport opstellen (page 593)*

### T-testen voor twee onafhankelijke steekproeven

De volgende functies zijn van toepassing op t-toetsen voor cursisten voor twee onafhankelijke steekproeven.

ttest_conf

**TTest_conf** retourneert het geaggregeerde vertrouwensinterval voor de t-testwaarde voor twee onafhankelijke steekproeven.

```
TTest_conf retourneert het geaggregeerde vertrouwensinterval voor de t-testwaarde voor twee onafhankelijke steekproeven. ( grp, value [, sig[, eq_var]])
```

ttest_df

**TTest_df()** retourneert de geaggregeerde waarde (vrijheidsgraden) van de t-toets van Student voor twee onafhankelijke series waarden.

```
TTest_df() retourneert de geaggregeerde waarde (vrijheidsgraden) van de t-toets van Student voor twee onafhankelijke series waarden. (grp, value [, eq_var)
```

ttest_dif

**TTest_dif()** is een numerieke functie die het geaggregeerde gemiddelde verschil van Student t-test voor twee onafhankelijke series waarden retourneert.

```
TTest_dif() is een numerieke functie die het geaggregeerde gemiddelde verschil van Student t-test voor twee onafhankelijke series waarden retourneert. (grp, value)
```

ttest_lower

**TTest_lower()** retourneert de geaggregeerde ondergrenswaarde van het·vertrouwensinterval voor twee onafhankelijke series waarden.

```
TTest_lower() retourneert de geaggregeerde ondergrenswaarde van het·vertrouwensinterval voor twee onafhankelijke series waarden. (grp, value [, sig[, eq_var]])
```

ttest_sig

**TTest_sig()** retourneert het geaggregeerde tweezijdige niveau van significantie van Students t-toets op voor twee onafhankelijke series waarden.

```
TTest_sig() retourneert het geaggregeerde tweezijdige niveau van significantie van Students t-toets op voor twee onafhankelijke series waarden. (grp, value [, eq_var])
```

ttest_sterr

**TTest_sterr()** retourneert de geaggregeerde standaardfout van de t-toets van Student voor twee onafhankelijke series waarden.

```
TTest_sterr() retourneert de geaggregeerde standaardfout van de t-toets van Student voor twee onafhankelijke series waarden. (grp, value [, eq_var])
```

ttest_t

**TTest_t()** retourneert de geaggregeerde t-waarde voor twee onafhankelijke waarden.

```
TTest_t() retourneert de geaggregeerde t-waarde voor twee onafhankelijke
waarden. (grp, value [, eq_var])
```

ttest_upper

**TTest_upper()** retourneert de geaggregeerde bovengrenswaarde van het·vertrouwensinterval voor twee onafhankelijke series waarden.

```
TTest_upper() retourneert de geaggregeerde bovengrenswaarde van
het·vertrouwensinterval voor twee onafhankelijke series waarden. (grp, value
[, sig [, eq_var]])
```

### T-testen voor twee onafhankelijke gewogen steekproeven

De volgende functies zijn van toepassing op t-testen voor cursisten voor twee onafhankelijke steekproeven waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

ttestw_conf

**TTestw_conf()** retourneert de geaggregeerde t-waarde voor twee onafhankelijke waarden.

```
TTestw_conf() retourneert de geaggregeerde t-waarde voor twee onafhankelijke
waarden. (weight, grp, value [, sig[, eq_var]])
```

ttestw_df

**TTestw_df()** retourneert de geaggregeerde df-waarde (vrijheidsgraden) van de t-toets van Student van twee onafhankelijke series waarden.

```
TTestw_df() retourneert de geaggregeerde df-waarde (vrijheidsgraden) van de
t-toets van Student van twee onafhankelijke series waarden. (weight, grp,
value [, eq_var])
```

ttestw_dif

**TTestw_dif()** retourneert het geaggregeerde gemiddelde verschil van de t-toets van Student van twee onafhankelijke series waarden.

```
TTestw_dif() retourneert het geaggregeerde gemiddelde verschil van de t-toets
van Student van twee onafhankelijke series waarden. ( weight, grp, value)
```

ttestw_lower

**TTestw_lower()** retourneert de geaggregeerde ondergrenswaarde van het·vertrouwensinterval voor twee onafhankelijke series waarden.

```
TTestw_lower() retourneert de geaggregeerde ondergrenswaarde van
het·vertrouwensinterval voor twee onafhankelijke series waarden. (weight,
grp, value [, sig[, eq_var]])
```

ttestw_sig

**TTestw_sig()** retourneert het geaggregeerde tweezijdige niveau van significantie van Students t-toets op voor twee onafhankelijke series waarden.

**TTestw_sig() retourneert het geaggregeerde tweezijdige niveau van significantie van Students t-toets op voor twee onafhankelijke series waarden.** ( weight, grp, value [, eq_var])

ttestw_sterr

**TTestw_sterr()** retourneert de geaggregeerde standaardfout van de t-toets van Student voor twee onafhankelijke series waarden.

**TTestw_sterr() retourneert de geaggregeerde standaardfout van de t-toets van Student voor twee onafhankelijke series waarden.** (weight, grp, value [, eq_var])

ttestw_t

**TTestw_t()** retourneert de geaggregeerde t-waarde voor twee onafhankelijke waarden.

**TTestw_t() retourneert de geaggregeerde t-waarde voor twee onafhankelijke waarden.** (weight, grp, value [, eq_var])

ttestw_upper

**TTestw_upper()** retourneert de geaggregeerde bovengrenswaarde van het·vertrouwensinterval voor twee onafhankelijke series waarden.

**TTestw_upper() retourneert de geaggregeerde bovengrenswaarde van het·vertrouwensinterval voor twee onafhankelijke series waarden.** (weight, grp, value [, sig [, eq_var]])

### T-testen voor één steekproef

De volgende functies zijn van toepassing op t-testen voor cursisten voor één steekproef.

ttest1_conf

**TTest1_conf()** retourneert de geaggregeerde waarde van het vertrouwensinterval voor een serie waarden.

**TTest1_conf() retourneert de geaggregeerde waarde van het vertrouwensinterval voor een serie waarden.** (value [, sig])

ttest1_df

**TTest1_df()** retourneert de geaggregeerde df-waarde (vrijheidsgraden) van de t-toets van Student van een serie waarden.

**TTest1_df() retourneert de geaggregeerde df-waarde (vrijheidsgraden) van de t-toets van Student van een serie waarden.** (value)

ttest1_dif

**TTest1_dif()** retourneert het geaggregeerde gemiddelde verschil van de t-toets van Student van een serie waarden.

**TTest1_dif() retourneert het geaggregeerde gemiddelde verschil van de t-toets van Student van een serie waarden.** (value)

ttest1_lower

**TTest1_lower()** retourneert de geaggregeerde ondergrenswaarde van het·vertrouwensinterval voor een serie waarden.

```
TTest1_lower() retourneert de geaggregeerde ondergrenswaarde van
het·vertrouwensinterval voor een serie waarden. (value [, sig])
```

ttest1_sig

**TTest1_sig()** retourneert het geaggregeerde tweezijdige niveau van significantie van Students t-toets op voor een serie waarden.

```
TTest1_sig() retourneert het geaggregeerde tweezijdige niveau van
significantie van Students t-toets op voor een serie waarden. (value)
```

ttest1_sterr

**TTest1_sterr()** retourneert de geaggregeerde standaardfout van de t-toets van Student voor een serie waarden.

```
TTest1_sterr() retourneert de geaggregeerde standaardfout van de t-toets van
Student voor een serie waarden. (value)
```

ttest1_t

**TTest1_t()** retourneert de geaggregeerde t-waarde voor een serie waarden.

```
TTest1_t() retourneert de geaggregeerde t-waarde voor een serie waarden.
(value)
```

ttest1_upper

**TTest1_upper()** retourneert de geaggregeerde bovengrenswaarde van het·vertrouwensinterval voor een serie waarden.

```
TTest1_upper() retourneert de geaggregeerde bovengrenswaarde van
het·vertrouwensinterval voor een serie waarden. (value [, sig])
```

### T-testen voor één gewogen steekproef

De volgende functies zijn van toepassing op t-testen voor cursisten voor één steekproef waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

ttest1w_conf

**TTest1w_conf()** is een **numerieke** functie die de geaggregeerde waarde van het vertrouwensinterval voor een serie waarden retourneert.

```
TTest1w_conf() is een numerieke functie die de geaggregeerde waarde van het
vertrouwensinterval voor een serie waarden retourneert. (weight, value [,
sig])
```

ttest1w_df

**TTest1w_df()** retourneert de geaggregeerde df-waarde (vrijheidsgraden) van de t-toets van Student van een serie waarden.

**TTest1w_df() retourneert de geaggregeerde df-waarde (vrijheidsgraden) van de t-toets van Student van een serie waarden.** (weight, value)

ttest1w_dif

**TTest1w_dif()** retourneert het geaggregeerde gemiddelde verschil van de t-toets van Student van een serie waarden.

**TTest1w_dif() retourneert het geaggregeerde gemiddelde verschil van de t-toets van Student van een serie waarden.** (weight, value)

ttest1w_lower

**TTest1w_lower()** retourneert de geaggregeerde ondergrenswaarde van het·vertrouwensinterval voor een serie waarden.

**TTest1w_lower() retourneert de geaggregeerde ondergrenswaarde van het·vertrouwensinterval voor een serie waarden.** (weight, value [, sig])

ttest1w_sig

**TTest1w_sig()** retourneert het geaggregeerde tweezijdige niveau van significantie van Students t-toets op voor een serie waarden.

**TTest1w_sig() retourneert het geaggregeerde tweezijdige niveau van significantie van Students t-toets op voor een serie waarden.** (weight, value)

ttest1w_sterr

**TTest1w_sterr()** retourneert de geaggregeerde standaardfout van de t-toets van Student voor een serie waarden.

**TTest1w_sterr() retourneert de geaggregeerde standaardfout van de t-toets van Student voor een serie waarden.** (weight, value)

ttest1w_t

**TTest1w_t()** retourneert de geaggregeerde t-waarde voor een serie waarden.

**TTest1w_t() retourneert de geaggregeerde t-waarde voor een serie waarden.** (weight, value)

ttest1w_upper

**TTest1w_upper()** retourneert de geaggregeerde bovengrenswaarde van het·vertrouwensinterval voor een serie waarden.

**TTest1w_upper() retourneert de geaggregeerde bovengrenswaarde van het·vertrouwensinterval voor een serie waarden.** (weight, value [, sig])

### TTest_conf

**TTest_conf** retourneert het geaggregeerde vertrouwensinterval voor de t-testwaarde voor twee onafhankelijke steekproeven.

Deze functie is van toepassing op student t-testen voor onafhankelijke steekproeven.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTest_conf ( grp, value [, sig [, eq_var]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in **group**. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |
| grp | Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Type**. |
| sig | Het tweezijdige niveau van significantie kan worden opgegeven in **sig**. Als dit wordt weggelaten, wordt **sig** ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%. |
| eq_var | Als **eq_var** wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als **eq_var** wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeelden:**

```
TTest_conf( Group, Value )
TTest_conf( Group, Value, Sig, false )
```

**Zie ook:**

p *Een typisch t-test-rapport opstellen (page 593)*

### TTest_df

**TTest_df()** retourneert de geaggregeerde waarde (vrijheidsgraden) van de t-toets van Student voor twee onafhankelijke series waarden.

Deze functie is van toepassing op student t-testen voor onafhankelijke steekproeven.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTest_df (grp, value [, eq_var])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in **group**. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |
| grp | Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Type**. |
| eq_var | Als **eq_var** wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als **eq_var** wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeelden:**

```
TTest_df( Group, Value )
TTest_df( Group, Value, false )
```

**Zie ook:**

p *Een typisch t-test-rapport opstellen (page 593)*

## TTest_dif

**TTest_dif()** is een numerieke functie die het geaggregeerde gemiddelde verschil van Student t-test voor twee onafhankelijke series waarden retourneert.

Deze functie is van toepassing op student t-testen voor onafhankelijke steekproeven.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTest_dif (grp, value [, eq_var] )
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in **group**. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |
| grp | Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Type**. |
| eq_var | Als **eq_var** wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als **eq_var** wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeelden:**

```
TTest_dif( Group, Value )
TTest_dif( Group, Value, false )
```

**Zie ook:**

p *Een typisch t-test-rapport opstellen (page 593)*

### TTest_lower

**TTest_lower()** retourneert de geaggregeerde ondergrenswaarde van het·vertrouwensinterval voor twee onafhankelijke series waarden.

Deze functie is van toepassing op student t-testen voor onafhankelijke steekproeven.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTest_lower (grp, value [, sig [, eq_var]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in **group**. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |
| grp | Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Type**. |
| sig | Het tweezijdige niveau van significantie kan worden opgegeven in **sig**. Als dit wordt weggelaten, wordt **sig** ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%. |
| eq_var | Als **eq_var** wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als **eq_var** wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeelden:**

```
TTest_lower( Group, Value )
TTest_lower( Group, Value, Sig, false )
```

**Zie ook:**

p *Een typisch t-test-rapport opstellen (page 593)*

### TTest_sig

**TTest_sig()** retourneert het geaggregeerde tweezijdige niveau van significantie van Students t-toets op voor twee onafhankelijke series waarden.

Deze functie is van toepassing op student t-testen voor onafhankelijke steekproeven.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTest_sig (grp, value [, eq_var])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in **group**. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |
| grp | Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Type**. |
| eq_var | Als **eq_var** wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als **eq_var** wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeelden:**

```
TTest_sig( Group, Value )
TTest_sig( Group, Value, false )
```

**Zie ook:**

p *Een typisch t-test-rapport opstellen (page 593)*

### TTest_sterr

**TTest_sterr()** retourneert de geaggregeerde standaardfout van de t-toets van Student voor twee onafhankelijke series waarden.

Deze functie is van toepassing op student t-testen voor onafhankelijke steekproeven.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTest_sterr (grp, value [, eq_var])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in **group**. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |
| grp | Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Type**. |
| eq_var | Als **eq_var** wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als **eq_var** wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeelden:**

```
TTest_sterr( Group, Value )
TTest_sterr( Group, Value, false )
```

**Zie ook:**

p *Een typisch t-test-rapport opstellen (page 593)*

### TTest_t

**TTest_t()** retourneert de geaggregeerde t-waarde voor twee onafhankelijke waarden.

Deze functie is van toepassing op student t-testen voor onafhankelijke steekproeven.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTest_t(grp, value[, eq_var])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in **group**. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |
| grp | Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Type**. |
| eq_var | Als **eq_var** wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als **eq_var** wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeeld:**

```
TTest_t( Group, Value, false )
```

**Zie ook:**

p *Een typisch t-test-rapport opstellen (page 593)*

### TTest_upper

**TTest_upper()** retourneert de geaggregeerde bovengrenswaarde van het·vertrouwensinterval voor twee onafhankelijke series waarden.

Deze functie is van toepassing op student t-testen voor onafhankelijke steekproeven.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTest_upper (grp, value [, sig [, eq_var]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in **group**. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |
| grp | Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Type**. |
| sig | Het tweezijdige niveau van significantie kan worden opgegeven in **sig**. Als dit wordt weggelaten, wordt **sig** ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%. |
| eq_var | Als **eq_var** wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als **eq_var** wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeelden:**

```
TTest_upper( Group, Value )
TTest_upper( Group, Value, sig, false )
```

**Zie ook:**

p *Een typisch t-test-rapport opstellen (page 593)*

### TTestw_conf

**TTestw_conf()** retourneert de geaggregeerde t-waarde voor twee onafhankelijke waarden.

Deze functie is van toepassing op student t-testen voor twee onafhankelijke steekproeven waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTestw_conf (weight, grp, value [, sig [, eq_var]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in **group**. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |
| weight | Elke waarde in **value** kan een of meer keer worden geteld, afhankelijk van de corresponderende wegingsfactor in **weight**. |
| grp | Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Type**. |
| sig | Het tweezijdige niveau van significantie kan worden opgegeven in **sig**. Als dit wordt weggelaten, wordt **sig** ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%. |
| eq_var | Als **eq_var** wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als **eq_var** wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeelden:**

```
TTestw_conf( Weight, Group, Value )
TTestw_conf( Weight, Group, Value, sig, false )
```

**Zie ook:**

p *Een typisch t-test-rapport opstellen (page 593)*

### TTestw_df

**TTestw_df()** retourneert de geaggregeerde df-waarde (vrijheidsgraden) van de t-toets van Student van twee onafhankelijke series waarden.

Deze functie is van toepassing op student t-testen voor twee onafhankelijke steekproeven waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTestw_df (weight, grp, value [, eq_var])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| weight | Elke waarde in **value** kan een of meer keer worden geteld, afhankelijk van de corresponderende wegingsfactor in **weight**. |
| grp | Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Type**. |
| value | De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in **group**. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |
| eq_var | Als **eq_var** wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als **eq_var** wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeelden:**

```
TTestw_df( Weight, Group, Value )
TTestw_df( Weight, Group, Value, false )
```

**Zie ook:**

p *Een typisch t-test-rapport opstellen (page 593)*

## TTestw_dif

**TTestw_dif()** retourneert het geaggregeerde gemiddelde verschil van de t-toets van Student van twee onafhankelijke series waarden.

Deze functie is van toepassing op student t-testen voor twee onafhankelijke steekproeven waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTestw_dif (weight, grp, value)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| weight | Elke waarde in **value** kan een of meer keer worden geteld, afhankelijk van de corresponderende wegingsfactor in **weight**. |
| grp | Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Type**. |
| value | De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in **group**. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeelden:**

```
TTestw_dif( Weight, Group, Value )
TTestw_dif( Weight, Group, Value, false )
```

**Zie ook:**

p *Een typisch t-test-rapport opstellen (page 593)*

## TTestw_lower

**TTestw_lower()** retourneert de geaggregeerde ondergrenswaarde van het·vertrouwensinterval voor twee onafhankelijke series waarden.

Deze functie is van toepassing op student t-testen voor twee onafhankelijke steekproeven waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTestw_lower (weight, grp, value [, sig [, eq_var]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| weight | Elke waarde in **value** kan een of meer keer worden geteld, afhankelijk van de corresponderende wegingsfactor in **weight**. |
| grp | Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Type**. |
| value | De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in **group**. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |
| sig | Het tweezijdige niveau van significantie kan worden opgegeven in **sig**. Als dit wordt weggelaten, wordt **sig** ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%. |
| eq_var | Als **eq_var** wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als **eq_var** wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeelden:**

```
TTestw_lower( Weight, Group, Value )
```

```
TTestw_lower( Weight, Group, Value, sig, false )
```

**Zie ook:**

p *Een typisch t-test-rapport opstellen (page 593)*

### TTestw_sig

**TTestw_sig()** retourneert het geaggregeerde tweezijdige niveau van significantie van Students t-toets op voor twee onafhankelijke series waarden.

Deze functie is van toepassing op student t-testen voor twee onafhankelijke steekproeven waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTestw_sig ( weight, grp, value [, eq_var])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| weight | Elke waarde in **value** kan een of meer keer worden geteld, afhankelijk van de corresponderende wegingsfactor in **weight**. |
| grp | Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Type**. |
| value | De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in **group**. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |
| eq_var | Als **eq_var** wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als **eq_var** wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeelden:**

```
TTestw_sig( Weight, Group, Value )
TTestw_sig( Weight, Group, Value, false )
```

**Zie ook:**

p *Een typisch t-test-rapport opstellen (page 593)*

### TTestw_sterr

**TTestw_sterr()** retourneert de geaggregeerde standaardfout van de t-toets van Student voor twee onafhankelijke series waarden.

Deze functie is van toepassing op student t-testen voor twee onafhankelijke steekproeven waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTestw_sterr (weight, grp, value [, eq_var])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| weight | Elke waarde in **value** kan een of meer keer worden geteld, afhankelijk van de corresponderende wegingsfactor in **weight**. |
| grp | Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Type**. |
| value | De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in **group**. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |
| eq_var | Als **eq_var** wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als **eq_var** wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeelden:**

```
TTestw_sterr( Weight, Group, Value )
TTestw_sterr( Weight, Group, Value, false )
```

**Zie ook:**

p *Een typisch t-test-rapport opstellen (page 593)*

### TTestw_t

**TTestw_t()** retourneert de geaggregeerde t-waarde voor twee onafhankelijke waarden.

Deze functie is van toepassing op student t-testen voor twee onafhankelijke steekproeven waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
ttestw_t (weight, grp, value [, eq_var])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in **group**. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |
| weight | Elke waarde in **value** kan een of meer keer worden geteld, afhankelijk van de corresponderende wegingsfactor in **weight**. |
| grp | Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Type**. |

| Argument | Beschrijving |
|---|---|
| eq_var | Als **eq_var** wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als **eq_var** wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeelden:**

```
TTestw_t( Weight, Group, Value )
TTestw_t( Weight, Group, Value, false )
```

**Zie ook:**

p *Een typisch t-test-rapport opstellen (page 593)*

## TTestw_upper

**TTestw_upper()** retourneert de geaggregeerde bovengrenswaarde van het·vertrouwensinterval voor twee onafhankelijke series waarden.

Deze functie is van toepassing op student t-testen voor twee onafhankelijke steekproeven waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTestw_upper (weight, grp, value [, sig [, eq_var]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| weight | Elke waarde in **value** kan een of meer keer worden geteld, afhankelijk van de corresponderende wegingsfactor in **weight**. |

| Argument | Beschrijving |
|---|---|
| grp | Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Type**. |
| value | De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in **group**. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |
| sig | Het tweezijdige niveau van significantie kan worden opgegeven in **sig**. Als dit wordt weggelaten, wordt **sig** ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%. |
| eq_var | Als **eq_var** wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als **eq_var** wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeelden:**

```
TTestw_upper( Weight, Group, Value )
TTestw_upper( Weight, Group, Value, sig, false )
```

**Zie ook:**

p *Een typisch t-test-rapport opstellen (page 593)*

## TTest1_conf

**TTest1_conf()** retourneert de geaggregeerde waarde van het vertrouwensinterval voor een serie waarden.

Deze functie is van toepassing op student t-testen voor één steekproef.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTest1_conf (value [, sig ])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproeven die worden geëvalueerd. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |
| sig | Het tweezijdige niveau van significantie kan worden opgegeven in **sig**. Als dit wordt weggelaten, wordt **sig** ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeelden:**

```
TTest1_conf( Value )
TTest1_conf( Value, 0.005 )
```

**Zie ook:**

p *Een typisch t-test-rapport opstellen (page 593)*

## TTest1_df

**TTest1_df()** retourneert de geaggregeerde df-waarde (vrijheidsgraden) van de t-toets van Student van een serie waarden.

Deze functie is van toepassing op student t-testen voor één steekproef.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTest1_df (value)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproeven die worden geëvalueerd. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeeld:**

```
TTest1_df( Value )
```

**Zie ook:**

p *Een typisch t-test-rapport opstellen (page 593)*

## TTest1_dif

**TTest1_dif()** retourneert het geaggregeerde gemiddelde verschil van de t-toets van Student van een serie waarden.

Deze functie is van toepassing op student t-testen voor één steekproef.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTest1_dif (value)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproeven die worden geëvalueerd. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeeld:**

```
TTest1_dif( Value )
```

**Zie ook:**

p *Een typisch t-test-rapport opstellen (page 593)*

## TTest1_lower

**TTest1_lower()** retourneert de geaggregeerde ondergrenswaarde van het·vertrouwensinterval voor een serie waarden.

Deze functie is van toepassing op student t-testen voor één steekproef.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTest1_lower (value [, sig])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproeven die worden geëvalueerd. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |
| sig | Het tweezijdige niveau van significantie kan worden opgegeven in **sig**. Als dit wordt weggelaten, wordt **sig** ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeelden:**

```
TTest1_lower( value )
TTest1_lower( value, 0.005 )
```

**Zie ook:**

p *Een typisch t-test-rapport opstellen (page 593)*

## TTest1_sig

**TTest1_sig()** retourneert het geaggregeerde tweezijdige niveau van significantie van Students t-toets op voor een serie waarden.

Deze functie is van toepassing op student t-testen voor één steekproef.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTest1_sig (value)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproeven die worden geëvalueerd. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeeld:**

```
TTest1_sig( Value )
```

**Zie ook:**

p *Een typisch t-test-rapport opstellen (page 593)*

## TTest1_sterr

**TTest1_sterr()** retourneert de geaggregeerde standaardfout van de t-toets van Student voor een serie waarden.

Deze functie is van toepassing op student t-testen voor één steekproef.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTest1_sterr (value)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproeven die worden geëvalueerd. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeeld:**

```
TTest1_sterr( Value )
```

**Zie ook:**

p *Een typisch t-test-rapport opstellen (page 593)*

## TTest1_t

**TTest1_t()** retourneert de geaggregeerde t-waarde voor een serie waarden.

Deze functie is van toepassing op student t-testen voor één steekproef.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTest1_t (value)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproeven die worden geëvalueerd. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeeld:**

```
TTest1_t( Value )
```

---

**Zie ook:**

p *Een typisch t-test-rapport opstellen (page 593)*

### TTest1_upper

**TTest1_upper()** retourneert de geaggregeerde bovengrenswaarde van het·vertrouwensinterval voor een serie waarden.

Deze functie is van toepassing op student t-testen voor één steekproef.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTest1_upper (value [, sig])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproeven die worden geëvalueerd. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |
| sig | Het tweezijdige niveau van significantie kan worden opgegeven in **sig**. Als dit wordt weggelaten, wordt **sig** ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeelden:**

```
TTest1_upper( Value )
TTest1_upper( Value, 0.005 )
```

**Zie ook:**

p *Een typisch t-test-rapport opstellen (page 593)*

### TTest1w_conf

**TTest1w_conf()** is een **numerieke** functie die de geaggregeerde waarde van het vertrouwensinterval voor een serie waarden retourneert.

Deze functie is van toepassing op student t-testen voor één steekproef waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTest1w_conf (weight, value [, sig ])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproeven die worden geëvalueerd. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |
| weight | Elke waarde in **value** kan een of meer keer worden geteld, afhankelijk van de corresponderende wegingsfactor in **weight**. |
| sig | Het tweezijdige niveau van significantie kan worden opgegeven in **sig**. Als dit wordt weggelaten, wordt **sig** ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeelden:**

```
TTest1w_conf( Weight, Value )
TTest1w_conf( Weight, Value, 0.005 )
```

**Zie ook:**

p *Een typisch t-test-rapport opstellen (page 593)*

### TTest1w_df

**TTest1w_df()** retourneert de geaggregeerde df-waarde (vrijheidsgraden) van de t-toets van Student van een serie waarden.

Deze functie is van toepassing op student t-testen voor één steekproef waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTest1w_df (weight, value)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproeven die worden geëvalueerd. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |
| weight | Elke waarde in **value** kan een of meer keer worden geteld, afhankelijk van de corresponderende wegingsfactor in **weight**. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeeld:**

```
TTest1w_df( Weight, Value )
```

**Zie ook:**

p *Een typisch t-test-rapport opstellen (page 593)*

### TTest1w_dif

**TTest1w_dif()** retourneert het geaggregeerde gemiddelde verschil van de t-toets van Student van een serie waarden.

Deze functie is van toepassing op student t-testen voor één steekproef waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTest1w_dif (weight, value)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproeven die worden geëvalueerd. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |
| weight | Elke waarde in **value** kan een of meer keer worden geteld, afhankelijk van de corresponderende wegingsfactor in **weight**. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeeld:**

```
TTest1w_dif( Weight, Value )
```

**Zie ook:**

p *Een typisch t-test-rapport opstellen (page 593)*

## TTest1w_lower

**TTest1w_lower()** retourneert de geaggregeerde ondergrenswaarde van het·vertrouwensinterval voor een serie waarden.

Deze functie is van toepassing op student t-testen voor één steekproef waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTest1w_lower (weight, value [, sig ])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproeven die worden geëvalueerd. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |
| weight | Elke waarde in **value** kan een of meer keer worden geteld, afhankelijk van de corresponderende wegingsfactor in **weight**. |
| sig | Het tweezijdige niveau van significantie kan worden opgegeven in **sig**. Als dit wordt weggelaten, wordt **sig** ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeelden:**

```
TTest1w_lower( Weight, Value )
TTest1w_lower( Weight, Value, 0.005 )
```

**Zie ook:**

p *Een typisch t-test-rapport opstellen (page 593)*

### TTest1w_sig

**TTest1w_sig()** retourneert het geaggregeerde tweezijdige niveau van significantie van Students t-toets op voor een serie waarden.

Deze functie is van toepassing op student t-testen voor één steekproef waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTest1w_sig (weight, value)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproeven die worden geëvalueerd. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |
| weight | Elke waarde in **value** kan een of meer keer worden geteld, afhankelijk van de corresponderende wegingsfactor in **weight**. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeeld:**

```
TTest1w_sig( Weight, Value )
```

**Zie ook:**

p *Een typisch t-test-rapport opstellen (page 593)*

### TTest1w_sterr

**TTest1w_sterr()** retourneert de geaggregeerde standaardfout van de t-toets van Student voor een serie waarden.

Deze functie is van toepassing op student t-testen voor één steekproef waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTest1w_sterr (weight, value)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproeven die worden geëvalueerd. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |
| weight | Elke waarde in **value** kan een of meer keer worden geteld, afhankelijk van de corresponderende wegingsfactor in **weight**. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeeld:**

```
TTest1w_sterr( Weight, Value )
```

---

**Zie ook:**

p *Een typisch t-test-rapport opstellen (page 593)*

### TTest1w_t

**TTest1w_t()** retourneert de geaggregeerde t-waarde voor een serie waarden.

Deze functie is van toepassing op student t-testen voor één steekproef waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTest1w_t ( weight, value)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproeven die worden geëvalueerd. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |
| weight | Elke waarde in **value** kan een of meer keer worden geteld, afhankelijk van de corresponderende wegingsfactor in **weight**. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeeld:**

```
TTest1w_t( Weight, Value )
```

---

**Zie ook:**

p *Een typisch t-test-rapport opstellen (page 593)*

## TTest1w_upper

**TTest1w_upper()** retourneert de geaggregeerde bovengrenswaarde van het·vertrouwensinterval voor een serie waarden.

Deze functie is van toepassing op student t-testen voor één steekproef waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTest1w_upper (weight, value [, sig])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproeven die worden geëvalueerd. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |
| weight | Elke waarde in **value** kan een of meer keer worden geteld, afhankelijk van de corresponderende wegingsfactor in **weight**. |
| sig | Het tweezijdige niveau van significantie kan worden opgegeven in **sig**. Als dit wordt weggelaten, wordt **sig** ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeelden:**

```
TTest1w_upper( Weight, Value )
TTest1w_upper( Weight, Value, 0.005 )
```

**Zie ook:**

p *Een typisch t-test-rapport opstellen (page 593)*

## Z-testfuncties

Een statistisch onderzoek van twee steekproefgemiddelden. Een z-test met twee steekproeven onderzoekt of twee steekproeven van elkaar verschillen. Dit wordt over het algemeen gebruikt als twee normale verdelingen onbekende varianties hebben en als een experiment een grote steekproefgrootte heeft.

De volgende statistische z-testfuncties zijn gegroepeerd volgens het type invoergegevensreeksen dat van toepassing is op de functie.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

*Voorbeelden van het gebruik van z-test-functies (page 597)*

### Functies voor notatie met één kolom

De volgende functies zijn van toepassing op z-tests met eenvoudige invoergegevensreeksen.

ztest_conf

**ZTest_conf()** retourneert de geaggregeerde z-waarde voor een serie waarden.

```
ZTest_conf() retourneert de geaggregeerde z-waarde voor een serie waarden.
(value [, sigma [, sig ])
```

ztest_dif

**ZTest_dif()** retourneert het geaggregeerde gemiddelde verschil van de z-toets van een serie waarden.

```
ZTest_dif() retourneert het geaggregeerde gemiddelde verschil van de z-toets
van een serie waarden. (value [, sigma])
```

ztest_sig

**ZTest_sig()** retourneert het geaggregeerde tweezijdige niveau van significantie van z-toets op voor een serie waarden.

```
ZTest_sig() retourneert het geaggregeerde tweezijdige niveau van
significantie van z-toets op voor een serie waarden. (value [, sigma])
```

ztest_sterr

**ZTest_sterr()** retourneert de geaggregeerde standaardfout van de z-toets voor een serie waarden.

```
ZTest_sterr() retourneert de geaggregeerde standaardfout van de z-toets voor
een serie waarden. (value [, sigma])
```

ztest_z

**ZTest_z()** retourneert de geaggregeerde z-waarde voor een serie waarden.

```
ZTest_z() retourneert de geaggregeerde z-waarde voor een serie waarden.
(value [, sigma])
```

ztest_lower

**ZTest_lower()** retourneert de geaggregeerde ondergrenswaarde van het·vertrouwensinterval voor twee onafhankelijke series waarden.

```
ZTest_lower() retourneert de geaggregeerde ondergrenswaarde van
het·vertrouwensinterval voor twee onafhankelijke series waarden. (grp, value
[, sig [, eq_var]])
```

ztest_upper

**ZTest_upper()** retourneert de geaggregeerde bovengrenswaarde van het·vertrouwensinterval voor twee onafhankelijke series waarden.

```
ZTest_upper() retourneert de geaggregeerde bovengrenswaarde van
het·vertrouwensinterval voor twee onafhankelijke series waarden. (grp, value
[, sig [, eq_var]])
```

### Functies voor notatie met twee gewogen kolommen

De volgende functies zijn van toepassing op z-tests waarbij de invoergegevensreeksen worden gegeven in de notatie met twee gewogen kolommen.

ztestw_conf

**ZTestw_conf()** retourneert de geaggregeerde z-waarde van het vertrouwensinterval voor een serie waarden.

```
ZTestw_conf() retourneert de geaggregeerde z-waarde van het
vertrouwensinterval voor een serie waarden. (weight, value [, sigma [, sig]])
```

ztestw_dif

**ZTestw_dif()** retourneert het geaggregeerde gemiddelde verschil van de z-toets van een serie waarden.

```
ZTestw_dif() retourneert het geaggregeerde gemiddelde verschil van de z-toets
van een serie waarden. (weight, value [, sigma])
```

ztestw_lower

**ZTestw_lower()** retourneert de geaggregeerde ondergrenswaarde van het·vertrouwensinterval voor twee onafhankelijke series waarden.

```
ZTestw_lower() retourneert de geaggregeerde ondergrenswaarde van
het·vertrouwensinterval voor twee onafhankelijke series waarden. (weight,
value [, sigma])
```

ztestw_sig

**ZTestw_sig()** retourneert het geaggregeerde tweezijdige niveau van significantie van z-toets op voor een serie waarden.

```
ZTestw_sig() retourneert het geaggregeerde tweezijdige niveau van
significantie van z-toets op voor een serie waarden. (weight, value [,
sigma])
```

ztestw_sterr

**ZTestw_sterr()** retourneert de geaggregeerde standaardfout van de z-toets voor een serie waarden.

```
ZTestw_sterr() retourneert de geaggregeerde standaardfout van de z-toets voor
een serie waarden. (weight, value [, sigma])
```

ztestw_upper

**ZTestw_upper()** retourneert de geaggregeerde bovengrenswaarde van het·vertrouwensinterval voor twee onafhankelijke series waarden.

```
ZTestw_upper() retourneert de geaggregeerde bovengrenswaarde van
het·vertrouwensinterval voor twee onafhankelijke series waarden. (weight,
value [, sigma])
```

ztestw_z

**ZTestw_z()** retourneert de geaggregeerde z-waarde voor een serie waarden.

**ZTestw_z() retourneert de geaggregeerde z-waarde voor een serie waarden.**
**(**weight, value [, sigma]**)**

## ZTest_z

**ZTest_z()** retourneert de geaggregeerde z-waarde voor een serie waarden.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

**ZTest_z(**value[, sigma]**)**

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproefwaarden die moeten worden geëvalueerd. Er wordt uitgegaan van een populatiegemiddelde van 0. Als u met een ander gemiddelde wilt werken, trekt u dat gemiddelde van de steekproefwaarden af. |
| sigma | De standaarddeviatie kan, indien bekend, worden opgegeven in **sigma**. Als **sigma** wordt weggelaten, wordt de huidige standaarddeviatie van de steekproef gebruikt. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeeld:**

```
ZTest_z( Value-TestValue )
```

**Zie ook:**

p *Voorbeelden van het gebruik van z-test-functies (page 597)*

## ZTest_sig

**ZTest_sig()** retourneert het geaggregeerde tweezijdige niveau van significantie van z-toets op voor een serie waarden.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
ZTest_sig(value[, sigma])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproefwaarden die moeten worden geëvalueerd. Er wordt uitgegaan van een populatiegemiddelde van 0. Als u met een ander gemiddelde wilt werken, trekt u dat gemiddelde van de steekproefwaarden af. |
| sigma | De standaarddeviatie kan, indien bekend, worden opgegeven in **sigma**. Als **sigma** wordt weggelaten, wordt de huidige standaarddeviatie van de steekproef gebruikt. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeeld:**

```
ZTest_sig(Value-TestValue)
```

---

**Zie ook:**

p *Voorbeelden van het gebruik van z-test-functies (page 597)*

## ZTest_dif

**ZTest_dif()** retourneert het geaggregeerde gemiddelde verschil van de z-toets van een serie waarden.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
ZTest_dif(value[, sigma])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproefwaarden die moeten worden geëvalueerd. Er wordt uitgegaan van een populatiegemiddelde van 0. Als u met een ander gemiddelde wilt werken, trekt u dat gemiddelde van de steekproefwaarden af. |
| sigma | De standaarddeviatie kan, indien bekend, worden opgegeven in **sigma**. Als **sigma** wordt weggelaten, wordt de huidige standaarddeviatie van de steekproef gebruikt. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeeld:**

```
ZTest_dif(Value-TestValue)
```

---

**Zie ook:**

p *Voorbeelden van het gebruik van z-test-functies (page 597)*

## ZTest_sterr

**ZTest_sterr()** retourneert de geaggregeerde standaardfout van de z-toets voor een serie waarden.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
ZTest_sterr(value[, sigma])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproefwaarden die moeten worden geëvalueerd. Er wordt uitgegaan van een populatiegemiddelde van 0. Als u met een ander gemiddelde wilt werken, trekt u dat gemiddelde van de steekproefwaarden af. |
| sigma | De standaarddeviatie kan, indien bekend, worden opgegeven in **sigma**. Als **sigma** wordt weggelaten, wordt de huidige standaarddeviatie van de steekproef gebruikt. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeeld:**

```
ZTest_sterr(Value-TestValue)
```

---

**Zie ook:**

p *Voorbeelden van het gebruik van z-test-functies (page 597)*

## ZTest_conf

**ZTest_conf()** retourneert de geaggregeerde z-waarde voor een serie waarden.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
ZTest_conf(value[, sigma[, sig]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproefwaarden die moeten worden geëvalueerd. Er wordt uitgegaan van een populatiegemiddelde van 0. Als u met een ander gemiddelde wilt werken, trekt u dat gemiddelde van de steekproefwaarden af. |
| sigma | De standaarddeviatie kan, indien bekend, worden opgegeven in **sigma**. Als **sigma** wordt weggelaten, wordt de huidige standaarddeviatie van de steekproef gebruikt. |
| sig | Het tweezijdige niveau van significantie kan worden opgegeven in **sig**. Als dit wordt weggelaten, wordt **sig** ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeeld:**

```
ZTest_conf(Value-TestValue)
```

**Zie ook:**

p *Voorbeelden van het gebruik van z-test-functies (page 597)*

## ZTest_lower

**ZTest_lower()** retourneert de geaggregeerde ondergrenswaarde van het·vertrouwensinterval voor twee onafhankelijke series waarden.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
ZTest_lower (grp, value [, sig [, eq_var]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in **group**. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |
| grp | Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Type**. |
| sig | Het tweezijdige niveau van significantie kan worden opgegeven in **sig**. Als dit wordt weggelaten, wordt **sig** ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%. |
| eq_var | Als **eq_var** wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als **eq_var** wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeelden:**

```
ZTest_lower( Group, Value )
ZTest_lower( Group, Value, sig, false )
```

**Zie ook:**

p *Voorbeelden van het gebruik van z-test-functies (page 597)*

## ZTest_upper

**ZTest_upper()** retourneert de geaggregeerde bovengrenswaarde van het·vertrouwensinterval voor twee onafhankelijke series waarden.

Deze functie is van toepassing op student t-testen voor onafhankelijke steekproeven.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
ZTest_upper (grp, value [, sig [, eq_var]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in **group**. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |
| grp | Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Type**. |
| sig | Het tweezijdige niveau van significantie kan worden opgegeven in **sig**. Als dit wordt weggelaten, wordt **sig** ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%. |
| eq_var | Als **eq_var** wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als **eq_var** wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeelden:**

```
ZTest_upper( Group, Value )
ZTest_upper( Group, Value, sig, false )
```

**Zie ook:**

p *Voorbeelden van het gebruik van z-test-functies (page 597)*

## ZTestw_z

**ZTestw_z()** retourneert de geaggregeerde z-waarde voor een serie waarden.

Deze functie is van toepassing op z-testen waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
ZTestw_z (weight, value [, sigma])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De waarden moeten op **value** worden geretourneerd. Er wordt uitgegaan van een steekproefgemiddelde van 0. Als u met een ander gemiddelde wilt werken, trekt u die waarde van de steekproefwaarden af. |
| weight | Elke steekproefwaarde in **value** kan één of meer keer worden geteld, afhankelijk van de corresponderende waarde van **weight**. |
| sigma | De standaarddeviatie kan, indien bekend, worden opgegeven in **sigma**. Als **sigma** wordt weggelaten, wordt de huidige standaarddeviatie van de steekproef gebruikt. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeeld:**

```
ZTestw_z( Weight, Value-TestValue)
```

---

**Zie ook:**

p *Voorbeelden van het gebruik van z-test-functies (page 597)*

## ZTestw_sig

**ZTestw_sig()** retourneert het geaggregeerde tweezijdige niveau van significantie van z-toets op voor een serie waarden.

Deze functie is van toepassing op z-testen waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
ZTestw_sig (weight, value [, sigma])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De waarden moeten op **value** worden geretourneerd. Er wordt uitgegaan van een steekproefgemiddelde van 0. Als u met een ander gemiddelde wilt werken, trekt u die waarde van de steekproefwaarden af. |
| weight | Elke steekproefwaarde in **value** kan één of meer keer worden geteld, afhankelijk van de corresponderende waarde van **weight**. |
| sigma | De standaarddeviatie kan, indien bekend, worden opgegeven in **sigma**. Als **sigma** wordt weggelaten, wordt de huidige standaarddeviatie van de steekproef gebruikt. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeeld:**

```
ZTestw_sig( Weight, Value-TestValue)
```

---

**Zie ook:**

p *Voorbeelden van het gebruik van z-test-functies (page 597)*

## ZTestw_dif

**ZTestw_dif()** retourneert het geaggregeerde gemiddelde verschil van de z-toets van een serie waarden.

Deze functie is van toepassing op z-testen waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
ZTestw_dif ( weight, value [, sigma])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De waarden moeten op **value** worden geretourneerd. Er wordt uitgegaan van een steekproefgemiddelde van 0. Als u met een ander gemiddelde wilt werken, trekt u die waarde van de steekproefwaarden af. |
| weight | Elke steekproefwaarde in **value** kan één of meer keer worden geteld, afhankelijk van de corresponderende waarde van **weight**. |
| sigma | De standaarddeviatie kan, indien bekend, worden opgegeven in **sigma**. Als **sigma** wordt weggelaten, wordt de huidige standaarddeviatie van de steekproef gebruikt. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeeld:**

```
ZTestw_dif( Weight, Value-TestValue)
```

---

**Zie ook:**

p *Voorbeelden van het gebruik van z-test-functies (page 597)*

## ZTestw_sterr

**ZTestw_sterr()** retourneert de geaggregeerde standaardfout van de z-toets voor een serie waarden.

Deze functie is van toepassing op z-testen waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
ZTestw_sterr (weight, value [, sigma])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De waarden moeten op **value** worden geretourneerd. Er wordt uitgegaan van een steekproefgemiddelde van 0. Als u met een ander gemiddelde wilt werken, trekt u die waarde van de steekproefwaarden af. |
| weight | Elke steekproefwaarde in **value** kan één of meer keer worden geteld, afhankelijk van de corresponderende waarde van **weight**. |
| sigma | De standaarddeviatie kan, indien bekend, worden opgegeven in **sigma**. Als **sigma** wordt weggelaten, wordt de huidige standaarddeviatie van de steekproef gebruikt. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeeld:**

```
ZTestw_sterr( Weight, Value-TestValue)
```

---

**Zie ook:**

p *Voorbeelden van het gebruik van z-test-functies (page 597)*

## ZTestw_conf

**ZTestw_conf()** retourneert de geaggregeerde z-waarde van het vertrouwensinterval voor een serie waarden.

Deze functie is van toepassing op z-testen waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
ZTest_conf(weight, value[, sigma[, sig]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproefwaarden die moeten worden geëvalueerd. Er wordt uitgegaan van een populatiegemiddelde van 0. Als u met een ander gemiddelde wilt werken, trekt u dat gemiddelde van de steekproefwaarden af. |
| weight | Elke steekproefwaarde in **value** kan één of meer keer worden geteld, afhankelijk van de corresponderende waarde van **weight**. |
| sigma | De standaarddeviatie kan, indien bekend, worden opgegeven in **sigma**. Als **sigma** wordt weggelaten, wordt de huidige standaarddeviatie van de steekproef gebruikt. |
| sig | Het tweezijdige niveau van significantie kan worden opgegeven in **sig**. Als dit wordt weggelaten, wordt **sig** ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeeld:**

```
ZTestw_conf( Weight, Value-TestValue)
```

---

**Zie ook:**

p *Voorbeelden van het gebruik van z-test-functies (page 597)*

## ZTestw_lower

**ZTestw_lower()** retourneert de geaggregeerde ondergrenswaarde van het·vertrouwensinterval voor twee onafhankelijke series waarden.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
ZTestw_lower (grp, value [, sig [, eq_var]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

| Argument | Beschrijving |
|---|---|
| value | De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in **group**. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |
| grp | Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Type**. |
| sig | Het tweezijdige niveau van significantie kan worden opgegeven in **sig**. Als dit wordt weggelaten, wordt **sig** ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%. |
| eq_var | Als **eq_var** wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als **eq_var** wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeelden:**

```
ZTestw_lower( Group, Value )
ZTestw_lower( Group, Value, sig, false )
```

**Zie ook:**

p *Voorbeelden van het gebruik van z-test-functies (page 597)*

## ZTestw_upper

**ZTestw_upper()** retourneert de geaggregeerde bovengrenswaarde van het·vertrouwensinterval voor twee onafhankelijke series waarden.

Deze functie is van toepassing op student t-testen voor onafhankelijke steekproeven.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
ZTestw_upper (grp, value [, sig [, eq_var]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in **group**. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Value**. |
| grp | Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam **Type**. |
| sig | Het tweezijdige niveau van significantie kan worden opgegeven in **sig**. Als dit wordt weggelaten, wordt **sig** ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%. |
| eq_var | Als **eq_var** wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als **eq_var** wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeelden:**

```
ZTestw_upper( Group, Value )
ZTestw_upper( Group, Value, sig, false )
```

**Zie ook:**

p *Voorbeelden van het gebruik van z-test-functies (page 597)*

## Voorbeelden van statistische testfuncties

Dit gedeelte bevat voorbeelden van statistische testfuncties zoals deze worden toegepast op diagrammen en het script voor het laden van gegevens.

## Voorbeelden van het gebruik van chi2-test-functies in grafieken

De chi2-test-functies worden gebruikt om waarden te zoeken die verband houden met chi2 statistische analyse.

In deze sectie wordt beschreven hoe u visualisaties kunt samenstellen met behulp van steekproefgegevens om de waarden van de chi2-distributietestfuncties te vinden die beschikbaar zijn in Qlik Sense. Raadpleeg de individuele onderwerpen over chi2-test-diagramfuncties voor een beschrijving van syntaxis en argumenten.

### De gegevens voor de steekproeven laden

Er zijn drie sets steekproefgegevens die drie verschillende statistische steekproeven beschrijven die in het script moeten worden geladen.

Doe het volgende:

1. Maak een nieuwe app.

   Voer het volgende in bij het laden van gegevens:

   ```
   // 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. Klik op om gegevens te laden.

### De visualisaties van de chi2-test-grafiekfuncties maken

#### Voorbeeld: Steekproef 1

Doe het volgende:

1. Klik in de editor voor het laden van gegevens op om naar het app-overzicht te gaan en vervolgens op het werkblad te klikken dat u eerder hebt gemaakt.
   De werkbladweergave wordt geopend.
2. Klik op **Werkblad bewerken** om het werkblad te bewerken.
3. Voeg vanuit **Diagrammen** een tabel toe en voeg vanuit **Velden** Grp, Grade en Count toe als dimensies.
   In deze tabel worden de steekproefgegevens weergegeven.
4. Voeg nog een tabel toe met de volgende uitdrukking als dimensie.
   `ValueList('p','df','Chi2')`
   Hierbij wordt de functie voor synthetische dimensies gebruikt om labels voor de dimensies te maken met de namen van de drie chi2-test-functies.
5. Voeg de volgende uitdrukking als meting toe aan de tabel.

```
IF(ValueList('p','df','Chi2')='p',Chi2Test_p(Grp,Grade,Count),

IF(ValueList('p','df','Chi2')='df',Chi2Test_df(Grp,Grade,Count),

Chi2Test_Chi2(Grp,Grade,Count)))
```

   Dit heeft als effect dat de resulterende waarde van elke chi2-test-functie in de tabel naast de bijbehorende synthetische dimensie wordt geplaatst.
6. Stel de **getalnotatie** van de meting in op **Getal** en **3 significante cijfers**.

*In de uitdrukking voor de meting zou u ook de volgende uitdrukking kunnen gebruiken:* `Pick (Match(ValueList('p','df','Chi2'),'p','df','Chi2'),Chi2Test_p (Grp,Grade,Count),Chi2Test_df(Grp,Grade,Count),Chi2Test_Chi2(Grp,Grade,Count))`

#### Resultaat

De resulterende tabel voor de chi2-test-functies voor de gegevens van Steekproef 1 bevat de volgende waarden:

Resultatentabel

| p | df | Chi2 |
|---|---|---|
| 0.820 | 5 | 2.21 |

### Voorbeeld: Steekproef 2

Doe het volgende:

1. In het werkblad dat u aan het bewerken was in het voorbeeld Steekproef 1, voegt u vanuit **Grafieken** een tabel toe en vanuit **Velden** Sex, Opinion en OpCount als dimensies.
2. Maak een kopie van de resultatentabel uit Steekproef 1 met behulp van de opdrachten **Kopiëren** en **Plakken**. Bewerk de uitdrukking in de meting en vervang de argumenten in alle drie de chi2-test-functies door de namen van de velden die worden gebruikt in de gegevens voor Steekproef 2, bijvoorbeeld: `Chi2Test_p(Sex,Opinion,OpCount)`.

### Resultaat

De resulterende tabel voor de chi2-test-functies voor de gegevens van Steekproef 2 bevat de volgende waarden:

Resultatentabel

| p | df | Chi2 |
|---|---|---|
| 0.000309 | 2 | 16.2 |

### Voorbeeld: Steekproef 3

Doe het volgende:

1. Maak nog twee extra tabellen op dezelfde manier als in de voorbeelden voor de gegevens van Steekproef 1 en Steekproef 2. Gebruik in de dimensietabel de volgende velden als dimensies: Gender, Description, Actual en Expected.
2. Gebruik in de resultatentabel de namen van de velden die worden gebruikt in de gegevens voor Steekproef 3, bijvoorbeeld: `Chi2Test_p(Gender,Description,Actual,Expected)`.

### Resultaat

De resulterende tabel voor de chi2-test-functies voor de gegevens van Steekproef 3 bevat de volgende waarden:

Resultatentabel

| p | df | Chi2 |
|---|---|---|
| 0.000308 | 2 | 16.2 |

### Voorbeelden van het gebruik van chi2-test-functies in het load-script voor gegevens

De chi2-test-functies worden gebruikt om waarden te zoeken die verband houden met chi2 statistische analyse. In deze sectie wordt beschreven hoe u de chi2-distributietestfuncties die beschikbaar zijn in Qlik Sense kunt gebruiken in het load-script voor gegevens. Raadpleeg de individuele onderwerpen over chi2-test-scriptfuncties voor een beschrijving van syntaxis en argumenten.

In dit voorbeeld wordt een tabel gebruikt die het aantal studenten die een graad hebben behaald (A-F) voor twee groepen studenten (I en II) bevat.

Data table

| Group | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| I | 15 | 7 | 9 | 20 | 26 | 19 |
| II | 10 | 11 | 7 | 15 | 21 | 16 |

#### De steekproefgegevens laden

Doe het volgende:

1. Maak een nieuwe app.

   Voer het volgende in de editor voor het laden van gegevens in:

   ```
   // 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. Klik op om gegevens te laden.

U hebt nu de steekproefgegevens geladen.

**De waarden van de chi2-test-functie laden**

Nu gaan we de chi2-test-waarden laden op basis van de steekproefgegevens in een nieuwe tabel, gegroepeerd op Grp.

Doe het volgende:

1. Voeg in de editor voor het laden van gegevens het volgende toe aan het uiteinde van het script:

```
// Sample_1 data is pre-aggregated... Note: make sure you set your DecimalSep='.' at the top of the script.

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. Klik op om gegevens te laden.

U hebt nu de chi2-test-waarden geladen in een tabel met de naam Chi2_table.

**Resultaten**

U kunt de resulterende chi2-test-waarden bekijken in de gegevensmodelviewer onder **Voorbeeld**. Deze zouden er als volgt moeten uitzien:

Results

| Grp | chi2 | df | p |
|---|---|---|---|
| I | 16.00 | 5 | 0.007 |
| II | 9.40 | 5 | 0.094 |

## Een typisch t-test-rapport opstellen

Een typisch t-test-rapport voor cursisten kan tabellen met **Group Statistics** en resultaten van **Independent Samples Test** bevatten.

In de volgende secties gaan we deze tabellen bouwen met Qlik Senset-test-functies die worden toegepast op twee onafhankelijke groepen steekproeven, Observation en Comparison. De bijbehorende tabellen voor deze steekproeven zouden er als volgt uitzien:

Groepsstatistieken

| Type | N | Mean | Standard Deviation | Standard Error Mean |
|---|---|---|---|---|
| Comparison | 20 | 11.95 | 14.61245 | 3.2674431 |
| Observation | 20 | 27.15 | 12.507997 | 2.7968933 |

Onafhankelijke voorbeeldtest

| 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 |

**De steekproefgegevens laden**

Doe het volgende:

1. Maak een nieuwe app met een nieuw werkblad.
2. Voer het volgende in de editor voor laden van gegevens in:

```
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 '|');
```

In dit load-script is **recno()** opgenomen omdat voor **crosstable** drie argumenten vereist zijn. **recno()** levert daarom niet meer dan een extra argument. In dit geval een ID voor elke rij. Zonder dit argument zouden geen steekproefwaarden voor **Comparison** worden geladen.

3. Klik op om gegevens te laden.

**De tabel Group statistics maken**

Doe het volgende:

1. Klik in de editor voor laden van gegevens op om naar het app-overzicht te gaan en vervolgens op het werkblad te klikken dat u eerder hebt gemaakt.
   Hiermee wordt de werkbladweergave geopend.
2. Klik op **Werkblad bewerken** om het werkblad te bewerken.
3. Voeg vanuit **Diagrammen** een tabel toe en vanuit **Velden** een Type als een dimensie aan de tabel.
4. Voeg de volgende uitdrukkingen als metingen toe.

Voorbeelduitdrukkingen

| Label | Uitdrukking |
|---|---|
| N | Count(Value) |
| Mean | Avg(Value) |
| Standard Deviation | Stdev(Value) |
| Standard Error Mean | Sterr(Value) |

5. Klik op **Sorteren** en zorg dat Type aan het begin van de sorteerlijst staat.

**Resultaat**

Een tabel Group statistics voor deze steekproeven zou er als volgt uitzien:

Groepsstatistieken

| Type | N | Mean | Standard Deviation | Standard Error Mean |
|---|---|---|---|---|
| Comparison | 20 | 11.95 | 14.61245 | 3.2674431 |
| Observation | 20 | 27.15 | 12.507997 | 2.7968933 |

**De tabel Independent sample test maken**

Doe het volgende:

1. Klik op **Werkblad bewerken** om het werkblad te bewerken.
2. Voeg vanuit **Diagrammen** een tabel toe met de volgende uitdrukking aan de tabel als een dimensie. `=ValueList (Dual('Equal Variance not Assumed', 0), Dual('Equal Variance Assumed', 1))` en geef het het label Type.
3. Voeg de volgende uitdrukkingen als metingen toe:

Voorbeelduitdrukkingen

| Label | Uitdrukking |
|---|---|
| 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)) |

**Resultaat**

Onafhankelijke voorbeeldtest

| 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 |

## Voorbeelden van het gebruik van z-test-functies

De z-test-functies worden gebruikt om waarden te vinden die zijn gekoppeld aan de statistische z-test-analyse voor grote testgegevensvoorbeelden, gewoonlijk groter dan 30, waarbij de variantie bekend is.

In dit gedeelte wordt beschreven hoe visualisaties kunnen worden gebouwd aan de hand van de voorbeeldgegevens, om de waarden te vinden van de z-test-functies die beschikbaar zijn in Qlik Sense. Raadpleeg de individuele onderwerpen over z-test-diagramfuncties voor een beschrijving van syntaxis en argumenten.

### De steekproefgegevens laden

De voorbeeldgegevens die hier worden gebruikt, zijn dezelfde die worden gebruikt in de voorbeelden voor t-test-functies. De grootte van de hoeveelheid voorbeeldgegevens wordt gewoonlijk als te klein beschouwd voor z-testanalyse, maar is voldoende om het gebruik van de verschillende z-test-functies te illustreren in Qlik Sense.

Doe het volgende:

1. Maak een nieuwe app met een nieuw werkblad.

   *Als u een app gemaakt hebt voor de t-test-functies, kunt u die gebruiken en een nieuw werkblad voor deze functies maken.*

2. Voer het volgende in de editor voor laden van gegevens in:

```
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 '|');
```

In dit load-script is **recno()** opgenomen omdat voor **crosstable** drie argumenten vereist zijn. **recno()** levert daarom niet meer dan een extra argument. In dit geval een ID voor elke rij. Zonder dit argument zouden geen steekproefwaarden voor **Comparison** worden geladen.

3. Klik op om gegevens te laden.

**De tabel z-test maken**

Doe het volgende:

1. Klik in de editor voor laden van gegevens op om naar het app-overzicht te gaan en vervolgens op het werkblad te klikken dat u eerder hebt gemaakt.
   De werkbladweergave wordt geopend.
2. Klik op **Werkblad bewerken** om het werkblad te bewerken.
3. Voeg vanuit **Diagrammen** een tabel toe en vanuit **Velden** een Type als een dimensie.
4. Voeg de volgende uitdrukkingen aan de tabel toe als metingen.

Voorbeelduitdrukkingen

| Label | Uitdrukking |
|---|---|
| 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) |

*Mogelijk wilt u de getalnotatie van de metingen aanpassen om betekenisvolle waarden weer te geven. De tabel is gemakkelijker te lezen als u getalnotatie voor de meeste metingen instelt op **Getal>Eenvoudig**, in plaats van op **Auto**. Maar voor ZTest Sig gebruikt u bijvoorbeeld de getalnotatie: **Aangepast** en past u vervolgens het opmaakpatroon aan naar **#.######**.*

**Resultaat**

De resulterende tabel voor de z-test-functies voor de voorbeeldgegevens bevat de volgende waarden:

z-test resultatentabel

| 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 |

**De tabel z-testw maken**

De z-testw-functies kunnen worden gebruikt als de invoergegevensreeks wordt opgegeven in twee gewogen kolommen. De expressies vereisen een waarde voor het argument weight.

In de voorbeelden hier wordt overal de waarde 2 gebruikt, maar u kunt ook een uitdrukking gebruiken, die een waarde definieert voor weight voor elke observatie.

Doe het volgende:

1. Klik in de editor voor laden van gegevens op om naar het app-overzicht te gaan en vervolgens op het werkblad te klikken dat u eerder hebt gemaakt.
   De werkbladweergave wordt geopend.
2. Klik op **Werkblad bewerken** om het werkblad te bewerken.
3. Voeg vanuit **Diagrammen** een tabel toe en vanuit **Velden** een Type als een dimensie.
4. Voeg de volgende uitdrukkingen aan de tabel toe als metingen.

   Voorbeelduitdrukkingen

   | Label | Uitdrukking |
   |---|---|
   | 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) |

   Gebruik dezelfde getalnotatie als in het voorbeeld van z-test-functies.

**Resultaat**

De resulterende tabel voor de z-testw-functies bevat de volgende waarden:

z-testw resultatentabel

| 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 |

## Aggregatiefuncties voor tekenreeksen

In deze sectie worden de aggregatiefuncties voor tekenreeksen beschreven.

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

### Tekenreeksaggregatiefuncties in het script voor het laden van gegevens

#### Concat

**Concat()** wordt gebruikt om tekenreekswaarden te combineren. Deze scriptfunctie retourneert de geaggregeerde aaneengeschakelde tekenreeks van alle waarden van de uitdrukking, herhaald over een aantal records, zoals vastgelegd door een **group by**-clausule.

```
Concat ([ distinct ] expression [, delimiter [, sort-weight]])
```

#### FirstValue

**FirstValue()** retourneert de waarde die als eerste werd geladen vanuit de records die zijn gedefinieerd door de uitdrukking, gesorteerd door een **group by**-clausule.

> *Deze functie is alleen beschikbaar als scriptfunctie.*

```
FirstValue (expression)
```

#### LastValue

**LastValue()** retourneert de waarde die als laatste werd geladen vanuit de records die zijn gedefinieerd door de uitdrukking, gesorteerd door een **group by**-clausule.

> *Deze functie is alleen beschikbaar als scriptfunctie.*

```
LastValue (expression)
```

#### MaxString

**MaxString()** vindt tekenreekswaarden in de uitdrukking en retourneert de laatste tekstwaarde alfabetisch gesorteerd over een aantal records, zoals gedefinieerd door een **group by**-clausule.

```
MaxString (expression )
```

### MinString

**MinString()** vindt tekenreekswaarden in de uitdrukking en retourneert de eerste tekstwaarde alfabetisch gesorteerd over een aantal records, zoals gedefinieerd door een **group by**-clausule.

```
MinString (expression )
```

## Tekenreeksaggregatiefuncties in diagrammen

De volgende diagramfuncties zijn beschikbaar voor het aggregeren van tekenreeksen in diagrammen.

### Concat

**Concat()** wordt gebruikt om tekenreekswaarden te combineren. De functie retourneert de geaggregeerde aaneenschakeling van tekenreeksen van alle waarden van de uitdrukking in elke dimensie.

```
Concat - diagramfunctie({[SetExpression] [DISTINCT] [TOTAL [<fld{, fld}>]]
string[, delimiter[, sort_weight]])
```

### MaxString

**MaxString()** vindt tekenreekswaarden in de uitdrukking of het veld en retourneert de laatste tekstwaarde in alfabetische sorteervolgorde.

```
MaxString - diagramfunctie({[SetExpression] [TOTAL [<fld{, fld}>]]} expr)
```

### MinString

**MinString()** vindt tekenreekswaarden in de uitdrukking of het veld en retourneert de eerste tekstwaarde in alfabetische sorteervolgorde.

```
MinString - diagramfunctie({[SetExpression] [TOTAL [<fld {, fld}>]]} expr)
```

## Concat

**Concat()** wordt gebruikt om tekenreekswaarden te combineren. Deze scriptfunctie retourneert de geaggregeerde aaneengeschakelde tekenreeks van alle waarden van de uitdrukking, herhaald over een aantal records, zoals vastgelegd door een **group by**-clausule.

**Syntaxis:**

```
Concat ([ distinct ] string [, delimiter [, sort-weight]])
```

**Retourgegevenstypen:** tekenreeks

**Argumenten:**

De uitdrukking die of het veld dat de tekenreeks bevat die moet worden verwerkt.

Argumenten

| Argument | Beschrijving |
|---|---|
| string | De uitdrukking die of het veld dat de tekenreeks bevat die moet worden verwerkt. |
| delimiter | Elke waarde kan worden gescheiden door de tekenreeks die is opgegeven in delimiter. |

| Argument | Beschrijving |
|---|---|
| sort-weight | De volgorde bij aaneenschakelen kan worden bepaald door de waarde van de dimensie **sort-weight**, indien aanwezig, waarbij de tekenreeks die overeenkomt met de laagste waarde als eerste in de aaneenschakeling verschijntzijn. |
| distinct | Als het woord **distinct** voor de uitdrukking staat, worden alle duplicaten genegeerd. |

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

Voorbeelden en resultaten

| Voorbeeld | Resultaat | Resultaten nadat ze zijn toegevoegd aan een werkblad |
|---|---|---|
| `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,Concat(Team) as TeamConcat1`<br>`Resident TeamData Group By SalesGroup;` | SalesGroup<br><br>East<br><br>West | TeamConcat1<br><br>AlphaBetaDeltaGammaGamma<br><br>EpsilonEtaThetaZeta |
| Stel dat de tabel **TeamData** is geladen zoals in het vorige voorbeeld:<br><br>`LOAD SalesGroup,Concat(distinct Team,'-')`<br>`as TeamConcat2 Resident TeamData Group By`<br>`SalesGroup;` | SalesGroup<br><br>East<br><br>West | TeamConcat2<br><br>Alpha-Beta-Delta-Gamma<br><br>Epsilon-Eta-Theta-Zeta |
| Stel dat de tabel **TeamData** is geladen zoals in het vorige voorbeeld. Doordat het argument voor **sort-weight** wordt toegevoegd, worden de resultaten gerangschikt op basis van de waarde van de dimensie Amount:<br><br>`LOAD SalesGroup,Concat(distinct Team,'-`<br>`',Amount) as TeamConcat2 Resident TeamData`<br>`Group By SalesGroup;` | SalesGroup<br><br>East<br><br>West | TeamConcat2<br><br>Delta-Beta-Gamma-Alpha<br><br>Eta-Epsilon-Zeta-Theta |

## Concat - diagramfunctie

**Concat()** wordt gebruikt om tekenreekswaarden te combineren. De functie retourneert de geaggregeerde aaneenschakeling van tekenreeksen van alle waarden van de uitdrukking in elke dimensie.

**Syntaxis:**

```
Concat({[SetExpression] [DISTINCT] [TOTAL [<fld{, fld}>]]} string[, delimiter
[, sort_weight]])
```

**Retourgegevenstypen:** tekenreeks

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| string | De uitdrukking die of het veld dat de tekenreeks bevat die moet worden verwerkt. |
| delimiter | Elke waarde kan worden gescheiden door de tekenreeks die is opgegeven in delimiter. |
| sort-weight | De volgorde bij aaneenschakelen kan worden bepaald door de waarde van de dimensie **sort-weight**, indien aanwezig, waarbij de tekenreeks die overeenkomt met de laagste waarde als eerste in de aaneenschakeling verschijntzijn. |
| SetExpression | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| DISTINCT | Als het woord **DISTINCT** voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd. |
| TOTAL | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br><br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

**Voorbeelden en resultaten:**

Results table

| SalesGroup | Amount | Concat(Team) | Concat(TOTAL <SalesGroup> Team) |
|---|---|---|---|
| East | 25000 | Alpha | AlphaBetaDeltaGammaGamma |

| SalesGroup | Amount | Concat(Team) | Concat(TOTAL <SalesGroup> Team) |
|---|---|---|---|
| East | 20000 | BetaGammaGamma | AlphaBetaDeltaGammaGamma |
| East | 14000 | Delta | AlphaBetaDeltaGammaGamma |
| West | 17000 | Epsilon | EpsilonEtaThetaZeta |
| West | 14000 | Eta | EpsilonEtaThetaZeta |
| West | 23000 | Theta | EpsilonEtaThetaZeta |
| West | 19000 | Zeta | EpsilonEtaThetaZeta |

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `Concat(Team)` | De tabel wordt gconstrueerd op basis van de dimensies SalesGroup en Amount, en variaties op de meting Concat(Team). Wanneer het totalenresultaat wordt genegeerd, let dan op dat, hoewel er gegevens zijn voor acht waarden van Team verspreid over twee waarden van SalesGroup, is het enige resultaat van de meting Concat(Team) waarbij meer dan één tekenreekswaarde voor Team wordt aaneengeschakeld in de tabel de rij die de dimensie Amount 20000 bevat, met BetaGammaGamma als resultaat. Dit komt doordat er drie waarden zijn voor het Amount 20000 in de invoergegevens. Alle andere resultaten worden niet aaneengeschakeld als de meting de dimensies omvat omdat er slechts één waarde van Team is voor elke combinatie van SalesGroup en Amount. |
| `Concat (DISTINCT Team,', ')` | Beta, Gamma, omdat de kwalificatie DISTINCT betekent dat het dubbele resultaat van Gamma buiten beschouwing wordt gelaten. Tevens wordt het argument voor het scheidingsteken gedefinieerd als een komma gevolgd door een spatie. |
| `Concat (TOTAL <SalesGroup> Team)` | Alle tekenreekswaarden voor alle waarden van Team worden aaneengeschakeld als de kwalificatie TOTAL wordt gebruikt. Als de veldselectie <SalesGroup> is opgegeven, worden de resultaten opgesplitst in de twee waarden van de dimensie SalesGroup. Voor de SalesGroupEast zijn de resultaten AlphaBetaDeltaGammaGamma. Voor de SalesGroupWest zijn de resultaten EpsilonEtaThetaZeta. |
| `Concat (TOTAL <SalesGroup> Team,';', Amount)` | Door het argument voor **sort-weight** toe te voegen: Amount, worden de resultaten gerangschikt op basis van de waarde van de dimensie Amount. De resultaten worden DeltaBetaGammaGammaAlpha en EtaEpsilonZEtaTheta. |

Gebruikte gegevens in voorbeeld:

```
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()** retourneert de waarde die als eerste werd geladen vanuit de records die zijn gedefinieerd door de uitdrukking, gesorteerd door een **group by**-clausule.

*Deze functie is alleen beschikbaar als scriptfunctie.*

**Syntaxis:**

```
FirstValue ( expr)
```

**Retourgegevenstypen:** dual

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |

**Beperkingen:**

Als er geen tekstwaarde wordt gevonden, is het resultaat NULL.

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

Resulterende gegevens

| Voorbeeld | Resultaat | Resultaten op een werkblad |
|---|---|---|
| `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>East<br>West | FirstTeamLoaded<br>Gamma<br>Zeta |

## LastValue

**LastValue()** retourneert de waarde die als laatste werd geladen vanuit de records die zijn gedefinieerd door de uitdrukking, gesorteerd door een **group by**-clausule.

*Deze functie is alleen beschikbaar als scriptfunctie.*

**Syntaxis:**

```
LastValue ( expr )
```

**Retourgegevenstypen:** dual

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |

**Beperkingen:**

Als er geen tekstwaarde wordt gevonden, is het resultaat NULL.

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens ten minste de velden die in de resultatenkolom staan aan een werkblad in onze app toe om het resultaat te bekijken.

Als u dezelfde weergave wilt als in de onderstaande resultatenkolom, schakelt u in het eigenschappenvenster onder Sorteren over van Automatisch naar Aangepast en schakelt u numeriek en alfabetisch sorteren uit.

| Voorbeeld | Resultaat | Resulaten met aangepast sorteren |
|---|---|---|
| `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>East<br>West | LastTeamLoaded<br>Beta<br>Theta |

## MaxString

**MaxString()** vindt tekenreekswaarden in de uitdrukking en retourneert de laatste tekstwaarde alfabetisch gesorteerd over een aantal records, zoals gedefinieerd door een **group by**-clausule.

**Syntaxis:**

```
MaxString ( expr )
```

**Retourgegevenstypen:** dual

**Argumenten:**

| Argument | Beschrijving |
|---|---|
| expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |

**Beperkingen:**

Als er geen tekstwaarde wordt gevonden, is het resultaat NULL.

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

| Voorbeeld | Resultaat | |
|---|---|---|
| `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,MaxString(Team) as MaxString1 Resident TeamData Group By SalesGroup;` | SalesGroup<br><br>East<br><br>West | MaxString1<br><br>Gamma<br><br>Zeta |
| Stel dat de tabel **TeamData** wordt geladen zoals in het voorafgaande voorbeeld en het load-script voor gegevens de opdracht SET bevat: `SET DateFormat='DD/MM/YYYY';`:<br><br>`LOAD SalesGroup,MaxString(Date) as MaxString2 Resident TeamData Group By SalesGroup;` | SalesGroup<br><br>East<br><br>West | MaxString2<br><br>01/11/2013<br><br>01/12/2013 |

## MaxString - diagramfunctie

**MaxString()** vindt tekenreekswaarden in de uitdrukking of het veld en retourneert de laatste tekstwaarde in alfabetische sorteervolgorde.

**Syntaxis:**

```
MaxString({[SetExpression] [TOTAL [<fld{, fld}>]]} expr)
```

**Retourgegevenstypen:** dubbele waarde

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| SetExpression | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |

| Argument | Beschrijving |
|---|---|
| TOTAL | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br><br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

**Beperkingen:**

Als de uitdrukking geen waarden bevat met een tekenreeksrepresentatie, wordt NULL geretourneerd.

**Voorbeelden en resultaten:**

Resultatentabel

| 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 |

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `MaxString (Team)` | Er zijn drie waarden van 20000 voor de dimensie Amount: twee voor Gamma (op verschillende datums) en één voor Beta. Het resultaat van de meting MaxString (Team) is daarom Gamma, omdat dit de hoogste waarde in de gesorteerde tekenreeksen is. |
| `MaxString (Date)` | 2013/11/01 is de grootste waarde voor Date van de drie die aan de dimensie Amount zijn gekoppeld. Hierbij wordt ervan uitgegaan dat uw script de SET-opdracht `SET DateFormat='YYYY-MM-DD';` bevat |

Gebruikte gegevens in voorbeeld:

```
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()** vindt tekenreekswaarden in de uitdrukking en retourneert de eerste tekstwaarde alfabetisch gesorteerd over een aantal records, zoals gedefinieerd door een **group by**-clausule.

**Syntaxis:**

```
MinString ( expr )
```

**Retourgegevenstypen:** dual

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |

**Beperkingen:**

Als er geen tekstwaarde wordt gevonden, is het resultaat NULL.

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

Resulterende gegevens

| Voorbeeld | Resultaat | |
|---|---|---|
| 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 |
| Stel dat de tabel **TeamData** wordt geladen zoals in het voorafgaande voorbeeld en het load-script voor gegevens de opdracht SET bevat: `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 - diagramfunctie

**MinString()** vindt tekenreekswaarden in de uitdrukking of het veld en retourneert de eerste tekstwaarde in alfabetische sorteervolgorde.

**Syntaxis:**

```
MinString({[SetExpression] [TOTAL [<fld {, fld}>]]} expr)
```

**Retourgegevenstypen:** dubbele waarde

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| SetExpression | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |

| Argument | Beschrijving |
|---|---|
| TOTAL | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.<br><br>Wanneer u **TOTAL [<fld {.fld}>]** gebruikt en de **TOTAL**-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. |

**Voorbeelden en resultaten:**

Voorbeeldgegevens

| 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 |

Voorbeelden van functies

| Voorbeelden | Resultaten |
|---|---|
| `MinString (Team)` | Er zijn drie waarden van 20000 voor de dimensie Amount: twee voor Gamma (op verschillende datums) en één voor Beta. Het resultaat van de meting MinString (Team) is daarom Beta, omdat dit de eerste waarde in de gesorteerde tekenreeksen is. |
| `MinString (Date)` | 2013/11/01 is de eerste waarde voor Date van de drie die aan de dimensie Amount zijn gekoppeld. Hierbij wordt ervan uitgegaan dat uw script de SET-opdracht `SET DateFormat='YYYY-MM-DD';` bevat |

Gebruikte gegevens in voorbeeld:

```
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 '|');
```

## Functies voor synthetische dimensies

Een synthetische dimensie wordt in de app gemaakt met waarden die zijn gegenereerd via de functies voor synthetische dimensies en niet rechtstreeks met behulp van velden in het gegevensmodel. Wanneer waarden die zijn gegenereerd door een functie voor synthetische dimensies in een diagram worden gebruikt als een berekende dimensie, levert dit een synthetische dimensie op. Met synthetische dimensies kunt u bijvoorbeeld diagrammen maken met dimensies met waarden die voortkomen uit uw gegevens, oftewel dynamische dimensies.

*Synthetische dimensies worden niet beïnvloed door selecties.*

De volgende functies voor synthetische dimensies kunnen in diagrammen worden gebruikt.

ValueList

**ValueList()** retourneert een lijst met waarden die bij gebruik in een berekende dimensie een synthetische dimensie vormen.

```
ValueList - diagramfunctie (v1 {, Expression})
```

ValueLoop

ValueLoop() retourneert een set herhaalde waarden die bij gebruik in een berekende dimensie een synthetische dimensie vormen.

```
ValueLoop - diagramfunctie(from [, to [, step ]])
```

### ValueList - diagramfunctie

**ValueList()** retourneert een lijst met waarden die bij gebruik in een berekende dimensie een synthetische dimensie vormen.

*In diagrammen met een synthetische dimensie die is gemaakt met de functie **ValueList** kunt u verwijzen naar de dimensiewaarde die hoort bij een specifieke uitdrukkingscel, door de functie **ValueList** met dezelfde parameters in de diagramuitdrukking nogmaals op te nemen. De functie kan natuurlijk overal in de lay-out worden gebruikt, maar behalve voor synthetische dimensies heeft deze functie alleen nut binnen een aggregatiefunctie.*

*Synthetische dimensies worden niet beïnvloed door selecties.*

**Syntaxis:**

```
ValueList(v1 {,...})
```

**Retourgegevenstypen:** dual

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| v1 | Statische waarde (gewoonlijk een tekenreeks, maar kan een getal zijn). |
| {,...} | Optionele lijst van statische waarden. |

**Voorbeelden en resultaten:**

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `ValueList ('Number of Orders', 'Average Order Size', 'Total Amount')` | Als deze bijvoorbeeld worden gebruikt voor het maken van een dimensie in een tabel, leidt dit tot de drie tekensreekswaarden als rijlabels in de tabel. Hier kan vervolgens naar worden verwezen in een uitdrukking. |
| `=IF( ValueList ('Number of Orders', 'Average Order Size', 'Total Amount') = 'Number of Orders', count (SaleID), IF( ValueList ('Number of Orders', 'Average Order Size', 'Total Amount') = 'Average Order Size', avg (Amount), sum (Amount) ))` | Deze uitdrukking haalt de waarden uit de gemaakte dimensie en verwijst ernaar in een geneste IF-opdracht als invoer voor drie aggregatiefuncties:<br>**ValueList()**<br>Created dimension \| Year \| Added expression<br>**522.00**<br>Number of Orders \| 2012 \| 5.00<br>Number of Orders \| 2013 \| 7.00<br>Average Order Size \| 2012 \| 13.20<br>Average Order Size \| 2013 \| 15.43<br>Total Amount \| 2012 \| 66.00<br>Total Amount \| 2013 \| 108.00 |

Gegevens die worden gebruikt in voorbeelden:

```
SalesPeople:
LOAD * INLINE [
SaleID|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 - diagramfunctie

ValueLoop() retourneert een set herhaalde waarden die bij gebruik in een berekende dimensie een synthetische dimensie vormen.
De gegenereerde waarden beginnen met de **from**-waarde en eindigen met de **to** -waarde, met tussenliggende waarden gemaakt in stappen ter grootte van step.

*In diagrammen met een synthetische dimensie die is gemaakt met de functie **ValueLoop** kunt u verwijzen naar de dimensiewaarde die hoort bij een specifieke uitdrukkingscel, door de functie **ValueLoop** met dezelfde parameters in de diagramuitdrukking nogmaals op te nemen. De functie kan natuurlijk overal in de lay-out worden gebruikt, maar behalve voor synthetische dimensies heeft deze functie alleen nut binnen een aggregatiefunctie.*

*Synthetische dimensies worden niet beïnvloed door selecties.*

**Syntaxis:**

```
ValueLoop(from [, to [, step ]])
```

**Retourgegevenstypen:** dual

**Argumenten:**

Argumenten

| Argumenten | Beschrijving |
|---|---|
| from | Beginwaarde in de waardeset die wordt gegenereerd. |
| to | Eindwaarde in de waardeset die wordt gegenereerd. |
| step | Grootte van de stap tussen waarden. |

**Voorbeelden en resultaten:**

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `ValueLoop (1, 10)` | Hiermee wordt een dimensie in een tabel gemaakt, bijvoorbeeld om te gebruiken als genummerde labels. Dit voorbeeld leidt tot waarden genummerd van 1 tot 10. Naar deze waarden kan vervolgens in een uitdrukking worden verwezen. |
| `ValueLoop (2, 10,2)` | Dit voorbeeld leidt tot waarden met de nummers 2, 4, 6, 8 en 10 omdat het argument step een waarde van 2 heeft. |

## Geneste aggregaties

U kunt situaties tegengekomen waarbij u een aggregatie moet toepassen op het resultaat van een andere aggregatie. Dit wordt het nesten van aggregaties genoemd.

U kunt aggregaties in de meeste diagramuitdrukkingen niet nesten. U kunt aggregaties echter wel nesten als u de kwalificatie **TOTAL** gebruikt in de binnenste aggregatiefunctie.

*Er zijn geneste aggregaties van maximaal 100 niveaus toegestaan.*

### Geneste aggregaties met de kwalificatie TOTAL

**Voorbeeld:**

U wilt de som van het veld **Sales** berekenen, maar alleen transacties opnemen met een **OrderDate** die gelijk zijn aan die van vorig jaar. Het afgelopen jaar kan worden verkregen via de aggregatiefunctie **`Max (TOTAL`** `Year(OrderDate)`**`)`**.

De volgende aggregatie zou het gewenste resultaat retourneren:

```
Sum(If(Year(OrderDate)=Max(TOTAL Year(OrderDate)), Sales))
```

Voor Qlik Sense moet de kwalificatie **TOTAL** worden opgenomen voor dit type van nesten. Dit is vereist voor de gewenste vergelijking. Dit soort geneste aggregaties komt vrij vaak voor en vormt een goede manier van werken.

---

**Zie ook:**

p *Aggr - diagramfunctie (page 616)*

# 8.3 Aggr - diagramfunctie

**Aggr()** retourneert een reeks van waarden voor de uitdrukking die is berekend via de opgegeven dimensie of dimensies. Bijvoorbeeld de maximale waarden van verkopen, per klant, per regio.

De functie **Aggr** wordt gebruikt voor geneste aggregaties, waarbij de eerste parameter (de binnenste aggregatie) één keer per dimensiewaarde wordt berekend. De dimensies worden opgegeven in de tweede parameter (en opeenvolgende parameters).

Bovendien moet de functie **Aggr** worden ingesloten in een buitenste aggregatiefunctie met behulp van de reeks resultaten van de functie **Aggr** als invoer voor de aggregatie waarin de functie is genest.

**Syntaxis:**

```
Aggr({SetExpression}[DISTINCT] [NODISTINCT ] expr, StructuredParameter{,
StructuredParameter})
```

**Retourgegevenstypen:** dubbele waarde

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| expr | Een uitdrukking die bestaat uit een aggregatiefunctie. De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. |
| StructuredParameter | StructuredParameter bestaat uit een `dimensie` en, optioneel, sorteercriteria in de indeling: (Dimension(Sort-type, Ordering))<br><br>De dimensie is een enkel veld en kan geen uitdrukking zijn. Aan de hand van de dimensie wordt de reeks waarden bepaald waarvoor de uitdrukking Aggr wordt berekend.<br><br>Als sorteercriteria zijn opgenomen, wordt de reeks waarden die door de functie Aggr wordt gemaakt en voor de dimensie wordt berekend, gesorteerd. Dit is belangrijk wanneer de sorteervolgorde van invloed is op het resultaat van de uitdrukking waarin de functie Aggr is opgenomen.<br><br>Voor details over het gebruik van sorteercriteria raadpleegt u Sorteercriteria toevoegen aan de dimensie in de gestructureerde parameter. |
| SetExpression | De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. |
| DISTINCT | Als het argument van de uitdrukking wordt voorafgegaan door de kwalificatie **distinct** of als er geen kwalificatie wordt gebruikt, genereert elke afzonderlijke combinatie van dimensiewaarden slechts één retourwaarde. Dit is de normale manier om aggregaties te maken: elke afzonderlijke combinatie van dimensiewaarden levert één lijn in het diagram op. |
| NODISTINCT | Als het argument van de uitdrukking wordt voorafgegaan door de kwalificatie **nodistinct**, kan elke combinatie van dimensiewaarden meer dan één retourwaarde genereren, afhankelijk van de onderliggende gegevensstructuur. Als er slechts één dimensie is, retourneert de functie **aggr** een reeks met hetzelfde aantal elementen als er rijen zijn in de brongegevens. |

Elementaire aggregatiefuncties, zoals **Sum**, **Min** en **Avg**, retourneren een enkele numerieke waarde, terwijl de functie Aggr() kan worden vergeleken met het maken van een tijdelijke resultatenset (een virtuele tabel) waarover een andere aggregatie kan worden uitgevoerd. Zo kunt u bijvoorbeeld een gemiddelde verkoopwaarde berekenen door met een **Aggr()**-instructie de verkopen per klant op te tellen en daarna het gemiddelde van de opgetelde resultaten te berekenen: **Avg(TOTAL Aggr(Sum(Sales),Customer))**zijn.

*Gebruik de functie Aggr() in berekende dimensies als u geneste diagramaggregaties op meerdere niveaus wilt maken.*

**Beperkingen:**

Elke dimensie in een Aggr()-functie moet een enkel veld zijn en kan geen uitdrukking (berekende dimensie) zijn.

**Sorteercriteria toevoegen aan de dimensie in de gestructureerde parameter**

In zijn basisvorm is het argument StructuredParameter in de syntaxis van de functie Aggr een enkele dimensie. De uitdrukking Aggr(Sum(Sales, Month)) levert de totale verkoopwaarde voor elke maand op. Bij insluiting in een andere aggregatiefunctie kunnen er onverwachte resultaten optreden tenzij sorteercriteria worden gebruikt. Dit komt doordat sommige dimensies numeriek of alfabetisch kunnen worden gesorteerd, enzovoort.

In het argument StructuredParameter in de functie Aggr kunt u sorteercriteria opgeven voor de dimensie in uw uitdrukking. Op die manier legt u een sorteervolgorde op aan de virtuele tabel die door de functie Aggr wordt gemaakt.

Het argument StructuredParameter heeft de volgende syntaxis:

```
(FieldName, (Sort-type, Ordering))
```

Gestructureerde parameters kunnen worden genest:

```
(FieldName, (FieldName2, (Sort-type, Ordering)))
```

Het sorteertype kan zijn: NUMERIC, TEXT, FREQUENCY of LOAD_ORDER.

De volgende ordeningstypen zijn aan elke sorteertype gekoppeld:

Toegestane ordeningstypen

| Sorteertype | Toegestane ordeningstypen |
|---|---|
| NUMERIC | ASCENDING, DESCENDING of REVERSE |
| TEXT | ASCENDING, A2Z, DESCENDING, REVERSE of Z2A |
| FREQUENCY | DESCENDING, REVERSE of ASCENDING |
| LOAD_ORDER | ASCENDING, ORIGINAL, DESCENDING of REVERSE |

De ordeningstypen REVERSE en DESCENDING zijn equivalent.

Voor sorteertype TEXT zijn de ordeningstypen ASCENDING en A2Z equivalent en zijn DESCENDING, REVERSE en Z2A equivalent.

Voor sorteertype LOAD_ORDER zijn de ordeningstypen ASCENDING en ORIGINAL equivalent.

# Voorbeelden: diagramuitdrukkingen met Aggr

Voorbeelden - Diagramuitdrukkingen

## Diagramuitdrukking voorbeeld 1

### Load-script

Laad de volgende gegevens als een inline-load in de editor voor laden van gegevens om de onderstaande voorbeelden van diagramuitdrukkingen te maken.

```
ProductData:

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|25
Canutility|AA|8|15
Canutility|CC|0|19
] (delimiter is '|');
```

### Diagramuitdrukking

Maak een KPI-visualisatie in een Qlik Sense-werkblad. Voeg de volgende uitdrukking als meting toe aan de KPI:

```
Avg(Aggr(Sum(UnitSales*UnitPrice), Customer))
```

### Resultaat

376.7

### Uitleg

De uitdrukking `Aggr(Sum(UnitSales*UnitPrice), Customer)` levert de totale verkoopwaarde per **Customer** op en retourneert een reeks waarden: 295, 715 en 120 voor de drie **Customer**-waarden.

Het komt erop neer dat wij een tijdelijke lijst met waarden hebben samengesteld zonder een expliciete tabel of kolom te hoeven maken die deze waarden bevat.

Deze waarden worden gebruikt als invoer voor de functie **Avg()** om de gemiddelde verkoopwaarde te berekenen: 376.7.

## Diagramuitdrukking voorbeeld 2

### Load-script

Laad de volgende gegevens als een inline-load in de editor voor laden van gegevens om de onderstaande voorbeelden van diagramuitdrukkingen te maken.

```
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 '|');
```

### Diagramuitdrukking

Maak een tabelvisualisatie in een Qlik Sense-werkblad met **Customer**, **Product**, **UnitPrice** en **UnitSales** als dimensies. Voeg de volgende uitdrukking als meting toe aan de tabel:

```
Aggr(NODISTINCT Max(UnitPrice), Customer, Product)
```

### Resultaat

| 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 |
| Canutility | AA | 11 | 5 | 15 |
| Canutility | AA | 15 | 8 | 15 |
| Canutility | CC | 19 | 0 | 19 |

### Uitleg

Een array van waarden: 16, 16, 15, 15, 12, 12, 22, 22, 25, 15, 15 en 19. De kwalificatie **nodistinct** betekent dat de reeks één element bevat voor elke rij in de brongegevens: elk element is de maximale **UnitPrice** per **Customer** en **Product**.

## Diagramuitdrukking voorbeeld 3

### Load-script

Laad de volgende gegevens als een inline-load in de editor voor laden van gegevens om de onderstaande voorbeelden van diagramuitdrukkingen te maken.

```
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;
```

### Diagramuitdrukkingen

Maak een tabelvisualisatie in een Qlik Sense-werkblad met **Year** en **Month** als dimensies. Voeg de volgende uitdrukkingen aan de tabel toe als metingen:

- `Sum(Sales)`
- `Sum(Aggr( Rangesum(Above(Sum(Sales),0,12)), (Year, (Numeric, Ascending)), (Month, (Numeric, Ascending)) ))` gelabeld als Structured Aggr() in de tabel.

### Resultaat

| Year | Month | Sum(Sales) | Structured Aggr() |
|---|---|---|---|
| 2013 | Jan | 53495 | 53495 |
| 2013 | Feb | 48580 | 102075 |

| Year | Month | Sum(Sales) | Structured Aggr() |
|---|---|---|---|
| 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 |

### Uitleg

Dit voorbeeld geeft de geaggregeerde waarden weer over een periode van twaalf maanden voor elk jaar in chronologisch oplopende volgorde, en daarmee het deel met de gestructureerde parameters (Numeriek, Oplopend) van de uitdrukking **Aggr()**. Twee specifieke dimensies zijn vereist als gestructureerde parameters: **Jaar** en **Maand** , gesorteerd op (1) **Jaar** (numeriek) en (2) **Maand** (numeriek). Deze twee dimensies moeten in de visualisatie van de tabel of het diagram worden gebruikt. Dit is nodig zodat de lijst met dimensies van de functie **Aggr()** overeenkomt met de dimensies van het object dat in de visualisatie wordt gebruikt.

U kunt de resultaten van deze metingen vergelijken in een tabel of in afzonderlijke lijndiagrammen:

- `Sum(Aggr( Rangesum(Above(Sum(Sales),0,12)), (Year), (Month) ))`
- `Sum(Aggr( Rangesum(Above(Sum(Sales),0,12)), (Year, (Numeric, Ascending)), (Month, (Numeric, Ascending)) ))`

Het moet duidelijk zijn dat de gewenste accumulatie van geaggregeerde waarden door de laatstgenoemde uitdrukking wordt uitgevoerd.

**Zie ook:**

p *Basisaggregatiefuncties (page 361)*

## 8.4 Kleurfuncties

Deze functies kunnen worden gebruikt in uitdrukkingen die verband houden met het instellen en evalueren van de kleureigenschappen van diagramobjecten, alsmede in load-scripts voor gegevens.

*Qlik Sense ondersteunt de kleurfuncties **Color()**, **qliktechblue** en **qliktechgray** om redenen van achterwaartse compatibiliteit, maar het gebruik ervan wordt niet aanbevolen.*

ARGB

**ARGB()** wordt gebruikt in uitdrukkingen om de kleureigenschappen van een diagramobject in te stellen of te evalueren, waarbij de kleur wordt gedefinieerd door een rode component **r**, een groene component **g** een blauwe component **b**, met een alfafactor (ondoorzichtigheid) van **alpha**.

```
ARGB (alpha, r, g, b)
```

HSL

**HSL()** wordt gebruikt in uitdrukkingen om de kleureigenschappen van een diagramobject in te stellen of te evalueren, waarbij de kleur wordt gedefinieerd door waarden van **hue**, **saturation** en **luminosity** tussen 0 en 1.

```
HSL (hue, saturation, luminosity)
```

RGB

**RGB()** retourneert een geheel getal dat overeenkomst met de kleurcode van de kleur die is gedefinieerd door de drie parameters: de rode component r, de groene component g en de blauwe component b. Deze componenten moeten beschikken over gehele getalwaarden tussen 0 en 255. De functie kan worden gebruikt in uitdrukkingen voor het instellen of evalueren van de kleureigenschappen van een diagramobject.

```
RGB (r, g, b)
```

Colormix1

**Colormix1()** wordt gebruikt in uitdrukkingen om een ARGB-kleurrepresentatie van een kleurovergang van twee kleuren te retourneren, gebaseerd op een waarde tussen 0 en 1.

```
Colormix1 (Value , ColorZero , ColorOne)
```

Value is een reëel getal tussen 0 en 1.

- Als Value = 0 wordt ColorZero geretourneerd.
- Als Value = 1 wordt ColorOne geretourneerd.
- Als 0 < Value< 1 is, wordt de bijbehorende tussenliggende schaduwkleur geretourneerd.

ColorZero is een geldige RGB-kleurrepresentatie voor de kleur die moet worden toegewezen aan het lage uiteinde van het interval.

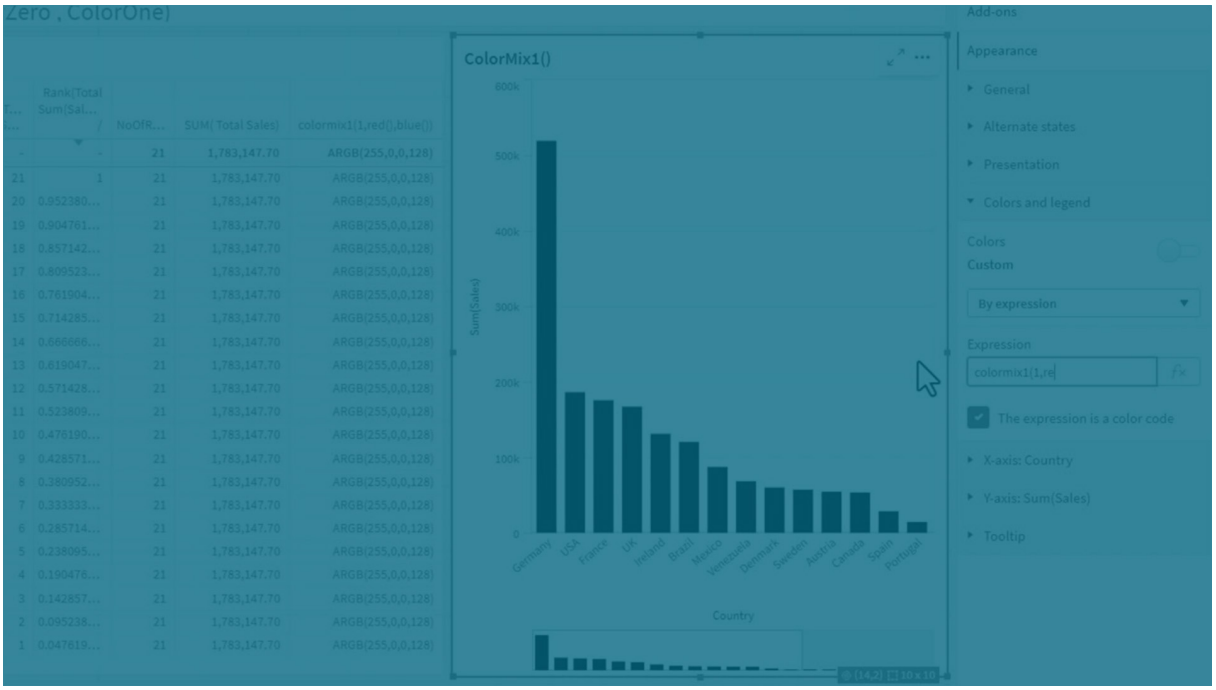
ColorOne is een geldige RGB-kleurrepresentatie voor de kleur die moet worden toegewezen aan het hoge uiteinde van het interval.

**Voorbeeld:**

```
Colormix1(0.5, red(), blue())
```

retourneert:

ARGB(255,64,0,64) (purple)




### Colormix2

**Colormix2()** wordt gebruikt in om een ARGB-kleurrepresentatie van een kleurovergang van twee kleuren gebaseerd op een waarde tussen -1 en 1, waarbij het mogelijk is een tussenliggende kleur op te geven voor de centrale (0) positie.

```
Colormix2 (Value ,ColorMinusOne , ColorOne[ , ColorZero])
```

Value is een reëel getal tussen -1 en 1.

- Als Value = -1, wordt de eerste kleur geretourneerd.
- Als Value = 1, wordt de tweede kleur geretourneerd.
- Als -1 < Value< 1 is, wordt de bijbehorende tussenliggende kleurenmix geretourneerd.

ColorMinusOne is een geldige RGB-kleurrepresentatie voor de kleur die moet worden toegewezen aan het lage uiteinde van het interval.

ColorOne is een geldige RGB-kleurrepresentatie voor de kleur die moet worden toegewezen aan het hoge uiteinde van het interval.

ColorZero is een optionele geldige RGB-kleurrepresentatie voor de kleur die moet worden toegewezen aan het midden van de interval.

### SysColor

**SysColor()** retourneert de ARGBkleurrepresentatie voor de Windows-systeemkleur nr, waarbij nr overeenkomt met de Windows API-functie **GetSysColor(nr)**.

```
SysColor (nr)
```

ColorMapHue

**ColorMapHue()** retourneert een ARGB-waarde van een kleur uit een kleurenkaart waarbij de tintcomponent van het HSV-kleurmodel wordt gevarieerd. De kleurenkaart begint met rood, doorloopt geel, groen, cyaan, blauw en magenta, en gaat terug naar rood. Voor x moet een waarde tussen 0 en 1 worden opgegeven.

```
ColorMapHue (x)
```

ColorMapJet

**ColorMapJet()** retourneert de ARGB-waarde van een kleur vanuit een kleurenkaart die begint bij blauw en doorloopt van cyaan en geel naar oranje en die daarna terugkeert naar rood. Voor x moet een waarde tussen 0 en 1 worden opgegeven.

```
ColorMapJet (x)
```

## Vooraf gedefinieerde kleurfuncties

De volgende functies kunnen worden gebruikt in uitdrukkingen voor vooraf gedefinieerde kleuren. Elke functie retourneert een RGB-kleurrepresentatie.

Optioneel kan een parameter voor alfafactor worden opgegeven. In dat geval wordt een ARGB-kleurrepresentatie geretourneerd. Een alfafactor van 0 komt overeen met volledige transparantie en een alfafactor van 255 komt overeen met volledige ondoorzichtigheid. Als geen waarde voor alfa is ingevoerd, wordt uitgegaan van 255.

Vooraf gedefinieerde kleurfuncties

| Kleurfunctie | RGB -waarde |
|---|---|
| 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) |

**Voorbeelden en resultaten:**

Voorbeelden en resultaten

| Voorbeelden | Resultaten |
|---|---|
| `Blue()` | RGB(0,0,128) |
| `Blue(128)` | ARGB(128,0,0,128) |

## ARGB

**ARGB()** wordt gebruikt in uitdrukkingen om de kleureigenschappen van een diagramobject in te stellen of te evalueren, waarbij de kleur wordt gedefinieerd door een rode component **r**, een groene component **g** een blauwe component **b**, met een alfafactor (ondoorzichtigheid) van **alpha**.

**Syntaxis:**

```
ARGB(alpha, r, g, b)
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **alpha** | Transparantiewaarde in het bereik 0 tot 255. Een 0 komt overeen met volledige transparantie en 255 komt overeen met volledige ondoorzichtigheid. |
| **r, g, b** | De componentwaarden voor rood, groen en blauw. Een kleurcomponent van 0 komt overeen met geen bijdrage en een kleurcomponent van 255 met volledige bijdrage. |

*Alle argumenten moeten uitdrukkingen zijn die resulteren in een geheel getal in het bereik van 0 tot 255.*

Bij het interpreteren van de numerieke component en het opmaken hiervan in een hexadecimale notatie, zijn de waarden van de kleurcomponenten gemakkelijker te zien. Lichtgroen heeft bijvoorbeeld het getal 4 278 255 360. In hexadecimale notatie is dit FF00FF00. De eerste twee posities 'FF' (255) geven het kanaal **alpha** aan. De volgende twee posities '00' geven de hoeveelheid **red** aan, de volgende twee posities 'FF' geven de hoeveelheid **green** aan en de laatste twee posities 00' geven de hoeveelheid **blue** aan.

Voorbeeld: diagramuitdrukking

| Voorbeeld | Resultaten |
|---|---|
| `ARGB(128, 0, 128, 0)` | Geeft de kleur groen met semitransparantie. |

## Voorbeeld - Basisprincipes voor ARGB

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Product`
  - `Sales`

### Load-script

```
Example:
LOAD * Inline [
Product, Sales
Product A, 200
Product B, 500
Product C, 100
Product D, 300
Product E, 700
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Product`

Maak de volgende meting:

- `=Sum(Sales)`, om de som van verkoop te berekenen.

Voer in het deelvenster **Eigenschappen** de volgende uitdrukking in als de **Uitdrukking achtergrondkleur** voor de `Sum(Sales)`-meting: Hierdoor wordt de celkleur toegepast volgens de verkoopberekening voor het product.

- `=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 | Sum(Sales) |
|---|---|
| Product C | 100 |
| Product D | 300 |
| Product E | 700 |

Volg in dit voorbeeld het volgende:

- Verkoopwaarden boven 400 zijn gemarkeerd met een groene achtergrond.
- Verkoopwaarden tussen 200 en 400 zijn gemarkeerd met een gele achtergrond.
- Alle andere verkoopwaarden zijn gemarkeerd met een rode achtergrond.

## Voorbeeld - Winstgegevens in een diagram markeren

Diagramuitdrukking

### Overzicht

Een gegevensverzameling bevat verkoopcijfers per klant. We willen de gegevens analyseren en de winstmarges markeren naargelang ze laag, middelhoog of hoog zijn.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Customer`
  - `SalesValue`
  - `Cost`
  - `ProfitMargin`

### Load-script

```
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%
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Customer`

Maak de volgende metingen:

- `=Sum(SalesValue)`, om de som van verkoop te berekenen.
- `=Sum(Cost)`, om de totale waarde van de verkoopkosten te berekenen.
- `=Sum(ProfitMargin)`, om de winstmarge op te tellen. Om deze waarde als percentage weer te geven, selecteert u onder **GetalnotatieGetal > Eenvoudige opmaak > 12,34%**.

Voer in het deelvenster **Eigenschappen** de volgende uitdrukking in als de **Uitdrukking achtergrondkleur** voor de `Sum(ProfitMargin)`-meting:

```
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
  )
)
```

Schakel in het venster **Uiterlijk**, onder **Presentatie**, **Totalen** uit.

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% |

De uitvoer van de functie `ARGB` geeft de volgende kleuren weer volgens de winstmarge van de klant:

- Groen, `ARGB(128, 0, 128, 0)`, voor klanten met winstmarges hoger dan 30%.
- Oranje, `ARGB(128, 255, 165, 0)`, voor klanten met middelhoge winstmarges tussen 10% en 30%.
- Rood, `ARGB(128, 255, 0, 0)`, voor klanten met een winstmarge onder 10%.

## Voorbeeld 5 – Scenario met ARGB-diagram

Diagramuitdrukking

**Overzicht**

Een gegevensverzameling bevat actuele en geschatte verkoopgegevens per land. Een verkoopmanager wil de gegevens in een staafdiagram weergeven en de verkoopcijfers markeren die hoger zijn dan de geschatte waarde voor de periode .

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
    - `Country`
    - `ActualSales`
    - `ProjectedSales`

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe staafdiagram en voeg dit veld toe als dimensie:

- `Country`

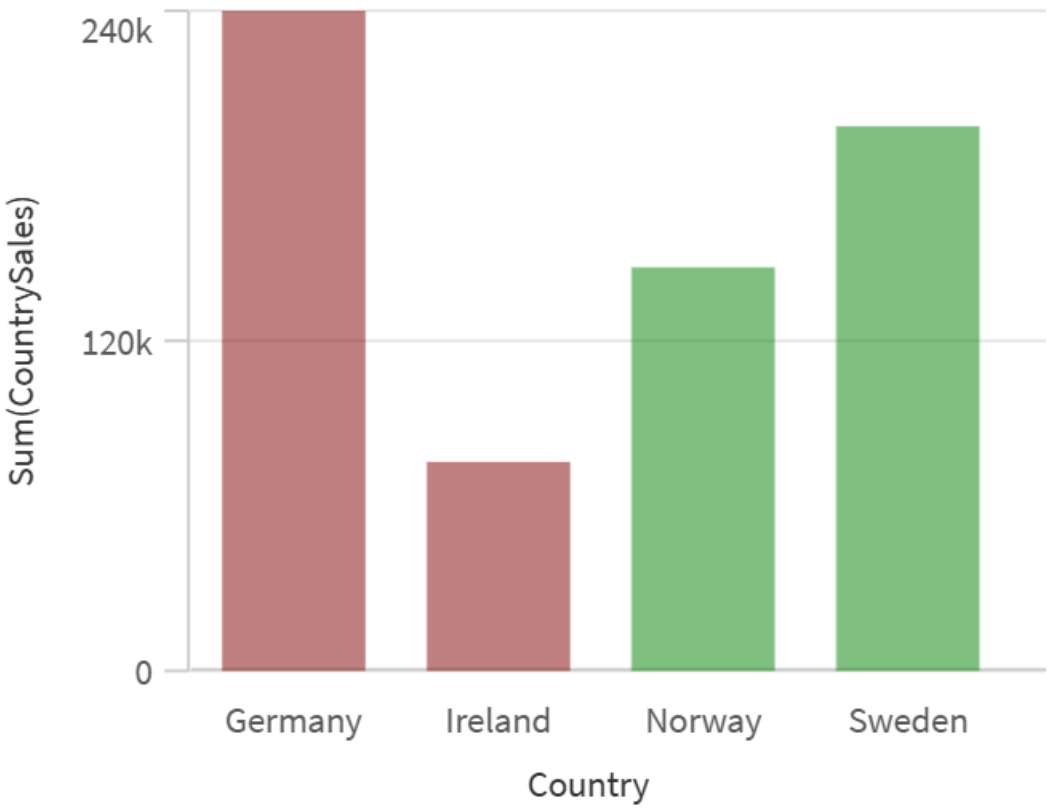
Maak de volgende meting:

- `=Sum(ActualSales)`, om de totale verkoopwaarde voor het land te berekenen.

Voeg een kleuruitdrukking toe om de werkelijke verkoop met de geschatte verkoop te vergelijken. Schakel in het eigenschappenvenster onder **Uiterlijk** > **Kleuren en legenda Kleuren**: **Automatisch** ouit m over te schakelen naar **Aangepast** en selecteer vervolgens **Op uitdrukking** in de lijst. Voer de volgende uitdrukking in:

- `=If(Sum(ActualSales)>Sum(ProjectedSales),ARGB(128,0,128,0),ARGB(128,128,0,0))`, om groen te markeren wanneer de werkelijke verkoop hoger is dan de geschatte verkoop en rood wanneer de geschatte verkoop hoger is dan de werkelijke verkoop.

*Resultaten: staafdiagram van de verkoop per land*




We kunnen zien dat de kleur van de staaf groen is voor de landen met een verkoop die de geschatte waarde overschrijdt, terwijl de landen met een verkoop onder de geschatte waarde rood worden weergegeven. `Sweden` is bijvoorbeeld groen omdat de verkoopwaarde van `98,000` groter is dan de geschatte waarde van `50,000`.

## RGB

**RGB()** retourneert een geheel getal dat overeenkomst met de kleurcode van de kleur die is gedefinieerd door de drie parameters: de rode component r, de groene component g en de blauwe component b. Deze componenten moeten beschikken over gehele getalwaarden tussen 0 en 255. De functie kan worden gebruikt in uitdrukkingen voor het instellen of evalueren van de kleureigenschappen van een diagramobject.

**Syntaxis:**

```
RGB (r, g, b)
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **r**, **g**, **b** | De componentwaarden voor rood, groen en blauw. Een kleurcomponent van 0 komt overeen met geen bijdrage en een kleurcomponent van 255 met volledige bijdrage. |

*Alle argumenten moeten uitdrukkingen zijn die resulteren in een geheel getal in het bereik van 0 tot 255.*

Bij het interpreteren van de numerieke component en het opmaken hiervan in een hexadecimale notatie, zijn de waarden van de kleurcomponenten gemakkelijker te zien. Lichtgroen heeft bijvoorbeeld het getal 4 278 255 360. In hexadecimale notatie is dit FF00FF00. De eerste twee posities 'FF' (255) geven het kanaal **alpha** aan. In de functies **RGB** en **HSL** is dit altijd 'FF' (ondoorzichtig). De volgende twee posities '00' geven de hoeveelheid **red** aan, de volgende twee posities 'FF' geven de hoeveelheid **green** aan en de laatste twee posities 00' geven de hoeveelheid **blue** aan.

Voorbeeld: diagramuitdrukking

| Voorbeeld | Resultaten |
|---|---|
| `If(Sum(Sales)>Sum(Budget),RGB (255,0,0),RGB(100,80,120))` | Retourneert paars of rood, afhankelijk van de waarde van de vergelijking tussen `Sales` en `Budget`. |

## Voorbeeld - Basisprincipes voor RGB

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Product`
  - `Sales`

### Load-script

```
Example:
LOAD * Inline [
Product, Sales
Product A, 200
Product B, 500
Product C, 100
```

```
Product D, 300
Product E, 700
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Product`

Maak de volgende meting:

- `=Sum(Sales)`, om de som van verkoop te berekenen.

Voer in het deelvenster **Eigenschappen** de volgende uitdrukking in als de **Uitdrukking achtergrondkleur** voor de `Sum(Sales)`-meting: Hierdoor wordt de celkleur toegepast volgens de verkoopberekening voor het product.

- `=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 |

Volg in dit voorbeeld het volgende:

- Verkoopwaarden boven `400` zijn gemarkeerd met een groene achtergrond.
- Verkoopwaarden tussen `200` en `400` zijn gemarkeerd met een gele achtergrond.
- Alle andere verkoopwaarden zijn gemarkeerd met een rode achtergrond.

## Voorbeeld - Winstgegevens in een diagram markeren

Diagramuitdrukking

#### Overzicht

Een gegevensverzameling bevat verkoopcijfers per klant. We willen de gegevens analyseren en de winstmarges markeren naargelang ze laag, middelhoog of hoog zijn.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Customer`
  - `SalesValue`
  - `Cost`
  - `ProfitMargin`

### Load-script

```
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%
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Customer`

Maak de volgende metingen:

- `=Sum(SalesValue)`, om de som van verkoop te berekenen.
- `=Sum(Cost)`, om de totale waarde van de verkoopkosten te berekenen.
- `=Sum(ProfitMargin)`, om de winstmarge op te tellen. Om deze waarde als percentage weer te geven, selecteert u onder **GetalnotatieGetal > Eenvoudige opmaak > 12,34%**.

Voer in het deelvenster **Eigenschappen** de volgende uitdrukking in als de **Uitdrukking achtergrondkleur** voor de `Sum(ProfitMargin)`-meting:

```
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
  )
)
```

Schakel in het venster **Uiterlijk**, onder **Presentatie**, **Totalen** uit.

Results table

| Customer | Sum(SalesValue) | Sum(Cost) | Sum(Profit Margin) |
|---|---|---|---|
| Customer A | 500 | 300 | 40.00% |
| Customer B | 800 | 750 | 6.25% |

| Customer | Sum(SalesValue) | Sum(Cost) | Sum(Profit Margin) |
|---|---|---|---|
| Customer C | 400 | 300 | 25.00% |
| Customer D | 600 | 550 | 8.33% |
| Customer E | 900 | 700 | 22.22% |

De uitvoer van de functie RGB geeft de volgende kleuren weer volgens de winstmarge van de klant:

- Groen, `RGB(0, 128, 0)`, voor klanten met winstmarges hoger dan 30%.
- Oranje, `RGB(255, 165, 0)`, voor klanten met middelhoge winstmarges tussen 10% en 30%.
- Rood, `RGB(255, 0, 0)`, voor klanten met een winstmarge onder 10%.

## Voorbeeld – Scenario met RGB-diagram

Diagramuitdrukking

### Overzicht

Een gegevensverzameling bevat actuele en geschatte verkoopgegevens per land. Een verkoopmanager wil de gegevens in een staafdiagram weergeven en de verkoopcijfers markeren die hoger zijn dan de geschatte waarde voor de periode.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Country`
  - `ActualSales`
  - `ProjectedSales`

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe staafdiagram en voeg dit veld toe als dimensie:

- `Country`

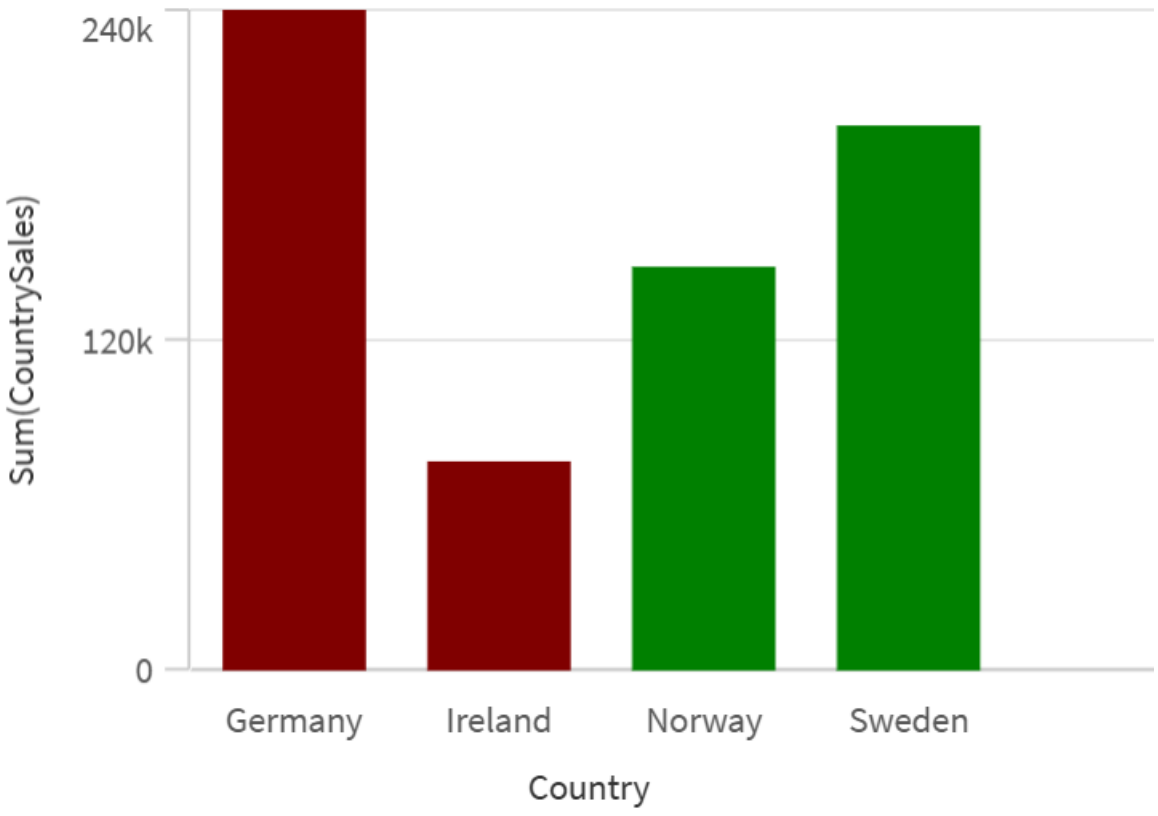
Maak de volgende meting:

- `=Sum(ActualSales)`, om de totale verkoopwaarde voor het land te berekenen.

Voeg een aangepaste kleuruitdrukking toe om de werkelijke verkoop met de geschatte verkoop te vergelijken. Schakel in het eigenschappenvenster onder **Uiterlijk** > **Kleuren en legenda** **Kleuren**: **Automatisch** ouit m over te schakelen naar **Aangepast** en selecteer vervolgens **Op uitdrukking** in de lijst. Voer de volgende uitdrukking in:

- `=If(Sum(ActualSales)>Sum(ProjectedSales),RGB(0,128,0),RGB(128,0,0))`, om groen te markeren wanneer de werkelijke verkoop hoger is dan de geschatte verkoop en rood wanneer de geschatte verkoop hoger is dan de werkelijke verkoop.

*Resultaten: staafdiagram van de verkoop per land*



We kunnen zien dat de kleur van de staaf groen is voor de landen met een verkoop die de geschatte waarde overschrijdt, terwijl de landen met een verkoop onder de geschatte waarde rood worden weergegeven. `Sweden` is bijvoorbeeld groen omdat de verkoopwaarde van `98,000` groter is dan de geschatte waarde van `50,000`.

## HSL

**HSL()** wordt gebruikt in uitdrukkingen om de kleureigenschappen van een diagramobject in te stellen of te evalueren, waarbij de kleur wordt gedefinieerd door waarden van **hue**, **saturation** en **luminosity** tussen 0 en 1.

**Syntaxis:**

```
HSL (hue, saturation, luminosity)
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **hue**, **saturation**, **luminosity** | De componentwaarden voor Hue, saturation en luminosity lopen uiteen van 0 tot 1. |

*Alle argumenten moeten uitdrukkingen zijn die resulteren in een geheel getal in het bereik van 0 tot 1.*

Bij het interpreteren van de numerieke component en het opmaken hiervan in een hexadecimale notatie, zijn de RGB -waarden van de kleurcomponenten gemakkelijker te zien. Lichtgroen heeft bijvoorbeeld het getal 4 278 255 360. In hexadecimale notatie is dit FF00FF00 en in RGB (0,255,0). Dit is equivalent aan HSL (80/240, 240/240, 120/240) - een HSL -waarde van (0.33, 1, 0.5).

Voorbeeld: diagramuitdrukking

| Voorbeeld | Resultaten |
|---|---|
| `If(Sum(Sales)>Sum(Budget),HSL( 0.33, 1, 0.25),HSL(0, 1, 0.5))` | Retourneert paars of rood, afhankelijk van de waarde van de vergelijking tussen `Sales` en `Budget`. |

### Voorbeeld - Basisprincipes voor HSL

Diagramuitdrukking

**Overzicht**

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Product`
  - `Sales`

### Load-script

```
Example:
LOAD * Inline [
Product, Sales
Product A, 200
Product B, 500
Product C, 100
Product D, 300
Product E, 700
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Product`

Maak de volgende meting:

- `=Sum(Sales)`, om de som van verkoop te berekenen.

Voer in het deelvenster **Eigenschappen** de volgende uitdrukking in als de **Uitdrukking achtergrondkleur** voor de `Sum(Sales)`-meting: Hierdoor wordt de celkleur toegepast volgens de verkoopberekening voor het product.

- `=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 |

Volg in dit voorbeeld het volgende:

- Verkoopwaarden boven `400`, zoals `Product B` en `E` zijn gemarkeerd met een groene achtergrond.
- Verkoopwaarden tussen `200` en `400`, zoals `Product A` en `D` zijn gemarkeerd met een gele

achtergrond.

- Alle andere verkoopwaarden zijn gemarkeerd met een rode achtergrond.

## Voorbeeld - Winstgegevens in een diagram markeren

Diagramuitdrukking

### Overzicht

Een gegevensverzameling bevat verkoopcijfers. We willen de gegevens analyseren en de winstmarges markeren naargelang ze laag, middelhoog of hoog zijn.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Customer`
  - `SalesValue`
  - `Cost`
  - `ProfitMargin`

### Load-script

```
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%
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Customer`

Maak de volgende metingen:

- `=Sum(SalesValue)`, om de som van verkoop te berekenen.
- `=Sum(Cost)`, om de totale waarde van de verkoopkosten te berekenen.
- `=Sum(ProfitMargin)`, om de winstmarge op te tellen. Om deze waarde als percentage weer te geven, selecteert u onder **GetalnotatieGetal > Eenvoudige opmaak > 12,34%**.

Voer in het deelvenster **Eigenschappen** de volgende uitdrukking in als de **Uitdrukking achtergrondkleur** voor de `Sum(ProfitMargin)`-meting:

```
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
  )
)
```

Schakel in het venster **Uiterlijk**, onder **Presentatie**, **Totalen** uit.

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% |

De uitvoer van de functie `HSL` geeft de volgende kleuren weer volgens de winstmarge van de klant:

- Groen, `HSL( 0.33, 1, 0.25)`, voor klanten met winstmarges hoger dan 30%.
- Oranje, `HSL( 0.11, 1, 0.5)`, voor klanten met middelhoge winstmarges tussen 10% en 30%.
- Rood, `HSL(0, 1, 0.5)`, voor klanten met een winstmarge onder 10%.

## Scenario met HSL-diagram

Diagramuitdrukking

**Overzicht**

Een gegevensverzameling bevat actuele en geschatte verkoopgegevens per land. Een verkoopmanager wil de gegevens in een staafdiagram weergeven en de verkoopcijfers markeren die hoger zijn dan de geschatte waarde voor de periode .

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Country`
  - `ActualSales`
  - `ProjectedSales`

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe staafdiagram en voeg dit veld toe als dimensie:

- `Country`

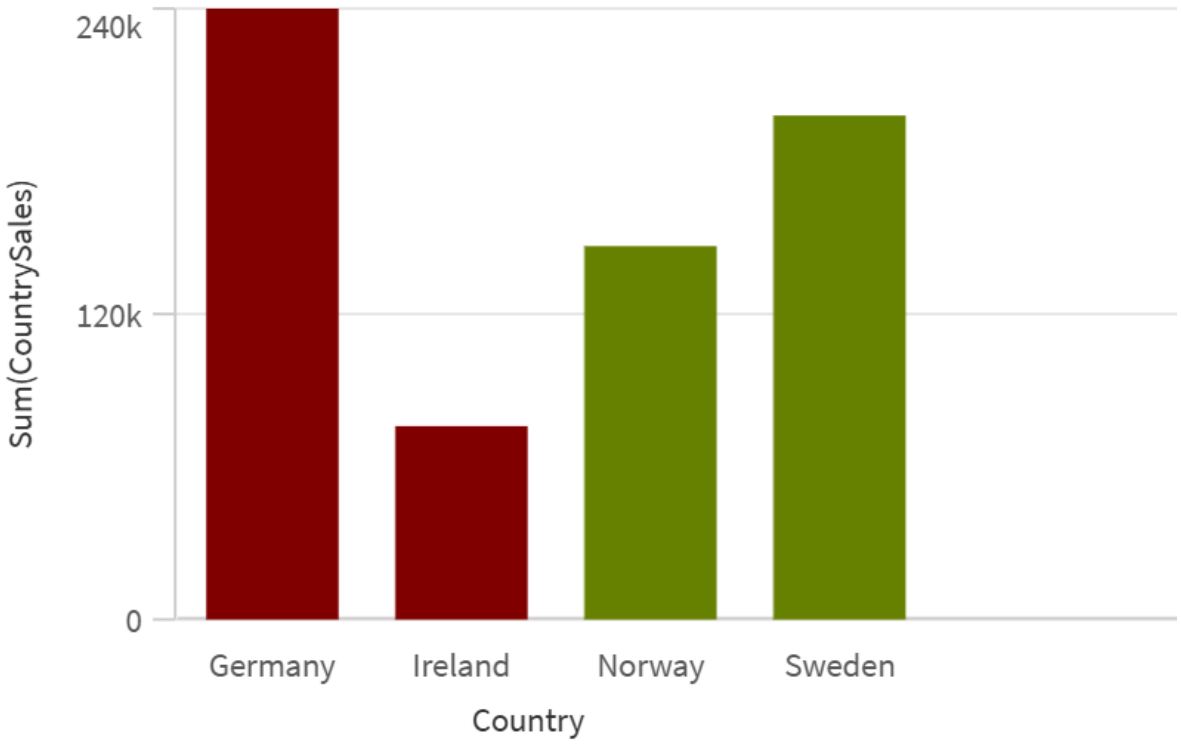
Maak de volgende meting:

- `=Sum(ActualSales)`, om de totale verkoopwaarde voor het land te berekenen.

Voeg een kleuruitdrukking toe om de werkelijke verkoop met de geschatte verkoop te vergelijken. Schakel in het eigenschappenvenster onder **Uiterlijk** > **Kleuren en legenda Kleuren**: **Automatisch** ouit m over te schakelen naar **Aangepast** en selecteer vervolgens **Op uitdrukking** in de lijst. Voer de volgende uitdrukking in:

- `=If(Sum(ActualSales)>Sum(ProjectedSales),HSL(1.2,1.0,0.25),HSL(0, 1.0, 0.25))`, om groen te markeren wanneer de werkelijke verkoop hoger is dan de geschatte verkoop en rood wanneer de geschatte verkoop hoger is dan de werkelijke verkoop.

*Resultaten: staafdiagram van de verkoop per land*




We kunnen zien dat de kleur van de staaf groen is voor de landen met een verkoop die de geschatte waarde overschrijdt, terwijl de landen met een verkoop onder de geschatte waarde rood worden weergegeven. `Sweden` is bijvoorbeeld groen omdat de verkoopwaarde van `98,000` groter is dan de geschatte waarde van `50,000`.

## 8.5 Voorwaardenfuncties

De voorwaardefuncties evalueren allemaal een voorwaarde en retourneren verschillende antwoorden, afhankelijk van de waarde van de voorwaarde. De functies kunnen zowel worden gebruikt in het load-script als in diagramuitdrukkingen.

### Overzicht voorwaardenfuncties

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

**alt**

De functie **alt** retourneert de eerste parameter waaraan een geldige numerieke waarde kan worden toegewezen. Als deze niet wordt gevonden, wordt de laatste parameter geretourneerd. U kunt een willekeurig aantal parameters gebruiken.

```
alt (expr1[ , expr2 , expr3 , ...] , else)
```

### class

De functie **class** wijst de eerste parameter toe aan een klasse-interval. Het resultaat is een duale waarde met a<=x<b als de tekstwaarde, waarbij a en b de boven- en ondergrens van het bereik zijn, en met de ondergrens als numerieke waarde.

```
class (expression, interval [ , label [ , offset ]])
```

### coalesce

De functie **coalesce** retourneert de eerste parameter waaraan een geldige non-NULL kan worden toegewezen. U kunt een willekeurig aantal parameters gebruiken.

```
coalesce(expr1[ , expr2 , expr3 , ...])
```

### if

De functie **if** retourneert een waarde afhankelijk van het feit of de opgegeven voorwaarde wordt geëvalueerd als True of False.

```
if (condition , then , else)
```

### match

Met de functie **match** vergelijkt u de eerste parameter met alle volgende parameters en krijgt u de numerieke locatie van de overeenkomende uitdrukkingen als resultaat. De vergelijking is hoofdlettergevoelig.

```
match ( str, expr1 [ , expr2,...exprN ])
```

### mixmatch

Met de functie **mixmatch** vergelijkt u de eerste parameter met alle volgende parameters en krijgt u de numerieke locatie van de overeenkomende uitdrukkingen als resultaat. De vergelijking is niet hoofdlettergevoelig en maakt geen onderscheid tussen de Japanse Hiragana- en Katakana-tekensystemen.

```
mixmatch ( str, expr1 [ , expr2,...exprN ])
```

### pick

De functie pick retourneert de *n*-de uitdrukking in de lijst.

```
pick (n, expr1[ , expr2,...exprN])
```

### wildmatch

Met de functie **wildmatch** vergelijkt u de eerste parameter met alle volgende parameters en krijgt u het aantal overeenkomende uitdrukkingen als resultaat. Bij deze functie kunt u jokertekens ( ***** en **?**) gebruiken in de tekenreeksen die worden vergeleken. ***** komt overeen met een willekeurige reeks tekens. **?** komt overeen met een willekeurig enkel teken. De vergelijking is niet hoofdlettergevoelig en maakt geen onderscheid tussen de Japanse Hiragana- en Katakana-tekensystemen.

```
wildmatch ( str, expr1 [ , expr2,...exprN ])
```

## alt

De functie **alt** retourneert de eerste parameter waaraan een geldige numerieke waarde kan worden toegewezen. Als deze niet wordt gevonden, wordt de laatste parameter geretourneerd. U kunt een willekeurig aantal parameters gebruiken.

**Syntaxis:**

```
alt(expr1[ , expr2 , expr3 , ...] , else)
```

**Retourgegevenstypen:** geheel getal

Argumenten

| Argument | Beschrijving |
|---|---|
| **expr1** | De eerste uitdrukking voor controle op een geldige getalsnotatie. |
| **expr2** | De tweede uitdrukking voor controle op een geldige getalsnotatie. |
| **expr3** | De derde uitdrukking voor controle op een geldige getalsnotatie. |
| **else** | Waarde die moet worden geretourneerd als geen van de voorafgaande parameters een geldige getalsnotatie heeft. |

De alt -functie wordt vaak gebruikt met interpretatiefuncties voor getallen of datums. Op die manier kan Qlik Sense verschillende datumnotaties in een voorkeursvolgorde testen. De functie kan tevens worden gebruikt voor het afhandelen van NULL-waarden in numerieke uitdrukkingen.

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
| --- | --- |
| `alt( date#( dat , 'YYYY/MM/DD' ),`<br><br>`date#( dat , 'MM/DD/YYYY' ),`<br><br>`date#( dat , 'MM/DD/YY' ),`<br><br>`'No valid date' )` | Deze uitdrukking test of het veld Datum een datum bevat volgens een van de drie opgegeven datumnotaties. Als dit het geval is, wordt een dubbele waarde met de oorspronkelijke tekenreeks en een geldige numerieke representatie van een datum weergegeven. Als geen geldige datum wordt gevonden, wordt de tekst `No valid date` geretourneerd (zonder geldige numerieke representatie). |
| `alt(Sales,0) + alt(Margin,0)` | Deze uitdrukking voegt de velden `Sales` en `Margin` toe, waarbij eventuele ontbrekende waarden (NULL) worden vervangen door een `0`. |

## Voorbeeld - Basisprincipes voor alt

Diagramuitdrukking

### Overzicht

Een gegevensverzameling bevat onvolledige verkoop- en prognosegegevens. U wilt ontbrekende waarden vervangen door geldige alternatieve waarden om de gaten op te vullen.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `ID`
  - `Sales`
  - `Forecast`

### Load-script

```
Example:
LOAD * INLINE [
ID, Sales, Forecast
1, 500, 450
2, , 300
3, 200, 250
4, , 400
5, 350,
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `ID`
- `Sales`
- `Forecast`

Maak de volgende meting:

- `=Sum(alt(Sales, Forecast))` om een geldige getalweergave te retourneren door ontbrekende waarden te vervangen door geldige alternatieve waarden.

Resultatentabel

| 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 |

De functie `alt` retourneert de waarde `Forecast` in de metingberekening voor de rijen `ID` met ontbrekende waarden van `Sales`. Door ontbrekende waarden te vervangen door geldige alternatieve waarden kunt u een volledig gevulde gegevensverzameling genereren.

## Voorbeeld - alt gebruiken om ontbrekende waarden te vervangen

Diagramuitdrukking

### Overzicht

Een bedrijf verzamelt temperatuurgegevens van meerdere sensoren. U wilt ontbrekende gegevens van de ene sensor vervangen door de gegevens van de andere sensor.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Time`
  - `Temp_Sensor1`
  - `Temp_Sensor2`

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `Time`
- `Temp_Sensor1`
- `Temp_Sensor2`

Maak de volgende meting:

- `=Sum(alt(Temp_Sensor1, Temp_Sensor2))` om een geldige getalrepresentatie terug te geven.

Resultatentabel

| Time | Temp_Sensor1 | Temp_Sensor2 | Sum(alt(Temp_Sensor1, Temp_Sensor2)) |
|---|---|---|---|
| **Totals** | | | **119.8** |
| 08:00 | 23.5 | 24.0 | 23.5 |
| 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 |

Als u naar de resultaten kijkt, heeft de functie `alt` ontbrekende sensorgegevens vervangen door geldige alternatieve resultaten van de andere sensor. Dit vult de gegevensverzameling aan door gaten op te vullen.

## Voorbeeld 5 – Scenario met alt

Diagramuitdrukking

**Overzicht**

Dit scenario toont verschillende manieren waarop u de functie `alt` kunt gebruiken om ontbrekende of null-waarden te vervangen.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `ProductID`
  - `ProductName`
  - `SalesAmount`
  - `ReturnAmount`
  - `Profit`
  - `Discount`
  - `Category`
  - `Year`

- Region
- CustomerRating

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- ProductID

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- =Coalesce( ProductName, 'Unknown Product'), om de beschrijving Unknown Product te geven voor alle producten die geen waarde hebben voor ProductName.

Maak de volgende metingen:

- =Sum(alt(SalesAmount, ReturnAmount, Profit)), om een geldig getal te retourneren als er geen SalesAmount wordt gevonden.
- =avg(alt(CustomerRating, 3)), om een geldige gestandaardiseerde klantwaardering te retourneren als er oorspronkelijk geen was opgegeven.
- =Sum(alt(Discount, SalesAmount * 0.05)), om een geldige korting te retourneren.
- =Sum(alt(Profit, SalesAmount * 0.1)), om een geldige winstwaarde te retourneren.
- =Sum(alt(SalesAmount, avg(TOTAL SalesAmount))), om een geldige verkoopwaarde te retourneren.
- =Sum(alt(Discount, avg(TOTAL Discount))), om een geldige kortingswaarde te retourneren.

Resultatentabel

| 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))) |
|---|---|---|---|---|---|---|---|
| **Totalen** | | **6200** | **3.6** | **310** | **2030** | **7500** | **410** |
| 101 | Laptop | 1200 | 3 | 60 | 500 | 1200 | 41 |
| 102 | Tablet | 800 | 4 | 50 | 300 | 500 | 50 |
| 103 | Smartphone | 1500 | 5 | 100 | 600 | 800 | 100 |
| 104 | Printer | 400 | 3 | 20 | 40 | 400 | 20 |
| 105 | Onbekend product | 1000 | 4 | 30 | 100 | 1000 | 30 |
| 106 | Bewaken | 150 | 5 | 0 | 250 | 750 | 41 |
| 107 | Toetsenbord | 200 | 2 | 5 | 80 | 200 | 5 |
| 108 | Muis | 50 | 3 | 0 | 0 | 750 | 41 |
| 109 | Onbekend product | 600 | 4 | 30 | 60 | 600 | 41 |
| 110 | Hoofdtelefoon | 300 | 3 | 15 | 100 | 300 | 41 |

De functie `alt` zorgt ervoor dat wanneer waarden ontbreken, alternatieve waarden kunnen worden berekend en geretourneerd. Bekijk de volgende resultaten:

- Kolom 2 gebruikt de functie `Coalesce` om de waarde `Unknown Product` te retourneren voor alle null-waarden. De functie `Alt` kan niet worden gebruikt omdat deze alleen van toepassing is op numerieke waarden.
- Kolom 3 retourneert de waarde voor `SalesAmount`. Als er geen waarde wordt gevonden, wordt de waarde `ReturnAmount` geretourneerd. Als er geen waarde voor `ReturnAmount` wordt gevonden, wordt de waarde `Profit` geretourneerd. Dit helpt ervoor te zorgen dat de kolom wordt gevuld met een waarde als `SalesAmount` geen waarde heeft.
- Kolom 4 retourneert de `CustomerRating`. Als er geen waarde wordt gevonden, wordt standaard `3` gebruikt. Het product `Laptop` heeft bijvoorbeeld geen klantbeoordeling, dus wordt `3` geretourneerd. De meting berekent vervolgens het gemiddelde voor de kolom.
- Kolom 5 retourneert de kortingswaarde. Als er geen waarde beschikbaar is, wordt deze berekend als 5% van de waarde voor `SalesAmount`.

- Kolom 6 retourneert de waarde voor `Profit`. Als er geen waarde beschikbaar is, wordt 10% van de `SalesAmount` geretourneerd.
- Kolom 7 retourneert de `SalesAmount`. Als er geen waarde wordt opgegeven, wordt de gemiddelde `SalesAmount` van de gehele gegevensverzameling geretourneerd. De producten `Monitor` en `Mouse` hebben bijvoorbeeld geen waarde voor `SalesAmount`, dus beide retourneren `750`, wat het gemiddelde is van alle waarden voor `SalesAmount`.
- Kolom 8 retourneert de waarde voor `Discount`. Als er geen waarde wordt gevonden, berekent het systeem de gemiddelde korting van de hele gegevensverzameling en retourneert deze.

## class

De functie **class** wijst de eerste parameter toe aan een klasse-interval. Het resultaat is een duale waarde met a<=x<b als de tekstwaarde, waarbij a en b de boven- en ondergrens van het bereik zijn, en met de ondergrens als numerieke waarde.

**Syntaxis:**

```
class(expression, interval [ , label [ , offset ]])
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **interval** | Een getal dat de bin-breedte aanduidt. |
| **label** | Een willekeurige tekenreeks die de 'x' in de resultaattekst kan vervangen. |
| **offset** | Een getal die kan worden gebruikt als verschil met het standaardbeginpunt van de classificatie. Het standaardbeginpunt is meestal 0. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `class( var,10 )` met `var = 23` | Retourneert `20<=x<30` |
| `class( var,5,'value' )` met `var = 23` | Retourneert `20<= value <25` |
| `class( var,10,'x',5 )` met `var = 23` | Retourneert `15<=x<25` |

### Voorbeeld - Basisprincipes voor class

Diagramuitdrukking

**Overzicht**

Er wordt een gegevenstabel geladen die namen en leeftijden bevat. U wilt de mensen indelen per leeftijdsgroep (met een interval van tien jaar) en een telling geven van het aantal mensen in elke groep.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Age`
  - `Name`

### Load-script

```
Example:
LOAD * INLINE [
Age, Name
25, John
42, Karen
53, Yoshi
43, Bob
27, Charles
];
```

### Resultaten

1. Laad de gegevens en open een werkblad. Maak twee nieuwe tabellen.
2. Doe het volgende in de eerste tabel:
   a. Voeg deze dimensies toe:
      - `Age`
      - `Name`
   b. Maak de volgende meting:
      - `=class(Age, 10, 'age')`, om de gegevens in leeftijdscategorieën in te delen.
3. Doe het volgende in de tweede tabel:
   a. Voeg deze uitdrukking toe als dimensie
      - `=class(Age, 10, 'age')`
   b. Maak de volgende meting:
      - `=Count(Age)`, om het aantal mensen met een leeftijdsclassificatie te tellen.

De eerste tabel toont de onbewerkte gegevens voor `Name` en `Age`, en in welke leeftijdscategorie ze zijn ingedeeld.

Resultaten tabel 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 |

In de tweede tabel kunt u zien hoe de gegevens nu zijn geclassificeerd en samengevat. De resultaten van de functie `class` creëerden 3 leeftijdscategorieën in de eerste meting. De tweede meting telt het aantal mensen in elke leeftijdscategorie.

Resultaten tabel 2

| class(Age, 10, 'age') | Count(Age) |
|---|---|
| **Totals** | **5** |
| 20 <= age < 30 | 2 |
| 40 <= age < 50 | 2 |
| 50 <= age < 60 | 1 |

Load-script

### Overzicht

Er wordt een gegevenstabel geladen die namen en leeftijden bevat. U wilt een veld toevoegen dat elke persoon classificeert op basis van een leeftijdsgroep met een interval van tien jaar. De originele brontabel ziet er als volgt uit.

Oorspronkelijke brontabel

| Name | Age |
|---|---|
| John | 25 |
| Karen | 42 |
| Yoshi | 53 |

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een voorafgaande lading opdracht die de **class** functie gebruikt om een nieuw veld genaamd `Agegroup` aan te maken.

### Load-script

```
LOAD *,
class(Age, 10, 'age') As Agegroup;

LOAD * INLINE
[ Age, Name
25, John
42, Karen
53, Yoshi];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `Name`
- `Age`
- `Agegroup`

Resultatentabel

| Name | Age | Agegroup |
|---|---|---|
| John | 25 | 20 <= age < 30 |
| Karen | 42 | 40 <= age < 50 |
| Yoshi | 53 | 50 <= age < 60 |

Het nieuwe veld `Agegroup` classificeert elke persoon per leeftijdscategorie.

## Voorbeeld 5 – Scenario met class

#### Diagramuitdrukking

#### Overzicht

Er wordt een gegevenstabel geladen die de klantgegevens bevat. U wilt klanten classificeren op basis van het aantal verdiende lifetime value-punten en een telling geven van het aantal klanten in elke categorie.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `CustomerID`
  - `CustomerName`
  - `LifetimeValue`

#### Load-script

```
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
];
```

**Resultaten**

1. Laad de gegevens en open een werkblad. Maak twee nieuwe tabellen.
2. Doe het volgende in de eerste tabel:
    a. Voeg deze dimensies toe:
        - `CustomerID`
        - `CustomerName`
        - `LifetimeValue`
    b. Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:
        - `=Class(LifetimeValue, 10000, 'Lifetime Value Group')`, om de gegevens te classificeren op basis van het veld `LifetimeValue`.
3. Doe het volgende in de tweede tabel:
    a. Voeg deze uitdrukking toe als dimensie
        - `=Class(LifetimeValue, 10000, 'Lifetime Value Group')`
    b. Maak de volgende meting:
        - `=Count(CustomerID)`, om het aantal mensen met een leeftijdsclassificatie te tellen.

De eerste tabel toont de onbewerkte gegevens die zijn ingevoerd en de resultaten van het gebruik van de klassefunctie.

Resultaten tabel 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 |

De tweede tabel toont hoe de gegevens geclassificeerd en samengevat kunnen worden. De gegevens worden in vier groepen ingedeeld met de functie `class`, en vervolgens geteld om het aantal klanten in elke groep weer te geven.

Resultaten tabel 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

De functie **coalesce** retourneert de eerste parameter waaraan een geldige non-NULL kan worden toegewezen. U kunt een willekeurig aantal parameters gebruiken.

**Syntaxis:**

```
coalesce(expr1[ , expr2 , expr3 , ...])
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **expr1** | De eerste uitdrukking voor controle op een geldige niet-null getalsnotatie. |
| **expr2** | De tweede uitdrukking voor controle op een geldige niet-null getalsnotatie. |
| **expr3** | De derde uitdrukking voor controle op een geldige niet-null getalsnotatie. |

Voorbeelden: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `Coalesce(ProductDescription, ProductName, ProductCode, 'no description available')` | Deze uitdrukking selecteert drie verschillende velden met een productomschrijving als sommige velden geen waarden voor het product bevatten. Volgt de volgorde die in de functie wordt vermeld, en het eerste veld met een niet-null waarde wordt geretourneerd. Als geen enkel veld een waarde bevat, is het resultaat `no description available`. |

| Voorbeeld | Resultaat |
| --- | --- |
| `Coalesce(TextBetween(FileName, '"', '"'), FileName)` | Deze uitdrukking zorgt ervoor dat de tekst tussen aanhalingstekens van het veld `FileName` wordt ingekort. Als de opgegeven `FileName` tussen aanhalingstekens staat, wordt deze verwijderd en wordt de ingesloten `FileName` zonder aanhalingstekens geretourneerd. Als de functie `TextBetween` de scheidingstekens niet kan vinden, wordt null geretourneerd. De **Coalesce** zal deze afwijzen en in plaats daarvan de onbewerkte `FileName` retourneren. |

## Voorbeeld - Basisprincipes voor Coalesce

Diagramuitdrukking

### Overzicht

Er wordt een gegevenstabel geladen die de product- en prijsgegevens bevat. U wilt ervoor zorgen dat eventuele null-waarden worden vervangen door een alternatieve waarde.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `ProductID`
  - `ProductName`
  - `Price`
- Een `NullInterpret` variabele opdracht om ervoor te zorgen dat ontbrekende waarden als NULL worden behandeld.

### Load-script

```
Set NullInterpret = '';
Example:
LOAD * INLINE [
```

```
ProductID, ProductName, Price
1, Laptop, 1000
2, , 800
3, Desktop, 1200
4, , 500
5, Tablet, 600
];
```

**Resultaten**

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:
    - `ProductID`
    - `ProductName`
2. Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:
    - `=Coalesce(ProductName, 'Unknown Product')`
3. Maak de volgende meting:
    - `=Sum(Price)`, om een geldige getalrepresentatie te retourneren.
4. Maak een nieuwe tabel en voeg de volgende uitdrukking toe als een berekende dimensie:
    - `=Coalesce(ProductName, 'Unknown Product')`
5. Maak de volgende meting:
    - `=Sum(Price)`, om een geldige getalrepresentatie te retourneren.

De resultaten in de volgende tabellen laten zien hoe de functie `Coalesce` de null-waarden in de oorspronkelijke gegevensverzameling kan vervangen door de alternatieve waarde `Unknown Product`.

Resultaten tabel 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 |

Resultaten tabel 2

| Coalesce(ProductName, 'Unknown Product') | Sum(Price) |
|---|---|
| **Totals** | **5** |
| Desktop | 600 |
| Laptop | 1000 |
| Tablet | 600 |
| Unknown Product | 1300 |

## Voorbeeld - Null-waarden vervangen door een alternatieve waarde met prioriteit

Diagramuitdrukking

### Overzicht

Dit voorbeeld gebruikt de functie `Coalesce` om ontbrekende locatiegegevens te vervangen door een alternatieve waarde volgens de opgegeven prioriteitslogica.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `LocationID`
  - `Country`
  - `State`
  - `City`
- Een `NullInterpret` variabele opdracht om ervoor te zorgen dat ontbrekende waarden als NULL worden behandeld.

### Load-script

```
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
];
```

### Resultaten

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:
   - `LocationID`
   - `Country`
   - `State`
   - `City`
2. Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:
   - `=Coalesce(Country, State, City)`, om eventuele null-waarden aan te vullen met alternatieven.

Resultatentabel

| 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 |

De resultaten laten zien dat er weliswaar ontbrekende waarden zijn, maar dat u de functie `Coalesce` kunt gebruiken om logica te verschaffen over hoe deze exemplaren moeten worden gevuld met een alternatieve waarde om de gegevensverzameling te voltooien. Bijvoorbeeld, `L004` mist een waarde voor `Country`. De functie `Coalesce` retourneert `New South Wales` omdat `State` het tweede veld is dat in de prioriteitslogica wordt vermeld om te gebruiken wanneer de waarde voor `Country` ontbreekt.

## if

De functie **if** retourneert een waarde afhankelijk van het feit of de opgegeven voorwaarde wordt geëvalueerd als True of False.

**Syntaxis:**

```
if(condition , then [, else])
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **condition** | Uitdrukking die logisch wordt geïnterpreteerd. |
| **then** | Uitdrukking die van elk willekeurig type kan zijn. Als de *condition* True is, retourneert de functie if de waarde van de uitdrukking *then*. |
| **else** | Uitdrukking die van elk willekeurig type kan zijn. Als de *condition* False is, retourneert de functie if de waarde van de uitdrukking *else*.<br>Deze parameter is optioneel. Als de *condition* False is, wordt NULL geretourneerd als u else niet hebt opgegeven. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `if( Amount>= 0, 'OK', 'Alarm' )` | Deze uitdrukking test of het aantal een positief getal (0 of groter) is en retourneert `OK` als dit het geval is. Als het aantal kleiner is dan 0, wordt `Alarm` geretourneerd. |

## Voorbeeld - Load-script dat gebruikmaakt van if

Load-script

### Overzicht

If kan worden gebruikt in een load-script met andere methoden en objecten, waaronder variabelen. Dit voorbeeld gebruikt de variabele `threshold` en neemt een nieuw veld op in het gegevensmodel dat gebaseerd is op de drempel.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in twee tabellen met de naam:
  - `Transactions`: bevat transactievelden
  - `Transaction_Buckets`: bevat een veld `transaction_id` en maakt een nieuw veld, `[Compared to Threshold]`, met behulp van de functie `if` om het veld `transaction_amount` te vergelijken met de variabele instelling `threshold`.
- Een variabele opdracht `threshold` is ingesteld op `100`.

### Load-script

```
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;
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `transaction_id`
- `[Compared to Threshold]`

Resultatentabel met de uitvoer van het gebruik van de functie *if* in het load-script.

| transaction_id | Vergeleken met drempelwaarde |
|---|---|
| 3750 | Kleiner dan 100 |
| 3751 | Groter dan 100 |
| 3752 | Kleiner dan 100 |
| 3753 | Groter dan 100 |
| 3754 | Groter dan 100 |
| 3756 | Kleiner dan 100 |
| 3757 | Groter dan 100 |

## Voorbeeld - Productgegevens evalueren met if

Diagramuitdrukking

### Overzicht

Een gegevensverzameling bevat productgegevens. U wilt producten identificeren en categoriseren op basis van of ze consumeerbaar zijn of niet.

### Load-script

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `ProductID`
  - `ProductName`
  - `ProductCategory`

```
Example:
LOAD * INLINE [
ProductID, ProductName, ProductCategory
1, "Laptop", "Electronics"
```

```
2, "Banana", "Fruits"
3, "Shampoo", "Toiletries"
4, "TV", "Electronics"
5, "Apple", "Fruits"
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `ProductID`
- `ProductName`
- `ProductCategory`

Maak de volgende meting:

- `=If(ProductCategory = 'Toiletries' or ProductCategory = 'Fruits', 'Consumable', 'Other')`, om de vereiste waarden te retourneren.

Resultatentabel

| 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 |

U kunt aan de resultaten zien dat de functie `if` de uitdrukking evalueert en de vereiste waarden retourneert. Producten in de categorie `Fruits` of `Toiletries` worden bijvoorbeeld geclassificeerd als `Consumable`.

## Voorbeeld - Klantgegevens evalueren en classificeren met behulp van if

Diagramuitdrukking

#### Overzicht

Een gegevensverzameling bevat klantgegevens. U wilt de klanten identificeren en categoriseren waarvan de regio en industriecategorie aan uw classificatievereisten voldoen.

### Load-script

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
    - `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"
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `ClientID`
- `ClientName`
- `Region`
- `Industry`

Maak de volgende meting:

- `=If((Region = 'North' or Region = 'East') AND (Industry= 'Technology' or Industry = 'Manufacturing'), 'Target Segment', 'Secondary Segment')`, om de vereiste waarde te retourneren.

Resultatentabel

| 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 | Sound | Healthcare | Secondary Segment |

U kunt aan de resultaten zien dat de functie `if` de uitdrukking evalueert en de vereiste waarden retourneert. Bijvoorbeeld, `Acme Corp` is geclassificeerd als `Target Segment` omdat het voldoet aan de vereisten om deel uit te maken van de `North` of `East` regio en de `Manufacturing` of `Technology` industrie.

## Voorbeeld - Incidentgegevens evalueren met if

Diagramuitdrukking

**Overzicht**

Gebruik de functie `if` om een gegevensverzameling van incidentrapporten te categoriseren.

**Load-script**

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `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];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- Datum
- Locatie
- Incidenten

Maak de volgende berekende dimensies:

- `=if(Incidents>=10, 'Critical', 'Ok' )`
- `=if(Incidents>=10, 'Critical', If( Incidents>=1 and Incidents<10, 'Warning', 'Ok'))`

Resultatentabel

| 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 |

## Voorbeeld - Datums categoriseren met if

Diagramuitdrukking

#### Overzicht

Gebruik de functie `if` om datums te categoriseren.

#### Load-script

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

```
SET FirstWeekDay=0;
Load
Date(MakeDate(2022)+RecNo()-1) as Date
Autogenerate 14;
```

**Resultaten**

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Date`

Maak de volgende berekende dimensies:

- `WeekDay(Date)`
- `=If(WeekDay(Date)>=5,'WeekEnd','Normal Day')`

Resultatentabel

| Date | WeekDay(Date) | If(WeekDay (Date)>=5,'WeekEnd','Normal 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

Met de functie **match** vergelijkt u de eerste parameter met alle volgende parameters en krijgt u de numerieke locatie van de overeenkomende uitdrukkingen als resultaat. De vergelijking is hoofdlettergevoelig.

**Syntaxis:**

```
match( str, expr1 [ , expr2,...exprN ])
```

**Retourgegevenstypen:** geheel getal

*Als u een vergelijking wilt gebruiken waarbij geen onderscheid wordt gemaakt tussen hoofdletters en kleine letters, gebruikt u de functie **mixmatch**. Als u een vergelijking wilt gebruiken met jokertekens en waarbij geen onderscheid wordt gemaakt tussen hoofdletters en kleine letters, gebruikt u de functie **wildmatch**.*

Argumenten

| Argument | Beschrijving |
|---|---|
| **str** | De tekenreeks die moet worden geëvalueerd. |
| **expr1** | De eerste uitdrukking voor controle op een geldige getalsnotatie. |
| **expr2** | De tweede uitdrukking voor controle op een geldige getalsnotatie. |
| **expr3** | De derde uitdrukking voor controle op een geldige getalsnotatie. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `match( ColorCode,'Blue','Black' )` | Deze uitdrukking retourneert de index van de overeenkomst met het veld `ColorCode`. Als de waarde van `ColorCode` is `Blue`, dan wordt `1` geretourneerd. Als de waarde van `ColorCode` is `Black`, dan wordt `2` geretourneerd. Als geen waarde wordt gevonden, wordt `0` geretourneerd. |
| `match( Region, 'North', 'East' )` | Deze uitdrukking retourneert de index van de overeenkomst met het veld `Region`. Als de waarde van `Region` is `North`, dan wordt `1` geretourneerd. Als de waarde van `Region` is `East`, dan wordt `2` geretourneerd. Als geen waarde wordt gevonden, wordt `0` geretourneerd. |

## Voorbeeld - Producten categoriseren met match

Diagramuitdrukking

**Overzicht**

Een gegevensverzameling bevat productgegevens. U wilt producten identificeren en categoriseren die consumeerbaar zijn.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
    - `ProductID`
    - `ProductName`
    - `ProductCategory`

### Load-script

```
Example:
LOAD * INLINE [
ProductID, ProductName, ProductCategory
1, "Laptop", "Electronics"
2, "Banana", "Fruits"
3, "Shampoo", "Toiletries"
4, "TV", "Electronics"
5, "Apple", "Fruits"
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `ProductID`
- `ProductName`
- `ProductCategory`

Maak de volgende metingen:

- `=match(ProductCategory,'Toiletries', 'Fruits')`, om de indexatie te retourneren waarbij `ProductCategory` de waarde `Toiletries` of `Fruits` heeft.
- `=If(match(ProductCategory, 'Toiletries', 'Fruits'), 'Consumable', 'Other')`, om de indexatie te retourneren waarbij `ProductCategory` de waarde `Toiletries` of `Fruits` heeft. Als een van beide wordt gevonden, dan wordt `Consumable` geretourneerd. Anders wordt `Other` geretourneerd.

Resultatentabel

| ProductID | ProductName | ProductCategory | match (ProductCategory,'Toiletries', 'Fruits') | If(match (ProductCategory, 'Toiletries', 'Fruits'), 'Consumable', 'Other') |
|---|---|---|---|---|
| 1 | Laptop | Electronics | 0 | Other |
| 2 | Banana | Fruits | 2 | Consumable |
| 3 | Shampoo | Toiletries | 1 | Consumable |

| ProductID | ProductName | ProductCategory | match (ProductCategory,'Toiletries', 'Fruits') | If(match (ProductCategory, 'Toiletries', 'Fruits'), 'Consumable', 'Other') |
|---|---|---|---|---|
| 4 | TV | Electronics | 0 | Other |
| 5 | Apple | Fruits | 2 | Consumable |

De functie `match` retourneert de indexatie van de zoekreeks in de eerste meting. In de tweede meting wordt dit resultaat vervolgens gebruikt in een `If` functie om producten te categoriseren als `Consumable` of `Other`.

## Voorbeeld - Klantgegevens evalueren en classificeren met behulp van match

Diagramuitdrukking

### Overzicht

Een gegevensverzameling bevat klantgegevens. U wilt de klanten identificeren en categoriseren waarvan de regio en industriecategorie aan uw classificatievereisten voldoen.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `ClientID`
  - `ClientName`
  - `Region`
  - `Industry`

### Load-script

```
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"
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `ClientID`
- `ClientName`
- `Region`
- `Industry`

Maak de volgende meting:

`=IF(match(Region, 'North', 'East') AND match(Industry, 'Technology', 'Manufacturing'), 'Target Segment', 'Secondary Segment')`, om de indexatie van de zoekreeks te retourneren. Als deze wordt gevonden, dan wordt `Target Segment` geretourneerd. Als deze niet wordt gevonden, dan wordt `Secondary Segment` geretourneerd.

Resultatentabel

| 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 |

De functie `match` zoekt naar tekst en stuurt een indexatie terug die vervolgens kan worden gebruikt om aanvullende analyses en gegevensclassificatie uit te voeren. Bijvoorbeeld, `Acme Corp` wordt geclassificeerd als een `Target Segment` omdat het voldoet aan de zoekvereisten van de `match` functie voor zowel het veld `Region` als `Industry`.

## Voorbeeld - Load-script met gebruik van match

Load-script

### Overzicht

U kunt match gebruiken om een gegevens-subset te laden. U kunt bijvoorbeeld een numerieke waarde als resultaat krijgen voor een uitdrukking in de functie. Vervolgens kunt u de gegevens beperken die zijn

geladen op basis van de numerieke waarde. De functie Match retourneert 0 als er geen overeenkomst is. Alle uitdrukkingen zonder overeenkomst retourneren in dit voorbeeld daarom 0 en worden uitgesloten van de gegevens die zijn geladen door de WHERE-instructie.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in twee tabellen met de naam:
  - Transactions: bevat transactievelden
  - Transaction_Buckets: maakt twee nieuwe velden, Customer en Color Code Blue and Black. De where-instructie laadt de resultaten van de functie match.

### Load-script

```
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;
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `[Color Code Blue and Black]`
- `Customer`

Resultatentabel met de uitvoer van het gebruik van de functie match in het load-script.

| Color Code Blue and Black | Customer |
|---|---|
| Black | 203521 |
| Black | 3036491 |
| Blue | 2038593 |

## Voorbeeld - Diagramuitdrukking met gebruik van match

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Cities`
  - `Count`

### Load-script

```
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];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Cities`

Maak de volgende metingen:

- `=match( Cities,'Toronto','Boston','Beijing','Zurich')`, om de indexatie van de zoekreeks te retourneren.
- `=match( Cities,'Toronto','Boston','Beijing','Stockholm','zurich')`, om de indexatie van de zoekreeks te retourneren.
- `=Cities & ' - ' & match ( Cities, 'Toronto','Boston', 'Beijing','Stockholm','zurich')`, om het veld `Cities` en de indexatie van de zoekreeks te retourneren.

De eerste meting in de onderstaande tabel retourneert `0` voor `Stockholm`, aangezien `Stockholm` niet is opgenomen in de lijst met uitdrukkingen in de functie `match`. Het retourneert ook `0` voor `Zurich` omdat de vergelijking `match` hoofdlettergevoelig is.

Resultatentabel

| 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 |

## Voorbeeld - Aangepaste sorteeruitdrukking met gebruik van match

### Diagramuitdrukking

### Overzicht

U kunt `match` gebruiken om een uitdrukking op aangepaste wijze te sorteren. Dit voorbeeld gebruikt dezelfde gegevensverzameling als het vorige voorbeeld.

Kolommen worden standaard numeriek of alfabetisch gesorteerd, afhankelijk van de gegevens.

Resultatentabel toont de standaardsorteervolgorde

| Cities |
| --- |
| Beijing |
| Boston |
| Stockholm |
| Toronto |
| zurich |

Doe het volgende om de sorteervolgorde te wijzigen:

1. Open de sectie **Sorteren** in het **eigenschappenvenster**.
2. Schakel automatisch sorteren uit voor de kolom die u aangepast wilt sorteren.
3. Schakel de selectievakjes **Numeriek sorteren** en **Alfabetisch sorteren** uit.
4. Selecteer **Sorteren op uitdrukking** en voer een uitdrukking in die hierop lijkt:
   `=match( Cities, 'Toronto','Boston','Beijing','Stockholm','zurich')`
   De sorteervolgorde van de kolom Cities wordt gewijzigd.

Resultatentabel toont de sorteervolgorde met de functie *match*.

| Cities |
| --- |
| Toronto |
| Boston |
| Beijing |
| Stockholm |
| zurich |

## mixmatch

Met de functie **mixmatch** vergelijkt u de eerste parameter met alle volgende parameters en krijgt u de numerieke locatie van de overeenkomende uitdrukkingen als resultaat. De vergelijking is niet hoofdlettergevoelig en maakt geen onderscheid tussen de Japanse Hiragana- en Katakana-tekensystemen.

**Syntaxis:**

```
mixmatch( str, expr1 [ , expr2,...exprN ])
```

**Retourgegevenstypen:** geheel getal

Als u een vergelijking wilt gebruiken waarbij onderscheid wordt gemaakt tussen hoofdletters en kleine letters, gebruikt u de functie **match**. Als u een vergelijking wilt gebruiken met jokertekens en waarbij geen onderscheid wordt gemaakt tussen hoofdletters en kleine letters, gebruikt u de functie **wildmatch**.

Argumenten

| Argument | Beschrijving |
|---|---|
| **str** | De tekenreeks die moet worden geëvalueerd. |
| **expr1** | De eerste uitdrukking voor controle op een geldige getalsnotatie. |
| **expr2** | De tweede uitdrukking voor controle op een geldige getalsnotatie. |
| **expr3** | De derde uitdrukking voor controle op een geldige getalsnotatie. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `mixmatch( ColorCode,'black','Blue' )` | Deze uitdrukking retourneert `1` als de waarde van `ColorCode` `Black` is of `2` als de waarde van `ColorCode` `Blue` is. |
| `mixmatch( Cities,'toronto','Boston','Beijing','Zurich' )` | Deze uitdrukking retourneert de indexatie van het item dat overeenkomt met de waarde in het veld `Cities`. |

## Voorbeeld - Producten categoriseren met mixmatch

Diagramuitdrukking

### Overzicht

Een gegevensverzameling bevat productgegevens. U wilt producten identificeren en categoriseren die consumeerbaar zijn.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `ProductID`
  - `ProductName`
  - `ProductCategory`

### Load-script

```
Example:
LOAD * INLINE [
ProductID, ProductName, ProductCategory
1, "Laptop", "Electronics"
2, "Banana", "Fruits"
3, "Shampoo", "toiletries"
4, "TV", "Electronics"
```

```
5, "Apple", "fruits"
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `ProductID`
- `ProductName`
- `ProductCategory`

Maak de volgende metingen:

- `=mixmatch(ProductCategory,'Toiletries', 'Fruits')`, om de indexatie te retourneren waarbij `ProductCategory` de waarde `Toiletries` of `Fruits` heeft.
- `=If(mixmatch(ProductCategory, 'Toiletries', 'Fruits'), 'Consumable', 'Other')`, om een overeenkomst van `ProductCategory` met `Toiletries` of `Fruits` te vertalen als `Consumable`. Als geen waarde wordt gevonden, wordt `Other` geretourneerd.

*De functie `mixmatch` retourneert een positieve waarde (niet nul) als er een overeenkomst is gevonden, anders nul als er geen overeenkomst is gevonden. De `if` instructie interpreteert op zijn beurt resultaten die niet nul zijn als `true` en resultaten die nul zijn als `false`. Door een `if` instructie te combineren met `mixmatch`, `wildmatch` of `match` functies, is het mogelijk om gemakkelijk overeenkomsten en mismatches te decoderen.*

Resultatentabel

| ProductID | ProductName | ProductCategory | mixmatch (ProductCategory,'Toiletries', 'Fruits') | If(mixmatch (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 |

De functie `mixmatch` retourneert de indexatie van de zoekreeks in de eerste meting. In de tweede meting wordt dit resultaat vervolgens gebruikt in een `If` functie om aangepaste resultaten te bieden. Let op dat de functie `mixmatch` hoofdletterongevoelig is.

## Voorbeeld - Klantgegevens evalueren en classificeren met behulp van mixmatch

Diagramuitdrukking

### Overzicht

Een gegevensverzameling bevat klantgegevens. U wilt de klanten identificeren en categoriseren waarvan de regio en industriecategorie aan uw classificatievereisten voldoen.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `ClientID`
  - `ClientName`
  - `Region`
  - `Industry`

### Load-script

```
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"
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `ClientID`
- `ClientName`
- `Region`
- `Industry`

Maak de volgende meting:

`=If(mixmatch(Region, 'North', 'East') AND mixmatch(Industry, 'Technology', 'Manufacturing'), 'Target Segment', 'Secondary Segment')`, om de indexatie van de zoekreeks te retourneren. Als deze wordt gevonden, dan wordt `Target Segment` geretourneerd. Als deze niet wordt gevonden, dan wordt `Secondary Segment` geretourneerd.

Resultatentabel

| ClientID | ClientName | Region | Industry | If(mixmatch (Region, 'North', 'East') AND mixmatch (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 |

De functie `mixmatch` zoekt naar tekst en stuurt een indexatie terug die vervolgens kan worden gebruikt om aanvullende analyses en gegevensclassificatie uit te voeren. Bijvoorbeeld, `Acme Corp` wordt geclassificeerd als een `Target Segment` omdat het voldoet aan de zoekvereisten van de `mixmatch` functie voor zowel het veld `Region` als `Industry`. Let op dat de functie `mixmatch` hoofdletterongevoelig is.

## Voorbeeld - Load-script dat gebruikmaakt van mixmatch

Load-script

**Overzicht**

U kunt mixmatch gebruiken om een gegevens-subset te laden. U kunt bijvoorbeeld een numerieke waarde als resultaat krijgen voor een uitdrukking in de functie. Vervolgens kunt u de gegevens beperken die zijn geladen op basis van de numerieke waarde. Mixmatch geeft 0 als resultaat als er geen overeenkomst is. Alle uitdrukkingen zonder overeenkomst retourneren in dit voorbeeld daarom 0 en worden uitgesloten van de gegevens die zijn geladen door de `WHERE`-instructie.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in twee tabellen met de naam:
  - `Transactions`: bevat transactievelden
  - `Transaction_Buckets`: maakt twee nieuwe velden, `Customer` en `Color Code - Black, Blue, blue`. De `where`-instructie laadt de resultaten van de functie `mixmatch`.

### Load-script

```
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;
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `[Color Code - Black, Blue, blue]`
- `Customer`

Resultatentabel met de uitvoer van het gebruik van de functie mixmatch in het load-script.

| Color Code - Black, Blue, blue | Customer |
|---|---|
| Black | 203521 |
| Black | 3036491 |
| Blue | 2038593 |
| blue | 5646471 |

## Voorbeeld - Diagramuitdrukking met gebruik van mixmatch

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Cities`
  - `Count`

### Load-script

```
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];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Cities`

Maak de volgende metingen:

- `=mixmatch( Cities,'Toronto','Boston','Beijing','Zurich')`, om de indexatie van de zoekreeks te retourneren.
- `=mixmatch( Cities,'Toronto','Boston','Beijing','Stockholm','Zurich')`, om de indexatie van de zoekreeks te retourneren.
- `=Cities & ' - ' & mixmatch ( Cities, 'Toronto','Boston', 'Beijing','Stockholm','Zurich')`, om het veld `Cities` en de indexatie van de zoekreeks te retourneren.

De eerste meting in de onderstaande tabel retourneert `0` voor `Stockholm`, aangezien `Stockholm` niet is opgenomen in de lijst met uitdrukkingen in de functie `mixmatch`. Het retourneert ook `4` voor `Zurich` omdat de vergelijking `mixmatch` niet hoofdlettergevoelig is.

Resultatentabel

| Cities | mixmatch( Cities,'Toronto','Boston','Beijing','Zurich') | mixmatch( Cities,'Toronto','Boston','Beijing','Stockholm','Zurich') | Cities & ' - ' & mixmatch ( 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 | 4 | 5 | zurich - 5 |

## Voorbeeld - Aangepaste sorteeruitdrukking met gebruik van mixmatch

Diagramuitdrukking

### Overzicht

U kunt `mixmatch` gebruiken om een uitdrukking op aangepaste wijze te sorteren. Dit voorbeeld gebruikt dezelfde gegevensverzameling als het vorige voorbeeld.

Kolommen worden standaard numeriek of alfabetisch gesorteerd, afhankelijk van de gegevens.

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Cities`

Resultatentabel toont de standaardsorteervolgorde

| Cities |
|---|
| Beijing |
| Boston |
| Stockholm |
| Toronto |
| zurich |

Doe het volgende om de sorteervolgorde te wijzigen:

1. Open de sectie **Sorteren** in het **eigenschappenvenster**.
2. Schakel automatisch sorteren uit voor de kolom die u aangepast wilt sorteren.
3. Schakel de selectievakjes **Numeriek sorteren** en **Alfabetisch sorteren** uit.
4. Selecteer **Sorteren op uitdrukking** en voer een uitdrukking in die hierop lijkt:
   `=mixmatch( Cities, 'Toronto','Boston','Beijing','Stockholm','Zurich')`
   De sorteervolgorde van de kolom Cities wordt gewijzigd.

Resultatentabel toont de sorteervolgorde met de functie *mixmatch*.

| Cities |
|---|
| Toronto |
| Boston |
| Beijing |
| Stockholm |
| zurich |

## pick

De functie pick retourneert de *n*-de uitdrukking in de lijst.

**Syntaxis:**

```
pick(n, expr1[ , expr2,...exprN])
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **n** | *n* is een geheel getal tussen 1 en N. |

Voorbeeld: diagramuitdrukking

| Voorbeeld | Resultaat |
|---|---|
| `pick( N, 'A','B',4, 6 )` | retourneert `B` als `N = 2`<br>retourneert `4` als `N = 3` |

## Voorbeeld - Labels aan producten toewijzen met pick

Diagramuitdrukking

### Overzicht

Een gegevensverzameling bevat productgegevens.  U wilt categorielabels aan producten toewijzen op basis van de categoriecode.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
    - `ProductID`
    - `ProductName`
    - `CategoryCode`

### Load-script

```
Example:
LOAD * INLINE [
ProductID, ProductName, CategoryCode
1, "Laptop", 1
2, "Headphones", 2
3, "Printer", 3
4, "Smartphone", 1
5, "Tablet", 1
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `ProductID`
- `ProductName`
- `CategoryCode`

Maak de volgende meting:

- `=Pick(CategoryCode, 'Electronics', 'Accessories', 'Office Equipment')`, om een geldige uitdrukking uit de lijst te retourneren.

Resultatentabel

| ProductID | ProductName | CategoryCode | Pick (CategoryCode, 'Electronics', 'Accessories', 'Office Equipment') |
|---|---|---|---|
| 1 | Laptop | 1 | Electronics |
| 2 | Headphone | 2 | Accessories |
| 3 | Printer | 3 | Office Equipment |
| 4 | Smartphone | 1 | Electronics |
| 5 | Tablet | 1 | Electronics |

De functie `pick` retourneert een lijst met waarden die bij de overeenkomstige indexatie van het numerieke veld `CategoryCode` horen. Het label `Electronics` is bijvoorbeeld de eerste parameter in de functie en wordt geretourneerd voor alle producten met de `CategoryCode` of `1`.

## Voorbeeld - Regiolabels toewijzen met pick

Diagramuitdrukking

### Overzicht

Een gegevensverzameling bevat land- en regiocodes. U wilt regiolabels aan landen toewijzen op basis van de regiocode.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Country`
  - `RegionCode`

### Load-script

```
Example:
LOAD * INLINE [
Country, RegionCode
"USA", 1
"Canada", 1
"Germany", 2
"Japan", 3
"Australia", 4
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `Country`
- `RegionCode`

Maak de volgende meting:

- `=Pick(RegionCode, 'North America', 'Europe', 'Asia', 'Australia')`, om een geldig item uit de lijst te retourneren.

Resultatentabel

| Country | RegionCode | Pick (RegionCode, 'North America', 'Europe', 'Asia', 'Australia') |
|---|---|---|
| Australia | 4 | Australia |
| Canada | 1 | North America |
| Germany | 2 | Europe |
| Japan | 3 | Asia |
| USA | 1 | North America |

De functie `pick` retourneert een lijst met waarden die bij de overeenkomstige indexatie van het numerieke veld `RegionCode` horen. Het label `North America` is bijvoorbeeld de eerste parameter in de functie en wordt geretourneerd voor alle landen met de `RegionCode` of `1`.

## Voorbeeld - pick en match combineren om lijsten met overeenkomsten in alternatieve tekst te vertalen

Diagramuitdrukking

#### Overzicht

Dit geavanceerde scenario beschrijft hoe u effectief een combinatie van de functies `pick` en `match` kunt gebruiken om tekstmatches inline toe te wijzen. In programmeertermen dient dit een soortgelijk doel als een `Case` instructie. Het scenario verkent deze functionaliteit door landnamen aan regio-equivalente namen toe te wijzen, terwijl het ook een standaard `Region` waarde geeft voor alle voorheen niet in kaart gebrachte `Country` namen.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Countries`.
    - De gegevenstabel bevat één veld met de naam `Country`.

### Load-script

```
Countries:
LOAD * INLINE [
Country
"USA"
"Canada"
"Germany"
"Japan"
"Australia"
"South Africa"
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `Country`
- `RegionCode`

Maak de volgende meting:

- `=Pick(1+match(Country,'USA','Canada','Germany','Japan','Australia'), 'Unmapped', 'North America', 'North America', 'Europe', 'Asia', 'Australia')`

De functie `match` retourneert de indexatie die overeenkomt met de opgegeven door komma's gescheiden alternatieven, te beginnen met `1 = 'USA'`, `2 = 'Canada'` enzovoort. Als geen waarde wordt gevonden, wordt nul geretourneerd. Dit resultaat wordt dan verhoogd met `1` en ingevoerd in de functie `pick`, met `1 = default value` als er geen overeenkomst wordt gevonden, `2 = 'USA'`, `3 = 'Canada'` enzovoort.

Deze combinatie van `Match` en `Pick` functie-aanroepen zal dan de overeenkomende `Country` namen decoderen in een gelijkwaardige `Region` naam. Alle niet-toegewezen `Country` waarden zullen de standaard `Region` waarde, `Unmapped`, retourneren.

Resultatentabel

| 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

Met de functie **wildmatch** vergelijkt u de eerste parameter met alle volgende parameters en krijgt u het aantal overeenkomende uitdrukkingen als resultaat. Bij deze functie kunt u jokertekens ( ***** en **?**) gebruiken in de tekenreeksen die worden vergeleken. ***** komt overeen met een willekeurige reeks tekens. **?** komt overeen met een willekeurig enkel teken. De vergelijking is niet hoofdlettergevoelig en maakt geen onderscheid tussen de Japanse Hiragana- en Katakana-tekensystemen.

**Syntaxis:**

```
wildmatch( str, expr1 [ , expr2,...exprN ])
```

**Retourgegevenstypen:** geheel getal

Als u een vergelijking wilt gebruiken zonder jokertekens, gebruikt u de functie **match** of **mixmatch**.

Argumenten

| Argument | Beschrijving |
|---|---|
| **str** | De tekenreeks die moet worden geëvalueerd. |
| **expr1** | De eerste uitdrukking voor controle op een geldige getalsnotatie. |
| **expr2** | De tweede uitdrukking voor controle op een geldige getalsnotatie. |
| **expr3** | De derde uitdrukking voor controle op een geldige getalsnotatie. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `wildmatch( ColorCode,'black','*Blue*' )` | Deze uitdrukking retourneert `1` als de waarde van `ColorCode Black` is of `2` als de waarde van `ColorCode Blue` is of `Light Blue` of `blue`. |
| `wildmatch( Cities,'toronto','Boston','Beijing','Zurich')` | Deze uitdrukking retourneert de indexatie van het item dat overeenkomt met de waarde in het veld `Cities`. |

### Voorbeeld - Producten categoriseren met wildmatch

Diagramuitdrukking

**Overzicht**

Een gegevensverzameling bevat productgegevens. U wilt producten identificeren en categoriseren die consumeerbaar zijn.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `ProductID`
  - `ProductName`
  - `ProductCategory`

### Load-script

```
Example:
LOAD * INLINE [
ProductID, ProductName, ProductCategory
1, "Laptop", "Electronics"
2, "Banana", "fruit"
3, "Shampoo", "toiletries"
4, "TV", "Electronics"
5, "Apple", "Fruits and vegetables"
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `ProductID`
- `ProductName`
- `ProductCategory`

Maak de volgende metingen:

- `=wildmatch(ProductCategory, 'Toiletries*', 'Fruit*')`, om de indexatie te retourneren waarbij `ProductCategory` de waarde heeft die begint met `Toiletries` of begint met `Fruit`.
- `=If(wildmatch(ProductCategory, 'Toiletries*', 'Fruit*'), 'Consumable', 'Other')`, om de indexatie te retourneren waarbij `ProductCategory` de waarde heeft die begint met `Toiletries` of begint met `Fruits`. Als een van beide wordt gevonden, dan wordt `Consumable` geretourneerd. Anders wordt `Other` geretourneerd.

*De functie `wildmatch` retourneert een positieve waarde (niet nul) als er een overeenkomst is gevonden, anders nul als er geen overeenkomst is gevonden. De `if` instructie interpreteert op zijn beurt resultaten die niet nul zijn als `true` en resultaten die nul zijn als `false`. Door een `if` instructie te combineren met `mixmatch`, `wildmatch` of `match` functies, is het mogelijk om gemakkelijk overeenkomsten en mismatches te decoderen.*

Resultatentabel

| ProductID | ProductName | ProductCategory | wildmatch (ProductCategory, 'Toiletries*', 'Fruit*') | If(wildmatch (ProductCategory, 'Toiletries*', 'Fruit*') |
|---|---|---|---|---|
| 1 | Laptop | Electronics | 0 | Other |
| 2 | Banana | fruit | 2 | Consumable |
| 3 | Shampoo | toiletries | 1 | Consumable |
| 4 | TV | Electronics | 0 | Other |
| 5 | Apple | Fruits and vegetables | 2 | Consumable |

De functie `wildmatch` retourneert de indexatie van de zoekreeks in de eerste meting. In de tweede meting wordt dit resultaat vervolgens gebruikt in een `If` functie om aangepaste resultaten te bieden. Let op dat de functie `wildmatch` hoofdletterongevoelig is en jokertekens (`*` en `?`) toestaat.

## Voorbeeld - Klantgegevens evalueren en classificeren met behulp van wildmatch

Diagramuitdrukking

### Overzicht

Een gegevensverzameling bevat klantgegevens. U wilt de klanten identificeren en categoriseren waarvan de regio en industriecategorie aan uw classificatievereisten voldoen.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `ClientID`
  - `ClientName`
  - `Region`
  - `Industry`

### Load-script

```
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"
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `ClientID`
- `ClientName`
- `Region`
- `Industry`

Maak de volgende meting:

`=If(wildmatch(Region, 'North*', 'East*') AND wildmatch(Industry, '*Technology*', '*Manufacturing*'), 'Target Segment', 'Secondary Segment')`, om de indexatie van de zoekreeks te retourneren. Als deze wordt gevonden, dan wordt `Target Segment` geretourneerd. Als deze niet wordt gevonden, dan wordt `Secondary Segment` geretourneerd.

Resultatentabel

| 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 |

De functie `wildmatch` zoekt naar tekst en stuurt een indexatie terug die vervolgens kan worden gebruikt om aanvullende analyses en gegevensclassificatie uit te voeren. Bijvoorbeeld, `Acme Corp` wordt geclassificeerd als een `Target Segment` omdat het voldoet aan de zoekvereisten van de `wildmatch` functie voor zowel het veld `Region` als `Industry`. Let op dat de functie `wildmatch` hoofdletterongevoelig is en jokertekens (`*` en `?`) toestaat.

## Voorbeeld - Load-script met gebruik van wildmatch

Load-script

### Overzicht

U kunt wildmatch gebruiken om een gegevens-subset te laden. U kunt bijvoorbeeld een numerieke waarde als resultaat krijgen voor een uitdrukking in de functie. Vervolgens kunt u de gegevens beperken die zijn geladen op basis van de numerieke waarde. Wildmatch geeft 0 als resultaat als er geen overeenkomst is. Alle uitdrukkingen zonder overeenkomst retourneren in dit voorbeeld daarom 0 en worden uitgesloten van de gegevens die zijn geladen door de `WHERE`-instructie.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in twee tabellen met de naam:
  - `Transactions`: bevat transactievelden
  - `Transaction_Buckets`: maakt twee nieuwe velden, `Customer` en `Color code - Black, Blue, blue, red`. De `where`-instructie laadt de resultaten van de functie `wildmatch`.

### Load-script

```
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;
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `[Color code - Black, Blue, blue, red]`
- `Customer`

Resultatentabel met de uitvoer van het gebruik van de functie wildmatch in het load-script.

| Color code - Black, Blue, blue, red | Customer |
|---|---|
| Black | 203521 |
| Black | 3036491 |
| Blue | 2038593 |
| blue | 5646471 |
| Red | 049681 |
| Red | 2038593 |

## Voorbeeld - Diagramuitdrukking met gebruik van wildmatch

Diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Cities`
  - `Count`

#### Load-script

```
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];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Cities`

Maak de volgende metingen:

- `=wildmatch( Cities,'Tor*','?ton','Beijing','*urich')` , om de indexatie van de zoekreeks te retourneren.
- `=wildmatch( Cities,'Tor*','???ton','Beijing','Stockholm','*urich')`, om de indexatie van de zoekreeks te retourneren.
- `=Cities & ' - ' & wildmatch ( Cities, 'Tor*','???ton','Beijing','Stockholm','*urich')`, om het veld `Cities` en de indexatie van de zoekreeks te retourneren.

De eerste meting in de onderstaande tabel retourneert `0` voor `Stockholm`, aangezien `Stockholm` niet is opgenomen in de lijst met uitdrukkingen in de functie `wildmatch`. Het retourneert ook `0` voor `Boston`, omdat `?` alleen overeenkomt met een enkel teken.

Resultatentabel

| Cities | wildmatch( Cities,'Tor*','?ton','Beijing','*urich') | wildmatch( Cities,'Tor*','???ton','Beijing','Stockholm','*urich') | Cities & ' - ' & wildmatch ( Cities, '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 |

## Voorbeeld - Aangepaste sorteeruitdrukking met gebruik van wildmatch

Diagramuitdrukking

### Overzicht

U kunt `wildmatch` gebruiken om een uitdrukking op aangepaste wijze te sorteren. Dit voorbeeld gebruikt dezelfde gegevensverzameling als het vorige voorbeeld.

Kolommen worden standaard numeriek of alfabetisch gesorteerd, afhankelijk van de gegevens.

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Cities`

Resultatentabel toont de standaardsorteervolgorde

| Cities |
|---|
| Beijing |
| Boston |
| Stockholm |
| Toronto |
| zurich |

Doe het volgende om de sorteervolgorde te wijzigen:

1. Open de sectie **Sorteren** in het **eigenschappenvenster**.
2. Schakel automatisch sorteren uit voor de kolom die u aangepast wilt sorteren.
3. Schakel de selectievakjes **Numeriek sorteren** en **Alfabetisch sorteren** uit.
4. Selecteer **Sorteren op uitdrukking** en voer een uitdrukking in die hierop lijkt:
   `=wildmatch( Cities, 'Tor*','???ton','Beijing','Stockholm','*urich')`
   De sorteervolgorde van de kolom Cities wordt gewijzigd.

Resultatentabel toont de sorteervolgorde met de functie *wildmatch*.

| Cities |
|---|
| Toronto |
| Boston |
| Beijing |
| Stockholm |
| zurich |

# 8.6 Tellerfuncties

In dit hoofdstuk worden functies beschreven waarmee u counters kunt opnemen tijdens de evaluatie van **LOAD** -opdrachten in het load-script voor gegevens. De enige functie die kan worden gebruikt in diagramuitdrukkingen is **RowNo()**.

Sommige tellerfuncties hebben geen parameters, maar er moeten altijd haakjes achter worden geplaatst.

## Overzicht tellerfuncties

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

### autonumber

Deze scriptfunctie retourneert een uniek geheel getal voor elke distinctieve geëvalueerde waarde van *expression* die is gevonden tijdens uitvoering van het script. Met deze functie kan bijvoorbeeld een compacte geheugenrepresentatie van een complexe sleutel worden gemaakt.

```
autonumber (expression[ , AutoID])
```

### autonumberhash128

Deze scriptfunctie berekent een 128-bits hash van de gecombineerde invoerwaarden van de uitdrukking en retourneert een uniek geheel getal voor elke distinctieve hash-waarde die wordt gevonden tijdens de uitvoering van het script. Met deze functie kan bijvoorbeeld een compacte geheugenrepresentatie van een complexe sleutel worden gemaakt.

```
autonumberhash128 (expression {, expression})
```

### autonumberhash256

Deze scriptfunctie berekent een 256-bits hash van de gecombineerde invoerwaarden van de uitdrukking en retourneert een uniek geheel getal voor elke distinctieve hash-waarde die wordt gevonden tijdens de uitvoering van het script. Met deze functie kan bijvoorbeeld een compacte geheugenrepresentatie van een complexe sleutel worden gemaakt.

```
autonumberhash256 (expression {, expression})
```

### IterNo

Deze scriptfunctie retourneert een geheel getal dat aangeeft hoe vaak één record wordt geëvalueerd in een **LOAD**-opdracht met een **while**-clausule. De eerste iteratie heeft nummer 1. De functie **IterNo** is alleen relevant als deze samen met een **while**-clausule wordt gebruikt.

```
IterNo ( )
```

### RecNo

Deze scriptfunctie retourneert een geheel getal voor het nummer van de momenteel gelezen rij van de interne tabel. De eerste record is nummer 1.

```
RecNo ( )
```

### RowNo - script function

Deze functie retourneert een geheel getal voor de positie van de huidige rij in de resulterende interne Qlik Sense-tabel. De eerste rij is nummer 1.

```
RowNo ( )
```

### RowNo - chart function

**RowNo()** retourneert het nummer van de huidige rij in het huidige kolomsegment in een tabel. Voor bitmapdiagrammen wordt met **RowNo()** het nummer van de huidige rij in het equivalent van de standaard tabel voor het diagram geretourneerd.

```
RowNo - diagramfunctie([TOTAL])
```

## autonumber

Deze scriptfunctie retourneert een uniek geheel getal voor elke distinctieve geëvalueerde waarde van *expression* die is gevonden tijdens uitvoering van het script. Met deze functie kan bijvoorbeeld een compacte geheugenrepresentatie van een complexe sleutel worden gemaakt.

> *U kunt alleen verbinding maken met **autonumber**-sleutels die zijn gegenereerd in dezelfde laadbewerking voor gegevens, aangezien het gehele getal wordt gegenereerd op basis van de volgorde waarin de tabel wordt gelezen. Als u sleutels moet gebruiken die ongewijzigd blijven tussen laadbewerkingen voor gegevens, onafhankelijk van de sortering van de brongegevens, moet u de functie **hash128**, **hash160** of **hash256** gebruiken.*

**Syntaxis:**

```
autonumber(expression[ , AutoID])
```

**Argumenten:**

| Argument | Beschrijving |
|---|---|
| AutoID | Als de functie **autonumber** voor verschillende sleutels in het script wordt gebruikt, kunt u meerdere instanties van de teller maken door met de optionele parameter *AutoID* elke teller een naam te geven. |

**Voorbeeld: Een samengestelde sleutel maken**

In dit voorbeeld maken we een samengestelde sleutel met de functie **autonumber** om geheugen te sparen. Het voorbeeld is kort vanwege demonstratiedoeleinden, maar zou relevant zijn bij een tabel met een groot aantal rijen.

Voorbeeldgegevens

| 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 |

De brongegevens worden geladen met behulp van inline-gegevens. Vervolgens voegen we een voorafgaande load-opdracht toe die een samengestelde sleutel maakt op basis van de velden Region, Year en 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
];
```

De resulterende tabel ziet er als volgt uit:

Resultatentabel

| 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 dit voorbeeld kunt u verwijzen naar de RYMkey, bijvoorbeeld 1, in plaats van de tekenreeks 'North2014May' als u een koppeling met een andere tabel tot stand moet brengen.

Nu laden we op vergelijkbare wijze een brontabel met kosten. De velden Region, Year en Month zijn uitgesloten in de voorafgaande laadbewerking om te voorkomen dat er een synthetische sleutel wordt gemaakt. We maken al een samengestelde sleutel met de functie **autonumber**, waardoor de tabellen worden gekoppeld.

```
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
];
```

Nu kunnen we een tabelvisualisatie toevoegen aan een werkblad en de velden Region, Year en Month toevoegen, alsmede metingen Som voor de verkopen en kosten. De tabel ziet er dan als volgt uit:

Resultatentabel

| 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

Deze scriptfunctie berekent een 128-bits hash van de gecombineerde invoerwaarden van de uitdrukking en retourneert een uniek geheel getal voor elke distinctieve hash-waarde die wordt gevonden tijdens de uitvoering van het script. Met deze functie kan bijvoorbeeld een compacte geheugenrepresentatie van een complexe sleutel worden gemaakt.

> *U kunt alleen verbinding maken met **autonumberhash128**-sleutels die zijn gegenereerd in dezelfde laadbewerking voor gegevens, aangezien het gehele getal wordt gegenereerd op basis van de volgorde waarin de tabel wordt gelezen. Als u sleutels moet gebruiken die ongewijzigd blijven tussen laadbewerkingen voor gegevens, onafhankelijk van de sortering van de brongegevens, moet u de functie **hash128**, **hash160** of **hash256** gebruiken.*

**Syntaxis:**

```
autonumberhash128(expression {, expression})
```

### Voorbeeld: Een samengestelde sleutel maken

In dit voorbeeld maken we een samengestelde sleutel met de functie **autonumberhash128** om geheugen te sparen. Het voorbeeld is kort vanwege demonstratiedoeleinden, maar zou relevant zijn bij een tabel met een groot aantal rijen.

Voorbeeldgegevens

| 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 |

De brongegevens worden geladen met behulp van inline-gegevens. Vervolgens voegen we een voorafgaande load-opdracht toe die een samengestelde sleutel maakt op basis van de velden Region, Year en 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
];
```

De resulterende tabel ziet er als volgt uit:

Resultatentabel

| 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 dit voorbeeld kunt u verwijzen naar de RYMkey, bijvoorbeeld 1, in plaats van de tekenreeks 'North2014May' als u een koppeling met een andere tabel tot stand moet brengen.

Nu laden we op vergelijkbare wijze een brontabel met kosten. De velden Region, Year en Month zijn uitgesloten in de voorafgaande laadbewerking om te voorkomen dat er een synthetische sleutel wordt gemaakt. We maken al een samengestelde sleutel met de functie **autonumberhash128**, waardoor de tabellen worden gekoppeld.

```
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
];
```

Nu kunnen we een tabelvisualisatie toevoegen aan een werkblad en de velden Region, Year en Month toevoegen, alsmede metingen Som voor de verkopen en kosten. De tabel ziet er dan als volgt uit:

Resultatentabel

| 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

Deze scriptfunctie berekent een 256-bits hash van de gecombineerde invoerwaarden van de uitdrukking en retourneert een uniek geheel getal voor elke distinctieve hash-waarde die wordt gevonden tijdens de uitvoering van het script. Met deze functie kan bijvoorbeeld een compacte geheugenrepresentatie van een complexe sleutel worden gemaakt.

> *U kunt alleen verbinding maken met **autonumberhash256**-sleutels die zijn gegenereerd in dezelfde laadbewerking voor gegevens, aangezien het gehele getal wordt gegenereerd op basis van de volgorde waarin de tabel wordt gelezen. Als u sleutels moet gebruiken die ongewijzigd blijven tussen laadbewerkingen voor gegevens, onafhankelijk van de sortering van de brongegevens, moet u de functie **hash128**, **hash160** of **hash256** gebruiken.*

**Syntaxis:**

```
autonumberhash256(expression {, expression})
```

### Voorbeeld: Een samengestelde sleutel maken

In dit voorbeeld maken we een samengestelde sleutel met de functie **autonumberhash256** om geheugen te sparen. Het voorbeeld is kort vanwege demonstratiedoeleinden, maar zou relevant zijn bij een tabel met een groot aantal rijen.

Voorbeeldtabel

| 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 |

De brongegevens worden geladen met behulp van inline-gegevens. Vervolgens voegen we een voorafgaande load-opdracht toe die een samengestelde sleutel maakt op basis van de velden Region, Year en 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
];
```

De resulterende tabel ziet er als volgt uit:

Resultatentabel

| Region | Year | Month | Sales | RYMkey |
|---|---|---|---|---|
| North | 2014 | May | 245 | 1 |
| North | 2014 | May | 347 | 1 |

| Region | Year | Month | Sales | RYMkey |
|---|---|---|---|---|
| North | 2014 | June | 127 | 2 |
| South | 2014 | June | 645 | 3 |
| South | 2013 | May | 367 | 4 |
| South | 2013 | May | 221 | 4 |

In dit voorbeeld kunt u verwijzen naar de RYMkey, bijvoorbeeld 1, in plaats van de tekenreeks 'North2014May' als u een koppeling met een andere tabel tot stand moet brengen.

Nu laden we op vergelijkbare wijze een brontabel met kosten. De velden Region, Year en Month zijn uitgesloten in de voorafgaande laadbewerking om te voorkomen dat er een synthetische sleutel wordt gemaakt. We maken al een samengestelde sleutel met de functie **autonumberhash256**, waardoor de tabellen worden gekoppeld.

```
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
];
```

Nu kunnen we een tabelvisualisatie toevoegen aan een werkblad en de velden Region, Year en Month toevoegen, alsmede metingen Som voor de verkopen en kosten. De tabel ziet er dan als volgt uit:

Resultatentabel

| 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

Deze scriptfunctie retourneert een geheel getal dat aangeeft hoe vaak één record wordt geëvalueerd in een **LOAD**-opdracht met een **while**-clausule. De eerste iteratie heeft nummer 1. De functie **IterNo** is alleen relevant als deze samen met een **while**-clausule wordt gebruikt.

**Syntaxis:**

```
IterNo( )
```

Voorbeelden en resultaten:

**Voorbeeld:**

```
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
];
```

Deze **LOAD**-opdracht genereert één record per datum binnen het bereik dat is gedefinieerd met **StartDate** en **EndDate**.

In dit voorbeeld ziet de resulterende tabel er als volgt uit:

Resultatentabel

| Day | Date |
|---|---|
| 1 | 2014-01-22 |
| 2 | 2014-01-23 |
| 3 | 2014-01-24 |
| 4 | 2014-01-25 |
| 5 | 2014-01-26 |

## RecNo

Deze scriptfunctie retourneert een geheel getal voor het nummer van de momenteel gelezen rij van de interne tabel. De eerste record is nummer 1.

**Syntaxis:**

```
RecNo( )
```

In tegenstelling tot bij **RowNo( )**, waarbij rijen in de resulterende Qlik Sense-tabel worden geteld, worden bij **RecNo( )** de records in de tabel met onbewerkte gegevens geteld en gereset als een tabel met onbewerkte gegevens wordt aaneengeschakeld met een andere tabel met onbewerkte gegevens.

**Voorbeeld: Load-script voor gegevens**

Laden van tabellen met onbewerkte gegevens:

```
Tab1:
LOAD * INLINE
[A, B
1, aa
2,cc
3,ee];

Tab2:
LOAD * INLINE
[C, D
5, xx
4,yy
6,zz];
```

Record- en rijnummers laden voor geselecteerde rijen:

```
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;
```

De resulterende interne Qlik Sense-tabel:

Resultatentabel

| A | B | RecNo( ) | RowNo( ) |
|---|---|---|---|
| 1 | aa | 1 | 1 |

| A | B | RecNo( ) | RowNo( ) |
|---|---|---|---|
| 3 | ee | 3 | 2 |
| 4 | yy | 2 | 3 |
| 6 | zz | 3 | 4 |

## RowNo

Deze functie retourneert een geheel getal voor de positie van de huidige rij in de resulterende interne Qlik Sense-tabel. De eerste rij is nummer 1.

**Syntaxis:**

```
RowNo( [TOTAL])
```

In tegenstelling tot **RecNo( )**, waarbij records in de tabel met onbewerkte gegevens worden geteld, telt de functie **RowNo( )** geen records mee die zijn uitgesloten met **where**-clausules. Daarnaast wordt deze functie niet opnieuw ingesteld als een tabel met onbewerkte gegevens aan een andere tabel wordt aaneengeschakeld.

> *Als u een voorafgaande load gebruikt, dat wil zeggen een aantal gestapelde **LOAD** -opdrachten die uit dezelfde tabel lezen, kunt u **RowNo( )** alleen in de bovenste **LOAD** -opdracht gebruiken. Als u **RowNo( )** in de volgende **LOAD** -opdrachten gebruikt, wordt 0 geretourneerd.*

**Voorbeeld: Load-script voor gegevens**

Laden van tabellen met onbewerkte gegevens:

```
Tab1:
LOAD * INLINE
[A, B
1, aa
2,cc
3,ee];

Tab2:
LOAD * INLINE
[C, D
5, xx
4,yy
6,zz];
```

Record- en rijnummers laden voor geselecteerde rijen:

```
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;
```

De resulterende interne Qlik Sense-tabel:

Resultatentabel

| A | B | RecNo( ) | RowNo( ) |
|---|---|---|---|
| 1 | aa | 1 | 1 |
| 3 | ee | 3 | 2 |
| 4 | yy | 2 | 3 |
| 6 | zz | 3 | 4 |

## RowNo - diagramfunctie

**RowNo()** retourneert het nummer van de huidige rij in het huidige kolomsegment in een tabel. Voor bitmapdiagrammen wordt met **RowNo()** het nummer van de huidige rij in het equivalent van de standaard tabel voor het diagram geretourneerd.

Als de tabel of het equivalent van de tabel meerdere verticale dimensies heeft, omvat het huidige kolomsegment alleen rijen met dezelfde waarden als de huidige rij in alle dimensiekolommen, met uitzondering van de kolom waarin de laatste dimensie wordt weergegeven in de onderlinge sorteervolgorde van de velden.

*Kolomsegmenten*

*Het is niet toegestaan te sorteren op y-waarden in grafieken of op uitdrukkingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van de grafiek. Deze sorteeropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie gebruikt in een visualisatie of tabel, wordt de sortering van de visualisatie teruggezet naar de gesorteerde invoer van deze functie.*

**Syntaxis:**

```
RowNo([TOTAL])
```

**Retourgegevenstypen:** geheel getal

**Argumenten:**

| Argument | Beschrijving |
|---|---|
| TOTAL | Als de tabel eendimensionaal is of als de kwalificatie **TOTAL** wordt gebruikt als argument, is het huidige kolomsegment altijd gelijk aan de hele kolom. |

## Voorbeeld: diagramuitdrukking met RowNo

Voorbeeld - diagramuitdrukking

### Load-script

Laad de volgende gegevens als een inline-load in de editor voor laden van gegevens om de onderstaande voorbeelden van diagramuitdrukkingen te maken.

```
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 '|');
```

### Diagramuitdrukking

Maak een tabelvisualisatie in een Qlik Sense-werkblad met **Customer** en **UnitSales** als dimensies. Voeg `RowNo( )` en `RowNo(TOTAL)` toe als metingen met respectievelijk het label **Rij in segment** en **Row Number**. Voeg de volgende uitdrukking als meting toe aan de tabel.

```
If( RowNo( )=1, 0, UnitSales / Above( UnitSales ))
```

### Resultaat

| 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 | 1.1111111111111 |
| 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 |

### Uitleg

De kolom **Row in Segment** toont de resultaten 1,2,3 voor het kolomsegment dat de waarden bevat van UnitSales voor klant Astrida. De rijnummering begint vervolgens opnieuw bij 1 voor het volgende kolomsegment, Betacab.

In de kolom **Row Number** worden de dimensies genegeerd vanwege het argument `TOTAL` voor `RowNo()` en het aantal rijen in de tabel worden geteld.

Deze uitdrukking retourneert 0 voor de eerste rij in elk kolomsegment, zodat de kolom er als volgt uitziet:

0, 2.25, 1.1111111, 0, 2.5, 5, 0, 2, 0 en 4.

**Zie ook:**

p *Above - diagramfunctie (page 1479)*

## 8.7 Datum- en tijdfuncties

Datum- en tijdfuncties in Qlik Sense worden gebruikt voor het transformeren en converteren van datum- en tijdwaarden. Alle functies kunnen zowel worden gebruikt in het load-script voor gegevens als in diagramuitdrukkingen.

Functies zijn gebaseerd op een serienummer voor datum-tijd dat gelijk is aan het aantal dagen sinds 30 december 1899. De waarde van het gehele getal geeft de dag aan en de fractionele waarde geeft de tijd van de dag aan.

In Qlik Sense wordt de numerieke waarde van de parameter gebruikt, dus een getal is geldig als parameter ook als het niet is opgemaakt als een datum of tijd. Als de parameter niet overeenkomt met een numerieke waarde, bijvoorbeeld als het een tekenreeks is, wordt in Qlik Sense geprobeerd om de tekenreeks te interpreteren op basis van de omgevingsvariabelen voor datum en tijd.

Als de tijdnotatie die in de parameter wordt gebruikt niet overeenkomt met de notatie die is ingesteld in de omgevingsvariabelen, is geen correcte interpretatie in Qlik Sense mogelijk. U kunt dit oplossen door de instellingen te wijzigen of gebruik te maken van een interpretatiefunctie.

In de voorbeelden voor elke functie wordt uitgegaan van de standaardnotatie voor tijd en datum, hh:mm:ss en YYYY-MM-DD (ISO 8601).

*Bij Het verwerken van een tijdsstempel met een datum- of tijdfunctie, negeert Qlik Sense zomertijdparameters tenzij de datum- of tijdfunctie een geografische positie bevat.*

*`ConvertToLocalTime( filetime('Time.qvd'), 'Paris')` gebruikt bijvoorbeeld wel zomertijdparameters terwijl `ConvertToLocalTime(filetime('Time.qvd'), 'GMT-01:00')` geen zomertijdparameters gebruikt.*

## Overzicht van datum- en tijdfuncties

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

### Uitdrukkingen van tijd in gehele getallen

#### second

Deze functie retourneert een geheel getal dat de seconde representeert als de breuk van de **expression** wordt geïnterpreteerd als een datum volgens de standaardinterpretatie van getallen.

```
second (expression)
```

#### minute

Deze functie retourneert een geheel getal dat de minuut representeert als de breuk van de **expression** wordt geïnterpreteerd als een datum volgens de standaardinterpretatie van getallen.

```
minute (expression)
```

#### hour

Deze functie retourneert een geheel getal dat het uur representeert als de breuk van de **expression** wordt geïnterpreteerd als een datum volgens de standaardinterpretatie van getallen.

```
hour (expression)
```

#### day

Deze functie retourneert een geheel getal dat de dag representeert als de breuk van **expression** wordt geïnterpreteerd als een datum volgens de standaardinterpretatie van getallen.

```
day (expression)
```

### week

Deze functie retourneert een geheel getal dat het weeknummer representeert conform ISO 8601. Het weeknummer wordt berekend volgens de datuminterpretatie van de uitdrukking, in overeenstemming met de standaardinterpretatie van getallen.

```
week (expression)
```

### month

Deze functie retourneert een duale waarde: een maandnaam zoals vastgelegd in de omgevingsvariabele **MonthNames**, en een geheel getal tussen 1-12. De maand wordt berekend volgens de datuminterpretatie van de uitdrukking, in overeenstemming met de standaardinterpretatie van getallen.

```
month (expression)
```

### year

Deze functie retourneert een geheel getal dat het jaar representeert als de **expression** wordt geïnterpreteerd als een datum volgens de standaardinterpretatie van getallen.

```
year (expression)
```

### weekyear

Deze functie retourneert het jaar waarbij het weeknummer hoort, op basis van de omgevingsvariabelen. Het weeknummer varieert van 1 tot ongeveer 52.

```
weekyear (expression)
```

### weekday

Deze functie retourneert een duale waarde met:

- De naam van een dag zoals gedefinieerd in de omgevingsvariabele **DayNames**.
- Een geheel getal tussen 0 en 6 dat overeenkomt met de nominale dag van de week (0-6).

```
weekday (date)
```

## Tijdstempelfuncties

### now

Deze functie retourneert een tijdstempel van de huidige tijd. Deze functie retourneert waarden in de notatie van de **TimeStamp**-systeemvariabele. De standaardwaarde voor **timer_mode** is 1.

```
now ([ timer_mode])
```

### today

Deze functie retourneert de huidige datum. Deze functie retourneert waarden in de indeling van de systeemvariabele `DateFormat`.

```
today ([timer_mode])
```

### LocalTime

Deze functie retourneert een tijdstempel met de huidige tijd voor een opgegeven tijdzone.

```
localtime ([timezone [, ignoreDST ]])
```

## Maakfuncties

### makedate

Deze functie retourneert een datum die is berekend voor het jaar **YYYY**, de maand **MM** en de dag **DD**.

```
makedate (YYYY [ , MM [ , DD ] ])
```

### makeweekdate

Deze functie retourneert een datum die is berekend voor het jaar, het weeknummer en de dag van de week.

```
makeweekdate (YYYY [ , WW [ , D ] ])
```

### maketime

Deze functie retourneert een tijd die is berekend voor het uur **hh**, de minuut **mm** en de seconde **ss**.

```
maketime (hh [ , mm [ , ss [ .fff ] ] ])
```

## Andere datumfuncties

### AddMonths

Deze functie retourneert de datum **n** maanden na **startdate** of, als **n** negatief is, de datum **n** maanden vóór **startdate**.

```
addmonths (startdate, n , [ , mode])
```

### AddYears

Deze functie retourneert de datum **n** jaar na **startdate** of, als **n** negatief is, de datum **n** jaar vóór **startdate**.

```
addyears (startdate, n)
```

### yeartodate

Deze functie bepaalt of de invoertijdstempel binnen het jaar valt waarin het script voor het laatst is geladen en retourneert True als dat het geval is en False als dat niet het geval is.

```
yeartodate (date [ , yearoffset [ , firstmonth [ , todaydate] ] ])
```

## TimeZone-functies

### timezone

De functie retourneert de tijdzone zoals gedefinieerd op de computer waarop de Qlik-engine wordt uitgevoerd.

```
timezone ( )
```

### GMT

Deze functie retourneert de huidige Greenwich Mean Time, zoals afgeleid uit de landinstellingen.

```
GMT ( )
```

### UTC

Retourneert de huidige Coordinated Universal Time.

```
UTC ( )
```

#### daylightsaving

Retourneert de huidige aanpassing voor zomertijd zoals die in Windows is gedefinieerd.

```
daylightsaving ( )
```

#### converttolocaltime

Converteert een UTC- of GMT-tijdstempel naar een lokale tijd met twee representaties. De plaats kan een reeks steden, plaatsen en tijdzones overal ter wereld zijn.

```
converttolocaltime (timestamp [, place [, ignore_dst=false]])
```

### Functies voor instellen van tijd

#### setdateyear

Deze functie neemt een **timestamp** en een **year** als invoer en werkt de **timestamp** bij met het **year** dat is opgegeven in de invoer.

```
setdateyear (timestamp, year)
```

#### setdateyearmonth

Deze functie neemt een **timestamp**, een **month** en een **year** als invoer en werkt de **timestamp** bij met het **year** en de **month** die zijn opgegeven in de invoer.

```
setdateyearmonth (timestamp, year, month)
```

### Functies In...

#### inyear

Deze functie retourneert True als **timestamp** binnen het jaar valt dat **base_date** bevat.

```
inyear (date, basedate , shift [, first_month_of_year = 1])
```

#### inyeartodate

Deze functie retourneert True als **timestamp** binnen het gedeelte van het jaar ligt dat **base_date** bevat tot en met de laatste milliseconde van **base_date**.

```
inyeartodate (date, basedate , shift [, first_month_of_year = 1])
```

#### inquarter

Deze functie retourneert True als **timestamp** binnen het kwartaal valt dat **base_date** bevat.

```
inquarter (date, basedate , shift [, first_month_of_year = 1])
```

#### inquartertodate

Deze functie retourneert True als **timestamp** binnen het gedeelte van het kwartaal ligt dat **base_date** bevat tot en met de laatste milliseconde van **base_date**.

```
inquartertodate (date, basedate , shift [, first_month_of_year = 1])
```

#### inmonth

Deze functie retourneert True als **timestamp** in de maand valt die **base_date** bevat.

```
inmonth (date, basedate , shift)
```

### inmonthtodate

Retourneert True als **date** binnen het gedeelte van de maand ligt dat **basedate** bevat tot en met de laatste milliseconde van **basedate**.

```
inmonthtodate (date, basedate , shift)
```

### inmonths

Deze functie bepaalt of een tijdstempel valt binnen dezelfde periode van de maand, twee maanden, kwartaal, tertiaal of half jaar als basisdatum. Het is tevens mogelijk om te bepalen of de tijdstempel binnen een voorafgaande of volgende tijdsperiode valt.

```
inmonths (n, date, basedate , shift [, first_month_of_year = 1])
```

### inmonthstodate

Deze functie bepaalt of een tijdstempel binnen het gedeelte van een periode van de maand, twee maanden, kwartaal, tertiaal of half jaar tot en met de laatste milliseconde van `base_date` valt. Het is tevens mogelijk om te bepalen of de tijdstempel binnen een voorafgaande of volgende tijdsperiode valt.

```
inmonthstodate (n, date, basedate , shift [, first_month_of_year = 1])
```

### inweek

Deze functie retourneert True als **timestamp** binnen de week valt die **base_date** bevat.

```
inweek (date, basedate , shift [, weekstart])
```

### inweektodate

Deze functie retourneert True als **timestamp** binnen het gedeelte van de week ligt dat **base_date** bevat tot en met de laatste milliseconde van **base_date**.

```
inweektodate (date, basedate , shift [, weekstart])
```

### inlunarweek

Deze functie bepaalt of **timestamp** binnen de maanweek valt die **base_date** bevat. Maanweken in Qlik Sense worden gedefinieerd door het feit dat 1 januari wordt gezien als de eerste dag van de week. Met uitzondering van de laatste week van het jaar, bevat elke week precies zeven dagen.

```
inlunarweek (date, basedate , shift [, weekstart])
```

### inlunarweektodate

Met deze functie wordt bepaald of **timestamp** binnen het gedeelte van de maanweek ligt tot en met de laatste milliseconde van **base_date**. Maanweken in Qlik Sense worden gedefinieerd door het feit dat 1 januari wordt gezien als de eerste dag van de week en, met uitzondering van de laatste week van het jaar, bevat elke week precies zeven dagen.

```
inlunarweektodate (date, basedate , shift [, weekstart])
```

### inday

Deze functie retourneert True als **timestamp** binnen de dag valt die **base_timestamp** bevat.

```
inday (timestamp, basetimestamp , shift [, daystart])
```

### indaytotime

Deze functie retourneert True als **timestamp** binnen het gedeelte van de dag ligt dat **base_timestamp** bevat tot en met de exacte milliseconde van **base_timestamp**.

```
indaytotime (timestamp, basetimestamp , shift [, daystart])
```

## Functies Begin ... Eind

### yearstart

Deze functie retourneert een tijdsstempel die overeenkomt met het begin van de eerste dag van het jaar dat **date** bevat. De standaarduitvoernotatie is de **DateFormat** die is ingesteld in het script.

```
yearstart ( date [, shift = 0 [, first_month_of_year = 1]])
```

### yearend

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de laatste milliseconde van de laatste dag van het jaar dat **date** bevat. De standaarduitvoernotatie is de **DateFormat** die is ingesteld in het script.

```
yearend ( date [, shift = 0 [, first_month_of_year = 1]])
```

### yearname

Deze functie retourneert een jaar in vier cijfers als weergavewaarde met een onderliggende numerieke waarde die overeenkomt met een tijdstempel van de eerste milliseconde van de eerste dag van het jaar dat **date** bevat.

```
yearname (date [, shift = 0 [, first_month_of_year = 1]] )
```

### quarterstart

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de eerste milliseconde van het kwartaal dat **date** bevat. De standaarduitvoernotatie is de **DateFormat** die is ingesteld in het script.

```
quarterstart (date [, shift = 0 [, first_month_of_year = 1]])
```

### quarterend

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de laatste milliseconde van het kwartaal dat **date** bevat. De standaarduitvoernotatie is de **DateFormat** die is ingesteld in het script.

```
quarterend (date [, shift = 0 [, first_month_of_year = 1]])
```

### quartername

Deze functie retourneert de maanden van het kwartaal (opgemaakt volgens de scriptvariabele **MonthNames**) en het jaar met een onderliggende numerieke waarde die overeenkomt met een tijdstempel van de eerste milliseconde van de eerste dag van het kwartaal.

```
quartername (date [, shift = 0 [, first_month_of_year = 1]])
```

### monthstart

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de eerste milliseconde van de eerste dag van de maand die **date** bevat. De standaarduitvoernotatie is de **DateFormat** die is ingesteld in het script.

```
monthstart (date [, shift = 0])
```

### monthend

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de laatste milliseconde van de laatste dag van de maand die `date` bevat. De standaarduitvoernotatie is de `DateFormat` die is ingesteld in het script.

```
monthend (date [, shift = 0])
```

### monthname

Deze functie retourneert de maand (opgemaakt volgens de scriptvariabele **MonthNames**) en het jaar met een onderliggende numerieke waarde die overeenkomt met een tijdstempel van de eerste milliseconde van de eerste dag van de maand.

```
monthname (date [, shift = 0])
```

### monthsstart

Deze functie retourneert een waarde die overeenkomt met de tijdstempel van de eerste milliseconde van de periode van een maand, twee maanden, kwartaal, tertiaal of half jaar die een basisdatum bevat. Het is tevens mogelijk om de tijdstempel voor een voorafgaande of volgende tijdsperiode te bepalen. De standaarduitvoernotatie is de **DateFormat** die is ingesteld in het script.

```
monthsstart (n, date [, shift = 0 [, first_month_of_year = 1]])
```

### monthsend

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de laatste milliseconde van de periode van een maand, twee maanden, kwartaal, tertiaal of half jaar die een basisdatum bevat. Het is tevens mogelijk om de tijdstempel voor een voorafgaande of volgende tijdsperiode te bepalen.

```
monthsend (n, date [, shift = 0 [, first_month_of_year = 1]])
```

### monthsname

Deze functie retourneert een weergavewaarde die het bereik vertegenwoordigt van de maanden van de periode (geformatteerd volgens de scriptvariabele **MonthNames**) alsmede het jaar. De onderliggende numerieke waarde komt overeen met een tijdstempel van de eerste milliseconde van de periode van een maand, twee maanden, kwartaal, tertiaal of half jaar die een basisdatum bevat.

```
monthsname (n, date [, shift = 0 [, first_month_of_year = 1]])
```

### weekstart

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de eerste milliseconde van de eerste dag van de kalenderweek die **date** bevat. De standaarduitvoernotatie is de **DateFormat** die is ingesteld in het script.

```
weekstart (date [, shift = 0 [,weekoffset = 0]])
```

### weekend

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de laatste milliseconde van de laatste dag (zondag) van de kalenderweek die **date** bevat. De standaarduitvoernotatie is de **DateFormat** die is ingesteld in het script.

```
weekend (date [, shift = 0 [,weekoffset = 0]])
```

### weekname

Deze functie retourneert het jaar en weeknummer met een onderliggende numerieke waarde die overeenkomt met een tijdstempel van de eerste milliseconde van de eerste dag van de week die **date** bevat.

```
weekname (date [, shift = 0 [,weekoffset = 0]])
```

### lunarweekstart

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de eerste milliseconde van de eerste dag van de maanweek die **date** bevat. Maanweken in Qlik Sense worden gedefinieerd door het feit dat 1 januari wordt gezien als de eerste dag van de week en, met uitzondering van de laatste week van het jaar, bevat elke week precies zeven dagen.

```
lunarweekstart (date [, shift = 0 [,weekoffset = 0]])
```

### lunarweekend

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de laatste milliseconde van de laatste dag van de maanweek die **date** bevat. Maanweken in Qlik Sense worden gedefinieerd door het feit dat 1 januari wordt gezien als de eerste dag van de week en, met uitzondering van de laatste week van het jaar, bevat elke week precies zeven dagen.

```
lunarweekend (date [, shift = 0 [,weekoffset = 0]])
```

### lunarweekname

Deze functie retourneert een weergavewaarde voor het jaar en maanweeknummer dat overeenkomt met een tijdstempel van de eerste milliseconde van de eerste dag van de week die **date** bevat. Maanweken in Qlik Sense worden gedefinieerd door het feit dat 1 januari wordt gezien als de eerste dag van de week en, met uitzondering van de laatste week van het jaar, bevat elke week precies zeven dagen.

```
lunarweekname (date [, shift = 0 [,weekoffset = 0]])
```

### daystart

Deze functie retourneert een waarde die overeenkomt met een tijdstempel met de eerste milliseconde van de dag in het argument **time**. De standaarduitvoernotatie is de **TimestampFormat** die is ingesteld in het script.

```
daystart (timestamp [, shift = 0 [, dayoffset = 0]])
```

### dayend

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de laatste milliseconde van de dag in **time**. De standaarduitvoernotatie is de **TimestampFormat** die is ingesteld in het script.

```
dayend (timestamp [, shift = 0 [, dayoffset = 0]])
```

### dayname

Deze functie retourneert een datum met een onderliggende numerieke waarde die overeenkomt met een tijdstempel van de eerste milliseconde van de dag met **time**.

```
dayname (timestamp [, shift = 0 [, dayoffset = 0]])
```

## Functies voor dagnummering

### age

De functie **age** retourneert de leeftijd op het tijdstip van de **timestamp** (in voltooide jaren) van iemand die is geboren op **date_of_birth**.

```
age (timestamp, date_of_birth)
```

### networkdays

De functie **networkdays** retourneert het aantal werkdagen (maandag-vrijdag) tussen **start_date** en **end_date**, waarbij rekening wordt gehouden met een eventueel opgegeven **holiday**.

```
networkdays (start:date, end_date {, holiday})
```

### firstworkdate

De functie **firstworkdate** retourneert de laatste begindatum waarbij **no_of_workdays** (maandag-vrijdag) kan worden gehaald als niet later wordt gestopt dan **end_date**, waarbij rekening wordt gehouden met een eventueel opgegeven vakantiedagen. **end_date** en **holiday** moeten geldige datums of tijdstempels zijn.

```
firstworkdate (end_date, no_of_workdays {, holiday} )
```

### lastworkdate

De functie **lastworkdate** retourneert de vroegste einddatum waarbij **no_of_workdays** (maandag-vrijdag) kan worden gehaald als wordt begonnen op **start_date** , waarbij rekening wordt gehouden met een eventueel opgegeven **holiday**. **start_date** en **holiday** moeten geldige datums of tijdstempels zijn.

```
lastworkdate (start_date, no_of_workdays {, holiday})
```

### daynumberofyear

Deze functie berekent het dagnummer van het jaar waarbinnen een tijdstempel valt. De berekening wordt uitgevoerd vanaf de eerste milliseconde van de eerste dag van het jaar, maar de beginmaand kan worden verschoven.

```
daynumberofyear (date[,firstmonth])
```

### daynumberofquarter

Deze functie berekent het dagnummer van het kwartaal waarbinnen een tijdstempel valt. Deze functie wordt gebruikt wanneer een masterkalender wordt gemaakt.

```
daynumberofquarter (date[,firstmonth])
```

# addmonths

Deze functie retourneert de datum **n** maanden na **startdate** of, als **n** negatief is, de datum **n** maanden vóór **startdate**.

**Syntaxis:**

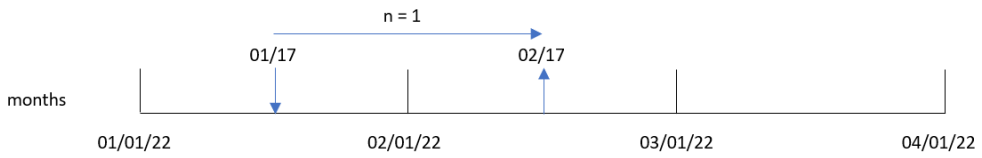
```
AddMonths(startdate, n , [ , mode])
```

**Retourgegevenstypen:** dubbele waarde

De `addmonths()`-functie voegt een gedefinieerd aantal maanden toe aan of trekt dit aantal af, `n`, van een `startdate` en retourneert de resulterende datum.

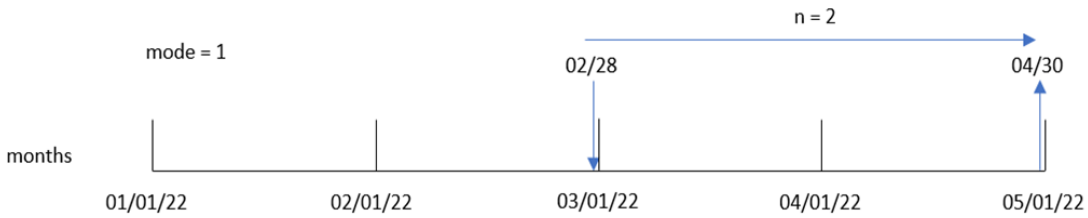
Het argument `mode` heeft invloed op `startdate`-waarden op of na de 28e van de maand. Door het argument `mode` in te stellen op 1, retourneert de `addmonths()`-functie een datum die gelijk is aan de relatieve afstand tot het eind van de maand als de `startdate`.

*Voorbeelddiagram van de `addmonths()`-functie*




Bijvoorbeeld: 28 februari is de laatste dag van de maand. Als de `addmonths()`-functie, met een `mode` van 1, wordt gebruikt om de datum twee maanden later te retourneren, retourneert de functie de laatste datum van april: 30 april.

*Voorbeelddiagram van de `addmonths()`-functie met `mode=1`*




Argumenten

| Argument | Beschrijving |
|---|---|
| startdate | De startdatum als tijdstempel, bijvoorbeeld '2012-10-12'. |
| n | Aantal maanden als een positief of negatief geheel getal. |
| mode | Geeft aan of de maand relatief tot het begin of het eind van de maand is toegevoegd. Standaardmodus is 0 voor toevoegingen relatief tot het begin van de maand. Stel modus in op 1 voor toevoegingen relatief tot het eind van de maand. Als de modus is ingesteld op 1 en de invoerdatum de 28ste of later is, controleert de functie hoeveel dagen er op de startdatum nog resteren voordat het eind van de maand is bereikt. Hetzelfde aantal dagen dat resteert voordat het eind van de maand is bereikt, wordt ingesteld op de geretourneerde datum. |

## Wanneer gebruiken

De `addmonths()`-functie wordt doorgaans gebruikt in een uitdrukking om een datum te vinden met een bepaald aantal maanden vóór of na een bepaalde periode.

Bijvoorbeeld: de `addmonths()`-functie kan worden gebruikt om de einddatum van mobieletelefooncontracten te identificeren.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `addmonths ('01/29/2003' ,3)` | Retourneert '04/29/2003'. |
| `addmonths ('01/29/2003',3,0)` | Retourneert '04/29/2003'. |
| `addmonths ('01/29/2003',3,1)` | Retourneert '04/28/2003'. |
| `addmonths ('01/29/2003',1,0)` | Retourneert '02/28/2003'. |
| `addmonths ('01/29/2003',1,1)` | Retourneert '02/26/2003'. |
| `addmonths ('02/28/2003',1,0)` | Retourneert '03/28/2003'. |
| `addmonths ('02/28/2003',1,1)` | Retourneert '03/31/2003'. |
| `addmonths ('01/29/2003',-3)` | Retourneert '10/29/2002'. |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – geen aanvullende argumenten

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties tussen 2020 en 2022 die wordt geladen in de tabel `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.
- Het maken van een veld, `two_months_later`, dat de datum retourneert voor twee maanden nadat de transactie plaatsvond.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

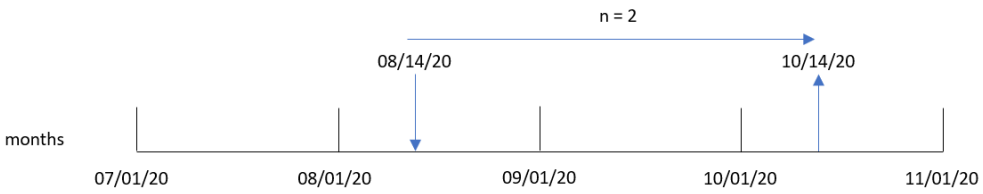
- `date`
- `two_months_later`

Resultatentabel

| 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 |

Het veld `two_months_later` wordt gemaakt in de vorige load-instructie met behulp van de functie `addmonths()`. Het eerste opgegeven argument geeft aan welke datum wordt geëvalueerd. Het tweede argument is het aantal maanden dat moet worden opgeteld bij of afgetrokken van de `startdate`. In dit geval is de waarde 2 opgegeven.

*Diagram van `addmonths()`-functie, voorbeeld zonder aanvullende argumenten*



Transactie 8193 vond plaats op 14 augustus. Daarom retourneert de `addmonths()`-functie 14 oktober 2020 voor het veld `two_months_later`.

## Voorbeeld 2 – Relatief einde van de maand

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met een set eindemaandstransacties in 2022 die wordt geladen in een tabel met de naam `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.
- Het maken van een veld, `relative_two_months_prior`, dat de datum retourneert voor de relatieve datum voor het einde van de maand voordat de transactie plaatsvond.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

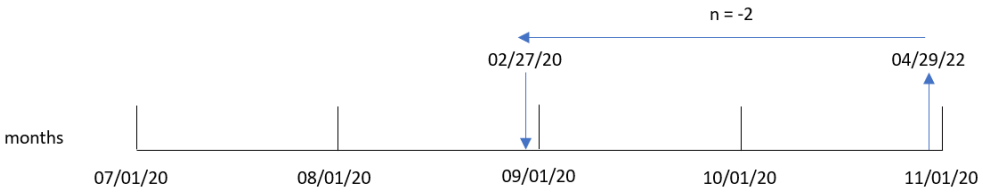
- `date`
- `relative_two_months_prior`

Resultatentabel

| 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 |

Het veld `relative_two_months_prior` wordt gemaakt in de vorige load-instructie met behulp van de functie `addmonths()`. Het eerste opgegeven argument geeft aan welke datum wordt geëvalueerd. Het tweede argument is het aantal maanden dat moet worden opgeteld bij of afgetrokken van de `startdate`. In dit geval is de waarde -2 opgegeven. Het laatste argument is de modus met de waarde 1, die de functie forceert de relatieve datum voor het einde van de maand te berekenen voor alle datums die groter zijn dan of gelijk zijn aan 28.

*Diagram van de `addmonths()`-functie, voorbeeld met `n=-2`*




Transactie 8191 vindt plaats op 29 april 2022. In eerste instantie zou twee maanden eerder de maand instellen op februari. Vervolgens berekent de functie de relatieve waarde voor het einde van de maand, omdat het derde argument van de functie de modus instelt op 1 en de dagwaarde later is dan de 27e. De functie identificeert dat de 29e de op één na laatste dag van april is en retourneert daarom de op één na laatste dag van februari, de 27e.

## Voorbeeld 3 – Diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die de datum retourneert voor twee maanden nadat de transactie plaatsvond, wordt gemaakt als een meting in een diagramobject.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `date`.

Maak de volgende meting:

```
=addmonths(date,2)
```

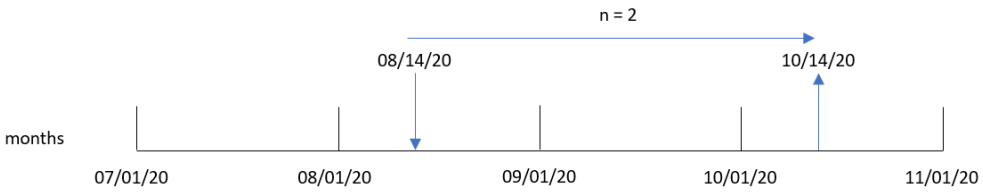
Resultatentabel

| date | =addmonths(date,2) |
|---|---|
| 01/10/2020 | 03/10/2020 |
| 02/28/2020 | 04/28/2020 |
| 04/09/2020 | 06/09/2020 |

| date | =addmonths(date,2) |
|---|---|
| 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 |

De meting `two_months_later` wordt in het diagramobject gemaakt met behulp van de functie `addmonths()`. Het eerste opgegeven argument geeft aan welke datum wordt geëvalueerd. Het tweede argument is het aantal maanden dat moet worden opgeteld bij of afgetrokken van de `startdate`. In dit geval is de waarde 2 opgegeven.

*Diagram van `addmonths()`-functie, diagramobjectvoorbeeld*




Transactie 8193 vond plaats op 14 augustus. Daarom retourneert de `addmonths()`-functie 14 oktober 2020 voor het veld `two_months_later`.

## Voorbeeld 4 – Scenario

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset die wordt geladen in een tabel met de naam `Mobile_Plans`.
- Informatie met de contract-id, begindatum, contractduur en maandelijkse kosten.

De eindgebruiker wil graag een diagramobject dat de einddatum van elk telefooncontract op contract-id weergeeft.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- contract_id
- start_date
- contract_length

Maak de volgende meting aan om de einddatum van elk contract te berekenen:

```
=addmonths(start_date,contract_length, 0)
```

Resultatentabel

| 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

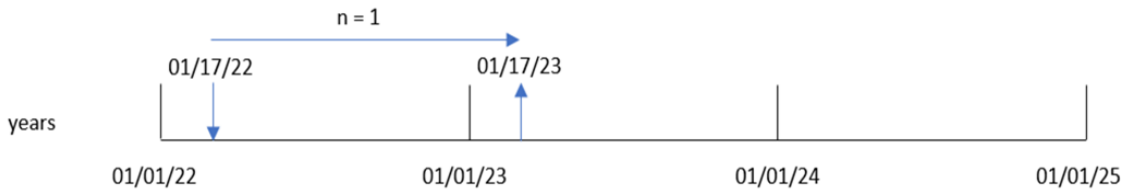
Deze functie retourneert de datum **n** jaar na **startdate** of, als **n** negatief is, de datum **n** jaar vóór **startdate**.

**Syntaxis:**

```
AddYears(startdate, n)
```

**Retourgegevenstypen:** dubbele waarde

*Voorbeelddiagram van de `addyears()`-functie*




De `addyears()`-functie voegt een gedefinieerd aantal jaar, `n`, toe aan of trekt dit aantal af van een `startdate`. Vervolgens wordt de resulterende datum geretourneerd.

Argumenten

| Argument | Beschrijving |
|---|---|
| startdate | De startdatum als tijdstempel, bijvoorbeeld '2012-10-12'. |
| n | Aantal jaren als een positief of negatief geheel getal. |

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `addyears ('01/29/2010',3)` | Retourneert '01/29/2013'. |
| `addyears ('01/29/2010',-1)` | Retourneert '01/29/2009'. |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – Eenvoudig voorbeeld

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties tussen 2020 en 2022 die wordt geladen in de tabel `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.
- Er wordt een veld gemaakt, `two_years_later`, dat de datum retourneert voor twee jaar nadat de transactie plaatsvond.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

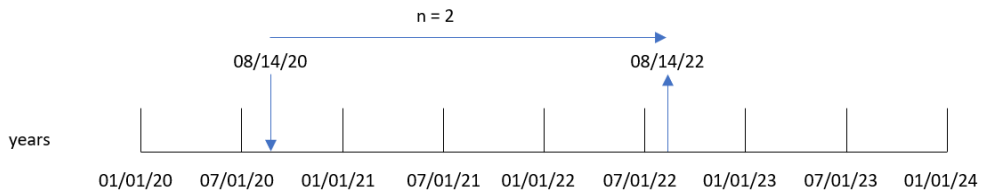
- `date`
- `two_years_later`

Resultatentabel

| date | two_years_later |
|---|---|
| 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 |

Het veld `two_years_later` wordt gemaakt in de vorige load-instructie met behulp van de functie `addyears()`. Het eerste opgegeven argument geeft aan welke datum wordt geëvalueerd. Het tweede argument is het aantal jaar dat moet worden opgeteld bij of afgetrokken van de begindatum. In dit geval is de waarde 2 opgegeven.

*Diagram van addyears()-functie, basisvoorbeeld*




Transactie 8193 vond plaats op 14 augustus 2020. Daarom retourneert de `addyears()`-functie 14 augustus 2022 voor het veld `two_years_later`.

## Voorbeeld 2 – diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met een set transacties tussen 2020 en 2022 die wordt geladen in een tabel met de naam `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.

Maak in een diagramobject een meting aan, `prior_year_date`, die de datum retourneert die één jaar eerder ligt dan de datum waarop de transactie heeft plaatsgevonden.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `date`.

Maak de volgende meting aan om de datum te berekenen die één jaar eerder ligt per transactie:

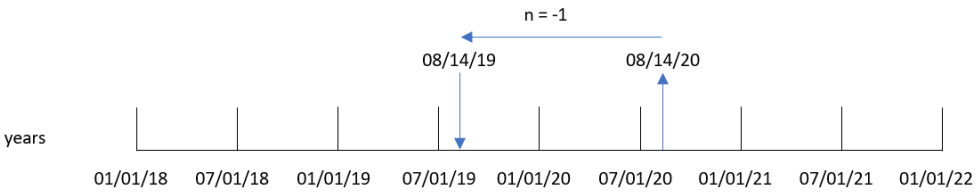
```
=addyears(date,-1)
```

Resultatentabel

| date | =addyears(date,-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 |
| 03/07/2022 | 03/07/2021 |
| 03/11/2022 | 03/11/2021 |

De meting `one_year_prior` wordt in het diagramobject gemaakt met behulp van de functie `addyears()`. Het eerste opgegeven argument geeft aan welke datum wordt geëvalueerd. Het tweede argument is het aantal jaar dat moet worden opgeteld bij of afgetrokken van de `startdate`. In dit geval is de waarde -1 opgegeven.

*Diagram van `addyears()`-functie, diagramobjectvoorbeeld*




Transactie 8193 vond plaats op 14 augustus. Daarom retourneert de `addyears()`-functie 14 augustus 2019 voor het veld `one_year_prior`.

## Voorbeeld 3 – Scenario

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset die wordt geladen in een tabel met de naam `Warranties`.
- Informatie met de product-id, aankoopdatum, garantieduur en aankoopprijs.

De eindgebruiker wil graag een diagramobject dat de einddatum van de garantieperiode van elk product op product-id weergeeft.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `product_id`
- `purchase_date`
- `warranty_length`

Maak de volgende meting aan om de einddatum van de garantie per product te berekenen:

```
=addyears(purchase_date,warranty_length)
```

Resultatentabel

| product_id | purchase_date | warranty_length | =addyears(purchase_date,warranty_length) |
|---|---|---|---|
| 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 | 05/21/2022 |
| 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 | 03/17/2023 |
| 8199 | 04/23/2021 | 3 | 04/23/2024 |
| 8200 | 05/04/2021 | 4 | 05/04/2025 |
| 8201 | 06/30/2021 | 3 | 06/30/2024 |
| 8202 | 07/26/2021 | 4 | 07/26/2025 |

| product_id | purchase_date | warranty_length | =addyears(purchase_date,warranty_length) |
|---|---|---|---|
| 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

De functie **age** retourneert de leeftijd op het tijdstip van de **timestamp** (in voltooide jaren) van iemand die is geboren op **date_of_birth**.

**Syntaxis:**

```
age(timestamp, date_of_birth)
```

Kan een uitdrukking zijn.

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| **timestamp** | De tijdstempel, of uitdrukking die wordt herleid tot een tijdstempel, tot waar het voltooide aantal jaren moet worden berekend. |
| **date_of_birth** | Geboortedatum van de persoon van wie de leeftijd wordt berekend. Kan een uitdrukking zijn. |

Voorbeelden en resultaten:

Deze voorbeelden gebruiken de datumopmaak **DD/MM/YYYY**. De datumopmaak wordt opgegeven in de opdracht **SET DateFormat** boven aan uw script voor het laden van gegevens. Wijzig de opmaak in de voorbeelden zodat deze in uw behoeften voorziet.

Scriptvoorbeelden

| Voorbeeld | Resultaat |
|---|---|
| `age('25/01/2014', '29/10/2012')` | Retourneert 1. |
| `age('29/10/2014', '29/10/2012')` | Retourneert 2. |

**Voorbeeld:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
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;
```

De resulterende tabel bevat de geretourneerde waarden van age voor elk van de records in de tabel.

Resultatentabel

| 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

Converteert een UTC- of GMT-tijdstempel naar een lokale tijd met twee representaties. De plaats kan een reeks steden, plaatsen en tijdzones overal ter wereld zijn.

**Syntaxis:**

```
ConvertToLocalTime(timestamp [, place [, ignore_dst=false]])
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **timestamp** | De tijdstempel of uitdrukking die wordt herleid tot een tijdstempel waarnaar moet worden geconverteerd. |
| **place** | Een plaats of tijdzone uit de tabel met geldige plaatsen en tijdzones hieronder. U kunt ook gebruikmaken van GMT of UTC om de lokale tijd te definiëren. De volgende waarden en tijdverschuivingsbereiken zijn geldig:<br>• GMT<br>• GMT-12:00 - GMT-01:00<br>• GMT+01:00 - GMT+14:00<br>• UTC<br>• UTC-12:00 - UTC-01:00<br>• UTC+01:00 - UTC+14:00<br><br>*Als u een zomertijdoffset (DST) (d.w.z. u specificeert een **ignore_dst**-argumentwaarde die evalueert naar False) gebruikt, moet u een plek specificeren in plaats van een GMT-offset, in het **place**-argument. De reden hiervoor is dat het aanpassen van de zomertijd breedtegegevens vereist naast de lengtegegevens die door een GMT-offset worden verstrekt. Zie GMT-offsets gebruiken in combinatie met zomertijd (page 741) voor meer informatie.*<br><br>*U kunt alleen standaard tijdverschuivingen gebruiken. Het is niet mogelijk een willekeurige tijdverschuiving te gebruiken, bijvoorbeeld GMT-04:27.* |
| **ignore_dst** | Als dit argument evalueert in True, wordt DST (zomertijd) genegeerd. Geldige argumentwaarden die in True evalueren, zijn onder meer `-1` en `True()`.<br><br>Als dit argument evalueert in False, wordt het tijdstempel aangepast voor de zomertijd. Geldige argumentwaarden die in False evalueren, zijn onder meer `0` en `False()`.<br><br>Als de **ignore_dst**-argumentwaarde ongeldig is, evalueert de functie de uitdrukking alsof de **ignore_dst**-waarde evalueert in True. Als de **ignore_dst**-argumentwaarde niet is gespecificeerd, evalueert de functie de uitdrukking alsof de **ignore_dst**-waarde evalueert in False. |

Geldige plaatsen en tijdzones

| 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 |
| 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 |

| A-C | D-K | L-R | S-Z |
|---|---|---|---|
| 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 | - | - | - |

Voorbeelden en resultaten:

Scriptvoorbeelden

| Voorbeeld | Resultaat |
|---|---|
| `ConvertToLocalTime('2023-08-14 08:39:47','Paris')` | Retourneert '2023-08-14 10:39:47' en het corresponderende interne tijdstempel. |
| `ConvertToLocalTime(UTC(), 'Stockholm')` | Retourneert de tijd voor Stockholm en past deze aan de zomertijd aan. |
| `ConvertToLocalTime(UTC(), 'Stockholm', -1)` | Retourneert de tijd voor Stockholm zonder deze aan de zomertijd aan te passen. |
| `ConvertToLocalTime(UTC(), 'GMT-05:00')` | Retourneert de tijd voor de Noord-Amerikaanse oostkust, bijvoorbeeld New York. Er wordt geen aanpassing gedaan voor zomertijd omdat er een GMT-offset is opgegeven in plaats van een plaats. |
| `ConvertToLocalTime(UTC(), 'New York', -1)` | Retourneert de tijd voor de Noord-Amerikaanse oostkust (New York) zonder aanpassing voor zomertijd. |
| `ConvertToLocalTime(UTC(), 'New York', True())` | Retourneert de tijd voor de Noord-Amerikaanse oostkust (New York) zonder aanpassing voor zomertijd. |
| `ConvertToLocalTime(UTC(), 'New York', 0)` | Retourneert de tijd voor de Noord-Amerikaanse oostkust (New York) met aanpassing voor zomertijd. |

| Voorbeeld | Resultaat |
|---|---|
| `ConvertToLocalTime(UTC(), 'New York', False())` | Retourneert de tijd voor de Noord-Amerikaanse oostkust (New York) met aanpassing voor zomertijd. |

### GMT-offsets gebruiken in combinatie met zomertijd

Sinds de implementatie van ICU-bibliotheken (International Components for Unicode) in Qlik Sense, vereist het gebruik van of GMT-offsets (Greenwich Mean Time) in combinatie met zomertijd (DST) aanvullende breedtegegevens.

GMT is een lengte-offset (oost-west) en zomertijd (DST) is a een breedte-offset (noord-zuid). Zo delen Helsinki (Finland) en Johannesburg (Zuid-Afrika) dezelfde GMT+02:00 offset, maar ze delen niet dezelfde zomertijdoffset. Dit betekent dat iedere zomertijdoffset naast de GMT-offset informatie vereist over de breedtepositie van de plaatselijke tijdzone (geografische tijdzoneinvoer) voor volledige informatie over plaatselijke zomertijdomstandigheden.

## day

Deze functie retourneert een geheel getal dat de dag representeert als de breuk van **expression** wordt geïnterpreteerd als een datum volgens de standaardinterpretatie van getallen.

De functie retourneert de dag van de maand voor een specifieke datum. Dit wordt vaak gebruikt om een dagveld af te leiden als onderdeel van een kalenderdimensie.

**Syntaxis:**

```
day(expression)
```

**Retourgegevenstypen:** geheel getal

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `day( 1971-10-12 )` | retourneert 12 |
| `day( 35648 )` | retourneert 6, omdat 35648 = 1997-08-06 |

### Voorbeeld 1 - gegevensverzameling DateFormat (script)

Load-script en resultaten

**Overzicht**

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensverzameling met de naam `Master_Calendar`. De `DateFormat` systeemvariabele is ingesteld op DD/MM/JJJJ.
- Een voorafgaande lading maakt een extra veld met de naam `day_of_month` met behulp van de `day()` functie.
- Een extra veld met de naam `long_date` met behulp van de `date()` functie voor het weergeven van de volledige naam van de maand.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `long_date`
- `day_of_month`

Resultatentabel

| date | long_date | day_of_month |
|---|---|---|
| 03/11/2022 | 11-March- 2022 | 11 |
| 03/12/2022 | 11-March- 2022 | 12 |
| 03/13/2022 | 13-March- 2022 | 13 |
| 03/14/2022 | 14-March- 2022 | 14 |

| date | long_date | day_of_month |
|---|---|---|
| 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 |

De dag van maand wordt juist geëvalueerd door de `day()` functie in het script.

## Voorbeeld 2 – ANSI-datums (script)

Load-script en resultaten

### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensverzameling met de naam `Master_Calendar`. De DateFormat systeemvariabele DD/MM/JJJJ wordt gebruikt. De datums die zijn opgenomen in de gegevensverzameling hebben echter een standaard ANSI-datumnotatie.
- Een voorafgaande lading maakt een extra veld met de naam `day_of_month` met behulp van de `date()`functie.
- Een extra veld met de naam `long_date` met behulp van de `date()` functie voor het weergeven van de datum met de volledige naam van de maand.

### Load-script

```
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
 ];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `long_date`
- `day_of_month`

Resultatentabel

| date | long_date | day_of_month |
|---|---|---|
| 03/11/2022 | 11-March- 2022 | 11 |
| 03/12/2022 | 11-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 |

De dag van maand wordt juist geëvalueerd door de `day()` functie in het script.

## Voorbeeld 3 – Datums zonder notatie (script)

Load-script en resultaten

#### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensverzameling met de naam `Master_Calendar`. De `DateFormat` systeemvariabele DD/MM/JJJJ wordt gebruikt.
- Een voorafgaande lading maakt een extra veld met de naam `day_of_month` met behulp van de `day()`functie.
- De originele datum zonder notatie, met de naam `unformatted_date`.
- Een extra veld met de naam `long_date` met de `date()` wordt gebruikt om de numerieke datum om te zetten in een datumveld met notatie.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `unformatted_date`
- `long_date`
- `day_of_month`

Resultatentabel

| unformatted_date | long_date | day_of_month |
|---|---|---|
| 44868 | 03-November- 2022 | 3 |
| 44898 | 03-December- 2022 | 3 |

| unformatted_date | long_date | day_of_month |
|---|---|---|
| 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 | 02-May- 2023 | 2 |
| 45068 | 22 mei 2023 | 22 |
| 45078 | 01-June- 2023 | 1 |

De dag van maand wordt juist geëvalueerd door de `day()` functie in het script.

## Voorbeeld 4 – Vervalmaand berekenen (diagram)

Load-script en diagramuitdrukking

### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensverzameling met bestellingen die in maart zijn geplaatst, met de naam `Orders`. De tabel bevat drie velden.
    - id
    - order_date
    - amount

### Load-script

```
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
];
```

**Resultaten**

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:`order_date`.

Om de leverdatum te berekenen, maakt u deze meting: `=day(order_date+5)`.

Resultatentabel

| 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 |

De `day()` functie bepaalt op correcte wijze dat een bestelling die op 11 maart is geplaatst op basis van een leveringsperiode van 5 dagen op 16 maart wordt bezorgd.

## dayend

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de laatste milliseconde van de dag in **time**. De standaarduitvoernotatie is de **TimestampFormat** die is ingesteld in het script.

**Syntaxis:**

```
DayEnd(time[, [period_no[, day_start]])
```

### Wanneer gebruiken

De functie `dayend()` wordt doorgaans gebruikt als onderdeel van een uitdrukking als de gebruiker wil dat in de berekening een deel van de dag wordt gebruikt dat nog niet heeft plaatsgevonden. Bijvoorbeeld om de totale kosten te berekenen die gedurende de dag nog gemaakt moeten worden.

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **time** | De te evalueren tijdstempel. |
| **period_no** | **period_no** is een geheel getal, of een uitdrukking die wordt herleid tot een geheel getal, waarbij de waarde 0 de dag aangeeft die **time** bevat. Negatieve waarden in **period_no** geven voorafgaande dagen aan en positieve waarden geven volgende dagen aan. |
| **day_start** | Als u wilt opgeven dat dagen niet om middernacht beginnen, geeft u een verschuiving aan als een fractie van een dag in **day_start**. Bijvoorbeeld 0.125 om 03.00 uur aan te geven.<br>Met andere woorden: om de offset te maken, verdeelt u de begintijd met 24 uur.<br>Bijvoorbeeld: om een dag te laten beginnen om 7.00 uur, gebruikt u de fractie 7/24. |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| dayend('01/25/2013 16:45:00') | Retourneert 01/25/2013 23:59:59. PM |
| dayend('01/25/2013 16:45:00', -1) | Retourneert 01/24/2013 23:59:59. PM |
| dayend('01/25/2013 16:45:00', 0, 0.5) | Retourneert 01/26/2013 11:59:59. PM |

## Voorbeeld 1 - Basisscript

Load-script en resultaten

### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensverzameling met een lijst met datums wordt geladen in een tabel met de naam "Calendar".
- De standaard `DateFormat`-systeemvariabele `(MM/DD/YYYY)`.
- Een voorafgaande lading om een extra veld te maken met de naam `EOD_timestamp` met behulp van de `dayend()`-functie.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `EOD_timestamp`

Resultatentabel

| date | Tijdstempel einde dag |
|---|---|
| 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 |

Zoals u in de bovenstaande tabel kunt zien, wordt het tijdstempel einde dag gegenereerd voor iedere datum in uw gegevensset. Het tijdstempel is de indeling van de systeemvariabele `TimestampFormat M/D/YYYY h:mm:ss[.fff] TT`.

## Voorbeeld 2 – period_no

### Load-script en resultaten

#### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

U laadt een gegevensset met serviceboekingen in een tabel genaamd Services.

De gegevensset bevat de volgende velden:

- `service_id`
- `service_date`
- `amount`

U maakt twee nieuwe velden in de tabel:

- `deposit_due_date`: de datum waarop de storting moet zijn ontvangen. Dit is het einde van de dag, drie dagen vóór de `service_date`.
- `final_payment_due_date`: de datum waarop de laatste betaling moet zijn ontvangen. Dit is het einde van de dag, zeven dagen na de `service_date`.

De twee velden hierboven zijn gemaakt in een voorafgaande lading met gebruik van de `dayend()`-functie. Ze geven de eerste twee parameters, `time` en `period_no`.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- service_date
- deposit_due_date
- final_payment_due_date

Resultatentabel

| service_date | deposit_due_date | final_payment_due_date |
|---|---|---|
| 03/11/2022 9:25:14 AM | 3/8/2022 11:59:59 PM | 3/18/2022 11:59:59 PM |
| 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 |

| service_date | deposit_due_date | final_payment_due_date |
|---|---|---|
| 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 |

De waarden van de nieuwe velden hebben de `TimestampFormat M/D/YYYY h:mm:ss[.fff] TT`. Omdat de functie `dayend()` is gebruikt, zijn de tijdstempelwaarden allemaal de laatste milliseconde van de dag.

De vervaldatumwaarden voor de storting zijn drie dagen vóór de servicedatum, omdat het tweede argument dat in de `dayend()`-functie wordt doorgegeven negatief is.

De vervaldatumwaarden voor de laatste betaling zijn zeven dagen na de servicevervaldatum, omdat het tweede argument dat in de `dayend()`-functie wordt doorgegeven negatief is.

## Voorbeeld 3 – day_start-script

### Load-script en resultaten

#### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

De gegevensset en het scenario die in dit voorbeeld worden gebruikt, zijn dezelfde als in het vorige voorbeeld.

Net als in het vorige voorbeeld, maakt u twee nieuwe velden:

- `deposit_due_date`: de datum waarop de storting moet zijn ontvangen. Dit is het einde van de dag, drie dagen vóór de `service_date`.
- `final_payment_due_date`: de datum waarop de laatste betaling moet zijn ontvangen. Dit is het einde van de dag, zeven dagen na de `service_date`.

Uw bedrijf wil echter volgens een beleid werken waarbij de werkdag om 5:00 PM begint en de volgende dag om 5:00 PM eindigt. Uw bedrijf kan dan de transacties bewaken die binnen die werkuren plaatsvinden.

Om deze vereisten te realiseren, worden de twee bovenstaande velden gemaakt in een voorafgaande load met de `dayend()`-functie en de drie parameters, `time`, `period_no` en `day_start`.

#### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- service_date
- deposit_due_date
- final_payment_due_date

Resultatentabel

| 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 |
| 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 |

De datums blijven zoals in voorbeeld 2, maar de datums hebben nu een tijdstempel van de laatste milliseconde vóór 5:00 PM, omdat de waarde van het derde argument, `day_start`, dat in de `dayend()`-functie is doorgegeven, 17/24 was.

## Voorbeeld 4 – diagramvoorbeeld

### Load-script en diagramuitdrukking

#### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

De gegevensset en het scenario die in dit voorbeeld worden gebruikt, zijn dezelfde als in de vorige twee voorbeelden. Uw bedrijf wil graag volgens een beleid werken waarbij de werkdag om 5:00 PM begint en de volgende dag om 5:00 PM eindigt.

Net als in het vorige voorbeeld, maakt u twee nieuwe velden:

- `deposit_due_date`: de datum waarop de storting moet zijn ontvangen. Dit is het einde van de dag, drie dagen vóór de `service_date`.
- `final_payment_due_date`: de datum waarop de laatste betaling moet zijn ontvangen. Dit is het einde van de dag, zeven dagen na de `service_date`.

#### Load-script

```
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
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

`service_date`.

Om het `deposit_due_date`-veld te maken, maakt u deze meting.

`=dayend(service_date,-3,17/24).`

Vervolgens maakt u om het `final_payment_due_date`-veld te maken, deze meting:

`=dayend(service_date,7,17/24).`

Resultatentabel

| 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 |

De waarden van de nieuwe velden hebben de `TimestampFormat M/D/YYYY h:mm:ss[.fff] TT`. Omdat de functie `dayend()` is gebruikt, zijn de tijdstempelwaarden allemaal de laatste milliseconde van de dag.

De vervaldatumwaarden voor de belating zijn drie dagen vóór de servicedatum, omdat het tweede argument dat in de `dayend()`-functie wordt doorgegeven negatief is.

De vervaldatumwaarden voor de laatste betaling zijn zeven dagen na de servicevervaldatum, omdat het tweede argument dat in de `dayend()`-functie wordt doorgegeven negatief is.

De datums hebben nu een tijdstempel van de laatste milliseconde vóór 5:00 PM omdat de waarde van het derde argument, `day_start`, dat in de `dayend()`-functie is doorgegeven, 17/24 was.

Argumenten

| Argument | Beschrijving |
|---|---|
| **time** | De te evalueren tijdstempel. |
| **period_no** | **period_no** is een geheel getal, of een uitdrukking die wordt herleid tot een geheel getal, waarbij de waarde 0 de dag aangeeft die **time** bevat. Negatieve waarden in **period_no** geven voorafgaande dagen aan en positieve waarden geven volgende dagen aan. |
| **day_start** | Als u wilt opgeven dat dagen niet om middernacht beginnen, geeft u een verschuiving aan als een fractie van een dag in **day_start**. Bijvoorbeeld 0.125 om 03.00 uur aan te geven. |

## daylightsaving

Retourneert de huidige aanpassing voor zomertijd zoals die in Windows is gedefinieerd.

**Syntaxis:**

```
DaylightSaving( )
```

**Retourgegevenstypen:** dual

**Voorbeeld:**

```
daylightsaving( )
```

## dayname

Deze functie retourneert een datum met een onderliggende numerieke waarde die overeenkomt met een tijdstempel van de eerste milliseconde van de dag met **time**.

**Syntaxis:**

```
DayName(time[, period_no [, day_start]])
```

**Retourgegevenstypen:** dual

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| **time** | De te evalueren tijdstempel. |
| **period_no** | **period_no** is een geheel getal, of een uitdrukking die wordt herleid tot een geheel getal, waarbij de waarde 0 de dag aangeeft die **time** bevat. Negatieve waarden in **period_no** geven voorafgaande dagen aan en positieve waarden geven volgende dagen aan. |
| **day_start** | Als u wilt opgeven dat dagen niet om middernacht beginnen, geeft u een verschuiving aan als een fractie van een dag in **day_start**. Bijvoorbeeld 0.125 om 03.00 uur aan te geven. |

Voorbeelden en resultaten:

Deze voorbeelden gebruiken de datumopmaak **DD/MM/YYYY**. De datumopmaak wordt opgegeven in de opdracht **SET DateFormat** boven aan uw script voor het laden van gegevens. Wijzig de opmaak in de voorbeelden zodat deze in uw behoeften voorziet.

Scriptvoorbeelden

| Voorbeeld | Resultaat |
|---|---|
| `dayname('25/01/2013 16:45:00')` | Retourneert 25/01/2013. |

| Voorbeeld | Resultaat |
| --- | --- |
| `dayname('25/01/2013 16:45:00', -1)` | Retourneert 24/01/2013. |
| `dayname('25/01/2013 16:45:00', 0, 0.5 )` | Retourneert 25/01/2013.<br><br>Als de volledige tijdstempel wordt weergegeven, verschijnt de onderliggende numerieke waarde die overeenkomt met '25/01/2013 12:00:00.000. |

**Voorbeeld:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

In dit voorbeeld wordt de naam van de dag gemaakt aan de hand van de tijdstempel die het einde van de dag na elke factuurdatum in de tabel aangeeft.

```
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;
```

De resulterende tabel bevat de oorspronkelijke datums en een kolom met de retourwaarde van de functie dayname(). U kunt de volledige tijdstempel weergeven door de opmaak op te geven in het eigenschappenvenster.

Resultatentabel

| 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

Deze functie berekent het dagnummer van het kwartaal waarbinnen een tijdstempel valt. Deze functie wordt gebruikt wanneer een masterkalender wordt gemaakt.

**Syntaxis:**

```
DayNumberOfQuarter(timestamp[,start_month])
```

**Retourgegevenstypen:** geheel getal

Argumenten

| Argument | Beschrijving |
|---|---|
| **timestamp** | De te evalueren datum of tijdstempel. |

| Argument | Beschrijving |
|---|---|
| **start_month** | Door een **start_month** tussen 2 en 12 op te geven (1 indien weggelaten), kunt u het begin van het jaar naar voren verplaatsen naar de eerste dag van een andere maand. Als u bijvoorbeeld wilt werken met een boekjaar dat begint op 1 maart, geeft u **start_month** = 3 op. |

Deze voorbeelden gebruiken de datumopmaak **DD/MM/YYYY**. De datumopmaak wordt opgegeven in de opdracht **SET DateFormat** boven aan uw script voor het laden van gegevens. Wijzig de opmaak in de voorbeelden zodat deze in uw behoeften voorziet.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `DayNumberOfQuarter('12/09/2014')` | Retourneert 74, het dagnummer van het huidige kwartaal. |
| `DayNumberOfQuarter ('12/09/2014',3)` | Retourneert 12, het dagnummer van het huidige kwartaal. In dit geval begint het eerste kwartaal met maart (omdat bij start_month 3 is opgegeven). Dit betekent dat het huidige kwartaal het derde kwartaal is, dat begint op 1 september. |

## Voorbeeld 1 – Januari begin van het jaar (script)

Load-script en resultaten

### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een eenvoudige gegevensverzameling met een lijst met datums, die wordt geladen in een tabel met de naam `Calendar`. De standaard `DateFormat`-systeemvariabele mm/dd/jjjj wordt gebruikt.
- Een voorafgaande lading maakt een extra veld met de naam `DayNrQtr` met behulp van de `DayNumberOfQuarter()`functie.

Naast de datum worden er geen extra parameters aan de functie toegevoegd.

### Load-script

```
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
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `daynrqtr`

Resultatentabel

| date | 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 |

De eerste dag van het jaar is 1 januari omdat er geen tweede argument is doorgegeven in de `DayNumberOfQuarter()`-functie.

1 Januari is de eerste dag van het kwartaal en 1 februari is de 32e dag van het kwartaal. 31 maart is de 91e en laatste dag van het kwartaal. 1 April is de eerste dag van het tweede kwartaal.

## Voorbeeld 2 – Februari begin van het jaar (script)

#### Load-script en resultaten

#### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset als in het eerste voorbeeld.
- De standaard `DateFormat`-systeemvariabele mm/dd/jjjj wordt gebruikt.
- Een `start_month`-argument dat begint op 1 februari. Dit stelt het financiële jaar in op 1 februari.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `daynrqtr`

Resultatentabel

| date | daynrqtr |
|---|---|
| 01/01/2022 | 62 |
| 01/10/2022 | 71 |
| 01/31/2022 | 92 |
| 02/01/2022 | 1 |
| 02/10/2022 | 10 |
| 02/28/2022 | 28 |

| date | daynrqtr |
|---|---|
| 03/01/2022 | 30 |
| 03/31/2022 | 60 |
| 04/01/2022 | 61 |

De eerste dag van het jaar is 1 februari omdat het tweede argument dat is doorgegeven in de `DayNumberOfQuarter()`-functie 2 is.

Het eerste kwartaal van het jaar loopt van februari tot en met april en het vierde kwartaal van november tot en met januari. Dit wordt weergegeven in de tabel met resultaten. Hierbij is 1 februari de eerste dag van het kwartaal en 31 januari is de 92e en laatste dag van het kwartaal.

## Voorbeeld 3 – Januari begin van het jaar (diagram)

Load-script en diagramuitdrukking

### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset als in het eerste voorbeeld.
- De standaard `DateFormat`-systeemvariabele mm/dd/jjjj wordt gebruikt.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De waarde van de dag van het kwartaal wordt berekend via een meting in een diagramobject.

### Load-script

```
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
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `date`.

Maak de volgende meting:

```
=daynumberofquarter(date)
```

Resultatentabel

| date | =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 |

De eerste dag van het jaar is 1 januari omdat er geen tweede argument is doorgegeven in de `DayNumberOfQuarter()`-functie.

1 Januari is de eerste dag van het kwartaal en 1 februari is de 32e dag van het kwartaal. 31 maart is de 91e en laatste dag van het kwartaal. 1 April is de eerste dag van het tweede kwartaal.

## Voorbeeld 4 – Februari begin van het jaar (diagram)

Load-script en diagramuitdrukking

#### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset als in het eerste voorbeeld.
- De standaard `DateFormat`-systeemvariabele mm/dd/jjjj wordt gebruikt.
- Het financiële jaar loopt van 1 februari tot en met 31 januari.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De waarde van de dag van het kwartaal wordt berekend via een meting in een diagramobject.

### Load-script

```
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
];
```

### Diagramobject

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `date`.

Maak de volgende meting:

```
=daynumberofquarter(date,2)
```

### Resultaten

Resultatentabel

| date | =daynumberofquarter(date,2) |
|---|---|
| 01/01/2022 | 62 |
| 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 |

De eerste dag van het jaar is 1 februari omdat het tweede argument dat is doorgegeven in de `DayNumberOfQuarter()`-functie 2 is.

Het eerste kwartaal van het jaar loopt van februari tot en met april en het vierde kwartaal van november tot en met januari. Dit wordt aangeduid in de tabel met resultaten. Hierbij is 1 februari de eerste dag van het kwartaal en 31 januari is de 92e en laatste dag van het kwartaal.

## daynumberofyear

Deze functie berekent het dagnummer van het jaar waarbinnen een tijdstempel valt. De berekening wordt uitgevoerd vanaf de eerste milliseconde van de eerste dag van het jaar, maar de beginmaand kan worden verschoven.

**Syntaxis:**

```
DayNumberOfYear(timestamp[,start_month])
```

**Retourgegevenstypen:** geheel getal

Argumenten

| Argument | Beschrijving |
|---|---|
| **timestamp** | De te evalueren datum of tijdstempel. |
| **start_month** | Door een **start_month** tussen 2 en 12 op te geven (1 indien weggelaten), kunt u het begin van het jaar naar voren verplaatsen naar de eerste dag van een andere maand. Als u bijvoorbeeld wilt werken met een boekjaar dat begint op 1 maart, geeft u **start_month** = 3 op. |

Deze voorbeelden gebruiken de datumopmaak **DD/MM/YYYY**. De datumopmaak wordt opgegeven in de opdracht **SET DateFormat** boven aan uw script voor het laden van gegevens. Wijzig de opmaak in de voorbeelden zodat deze in uw behoeften voorziet.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `DayNumberOfYear( '12/09/2014' )` | Retourneert 256, het dagnummer geteld vanaf de eerste dag van het jaar. |
| `DayNumberOfYear( '12/09/2014',3 )` | Retourneert 196, het nummer van de dag geteld vanaf 1 maart. |

### Voorbeeld 1 – Januari begin van het jaar (script)

Load-script en resultaten

**Overzicht**

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een eenvoudige gegevensverzameling met een lijst met datums, die wordt geladen in een tabel met de naam `Calendar`. De standaard `DateFormat`-systeemvariabele mm/dd/jjjj wordt gebruikt.
- Een voorafgaande lading maakt een extra veld met de naam `daynryear` met behulp van de `DayNumberOfYear()`functie.

Naast de datum worden er geen extra parameters aan de functie toegevoegd.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `daynryear`

Resultatentabel

| date | daynryear |
|---|---|
| 01/01/2022 | 1 |
| 01/10/2022 | 10 |
| 01/31/2022 | 31 |
| 02/01/2022 | 32 |
| 02/10/2022 | 41 |

| date | daynryear |
|---|---|
| 06/30/2022 | 182 |
| 07/26/2022 | 208 |
| 10/31/2022 | 305 |
| 11/01/2022 | 306 |
| 12/31/2022 | 366 |

De eerste dag van het jaar is 1 januari omdat er geen tweede argument is doorgegeven in de `DayNumberOfYear()`-functie.

1 Januari is de eerste dag van het kwartaal en 1 februari is de 32e dag van het jaar. 30 juni is de 182e dag en 31 december is de 366e en laatste dag van het jaar.

## Voorbeeld 2 – November begin van het jaar (script)

Load-script en resultaten

### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset als in het eerste voorbeeld.
- De standaard `DateFormat`-systeemvariabele mm/dd/jjjj wordt gebruikt
- Een `start_month`-argument dat begint op 1 november. Dit stelt het financiële jaar in op 1 november.

### Load-script

```
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
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `daynryear`

Resultatentabel

| date | 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 |

De eerste dag van het jaar is 1 november omdat het tweede argument dat is doorgegeven in de `DayNumberOfYear()`-functie 11 is.

1 Januari is de eerste dag van het kwartaal en 1 februari is de 32e dag van het jaar. 30 juni is de 182e dag en 31 december is de 366e en laatste dag van het jaar.

## Voorbeeld 3 – Januari begin van het jaar (diagram)

Load-script en diagramuitdrukking

#### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset als in het eerste voorbeeld.
- De standaard `DateFormat`-systeemvariabele mm/dd/jjjj wordt gebruikt.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De waarde van de dag van het kwartaal wordt berekend via een meting in een diagramobject.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `date`.

Maak de volgende meting:

```
=daynumberofyear(date)
```

Resultatentabel

| date | =daynumberofyear(date) |
|---|---|
| 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 |

De eerste dag van het jaar is 1 januari omdat er geen tweede argument is doorgegeven in de `DayNumberOfYear()`-functie.

1 Januari is de eerste dag van het jaar en 1 februari is de 32e dag van het jaar. 30 juni is de 182e dag en 31 december is de 366e en laatste dag van het jaar.

## Voorbeeld 4 – November begin van het jaar (diagram)

Load-script en diagramuitdrukking

### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset als in het eerste voorbeeld.
- De standaard `DateFormat`-systeemvariabele mm/dd/jjjj wordt gebruikt.
- Het financiële jaar loopt van 1 november tot en met 31 oktober.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De waarde van de dag van het jaar wordt berekend via een meting in een diagramobject.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `date`.

Maak de volgende meting:

```
=daynumberofyear(date)
```

Resultatentabel

| date | =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 |

De eerste dag van het jaar is 1 november omdat het tweede argument dat is doorgegeven in de `DayNumberOfYear()`-functie 11 is.

Het financiële jaar loopt tussen november en oktober. Dit wordt weergegeven in de tabel met resultaten. Hierbij is 1 november de eerste dag van het jaar en 31 oktober is de 366e en laatste dag van het jaar.

## daystart

Deze functie retourneert een waarde die overeenkomt met een tijdstempel met de eerste milliseconde van de dag in het argument **time**. De standaarduitvoernotatie is de **TimestampFormat** die is ingesteld in het script.

**Syntaxis:**

```
DayStart(time[, [period_no[, day_start]])
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **time** | De te evalueren tijdstempel. |
| **period_no** | **period_no** is een geheel getal, of een uitdrukking die wordt herleid tot een geheel getal, waarbij de waarde 0 de dag aangeeft die **time** bevat. Negatieve waarden in **period_no** geven voorafgaande dagen aan en positieve waarden geven volgende dagen aan. |
| **day_start** | Als u wilt opgeven dat dagen niet om middernacht beginnen, geeft u een verschuiving aan als een fractie van een dag in **day_start**. Bijvoorbeeld 0.125 om 03.00 uur aan te geven.<br>Met andere woorden: om de offset te maken, verdeelt u de begintijd met 24 uur.<br>Bijvoorbeeld: om een dag te laten beginnen om 7.00 uur, gebruikt u de fractie 7/24. |

## Wanneer gebruiken

De functie `daystart()` wordt doorgaans gebruikt als onderdeel van een uitdrukking als de gebruiker wil dat in de berekening het deel van de dag wordt gebruikt dat al is geweest. Het zou bijvoorbeeld kunnen worden gebruikt om het totale salaris te berekenen dat werknemers op een dag tot dusver hebben verdiend.

Deze voorbeelden gebruiken de tijdstempelopmaak `'M/D/YYYY h:mm:ss[.fff] TT'`. De tijdstempelopmaak wordt opgegeven in de opdracht `SET TimeStamp` boven aan uw load-script voor gegevens. Wijzig de opmaak in de voorbeelden zodat deze in uw behoeften voorziet.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `daystart('01/25/2013 4:45:00 PM')` | Retourneert 1/25/2013 12:00:00 AM. |
| `daystart('1/25/2013 4:45:00 PM', -1)` | Retourneert 1/24/2013 12:00:00 AM. |
| `daystart('1/25/2013 16:45:00',0,0.5 )` | Retourneert 1/25/2013 12:00:00 PM. |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 - Eenvoudig voorbeeld

Load-script en resultaten

**Overzicht**

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een eenvoudige gegevensverzameling met een lijst met datums, die wordt geladen in een tabel met de naam `Calendar`.

- De standaard `TimeStampFormat`-systeemvariabele (`(M/D/YYYY h:mm:ss[.fff] TT`) wordt gebruikt.
- Een voorafgaande lading die een extra veld maakt met de naam `SOD_timestamp` met behulp van de `daystart()`-functie.

Naast de datum worden er geen extra parameters aan de functie toegevoegd.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `SOD_timestamp`

Resultatentabel

| date | SOD_timestamp |
|---|---|
| 03/11/2022 1:47:15 AM | 3/11/2022 12:00:00 AM |
| 03/12/2022 4:34:58 AM | 3/12/2022 12:00:00 AM |
| 03/13/2022 5:15:55 AM | 3/13/2022 12:00:00 AM |
| 03/14/2022 9:25:14 AM | 3/14/2022 12:00:00 AM |

| date | SOD_timestamp |
|---|---|
| 03/15/2022 10:06:54 AM | 3/15/2022 12:00:00 AM |
| 03/16/2022 10:44:42 AM | 3/16/2022 12:00:00 AM |
| 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 | 3/19/2022 12:00:00 AM |
| 03/20/2022 6:42:15 PM | 3/20/2022 12:00:00 AM |
| 03/21/2022 7:41:16 PM | 3/21/2022 12:00:00 AM |

Zoals in de bovenstaande tabel te zien is, wordt de eind van de dag-tijdstempel gegenereerd voor iedere datum in uw gegevensset. Het tijdstempel is de indeling van de systeemvariabele `TimestampFormat M/D/YYYY h:mm:ss[.fff] TT`.

## Voorbeeld 2 – period_no

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met parkeerboetes, die wordt geladen in de tabel `Fines`. De gegevensset bevat de volgende velden:
  - `id`
  - `due_date`
  - `number_plate`
  - `amount`
- Een voorafgaande lading met de `daystart()`-functie en met de drie parameters: `time`, `period_no` en `day_start`. Deze voorafgaande lading maakt de volgende twee nieuwe datumvelden aan:
  - Een `early_repayment_period`-datumveld dat zeven dagen vóór de vervaldatum van de betaling begint.
  - Een `late_penalty_period`-datumveld dat veertien dagen na de vervaldatum van de betaling begint.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- due_date
- early_repayment_period
- late_penalty_period

Resultatentabel

| 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 | 6/21/2022 12:00:00 AM | 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 |

De waarden van de nieuwe velden hebben de `TimestampFormat M/DD/YYYY tt`. Omdat de `daystart()`-functie is gebruikt, zijn de tijdstempelwaarden allemaal de eerste milliseconde van de dag.

De vroege betaalperiode-waarden zijn zeven dagen vóór de vervaldatum, omdat het tweede argument dat in de `daystart()`-functie wordt doorgegeven negatief is.

De late betaalperiode-waarden zijn veertien dagen na de vervaldatum, omdat het tweede argument dat in de `daystart()`-functie wordt doorgegeven positief is.

## Voorbeeld 3 – day_start

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het vorige voorbeeld.
- Dezelfde voorafgaande lading als het vorige voorbeeld.

In dit voorbeeld laten we de werkdag elke dag beginnen en eindigen om 7:00 AM.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- due_date
- early_repayment_period
- late_penalty_period

Resultatentabel

| 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 |

De datums hebben nu een tijdstempel van 7:00 AM, omdat de waarde van het `day_start`-argument dat in de `daystart()`-functie is doorgegeven, 7/24 was. Dit stelt het begin van de dag in op 7:00 AM.

Omdat het `due_date`-veld geen tijdstempel heeft, wordt dit behandeld als 12:00 AM, waardoor de dag nog deel uitmaakt van de vorige dag omdat de dagen beginnen en eindigen om 7:00 AM. Daarom begint de vroege-terugbetalingsperiode voor een boete die op 11 februari betaald moet zijn, op 3 februari om 7:00 AM.

## Voorbeeld 4 – Diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Dit voorbeeld gebruikt dezelfde gegevensset en hetzelfde scenario als het vorige voorbeeld.

Maar alleen de originele `Fines`-tabel wordt in de applicatie geladen waarbij de twee extra vervaldatumwaarden worden berekend in een diagramobject.

### Load-script

```
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
];
```

**Resultaten**

**Doe het volgende:**

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `due_date`.
2. Om het `early_repayment_period`-veld te maken, maakt u de volgende meting. `=daystart(due_date,-7,7/24)`
3. Om het `late_penalty_period`-veld te maken, maakt u de volgende meting: =daystart(due_date,14,7/24)

Resultatentabel

| due_date | =daystart(due_date,-7,7/24) | =daystart(due_date,14,7/24) |
|---|---|---|
| 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 |

De waarden van de nieuwe velden hebben de `TimestampFormat M/D/YYYY h:mm:ss[.fff] TT`. Omdat de `daystart()`-functie is gebruikt, komen de tijdstempelwaarden allemaal overeen met de eerste milliseconde van de dag.

De vroege betaalperiode-waarden zijn zeven dagen vóór de vervaldatum, omdat het tweede argument dat in de `daystart()`-functie is doorgegeven, negatief was.

De late betaalperiode-waarden zijn veertien dagen na de vervaldatum, omdat het tweede argument dat in de `daystart()`-functie is doorgegeven, positief was.

De datums hebben een tijdstempel van 7:00 AM, omdat de waarde van het derde argument dat in de `daystart()`-functie, `day_start`, is doorgegeven, 7/24 was.

## firstworkdate

De functie **firstworkdate** retourneert de laatste begindatum waarbij **no_of_workdays** (maandag-vrijdag) kan worden gehaald als niet later wordt gestopt dan **end_date**, waarbij rekening wordt gehouden met een eventueel opgegeven vakantiedagen. **end_date** en **holiday** moeten geldige datums of tijdstempels zijn.

**Syntaxis:**

```
firstworkdate(end_date, no_of_workdays {, holiday} )
```

**Retourgegevenstypen:** geheel getal

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| **end_date** | De tijdstempel van de te evalueren einddatum. |
| **no_of_ workdays** | Het aantal te bereiken werkdagen. |
| **holiday** | Vakantieperioden die moeten worden uitgesloten van werkdagen. Een vakantie wordt aangegeven als een tekenreeks met constante datum. U kunt meerdere vakantiedatums opgegeven, gescheiden door komma's.<br><br>**Voorbeeld:** `'12/25/2013', '12/26/2013', '12/31/2013', '01/01/2014'` |

Voorbeelden en resultaten:

Deze voorbeelden gebruiken de datumopmaak **DD/MM/YYYY**. De datumopmaak wordt opgegeven in de opdracht **SET DateFormat** boven aan uw script voor het laden van gegevens. Wijzig de opmaak in de voorbeelden zodat deze in uw behoeften voorziet.

Scriptvoorbeelden

| Voorbeeld | Resultaat |
|---|---|
| `firstworkdate ('29/12/2014', 9)` | Retourneert '17/12/2014. |
| `firstworkdate ('29/12/2014', 9, '25/12/2014', '26/12/2014')` | Retourneert 15/12/2014 omdat rekening is gehouden met een vakantieperiode van twee dagen. |

**Voorbeeld:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
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;
```

De resulterende tabel bevat de geretourneerde waarden van FirstWorkDate voor elk van de records in de tabel.

Resultatentabel

| 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

Deze functie retourneert de huidige Greenwich Mean Time, zoals afgeleid uit de landinstellingen. Deze functie retourneert waarden in de notatie van de `TimestampFormat`-systeemvariabele.

Wanneer de app opnieuw wordt geladen, wordt iedere load-scripttabel, variabele of diagramobject die/dat gebruikmaakt van de `GMT`-functie aangepast aan de actuele Greenwich Mean Time zoals afgeleid uit de systeemklok.

**Syntaxis:**

```
GMT( )
```

**Retourgegevenstypen:** dubbele waarde

Deze voorbeelden gebruiken de tijdstempelopmaak `M/D/YYYY h:mm:ss[.fff] TT`. De datumopmaak wordt opgegeven in de opdracht `SET TimestampFormat` boven aan uw script voor het laden van gegevens. Wijzig de opmaak in de voorbeelden zodat deze in uw behoeften voorziet.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| GMT() | 3/28/2022 2:47:36 PM |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – Variabele (script)

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad. In dit voorbeeld voor de actuele Greenwich Mean Time met de `GMT`-functie ingesteld als een variabele in het load-script.

### Load-script

```
LET vGMT = GMT();
```

### Resultaten

Laad de gegevens en maak een werkblad. Gebruik het diagramobject **Tekst en afbeelding** om een tekstvak te maken.

Voeg deze meting toe aan het tekstvak:

```
=vGMT
```

Het tekstvak moet een regel tekst met een datum en een tijd bevatten, zoals hieronder:

```
3/28/2022 2:47:36 PM
```

## Voorbeeld 2 – November begin van het jaar (script)

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met bibliotheekboeken die al hadden moeten zijn ingeleverd, die wordt geladen in de tabel `Overdue`. De standaard `DateFormat`-systeemvariabele mm/dd/jjjj wordt gebruikt.
- Er wordt een nieuw veld gemaakt, `days_overdue`, waarmee wordt berekend hoeveel dagen elk boek te laat is.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `due_date`
- `book_id`
- `days_overdue`

Resultatentabel

| 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 |
| 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 |

De waarden in het veld `days_overdue` worden berekend door het verschil te vinden tussen de actuele Greenwich Mean Time, met behulp van de `GMT()`-functie, en de oorspronkelijke vervaldatum. Om alleen het aantal dagen te berekenen, worden de resultaten afgerond op het dichtstbijzijnde hele getal met behulp van de `Floor()`-functie.

## Voorbeeld 3 – Diagramobject (diagram)

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad. Het load-script bevat dezelfde gegevensset als het vorige voorbeeld. De standaard `DateFormat`-systeemvariabele mm/dd/jjjj wordt gebruikt.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De waarde van het aantal te laat-dagen wordt berekend via een meting in een diagramobject.

### Load-script

```
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
];
```

**Resultaten**

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- due_date
- book_id

Maak de volgende meting:

```
=Floor(GMT() - due_date)
```

Resultatentabel

| 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 |

De waarden in het veld `days_overdue` worden berekend door het verschil te vinden tussen de actuele Greenwich Mean Time, met behulp van de `GMT()`-functie, en de oorspronkelijke vervaldatum. Om alleen het aantal dagen te berekenen, worden de resultaten afgerond op het dichtstbijzijnde hele getal met behulp van de `Floor()`-functie.

## hour

Deze functie retourneert een geheel getal dat het uur representeert als de breuk van de **expression** wordt geïnterpreteerd als een datum volgens de standaardinterpretatie van getallen.

**Syntaxis:**

```
hour(expression)
```

**Retourgegevenstypen:** geheel getal

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `hour( '09:14:36' )` | De opgegeven tekenreeks wordt impliciet geconverteerd in een tijdstempel omdat het overeenkomt met de indeling van het tijdstempel dat is gedefinieerd in de variabele TimestampFormat. De expressie retourneert 9. |
| `hour( '0.5555' )` | De expressie retourneert 13 (omdat 0,5555 = 13:19:55 is) |

## Voorbeeld 1 – Variabele (script)

Load-script en resultaten

### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensverzameling met transactie per tijdstempel.
- De standaard `TimeStamp`-systeemvariabele `(M/D/YYYY h:mm:ss[.fff] TT)`.

Maak een veld, `hour`, dat berekent wanneer de aankopen hebben plaatsgevonden.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `hour`

Resultatentabel

| date | hour |
|---|---|
| 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 |

De waarden in het uurveld zijn gemaakt met de `hour()`-functie en geven de datum door als de uitdrukking in het voorafgaande load-instructie.

## Voorbeeld 2 – Diagramobject (diagram)

### Load-script en diagramuitdrukking

### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset als in het eerste voorbeeld.
- De standaard `TimeStamp`-systeemvariabele `(M/D/YYYY h:mm:ss[.fff] TT)`.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De `hour`-waarden worden berekend via een meting in een diagramobject.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `date`.

Maak de volgende meting om het `hour` te berekenen:

```
=hour(date)
```

Resultatentabel

| due_date | =uur(datum) |
|---|---|
| 2022-01-01 22:10:22 | 22 |
| 2022-01-02 08:35:54 | 8 |
| 2022-01-03 05:40:49 | 5 |

| due_date | =uur(datum) |
|---|---|
| 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 |

De waarden voor `hour` worden gemaakt met behulp van de `hour()`-functie en geven de datum door als de uitdrukking in een meting voor het diagramobject.

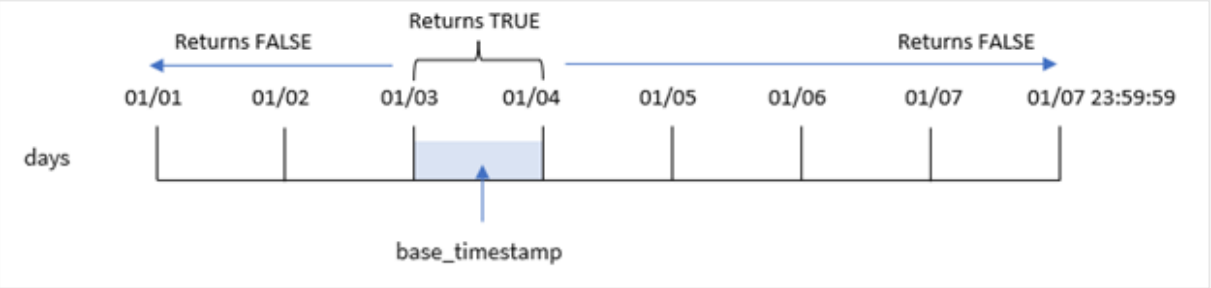
## inday

Deze functie retourneert True als **timestamp** binnen de dag valt die **base_timestamp** bevat.

**Syntaxis:**

```
InDay (timestamp, base_timestamp, period_no[, day_start])
```

*Diagram van de `inday`-functie*




De functie `inday()` gebruikt het argument `base_timestamp` om te identificeren onder welke dag het tijdstempel valt. De begintijd van de dag is standaard middernacht, maar u kunt de begintijd van de dag wijzigen door het argument `day_start` van de functie `inday()` te gebruiken. Zodra deze dag is gedefinieerd, retourneert de functie de booleaanse waarden bij het vergelijken van de opgegeven tijdstempelwaarden van die dag.

### Wanneer gebruiken

De functie `inday()` retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als voorwaarde in een `if expression`. Hierdoor wordt een aggregatie of berekening geretourneerd, afhankelijk van het feit of een geëvalueerde datum plaatsvond op de dag van de desbetreffende tijdstempel.

De functie `inday()` bijvoorbeeld kan worden gebruikt om alle materieel te identificeren die op een bepaalde dag is geproduceerd.

**Retourgegevenstypen:** Booleaanse waarde

In Qlik Sense wordt de booleaanse waarde Waar vertegenwoordigd door -1 en de waarde Onwaar door 0.

Argumenten

| Argument | Beschrijving |
|---|---|
| `timestamp` | De datum en tijd die u wilt vergelijken met `base_timestamp`. |
| `base_timestamp` | Datum en tijd op basis waarvan de tijdstempel wordt geëvalueerd. |
| `period_no` | De dag kan worden verschoven met `period_no`. `period_no` is een geheel getal, waarbij de waarde 0 de dag aangeeft die `base_timestamp` bevat. Negatieve waarden in `period_no` geven voorafgaande dagen aan en positieve waarden geven volgende dagen aan. |
| `day_start` | Als u wilt werken met dagen die niet om middernacht beginnen, kunt u een verschuiving als een fractie van een dag opgeven in `day_start`, bijvoorbeeld 0,125 voor 3 uur in de ochtend. |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `inday` ('01/12/2006 12:23:00 PM', '01/12/2006 12:00:00 AM', 0) | Retourneert True |
| `inday` ('01/12/2006 12:23:00 PM', '01/13/2006 12:00:00 AM', 0) | Retourneert False |
| `inday` ('01/12/2006 12:23:00 PM', '01/12/2006 12:00:00 AM', -1) | Retourneert False |
| `inday` ('01/11/2006 12:23:00 PM', '01/12/2006 12:00:00 AM', -1) | Retourneert True |
| `inday` ('01/12/2006 12:23:00 PM', '01/12/2006 12:00:00 AM', 0, 0.5) | Retourneert False |
| `inday` ('01/12/2006 11:23:00 AM', '01/12/2006 12:00:00 AM', 0, 0.5) | Retourneert True |

## Voorbeeld 1 – Load-instructie (script)

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met transacties per tijdstempel die wordt geladen in de tabel `Transactions`.
- Een datumveld dat wordt weergegeven in de `TimeStamp`-systeemvariabele `(M/D/YYYY h:mm:ss [.fff] TT)`-opmaak.
- Een voorafgaande lading met de functie `inday()` die is ingesteld als het veld `in_day`.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `in_day`

Resultatentabel

| date | in_dag |
|---|---|
| 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 | 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 |

Het veld `in_day` wordt gemaakt in de voorafgaande load-instructie met behulp van de functie `inday()` en geeft het datumveld door, een vastgelegde tijdstempel voor 5 januari en een `period_no` of `0` als de argumenten van de functie.

## Voorbeeld 2 – period_no

Load-script en resultaten

### Overzicht

Het load-script gebruikt dezelfde gegevensverzameling en hetzelfde scenario als die in het eerste voorbeeld.

Maar in dit voorbeeld is de taak om te berekenen of de transactiedatum twee dagen vóór 5 januari plaatsvond.

### Load-script

```
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
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `in_day`

Resultatentabel

| date | in_dag |
|---|---|
| 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 dit voorbeeld bepaalt de functie of elk van de transactiedatums op 3 januari plaatsvonden omdat een `period_no` of `-2` is gebruikt als het offset-argument in de functie `inday()`. Dit kan worden geverifieerd in de uitvoertabel waar één transactie een booleaans resultaat van WAAR retourneert.

## Voorbeeld 3 – day_start

Load-script en resultaten

#### Overzicht

De load-instructie gebruikt dezelfde gegevensset en hetzelfde scenario als die in de vorige voorbeelden.

Maar in dit voorbeeld is het bedrijfsbeleid dat de werkdag begint en eindigt om 07.00 uur.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- in_day

Resultatentabel

| date | in_dag |
|---|---|
| 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 |

De functie bepaalt of elk van de transactiedatums plaatsvonden op 4 januari vanaf 07.00 uur en 5 januari vóór 07.00 uur, omdat het argument `start_day` van 7/24, dat 07.00 uur is, is gebruikt in de functie `inday()`.

Dit kan worden geverifieerd in de uitvoertabel waarbij transacties die na 07.00 uur op 4 januari plaatsvinden een booleaans resultaat van WAAR retourneren en transacties die plaatsvinden na 07.00 uur op 5 januari een booleaans resultaat van ONWAAR.

## Voorbeeld 4 – diagramobject

Load-script en diagramuitdrukking

### Overzicht

De load-instructie gebruikt dezelfde gegevensset en hetzelfde scenario als die in de vorige voorbeelden.

In dit voorbeeld is de gegevensverzameling echter niet gewijzigd en wordt in de applicatie geladen. U maakt een berekening om te bepalen of een transactie plaatsvindt op 5 januari door een meting te maken in een diagramobject.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `date`

Maak de volgende meting om te berekenen of een transactie plaatsvindt op 5 januari:

```
=inday(date,'01/05/2022 12:00:00 AM',0)
```

Resultatentabel

| 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 |
| 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 |

## Voorbeeld 5 – Scenario

Load-script en resultaten

### Overzicht

In dit voorbeeld is vastgesteld dat producten die op 5 januari zijn geproduceerd defect zijn vanwege een materieelfout. De eindgebruiker wil graag een diagramobject dat op datum de status toont van geproduceerde producten die 'defect' of 'foutloos' zijn en de kosten van de producten die op 5 januari zijn geproduceerd.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam 'Producten'.
- De tabel bevat de volgende velden:
  - product-id
  - productiedatum
  - kostprijs

### Load-script

```
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
];
```

**Resultaten**

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

```
=dayname(manufacture_date)
```

Maak de volgende metingen:

- `=if(only(InDay(manufacture_date,makedate(2022,01,05),0)),'Defective','Faultless')`
- `=sum(cost_price)`

Stel de **Getalnotatie** van de meting in op **Geld**.

Onder **Uiterlijk** schakelt u **Totaal** uit.

Resultatentabel

| dagnaam (productiedatum) | =if(only(InDay(manufacture_date,makedate (2022,01,05),0)),'Defect','Foutloos') | =sum(cost_ price) |
|---|---|---|
| 01/01/2022 | Foutloos | 44.67 |
| 01/02/2022 | Foutloos | 36.34 |
| 01/03/2022 | Foutloos | 51.75 |
| 01/04/2022 | Foutloos | 89.69 |
| 01/05/2022 | Defect | 170.78 |
| 01/06/2022 | Foutloos | 74.23 |

De functie `inday()` retourneert een booleaanse waarde bij het evalueren van de productiedatums van elk product. Voor elk product dat op 5 januari is geproduceerd, retourneert de functie `inday()` een booleaanse waarde voor WAAR en markeert de producten als 'Defect'. Voor elk product dat de waarde voor ONWAAR retourneert, en dus niet op die dag is geproduceerd, wordt het product als 'Foutloos' gemarkeerd.

## indaytotime

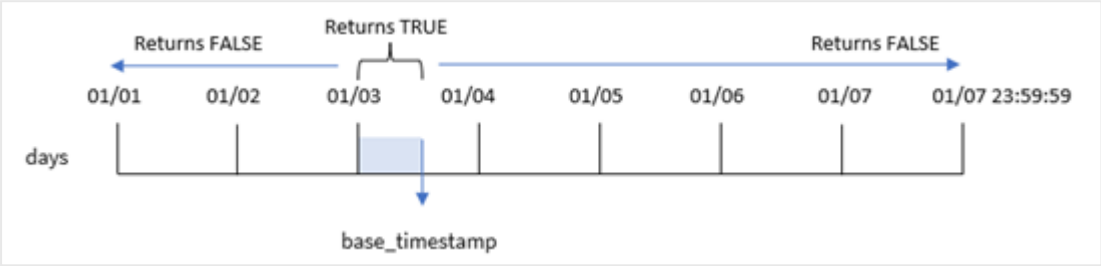
Deze functie retourneert True als **timestamp** binnen het gedeelte van de dag ligt dat **base_ timestamp** bevat tot en met de exacte milliseconde van **base_timestamp**.

**Syntaxis:**

```
InDayToTime (timestamp, base_timestamp, period_no[, day_start])
```

De `indaytotime()`-functie retourneert een booleaans resultaat afhankelijk van het feit wanneer een tijdstempelwaarde plaatsvindt tijdens het segment van de dag. De begingrens van dit segment is het begin van de dag, die standaard is ingesteld op middernacht. Het begin van de dag dit kan worden aangepast via het argument `day_start` van de functie `indaytotime()`. De eindgrens van het dagsegment wordt bepaald door een `base_timestamp`-argument van de functie.

*Diagram van `indaytotime`-functie.*




### Wanneer gebruiken

De functie `indaytotime()` retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als voorwaarde in een `if expression`. De functie `indaytotime()` retourneert een aggregatie of berekening afhankelijk van het feit of een tijdstempel aanwezig is in het segment van de dag tot en met de tijd van de basistijdstempel.

De functie `indaytotime()` kan bijvoorbeeld worden gebruikt om de som van ticketverkopen te tonen voor shows die tot nu toe vandaag hebben plaatsgevonden.

**Retourgegevenstypen:** Booleaanse waarde

In Qlik Sense wordt de booleaanse waarde Waar vertegenwoordigd door -1 en de waarde Onwaar door 0.

Argumenten

| Argument | Beschrijving |
|---|---|
| `timestamp` | De datum en tijd die u wilt vergelijken met `base_timestamp`. |
| `base_timestamp` | Datum en tijd op basis waarvan de tijdstempel wordt geëvalueerd. |
| `period_no` | De dag kan worden verschoven met `period_no`. `period_no` is een geheel getal, waarbij de waarde 0 de dag aangeeft die `base_timestamp` bevat. Negatieve waarden in `period_no` geven voorafgaande dagen aan en positieve waarden geven volgende dagen aan. |
| `day_start` | (Optioneel) Als u wilt werken met dagen die niet om middernacht beginnen, kunt u een verschuiving als een fractie van een dag opgeven in `day_start`. Bijvoorbeeld gebruik 0.125 voor 3 AM |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege

landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `indaytotime` ('01/12/2006 12:23:00 PM', '01/12/2006 11:59:00 PM', 0) | Retourneert True |
| `indaytotime` ('01/12/2006 12:23:00 PM', '01/12/2006 12:00:00 AM', 0) | Retourneert False |
| `indaytotime` '01/11/2006 12:23:00 PM', '01/12/2006 11:59:00 PM', -1) | Retourneert True |

## Voorbeeld 1 – geen aanvullende argumenten

Load-script en resultaten

### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensverzameling met transacties voor de periode tussen 4 en 5 januari wordt geladen in een tabel met de naam 'Transactions'.
- Een datumveld dat wordt weergegeven in de `TimeStamp`-systeemvariabele `(M/D/YYYY h:mm:ss [.fff] TT)`-opmaak.
- Een voorafgaande lading met de functie `indaytotime()` dat is ingesteld als `'in_day_to_time'`, het veld dat bepaalt of de transacties vóór 09.00 AM plaatsvinden.

### Load-script

```
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
];
```

### Resultaten

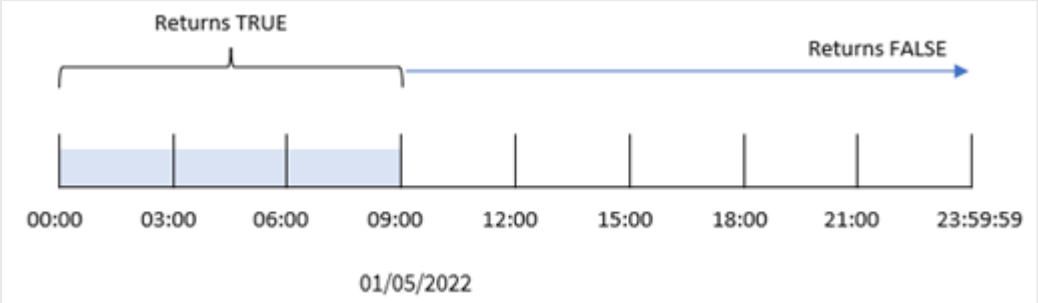
Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `in_day_to_time`

Resultatentabel

| 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 |

*Voorbeeld 1: diagram van `indaytotime`-functie met limiet van 9:00 AM.*




Het `in_day_to_time field` wordt gemaakt in de voorafgaande load-instructie met behulp van de `indaytotime()`-functie en geeft het datumveld door, een vastgelegde tijdstempel voor 09.00 uur op 5 januari en een offset van 0 als de argumenten van de functie. Alle transacties die plaatsvinden tussen middernacht en 09.00 uur op 5 januari retourneren TRUE.

## Voorbeeld 2 – period_no

Load-script en resultaten

### Overzicht

De load-instructie gebruikt dezelfde gegevensset en hetzelfde scenario als die in het eerste voorbeeld.

Maar in dit voorbeeld berekent u of de transactiedatum één dag vóór 09.00 uur op 5 januari plaatsvond.

### Load-script

```
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
];
```

### Resultaten

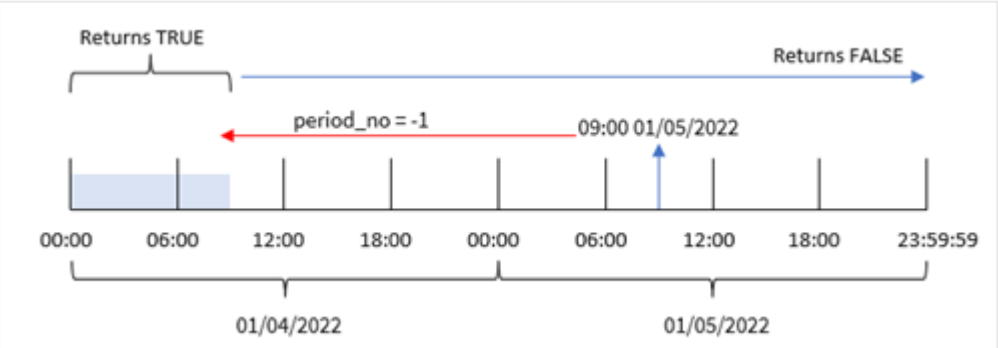
Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `in_day_to_time`

Resultatentabel

| 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 |

*Voorbeeld 2: diagram van `indaytotime`-functie met transacties vanaf 4 januari.*



In dit voorbeeld bepaalt de functie of elk van de transactiedatums vóór 09.00 uur op 4 januari plaatsvonden omdat een offset-argument van -1 is gebruikt in de `indaytotime()`-functie. Dit kan worden geverifieerd in de uitvoertabel waar een transactie een booleaans resultaat van TRUE retourneert.

## Voorbeeld 3 – day_start

Load-script en resultaten

### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

Maar in dit voorbeeld is het bedrijfsbeleid dat de werkdag begint en eindigt om 8 AM.

### Load-script

```
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
];
```

### Resultaten

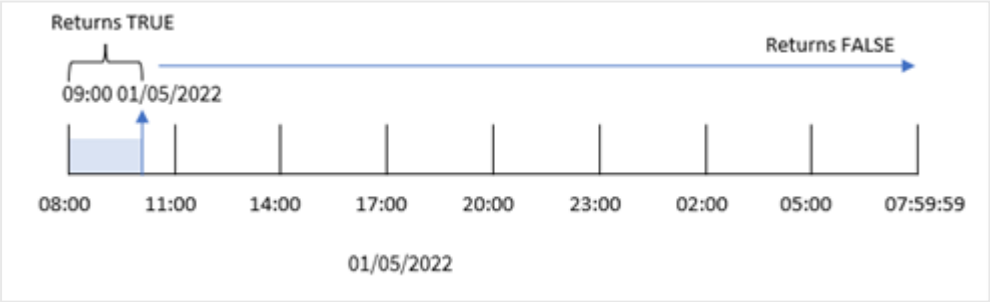
Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `in_day_to_time`

Resultatentabel

| 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 |

*Voorbeeld 3: diagram van `indaytotime`-functie met transacties vanaf 08.00 uur tot 09.00 uur.*




Omdat het argument `start_day` van 8/24, dat overeenkomt met 8:00 AM, wordt gebruikt in de functie `indaytotime()`, begint en eindigt elke dag om 8:00 AM. Daarom zal de functie `indaytotime()` een booleaanse waarde voor WAAR retourneren voor elke transactie die plaatsvond tussen 8:00 AM en 9:00 AM op 5 januari.

## Voorbeeld 4 – diagramobject

Load-script en diagramuitdrukking

### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is de gegevensverzameling echter niet gewijzigd en wordt in de applicatie geladen. U maakt een berekening om te bepalen of een transactie plaatsvindt op 5 januari vóór 09.00 uur door een meting te maken in een diagramobject.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

`date`.

Maak de volgende meting om te berekenen of een transactie plaatsvindt op 5 januari vóór 09.00 uur:

```
=indaytotime(date,'01/05/2022 9:00:00 AM',0)
```

Resultatentabel

| date | inday(date,'01/05/2022 12:00:00 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 |

| date | inday(date,'01/05/2022 12:00:00 AM',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 |

De meting `in_day_to_time` wordt gemaakt in het diagramobject met behulp van de functie `indaytotime()` en geeft het datumveld door, een vastgelegde tijdstempel voor 09.00 uur op 5 januari en een offset van `0` als de argumenten van de functie. Alle transacties die plaatsvinden tussen middernacht en 09.00 uur op 5 januari retourneren TRUE. Dit wordt gevalideerd in de resultatentabel.

## Voorbeeld 5 – Scenario

Load-script en resultaten

### Overzicht

In dit voorbeeld wordt een gegevensverzameling met ticketverkopen voor een lokale bioscoop geladen in een tabel met de naam Ticket_Sales. Vandaag is het 3 mei 2022 en het is 11:00 AM.

De gebruiker wil graag een KPI-diagramobject gebruiken om de omzet te tonen van alle shows die tot dusver vandaag hebben plaatsgevonden.

### Load-script

```
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
];
```

**Resultaten**

Doe het volgende:

1. Maak een KPI-object.
2. Maak met behulp van de functie `indaytotime()` een meting die de som toont van alle ticketverkopen voor shows die tot dusver vandaag hebben plaatsgevonden:

```
=sum(if(indaytotime([show time],'05/03/2022 11:00:00 AM',0),[ticket price],0))
```

3. Maak een label voor het KPI-object, 'Current Revenue'.
4. Stel de **Getalnotatie** van de meting in op **Geld**.

Het totaal van ticketverkopen tot 11:00 AM op 3 mei 2022 is $52,50.

De functie `indaytotime ()` retourneert een booleaanse waarde bij het vergelijken van de showtijden van elk van de ticketverkopen met de huidige tijd ('05/03/2022 11:00:00 AM'). Voor elke show op 3 mei vóór 11:00 AM retourneert de functie `indaytotime ()` een booleaanse waarde WAAR en de ticketprijs wordt meegenomen in het somtotaal.

## inlunarweek

Deze functie bepaalt of **timestamp** binnen de maanweek valt die **base_date** bevat. Maanweken in Qlik Sense worden gedefinieerd door het feit dat 1 januari wordt gezien als de eerste dag van de week. Met uitzondering van de laatste week van het jaar, bevat elke week precies zeven dagen.

**Syntaxis:**

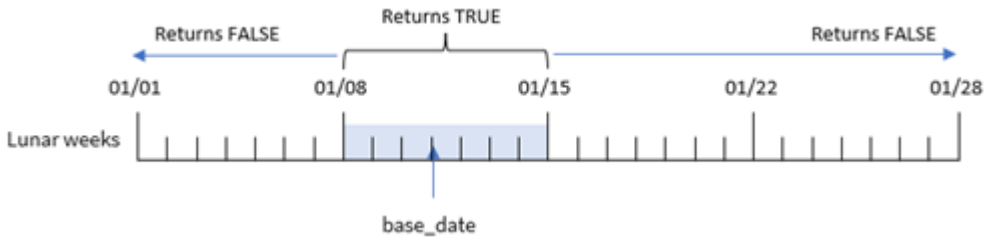
```
InLunarWeek (timestamp, base_date, period_no[, first_week_day])
```

**Retourgegevenstypen:** Booleaanse waarde

> *In Qlik Sense wordt de booleaanse waarde Waar vertegenwoordigd door -1 en de waarde Onwaar door 0.*

De `inlunarweek()`-functie bepaalt welke maanweek de `base_date` in valt. Er wordt vervolgens een booleaanse waarde geretourneerd zodra de functie heeft bepaald of iedere tijdstempelwaarde binnen dezelfde maanweek valt als de `base_date`.

*Diagram van de `inlunarweek()`-functie*




## Wanneer gebruiken

De functie `inlunarweek()` retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als voorwaarde in een IF-uitdrukking. Dit retourneert een aggregatie of berekening afhankelijk van of de geëvalueerde datum plaatsvond in de betreffende maanweek.

De functie `inlunarweek()` kan bijvoorbeeld worden gebruikt om alle materieel te identificeren die in een bepaalde maanweek is geproduceerd.

Argumenten

| Argument | Beschrijving |
|---|---|
| **timestamp** | De datum die u wilt vergelijken met **base_date**. |
| **base_date** | Datum op basis waarvan de maanweek wordt geëvalueerd. |
| **period_no** | De maanweek kan worden verschoven met **period_no**. period_no is een geheel getal, waarbij de waarde 0 de maanweek aangeeft die **base_date** bevat. Negatieve waarden in **period_no** geven voorafgaande maanweken aan en positieve waarden geven volgende maanweken aan. |
| **first_week_day** | Een verschuiving kan groter of kleiner zijn dan nul. Hiermee wordt het begin van het jaar gewijzigd met het opgegeven aantal dagen en/of fracties van een dag. |

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `inlunarweek ('01/12/2013', '01/14/2013', 0)` | Retourneert `TRUE`, omdat de waarde van `timestamp`, 01/12/2013, in de week van 01/08/2013 tot 01/14/2013 valt. |
| `inlunarweek ('01/12/2013', '01/07/2013', 0)` | Retourneert `FALSE`, omdat de `base_date` 01/07/2013 in de maanweek valt die is ingesteld op 01/01/2013 tot 01/07/2013. |

| Voorbeeld | Resultaat |
|---|---|
| `inlunarweek ('01/12/2013', '01/14/2013', -1)` | Retourneert `FALSE`. Door het opgeven van een waarde van period_no als -1 wordt de week naar de vorige week verschoven, van 01/01/2013 t/m 01/07/2013. |
| `inlunarweek ('01/07/2013', 01/14/2013', -1)` | Retourneert `TRUE`. In vergelijking met het voorafgaande voorbeeld ligt de `timestamp` in de volgende week daarna als wordt rekening gehouden met de terugwaartse verschuiving. |
| `inlunarweek ('01/11/2006', '01/08/2006', 0, 3)` | Retourneert `FALSE`. Door een waarde van 3 op te geven voor `first_week_day`, wordt het begin van het jaar berekend vanaf 01/04/2013. Daarom valt de waarde van `base_date` in de eerste week en de waarde van `timestamp` in de week van 01/11/2013 t/m 01/17/2013. |

De functie `inlunarweek()` wordt vaak gebruikt in combinatie met de volgende functies:

Gerelateerde functies

| Functie | Interactie |
|---|---|
| *lunarweekname (page 984)* | Deze functie wordt gebruikt om het maanweeknummer te bepalen van het jaar waarin een invoerdatum valt. |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – Geen aanvullende argumenten

Load-script en resultaten

**Overzicht**

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met transacties voor de maand januari die wordt geladen in de tabel `Transactions`.
- Het datumveld is opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.

Maak een veld, `in_lunar_week`, dat bepaalt of de transacties in dezelfde maanweek plaatsvonden als 10 januari.

### Load-script

```
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
];
```

### Resultaten

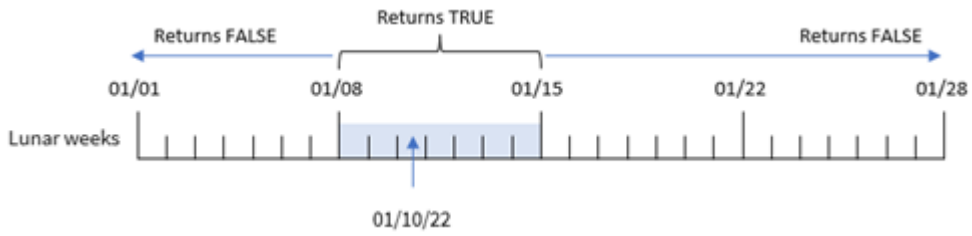
Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `in_lunar_week`

Resultatentabel

| 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 |

*`inlunarweek()`-functie, basisvoorbeeld*




Het veld `in_lunar_week` wordt gemaakt in de voorafgaande load-instructie met behulp van de `inlunarweek()`-functie en geeft vervolgens het volgende door als de argumenten van de functie:

- Het `date`-veld
- Een vastgelegde datum van 10 januari als de `base_date`
- Een `period_no` van 0

Omdat maanweken op 1 januari beginnen, zou 10 januari in de maanweek vallen die begint op 8 januari en eindigt op 14 januari. Daarom zouden alle transacties die tussen deze datums in januari plaatsvinden, de booleaanse waarde TRUE retourneren. Dit wordt gevalideerd in de resultatentabel.

## Voorbeeld 2 – period_no

Voorbeelden en resultaten:

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het datumveld is opgegeven in de DateFormat-systeemvariabele (MM/DD/YYYY) indeling.

Maak een veld, 2_lunar_weeks_later, dat bepaalt of de transacties in dezelfde maanweek plaatsvonden als 10 januari.

### Load-script

```
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
];
```

### Resultaten

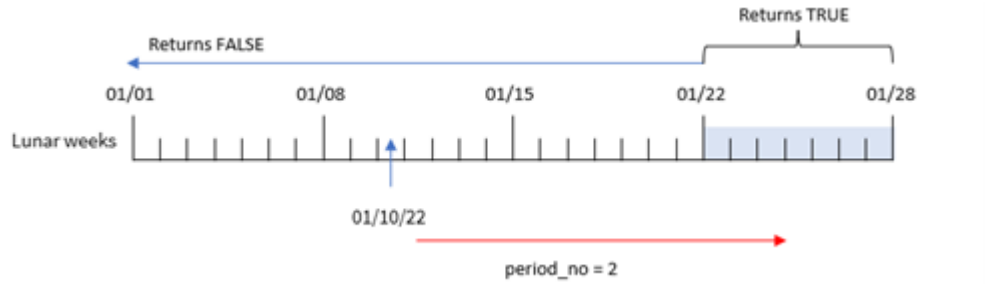
Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `2_lunar_weeks_later`

Resultatentabel

| 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 |

*inlunarweek()-functie, voorbeeld van period_no*




In dit voorbeeld definieert de functie de week die begint op 22 januari als de maanweek op basis waarvan transacties moeten worden gevalideerd omdat een `period_no` van 2 is gebruikt als het offset-argument in de `inlunarweek()`-functie. Daarom zal elke transactie die plaatsvindt tussen 22 januari en 28 januari een booleaans resultaat van `TRUE` retourneren.

## Voorbeeld 3 – first_week_day

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script gebruikt dezelfde gegevensset en scenario als het eerste voorbeeld. Maar in dit voorbeeld stellen we het begin van de maanweek in op 6 januari.

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- De standaard `DateFormat`-systeemvariabele mm/dd/jjjj wordt gebruikt.
- Een `first_week_day`-argument van 5. Dit stelt het begin van de maanweken in op 5 januari.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

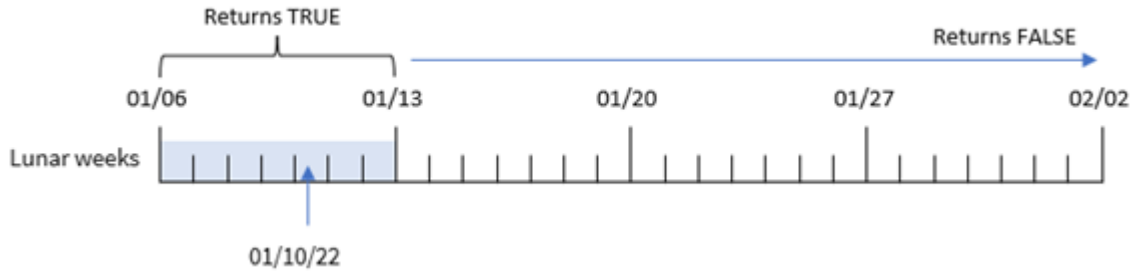
- `date`
- `in_lunar_week`

Resultatentabel

| 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 |

*inlunarweek()-functie, first_week_day example*




In dit geval wordt het begin van de maanweekkalender verschoven naar 6 januari omdat in het `first_week_date`-argument van 5 wordt gebruikt in de functie `inlunarweek()`. Daarom valt 10 januari in de maanweek die begint op 6 januari en eindigt op 12 januari. Iedere transactie die tussen deze twee datums valt, retourneert een booleaanse waarde voor `TRUE`.

## Voorbeeld 4 – Diagramobject

Load-script en diagramuitdrukking:

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het datumveld is opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die bepaalt of transacties plaatsvonden in dezelfde maanweek als 10 januari, is gemaakt als meting in een diagramobject van de toepassing.

### Load-script

```
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
];
```

**Resultaten**

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `date`.

Maak de volgende meting om te berekenen of een transactie plaatsvindt in de maanweek waarin 10 januari valt:

```
= inlunarweek(date,'01/10/2022', 0)
```

Resultatentabel

| 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 |

## Voorbeeld 5 – Scenario

Load-script en diagramuitdrukking:

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset die wordt geladen in een tabel met de naam `Products`.
- Informatie met daarin product-id, productiedatum en kostprijs.

Er is vastgesteld dat producten die in de maanweek van 12 januari zijn geproduceerd defect zijn vanwege een materieelfout. De eindgebruiker wil graag een diagramobject dat op maanweeknaam de status toont of de geproduceerde producten 'defect' of 'foutloos' zijn en de kosten van de producten die in die maand zijn geproduceerd.

### Load-script

```
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
];
```

**Resultaten**

**Doe het volgende:**

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel.
2. Maak een dimensie om de maandnamen weer te geven:
   `=lunarweekname(manufacture_date)`
3. Maak een dimensie om te identificeren welke van de producten defect zijn en welke foutloos zijn met behulp van de `inlunarweek()`-functie:
   `=if(only(inlunarweek(manufacture_date,makedate(2022,01,12),0)), 'Defective','Faultless')`
4. Maak een meting om de `cost_price` van de producten op te tellen:
   `=sum(cost_price)`
5. Stel de **Getalnotatie** van de meting in op **Geld**.
6. Onder **Uiterlijk** schakelt u **Totaal** uit.

Resultatentabel

| lunarweekname (manufacture_date) | =if(only(inlunarweek(manufacture_date,makedate (2022,01,12),0)), 'Defect','Foutloos') | sum(cost_ price) |
|---|---|---|
| 2022/01 | Foutloos | $125.79 |
| 2022/02 | Defect | $316.38 |
| 2022/03 | Foutloos | $455.75 |
| 2022/04 | Foutloos | $146.09 |

De functie `inlunarweek()` retourneert een booleaanse waarde bij het evalueren van de productiedatums van elk product. Voor elk product dat wordt geproduceerd in de maanweek waarin 10 januari valt, retourneert de `inlunarweek()`-functie een booleaanse waarde voor `TRUE` en markeert de producten als 'Defect'. Voor elk product dat de waarde voor `FALSE` retourneert, en dus niet in die week is geproduceerd, wordt het product als 'Foutloos' gemarkeerd.

## inlunarweektodate

Met deze functie wordt bepaald of **timestamp** binnen het gedeelte van de maanweek ligt tot en met de laatste milliseconde van **base_date**. Maanweken in Qlik Sense worden gedefinieerd door het feit dat 1 januari wordt gezien als de eerste dag van de week en, met uitzondering van de laatste week van het jaar, bevat elke week precies zeven dagen.

**Syntaxis:**

```
InLunarWeekToDate (timestamp, base_date, period_no [, first_week_day])
```

**Retourgegevenstypen:** Booleaanse waarde

> *In Qlik Sense wordt de booleaanse waarde Waar vertegenwoordigd door -1 en de waarde Onwaar door 0.*

*Voorbeelddiagram van de `inlunarweektodate()`-functie*

De `inlunarweektodate()`-functie functioneert als het eindpunt van de maanweek. Daarentegen bepaalt de `inlunarweek()`-functie in welke maanweek de `base_date` valt. Bijvoorbeeld: als de `base_date` 5 januari was, dan zou een tijdstempel tussen 1 januari en 5 januari een booleaans resultaat voor `TRUE` retourneren. Datums zoals 6 en 7 januari en zouden een booleaans resultaat voor `FALSE` retourneren.

Argumenten

| Argument | Beschrijving |
|---|---|
| **timestamp** | De datum die u wilt vergelijken met **base_date**. |
| **base_date** | Datum op basis waarvan de maanweek wordt geëvalueerd. |
| **period_no** | De maanweek kan worden verschoven met **period_no**. period_no is een geheel getal, waarbij de waarde 0 de maanweek aangeeft die **base_date** bevat. Negatieve waarden in **period_no** geven voorafgaande maanweken aan en positieve waarden geven volgende maanweken aan. |
| **first_week_day** | Een verschuiving kan groter of kleiner zijn dan nul. Hiermee wordt het begin van het jaar gewijzigd met het opgegeven aantal dagen en/of fracties van een dag. |

## Wanneer gebruiken

De functie `inlunarweektodate()` retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als voorwaarde in een IF-uitdrukking. De `inlunarweektodate()`-functie zou worden gebruikt wanneer de gebruiker wil dat de berekening een aggregatie of berekening retourneert, afhankelijk van of de geëvalueerde datum in een bepaald deel van de week in kwestie heeft plaatsgevonden.

De functie `inlunarweektodate()` kan bijvoorbeeld worden gebruikt om alle materieel te identificeren die in een bepaalde week t/m een bepaalde datum is geproduceerd.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `inlunarweektodate ('01/12/2013', '01/13/2013', 0)` | Retourneert `TRUE`, omdat de waarde van `timestamp`, 01/12/2013, in het deel van de week van 01/08/2013 tot 01/13/2013 valt. |
| `inlunarweektodate ('01/12/2013', '01/11/2013', 0)` | Retourneert `FALSE` omdat de waarde `timestamp` later is dan de waarde `base_date` hoewel de twee datums in dezelfde maanweek vallen vóór 01/12/2012. |
| `inlunarweektodate ('01/12/2006', '01/05/2006', 1)` | Retourneert `TRUE`. Als een waarde van 1 wordt opgegeven voor `period_no`, wordt de `base_date` één week vooruit geschoven, zodat de waarde van `timestamp` in het deel van de maanweek valt. |

De functie `inlunarweektodate()` wordt vaak gebruikt in combinatie met de volgende functies:

Gerelateerde functies

| Functie | Interactie |
|---|---|
| *lunarweekname (page 984)* | Deze functie wordt gebruikt om het maanweeknummer te bepalen van het jaar waarin een invoerdatum valt. |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – Geen aanvullende argumenten

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties voor de maand januari die wordt geladen in de tabel `Transactions`. De standaard `DateFormat`-systeemvariabele mm/dd/jjjj wordt gebruikt.
- Maak een veld, `in_lunar_week_to_date`, dat bepaalt welke transacties in de maanweek tot de datum 10 januari plaatsvonden.

### Load-script

```
SET DateFormat='MM/DD/YYYY';

Transactions:
    Load
        *,
        inlunarweektodate(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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

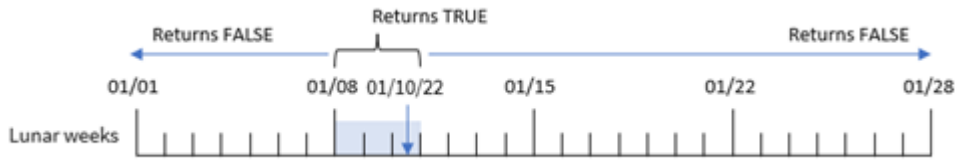
- `date`
- `in_lunar_week_to_date`

Resultatentabel

| 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 |

*`inlunarweektodate()`-functie, geen extra argumenten*




Het `in_lunar_week_to_date`-veld wordt gemaakt in de voorafgaande load-instructie met behulp van de `inlunarweektodate()`-functie en geeft het `date`-veld door, een vastgelegde tijdstempel voor 10 januari als onze `base_date` en een offset van 0 als de argumenten van de functie.

Omdat maanweken beginnen op 1 januari, zou 10 januari in de maakweek vallen die op 8 januari begint, en omdat we de `inlunarweektodate()`-functie gebruiken zou die maanweek eindigen op de 10e. Daarom zouden alle transacties die tussen deze datums in januari plaatsvinden, de booleaanse waarde `TRUE` retourneren. Dit wordt gevalideerd in de resultatentabel.

## Voorbeeld 2 – period_no

### Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld. Maar in dit voorbeeld, is de taak om een veld te maken, `2_lunar_weeks_later`, dat bepaalt of de transacties al dan niet twee weken na de maanweek van 1 januari plaatsvonden.

### Load-script

```
SET DateFormat='MM/DD/YYYY';
Transactions:
    Load
        *,
        inlunarweektodate(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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

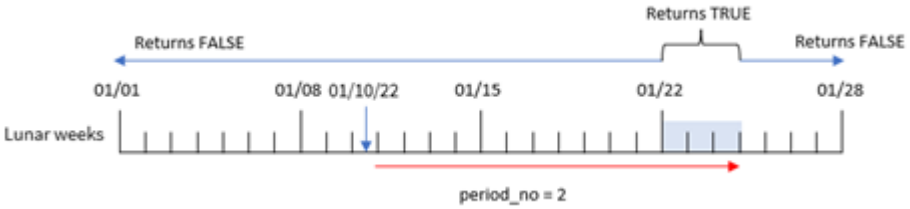
- `date`
- `2_lunar_weeks_later`

Resultatentabel

| 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 |

*`inlunarweektodate()`-functie, voorbeeld van period_no*




In deze instantie bepaalt de `inlunarweektodate()`-functie dat de maanweek tot 10 januari gelijk is aan drie dagen (8, 9 en 10 januari). Omdat er een `period_no` van 2 is gebruikt als het offset-argument, wordt deze maanweek met 14 dagen verschoven. Dit zorgt er daarom voor dat de definitie van deze driedaagse maanweek bestaat uit 22, 23 en 24 januari. Iedere transactie die plaatsvindt tussen 22 januari en 24 januari zal een booleaans resultaat van `TRUE` retourneren.

## Voorbeeld 3 – first_week_day

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- De standaard `DateFormat`-systeemvariabele mm/dd/jjjj wordt gebruikt.
- Een `first_week_date`-argument van 3. Dit stelt het begin van de maanweken in op 3 januari

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

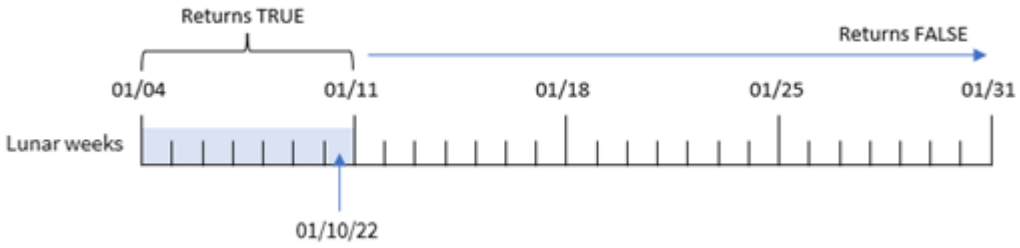
- `date`
- `in_lunar_week_to_date`

Resultatentabel

| 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 |

*`inlunarweektodate()`-functie, first_week_day example*




In deze instantie loopt de eerste maanweek van 3 januari t/m 10 januari omdat het `the first_week_date`-argument 3 is gebruikt in de `inlunarweek()`-functie. Omdat 10 januari tevens de `base_date` is, retourneert iedere transactie die tussen deze twee datums valt, een booleaanse waarde van `TRUE`.

## Voorbeeld 4 – Diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die bepaalt of transacties plaatsvonden in de maanweek tot 10 januari, is gemaakt als meting in een diagramobject van de toepassing.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `date`.

Maak de volgende meting:

```
=inlunarweektodate(date,'01/10/2022', 0)
```

Resultatentabel

| 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 |
| 1/23/2022 | 0 |

| date | =inlunarweektodate(date,'01/10/2022', 0) |
|---|---|
| 1/26/2022 | 0 |
| 1/27/2022 | 0 |
| 1/29/2022 | 0 |
| 1/31/2022 | 0 |

*`inlunarweektodate()`-functie, diagramobjectvoorbeeld*

De meting `in_lunar_week_to_date` wordt gemaakt in het diagramobject met behulp van de `inlunarweektodate()`-functie en geeft het datumveld door, een vastgelegde tijdstempel voor 10 januari als onze `base_date` en een offset van 0 als de argumenten van de functie.

Omdat maanweken beginnen op 1 januari, zou 10 januari in de maakweek vallen die op 8 januari begint. Omdat we de `inlunarweektodate()`-functie gebruiken zou die maanweek eindigen op de 10e. Daarom zouden alle transacties die tussen deze datums in januari plaatsvinden, de booleaanse waarde `TRUE` retourneren. Dit wordt gevalideerd in de resultatentabel.

## Voorbeeld 5 – Scenario

Load-script en diagramuitdrukkingen

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset die wordt geladen in een tabel met de naam `Products`.
- Informatie met daarin product-id, productiedatum en kostprijs.

Er is vastgesteld dat producten die in de maanweek van 12 januari zijn geproduceerd defect zijn vanwege een materieelfout. Op 13 januari is het probleem opgelost. De eindgebruiker wil graag een diagramobject dat op week de status toont of de geproduceerde producten 'defect' of 'foutloos' zijn en de kosten van de producten die in die week zijn geproduceerd.

### Load-script

```
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
];
```

**Resultaten**

**Doe het volgende:**

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel.
2. Maak een dimensie om de weeknamen weer te geven:
   `=weekname(manufacture_date)`
3. Maak vervolgens een dimensie om te maken die de `inlunarweektodate()`-functie gebruikt om te identificeren welke van de producten defect zijn en welke foutloos zijn:
   `=if(inlunarweektodate(manufacture_date,makedate(2022,01,12),0),'Defective','Faultless')`
4. Maak een meting om de `cost_price` van de producten op te tellen:
   `=sum(cost_price)`
5. Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

| =lunarweekname (manufacture_date) | =if(InLunarWeekToDate(manufacture_date,makedate (2022,01,12),0),'Defective','Faultless') | =Sum(cost_ price) |
|---|---|---|
| 2022/01 | Foutloos | $142.67 |
| 2022/02 | Defect | $320.88 |
| 2022/02 | Foutloos | $141.82 |

| =lunarweekname (manufacture_date) | =if(InLunarWeekToDate(manufacture_date,makedate (2022,01,12),0),'Defective','Faultless') | =Sum(cost_ price) |
|---|---|---|
| 2022/03 | Foutloos | $214.64 |
| 2022/04 | Foutloos | $147.46 |
| 2022/05 | Foutloos | $248.12 |

De functie `inlunarweektodate()` retourneert een booleaanse waarde bij het evalueren van de productiedatums van elk product. Voor degenen die een booleaanse waarde van `TRUE` retourneren, markeert het de producten als `'Defective'`. Voor elk product dat een waarde retourneert van `FALSE`, en dus niet gemaakt in de maanweek tot en met 12 januari, markeert het de producten als `'Faultless'`.

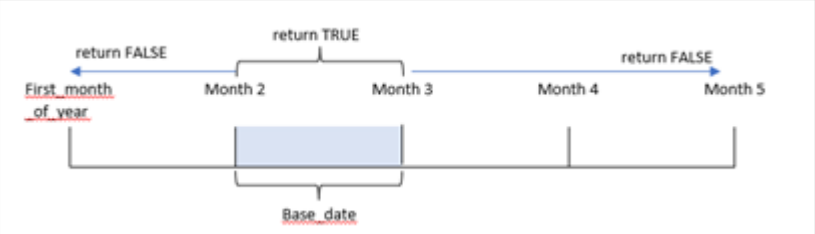
## inmonth

Deze functie retourneert True als **timestamp** in de maand valt die **base_date** bevat.

**Syntaxis:**

```
InMonth (timestamp, base_date, period_no)
```

*Diagram van `indaytotime`-functie.*




Met andere woorden, de functie `inmonth()` bepaalt of een reeks datums in deze maand valt en retourneert een booleaanse waarde op basis van een `base_date` die de maand identificeert.

### Wanneer gebruiken

De functie `inmonth()` retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als voorwaarde in een `if expression`. Hierdoor wordt een aggregatie of berekening geretourneerd, afhankelijk van of een datum plaatsvond in de maand, inclusief de desbetreffende datum.

De functie `inmonth()` bijvoorbeeld kan worden gebruikt om alle materieel te identificeren die in een specifieke maand is geproduceerd.

**Retourgegevenstypen:** Booleaanse waarde

In Qlik Sense wordt de booleaanse waarde Waar vertegenwoordigd door -1 en de waarde Onwaar door 0.

Argumenten

| Argument | Beschrijving |
|---|---|
| tijdstempel | De datum die u wilt vergelijken met `base_date`. |

| Argument | Beschrijving |
|---|---|
| base_date | Datum op basis waarvan de maand wordt geëvalueerd. Het is belangrijk op te merken dat de `base_date` elke dag binnen een maand kan zijn. |
| period_no | De maand kan worden verschoven met `period_no`. `period_no` is een geheel getal, waarbij de waarde 0 de maand aangeeft die `base_date` bevat. Negatieve waarden in `period_no` geven voorafgaande maanden aan en positieve waarden geven volgende maanden aan. |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `inmonth ('25/01/2013', '01/01/2013', 0)` | Retourneert True |
| `inmonth('25/01/2013', '23/04/2013', 0)` | Retourneert False |
| `inmonth ('25/01/2013', '01/01/2013', -1)` | Retourneert False |
| `inmonth ('25/12/2012', '17/01/2013', -1)` | Retourneert True |

## Voorbeeld 1 – geen aanvullende argumenten

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties voor de eerste helft van 2022.
- Een voorafgaande lading met een extra variabele, '`in_month`', die bepaalt of transacties in april hebben plaatsgevonden.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `in_month`

Voorbeelden van functies

| 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 |

Het veld '`in_month`' wordt gemaakt in de voorafgaande load-instructie met behulp van de functie `inmonth()` en geeft het datumveld door, een vastgelegde datum van 1 april, als onze `base_date` en een `period_no` van `0` als de argumenten van de functie.

De `base_date` identificeert de maand die het booleaanse resultaat TRUE zal opleveren. Daarom retourneren alle transacties die in april hebben plaatsgevonden TRUE, wat wordt gevalideerd in de resultatentabel.

## Voorbeeld 2 – period_no

Load-script en resultaten

### Overzicht

Dezelfde gegevensset en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld maakt u echter een veld '`2_months_prior`' aan dat bepaalt of de transacties twee maanden voor april hebben plaatsgevonden.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `2_months_prior`

Voorbeelden van functies

| 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 |

Als u -2 gebruikt als het `period_no`-argument in de `inmonth()`-functie, verschuift de maand gedefinieerd door het `base_date`-argument naar twee maanden eerder. In dit voorbeeld verandert het de gedefinieerde maand van april in februari.

Daarom zal elke transactie die in februari plaatsvindt een booleaans resultaat van TRUE retourneren.

## Voorbeeld 3 – Diagramobject

Load-script en diagramuitdrukking

### Overzicht

Dezelfde gegevensset en het scenario uit de vorige voorbeelden worden gebruikt.

In dit voorbeeld is de gegevensverzameling echter niet gewijzigd en wordt in de applicatie geladen. De berekening die bepaalt of transacties plaatsvonden in april is gemaakt als meting in een diagramobject van de toepassing.

### Load-script

```
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
];
```

### Diagramobject

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

```
date
```

Maak de volgende meting om te berekenen of een transactie plaatsvindt in april:

```
=inmonth(date,'04/01/2022', 0)
```

**Resultaten**

| | Voorbeelden van functies |
|---|---|
| **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 |

## Voorbeeld 4 – Scenario

Load-script en resultaten

**Overzicht**

In dit voorbeeld wordt een gegevensset geladen in de tabel '`Products`'. De tabel bevat de volgende velden:

- Product-id
- Productiedatum
- Kostprijs

Door een fout in de apparatuur waren producten die in de maand juli 2022 zijn geproduceerd defect. Het probleem is op 27 juli 2022 opgelost.

De eindgebruiker wil graag een diagram dat op maand de status toont van producten die zijn geproduceerd als 'defect' (booleaans WAAR) of 'foutloos' (booleaans NIET WAAR) en de kosten van de producten die in die maand zijn geproduceerd.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

```
=monthname(manufacture_date)
```

Maak de volgende metingen

- `=sum(cost_price)`
- `=if(only(inmonth(manufacture_date,makedate(2022,07,01),0)),'Defective','Faultless')`

1. Stel de **Getalnotatie** van de meting in op **Geld**.
2. Onder **Uiterlijk** schakelt u **Totaal** uit.

Resultatentabel

| monthname (manufacture_date) | =if(only(inmonth(manufacture_date,makedate (2022,07,01),0)),'Defective','Faultless') | sum(cost_ price) |
|---|---|---|
| januari 2022 | Foutloos | $54.40 |
| februari 2022 | Foutloos | $145.69 |
| maart 2022 | Foutloos | $53.80 |

| monthname (manufacture_date) | =if(only(inmonth(manufacture_date,makedate (2022,07,01),0)),'Defective','Faultless') | sum(cost_ price) |
|---|---|---|
| april 2022 | Foutloos | $82.06 |
| mei 2022 | Foutloos | $127.60 |
| juni 2022 | Foutloos | $141.82 |
| juli 2022 | Defect | $214.64 |
| augustus 2022 | Foutloos | $147.46 |
| september 2022 | Foutloos | $84.21 |
| oktober 2022 | Foutloos | $163.91 |

De functie `inmonth()` retourneert een booleaanse waarde bij het evalueren van de productiedatums van elk product. Voor elk product dat in juli 2022 is geproduceerd, retourneert de functie `inmonth()` een booleaanse waarde voor Waar en markeert de producten als 'Defect'. Voor elk product dat de waarde voor Niet waar retourneert, en dus niet op in juli is geproduceerd, wordt het product als 'Foutloos' gemarkeerd.

## inmonths

Deze functie bepaalt of een tijdstempel valt binnen dezelfde periode van de maand, twee maanden, kwartaal, tertiaal of half jaar als basisdatum. Het is tevens mogelijk om te bepalen of de tijdstempel binnen een voorafgaande of volgende tijdsperiode valt.

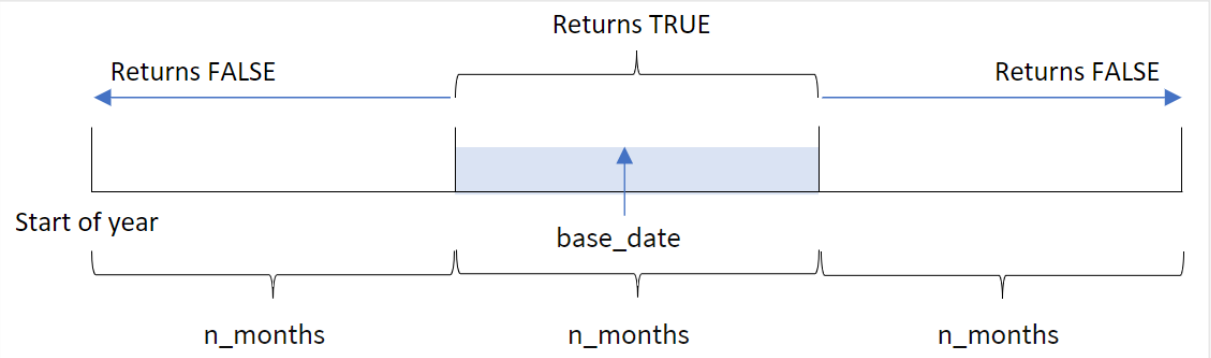
**Syntaxis:**

```
InMonths(n_months, timestamp, base_date, period_no [, first_month_of_year])
```

**Retourgegevenstypen:** Booleaanse waarde

In Qlik Sense wordt de booleaanse waarde Waar vertegenwoordigd door -1 en de waarde Onwaar door 0.

*Diagram van de `inmonths()`-functie*



De functie `inmonths()` verdeelt het jaar in segmenten op basis van het opgegeven `n_months`-argument. Het bepaalt vervolgens of een geëvalueerde tijdstempel in hetzelfde segment valt als het `base_date`-argument. Als er echter een `period_no`-argument is opgegeven, bepaalt de functie of de tijdstempels in een vorige of volgende periode vallen ten opzichte van de `base_date`.

De volgende segmenten van het jaar zijn beschikbaar in de functie als `n_month`-argumenten:

`n_month`-argumenten

| Periode | Aantal maanden |
|---|---|
| maandelijks | 1 |
| tweemaandelijks | 2 |
| kwartaal | 3 |
| tertiaal | 4 |
| halfjaarlijks | 6 |

### Wanneer gebruiken

De functie `inmonths()` retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als voorwaarde in een `if expression`. Door de `inmonths()`-functie te gebruiken, kunt u de periode selecteren die u wilt laten evalueren. Bijvoorbeeld: door de gebruiker producten te laten identificeren die in de maand, het kwartaal of de jaarhelft van een bepaalde periode zijn geproduceerd.

**Retourgegevenstypen:** Booleaanse waarde

In Qlik Sense wordt de booleaanse waarde Waar vertegenwoordigd door -1 en de waarde Onwaar door 0.

Argumenten

| Argument | Beschrijving |
|---|---|
| **n_months** | Het aantal maanden dat de periode definieert. Een geheel getal of uitdrukking die wordt herleid tot een geheel getal dat een van de volgende waarden moet hebben: 1 (equivalent aan de functie inmonth()), 2 (2 maanden), 3 (equivalent aan de functie inquarter()), 4 (tertiaal) of 6 (half jaar). |
| **timestamp** | De datum die u wilt vergelijken met **base_date**. |
| **base_date** | Datum op basis waarvan de periode wordt geëvalueerd. |
| **period_no** | De periode kan worden verschoven met **period_no**, een geheel getal of een uitdrukking die wordt herleid tot een geheel getal, waarbij de waarde 0 de periode aangeeft die **base_date** bevat. Negatieve waarden in **period_no** geven voorafgaande perioden aan en positieve waarden geven volgende perioden aan. |
| **first_month_of_year** | Als u wilt werken met (boek)jaren die niet in januari beginnen, geeft u een waarde op tussen 2 en 12 in **first_month_of_year**. |

U kunt de volgende waarden gebruiken om de eerste maand van het jaar in te stellen in het argument `first_month_of_year`:

waarden first_month_of_year

| Maand | Waarde |
|---|---|
| Februari | 2 |
| Maart | 3 |
| April | 4 |
| Mei | 5 |
| Juni | 6 |
| Juli | 7 |
| Augustus | 8 |
| September | 9 |
| Oktober | 10 |
| November | 11 |
| December | 12 |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `inmonths(4, '01/25/2013', '04/25/2013', 0)` | Retourneert TRUE. Omdat de waarde van timestamp, 25/01/2013, binnen de periode van vier maanden van 01/01/2013 t/m 30/04/2013 ligt, waarin de waarde van `base_date`, 25/04/2013 ligt. |
| `inmonths(4, '05/25/2013', '04/25/2013', 0)` | Retourneert FALSE. Omdat 25/05/2013 buiten dezelfde periode als het voorafgaande voorbeeld ligt. |

| Voorbeeld | Resultaat |
|---|---|
| `inmonths(4, '11/25/2012', '02/01/2013', -1)` | Retourneert TRUE. Omdat de waarde van `period_no`, `-1`, de zoekperiode één periode van vier maanden (de waarde van n-months) terugschuift, waardoor de zoekperiode valt van 01/09/2012 t/m 31/12/2012. |
| `inmonths(4, '05/25/2006', '03/01/2006', 0, 3)` | Retourneert TRUE. Aangezien de waarde van `first_month_of_year` is ingesteld op `3`, waardoor de zoekperiode 03/01/2006 tot 07/30/2006 wordt in plaats van 01/01/2006 tot 04/30/2006. |

## Voorbeeld 1 – Geen aanvullende argumenten

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties voor 2022 wordt in de tabel `Transactions` geladen.
- Een voorafgaande lading met een extra variabele, `in_months`, die bepaalt of transacties in hetzelfde kwartaal hebben plaatsgevonden als het kwartaal waarin 15 mei 2022 valt.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

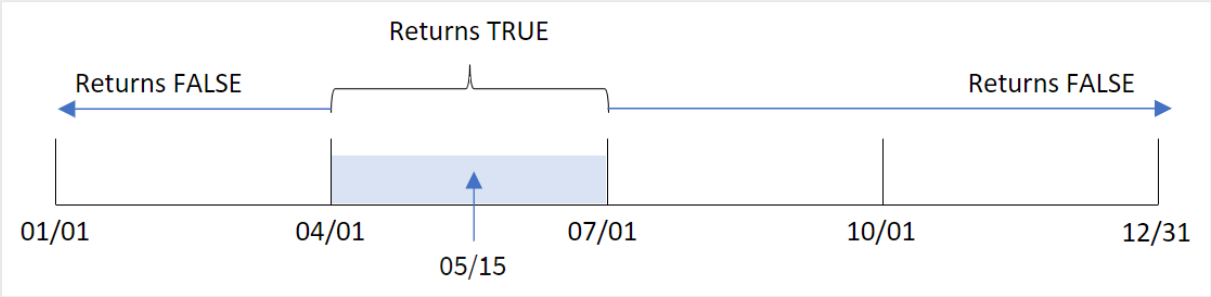
- `date`
- `in_months`

Resultatentabel

| 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 |

Het veld `in_months` wordt gemaakt in de vorige load-instructie met behulp van de functie `inmonths()`. Het eerste opgegeven argument is `3`, waarbij het jaar wordt verdeeld in kwartaalsegmenten. Het tweede argument geeft aan welk veld wordt geëvalueerd; het datumveld in dit voorbeeld. Het derde argument is een vastgestelde datum voor 15 mei. Dit is de `base_date` en een `period_no` van `0` is het laatste argument.

*Diagram van de `inmonths()`-functie met kwartaalsegmenten*




Mei valt in het tweede kwartaal van het jaar. Daarom zal elke transactie die plaatsvindt tussen 1 april en 30 juni een booleaans resultaat voor TRUE retourneren. Dit wordt gevalideerd in de resultatentabel.

## Voorbeeld 2 – period_no

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2022 wordt in de tabel `Transactions` geladen.
- Een voorafgaande lading met een extra variabele, `previous_quarter`, die bepaalt of transacties hebben plaatsgevonden in het kwartaal voorafgaand aan het kwartaal waarin 15 mei 2022 valt.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `previous_quarter`

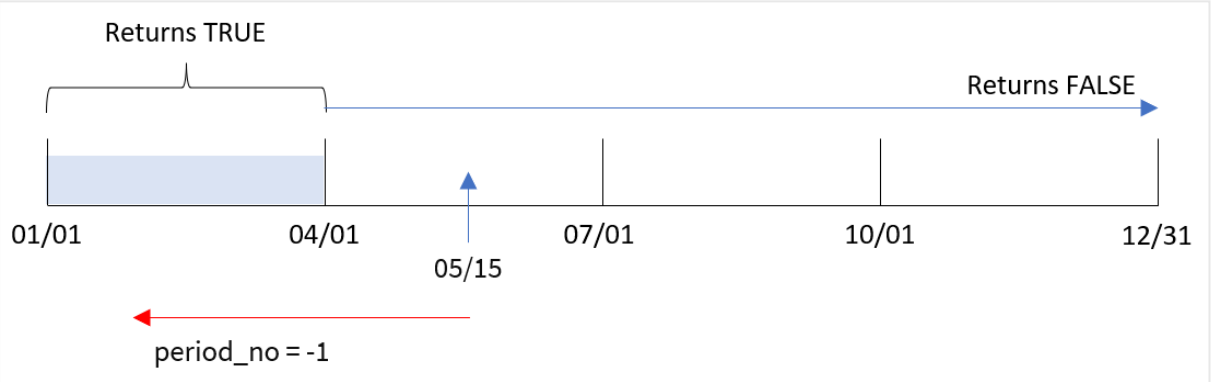
Resultatentabel

| date | vorig kwartaal |
|---|---|
| 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 | vorig kwartaal |
|---|---|
| 8/2/2022 | 0 |
| 8/8/2022 | 0 |
| 8/19/2022 | 0 |
| 9/26/2022 | 0 |
| 10/14/2022 | 0 |
| 10/29/2022 | 0 |

De functie stelt vast of transacties in het eerste kwartaal van het jaar hebben plaatsgevonden met `-1` als het `period_no`-argument in de `inmonths()`-functie. 15 Mei is de `base_date` en valt in het tweede kwartaal van het jaar (april-juni).

*Diagram van de `inmonths()`-functie met kwartaalsegmenten en de `period_no` ingesteld op `-1`*




Daarom zal elke transactie die plaatsvindt tussen januari en maart een booleaans resultaat voor TRUE retourneren.

## Voorbeeld 3 – first_month_of_year

### Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2022 wordt in de tabel `Transactions` geladen.
- Een voorafgaande lading met een extra variabele, `in_months`, die bepaalt of transacties in hetzelfde kwartaal hebben plaatsgevonden als het kwartaal waarin 15 mei 2022 valt.

In dit voorbeeld is maart volgens het organisatiebeleid de eerste maand van het boekjaar.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `in_months`

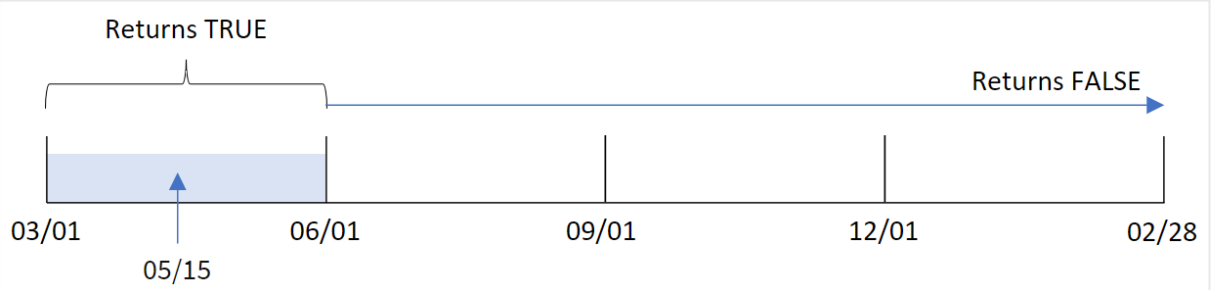
Resultatentabel

| 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 |

Door `3` te gebruiken als het `first_month_of_year`-argument in de `inmonths()`-functie, begint het jaar op 1 maart. De `inmonths()`-functie verdeelt het jaar vervolgens in kwartalen: Mrt-Mei, Jun-Aug, Sep-Nov, Dec-Feb. Daarom valt 15 mei in het eerste kwartaal van het jaar (maart-mei).

*Diagram van functie `inmonths()` met maart ingesteld als de eerste maand van het jaar*




Een transactie die in deze maanden plaatsvindt, retourneert een booleaans resultaat voor TRUE.

## Voorbeeld 4 – Diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Dezelfde gegevensset en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is de gegevensverzameling echter niet gewijzigd en wordt in de applicatie geladen. De berekening die bepaalt welke transacties plaatsvonden in hetzelfde kwartaal als 15 mei, wordt gemaakt als een meting in een diagram in de app.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `date`

Om te berekenen of transacties in hetzelfde kwartaal hebben plaatsgevonden als het kwartaal waarin 15 mei valt, maakt u de volgende meting:

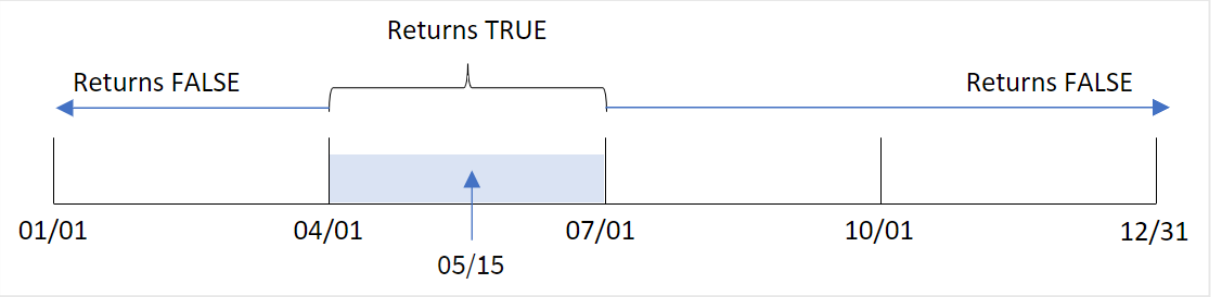
```
=inmonths(3,date,'05/15/2022', 0)
```

Resultatentabel

| 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 |

Het veld `in_months` wordt in een diagram gemaakt met behulp van de functie `inmonths()`. Het eerste opgegeven argument is `3`, waarbij het jaar wordt verdeeld in kwartaalsegmenten. Het tweede argument geeft aan welk veld wordt geëvalueerd; het datumveld in dit voorbeeld. Het derde argument is een vastgestelde datum voor 15 mei. Dit is de `base_date` en een `period_no` van `0` is het laatste argument.

*Diagram van de `inmonths()`-functie met kwartaalsegmenten*




Mei valt in het tweede kwartaal van het jaar. Daarom zal elke transactie die plaatsvindt tussen 1 april en 30 juni een booleaans resultaat voor TRUE retourneren. Dit wordt gevalideerd in de resultatentabel.

## Voorbeeld 5 – Scenario

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset die in de tabel `Products` wordt geladen.
- De tabel bevat de volgende velden:
  - product-id
  - producttype
  - productiedatum
  - kostprijs

De eindgebruiker wil een diagram dat de kosten per producttype weergeeft van de producten die in het eerste segment van 2021 zijn geproduceerd. De gebruiker wil de duur van dit segment kunnen definiëren.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad.

Aan het begin van het load-script is een variabele, `vPeriod`, gemaakt die wordt gekoppeld aan het besturingselement voor variabeleninvoer.

Doe het volgende:

1. Klik in het bedrijfsmiddelenvenster op **Aangepaste objecten**.
2. Selecteer **Qlik Dashboard-bundel** en maak een **Variabeleninvoer**-object.
3. Voer een titel in voor het diagramobject.
4. Selecteer onder **Variabele** de optie **vPeriod** als de naam en stel het object in om als een **Vervolgkeuzelijst** te worden weergegeven.
5. Klik onder **Waarden** op **Dynamische** waarden. Voer het volgende in:
   `='1~month|2~bi-month|3~quarter|4~tertial|6~half-year'.`
6. Voeg een nieuwe tabel toe aan het werkblad.
7. Voeg in het eigenschappenvenster onder **Gegevens** `product_type` toe als een dimensie.
8. Voeg de volgende uitdrukking als een meting toe:
   `=sum(if(inmonths($(vPeriod),manufacture_date,makedate(2022,01,01),0),cost_price,0))`
9. Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

| producttype | =sum(if(inmonths($(vPeriod),manufacture_date,makedate(2022,01,01),0),cost_price,0)) |
|---|---|
| product A | $88.27 |
| product B | $37.23 |
| product C | $17.17 |
| product D | $0.00 |

De `inmonths()`-functie gebruikt de invoer van de gebruiker als argument om de omvang van het beginsegment van het jaar te definiëren. De functie passeert in de productiedatum van elk van de producten als het tweede argument van de `inmonths()`-functie. Door 1 januari als het derde argument in de `inmonths()`-functie te gebruiken, zullen producten met productiedatums die in het beginsegment van het jaar vallen, een booleaanse waarde voor TRUE retourneren. Daarom zal de somfunctie de kosten van die producten optellen.

## inmonthstodate

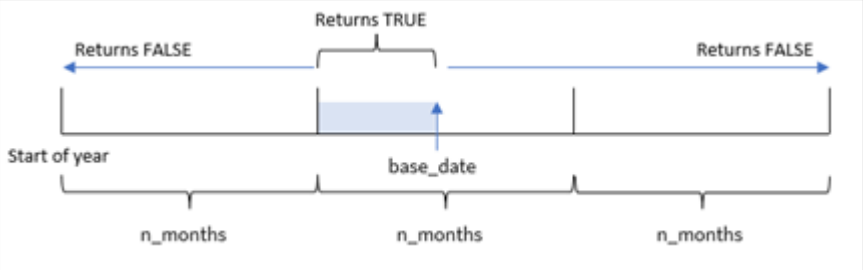
Deze functie bepaalt of een tijdstempel binnen het gedeelte van een periode van de maand, twee maanden, kwartaal, tertiaal of half jaar tot en met de laatste milliseconde van `base_date` valt. Het is tevens mogelijk om te bepalen of de tijdstempel binnen een voorafgaande of volgende tijdsperiode valt.

**Syntaxis:**

```
InMonths (n_months, timestamp, base_date, period_no[, first_month_of_year ])
```

**Retourgegevenstypen:** Booleaanse waarde

*Diagram van `inmonthstodate`-functie.*



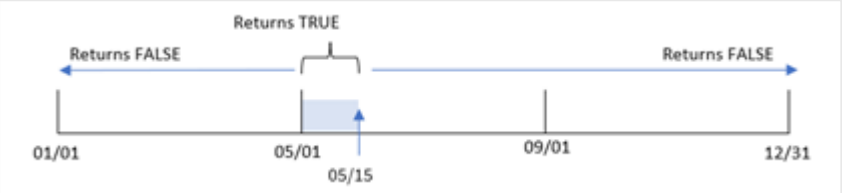

Argumenten

| Argument | Beschrijving |
|---|---|
| **n_months** | Het aantal maanden dat de periode definieert. Een geheel getal of uitdrukking die wordt herleid tot een geheel getal dat een van de volgende waarden moet hebben: 1 (equivalent aan de functie inmonth()), 2 (2 maanden), 3 (equivalent aan de functie inquarter()), 4 (tertiaal) of 6 (half jaar). |
| **timestamp** | De datum die u wilt vergelijken met **base_date**. |
| **base_date** | Datum op basis waarvan de periode wordt geëvalueerd. |
| **period_no** | De periode kan worden verschoven met **period_no**, een geheel getal of een uitdrukking die wordt herleid tot een geheel getal, waarbij de waarde 0 de periode aangeeft die **base_date** bevat. Negatieve waarden in **period_no** geven voorafgaande perioden aan en positieve waarden geven volgende perioden aan. |
| **first_month_of_year** | Als u wilt werken met (boek)jaren die niet in januari beginnen, geeft u een waarde op tussen 2 en 12 in **first_month_of_year**. |

In de `inmonthstodate()`-functie functioneert de `base_date` als het eindpunt van een specifiek jaarsegment waar het onderdeel van is.

Als het jaar bijvoorbeeld in drie segmenten is verdeeld en de `base_date` 15 mei was, zou iedere tijdstempel tussen het begin van januari en het eind van april een booleaans resultaat FALSE retourneren. Datums tussen 1 mei en 15 mei zouden TRUE retourneren. De rest van het jaar zou FALSE retourneren.

*Diagram van booleaans resultatenbereik van de `inmonthstodate`-functie.*




De volgende segmenten van het jaar zijn beschikbaar in de functie als `n_month`-argumenten:

`n_month`-argumenten

| Periode | Aantal maanden |
|---|---|
| maandelijks | 1 |
| tweemaandelijks | 2 |
| kwartaal | 3 |
| viermaandelijks | 4 |
| halfjaarlijks | 6 |

#### Wanneer gebruiken

De functie `inmonthstodate()` retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als voorwaarde in een `if expression`. Door de `inmonthstodate()`-functie te gebruiken, kunt u de periode selecteren die u wilt laten evalueren. Bijvoorbeeld: door een invoervariabele te geven waarmee de gebruiker producten kan identificeren die in de maand, het kwartaal of de jaarhelft van een periode tot een bepaalde datum zijn geproduceerd.

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze

landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `inmonthstodate(4, '01/25/2013', '04/25/2013', 0)` | Retourneert True omdat de waarde van de timestamp, 01/25/2013, binnen de periode van vier maanden van 01/01/2013 tot het einde van 04/25/2013 ligt, waarin de waarde van base_date, 04/25/2013 ligt. |
| `inmonthstodate(4, '04/26/2013', '04/25/2006', 0)` | Retourneert False, omdat 04/26/2013 buiten dezelfde periode als het voorafgaande voorbeeld ligt. |
| `inmonthstodate(4, '09/25/2005', '02/01/2006', -1)` | Retourneert True, omdat de waarde van period_no, -1, de zoekperiode één periode van vier maanden (de waarde van n-months) terugschuift, waardoor de zoekperiode van 01/09/2005 t/m 02/01/2006 valt. |
| `inmonthstodate(4, '04/25/2006', '06/01/2006', 0, 3)` | Retourneert True, omdat de waarde van first_month_of_year is ingesteld op 3, waardoor de zoekperiode van 03/01/2006 t/m 06/01/2006 valt in plaats van van 05/01/2006 t/m 06/01/2006. |

## Voorbeeld 1 – geen aanvullende argumenten

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties voor 2022 wordt in de tabel `Transactions` geladen.
- Een datumveld dat wordt weergegeven in de `DateFormat`-systeemvariabele met de notatie `(MM/DD/YYYY)`.
- Een voorgaande load-instructie met:
    - De `inmonthstodate()`-functie die is ingesteld als het veld `in_months_to_date`. Dit bepaalt welke transacties hebben plaatsgevonden in het kwartaal tot 15 mei 2022.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `in_months_to_date`

Resultatentabel

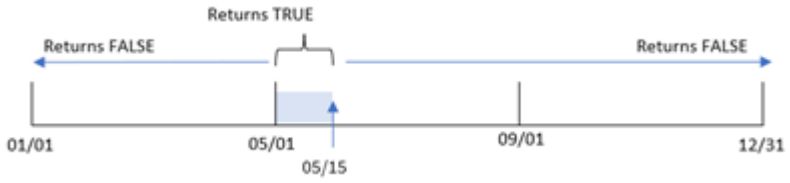
| date | in_months_to_date |
|---|---|
| 1/7/2022 | 0 |
| 1/19/2022 | 0 |
| 2/5/2022 | 0 |
| 2/28/2022 | 0 |
| 3/16/2022 | 0 |
| 4/1/2022 | -1 |
| 5/7/2022 | -1 |
| 5/16/2022 | 0 |
| 6/15/2022 | 0 |

| date | in_months_to_date |
|---|---|
| 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 |

Het veld `in_months_to_date` wordt gemaakt in de vorige load-instructie met behulp van de functie `inmonthstodate()`.

Het eerste opgegeven argument is `3`, waarbij het jaar wordt verdeeld in kwartaalsegmenten. Het tweede argument geeft aan welk veld wordt geëvalueerd. Het derde argument is een vastgestelde datum voor 15 mei. Dit is de `base_date` die de eindgrens van het segment definieert. Een `period_no` van `0` is het laatste argument.

*Diagram van `inmonthstodate`-functie zonder aanvullende argumenten.*




Elke transactie die tussen 1 april en 15 mei plaatsvindt, retourneert een booleaans resultaat voor TRUE. Transactiedatums buiten die periode retourneren FALSE.

## Voorbeeld 2 – period_no

Load-script en resultaten

### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld wordt echter een veld gemaakt, `previous_qtr_to_date`, dat bepaalt of de transacties plaatsvonden in een kwartaal vóór 15 mei.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `previous_qtr_to_date`

Resultatentabel

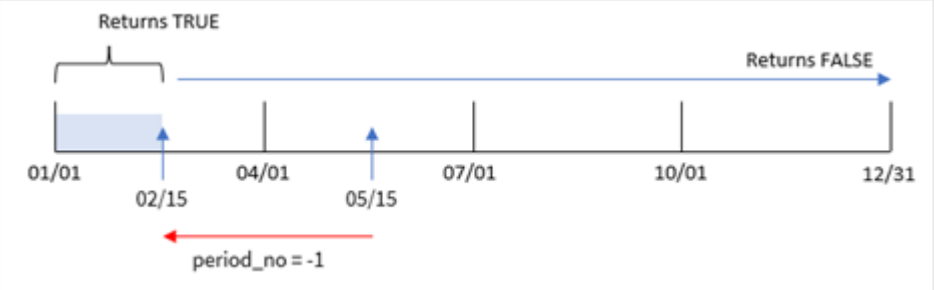
| date | previous_qtr_to_date |
|---|---|
| 1/7/2022 | -1 |
| 1/19/2022 | -1 |
| 2/5/2022 | -1 |
| 2/28/2022 | 0 |

| date | previous_qtr_to_date |
|---|---|
| 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 |

Door `-1` als het `period_no`-argument te gebruiken in de `inmonthstodate()`-functie, verschuift de functie de grenzen van het jaarsegment in de vergelijking met een kwartaal.

15 mei valt in het tweede kwartaal van het jaar en dus komt het segment aanvankelijk overeen met de ingestelde periode tussen 1 april en 15 mei. Het `period_no`-argument verschuift dit segment met een negatieve periode van drie maanden. De datumgrenzen worden 1 januari t/m 15 februari.

*Diagram van functie `inmonthstodate` waarbij period_no is ingesteld op -1.*




Daarom zal elke transactie die plaatsvindt tussen 1 januari en 15 februari een booleaans resultaat voor TRUE retourneren.

## Voorbeeld 3 – first_month_of_year

**Load-script en resultaten**

### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is maart volgens het organisatiebeleid de eerste maand van het boekjaar.

Maak een veld, `in_months_to_date`, dat bepaalt welke transacties hebben plaatsgevonden in hetzelfde kwartaal tot en met 15 mei 2022.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `in_months_to_date`

Resultatentabel

| date | previous_qtr_to_date |
|---|---|
| 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 |

Door `3` te gebruiken als het `first_month_of_year`-argument in de `inmonthstodate()`-functie, begint het jaar op 1 maart en wordt het jaar vervolgens in kwartalen verdeeld op basis van het eerste argument dat is opgegeven. Daarom zijn de kwartaalsegmenten:

- Maa-mei
- Jun-aug
- Sep-nov
- Dec-feb

De `base_date` van 15 mei segmenteert vervolgens het kwartaal van maart tot mei door de eindgrens in te stellen op 15 mei.

*Diagram van de `inmonthstodate`-functie waarbij maart is ingesteld als de eerste maand van het jaar.*

Daarom zal elke transactie die plaatsvindt tussen 1 maart en 15 mei een booleaans resultaat voor TRUE retourneren, terwijl transacties met datums buiten deze grenzen een waarde voor FALSE zullen retourneren.

## Voorbeeld 4 – diagramvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is de gegevensverzameling niet gewijzigd en wordt in de app geladen. Er moet een berekening worden gemaakt die bepaalt welke transacties plaatsvonden in hetzelfde kwartaal als 15 mei, wordt gemaakt als een meting in een diagram in de app.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

```
date
```

Om te berekenen of transacties in hetzelfde kwartaal hebben plaatsgevonden als het kwartaal waarin 15 mei valt, maakt u de volgende meting:

```
=inmonthstodate(3,date,'05/15/2022', 0)
```

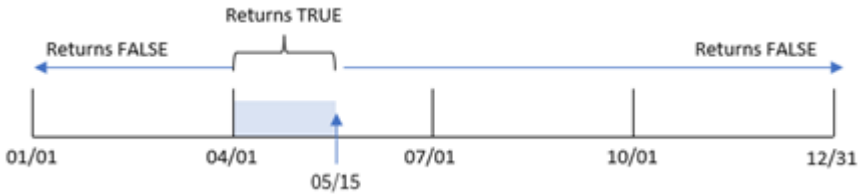
Resultatentabel

| 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 |

De meting ‘in_months_to_date wordt in het diagram gemaakt met behulp van de inmonthstodate()-functie.

Het eerste opgegeven argument is 3, waarbij het jaar wordt verdeeld in kwartaalsegmenten. Het tweede argument geeft aan welk veld wordt geëvalueerd. Het derde argument is een vastgestelde datum voor 15 mei. Dit is de base_date die de eindgrens van het segment definieert. Een period_no van 0 is het laatste argument.

*Diagram van de inmonthstodate-functie met kwartaalsegmenten.*




Elke transactie die tussen 1 april en 15 mei plaatsvindt, retourneert een booleaans resultaat voor TRUE. Transactiedatums buiten dat segment retourneren FALSE.

## Voorbeeld 5 – Scenario

Load-script en resultaten

### Overzicht

In dit voorbeeld wordt een gegevensset geladen in de tabel ‘Sales’. De tabel bevat de volgende velden:

- Product-id
- Producttype
- Verkoopdatum
- Verkoopprijs

De eindgebruiker wil een diagram dat de verkoop van producten op producttype weergeeft van producten die in de periode voorafgaand aan 24 december 2022 zijn verkocht. De gebruiker wil de duur van deze periode kunnen definiëren.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad.

Aan het begin van het load-script is een variabele, `vPeriod`, gemaakt die wordt gekoppeld aan het besturingselement voor variabeleninvoer.

Doe het volgende:

1. Klik in het bedrijfsmiddelenvenster op **Aangepaste objecten**.
2. Selecteer **Qlik Dashboard-bundel** en maak een **Variabeleninvoer**-object.
3. Voer een titel in voor het diagram.
4. Selecteer onder **Variabele** de optie **vPeriod** als de naam en stel het object in om als een **Vervolgkeuzelijst** te worden weergegeven.
5. Klik onder **Waarden** op **Dynamische** waarden. Voer het volgende in:
   `='1~month|2~bi-month|3~quarter|4~tertial|6~half-year'.`
6. Voeg een nieuwe tabel toe aan het werkblad.
7. Voeg in het eigenschappenvenster onder **Gegevens** `product_type` toe als een dimensie.
8. Voeg de volgende uitdrukking als een meting toe:
   `=sum(if(inmonthstodate($(vPeriod),sales_date,makedate(2022,12,24),0),sales_price,0))`
9. Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

| product_type | =sum(if(inmonthstodate($(vPeriod),sales_date,makedate(2022,12,24),0),sales_price,0)) |
|---|---|
| product A | $186.48 |
| product B | $190.52 |

| product_ type | =sum(if(inmonthstodate($(vPeriod),sales_date,makedate(2022,12,24),0),sales_ price,0)) |
|---|---|
| product C | $220.43 |
| product D | $261.46 |

De `inmonthstodate()`-functie gebruikt de invoer van de gebruiker als argument om de omvang van het beginsegment van het jaar te definiëren.

De functie passeert in de verkoopdatum van elk van de producten als het tweede argument van de `inmonthstodate()`-functie. Door 24 december te gebruiken als het derde argument in de `inmonthstodate()`-functie, retourneren producten met een verkoopdatum die binnen de gedefinieerde periode tot en met 24 december valt, een booleaanse waarde voor TRUE. De somfunctie telt de verkoop van deze producten bij elkaar op.

## inmonthtodate

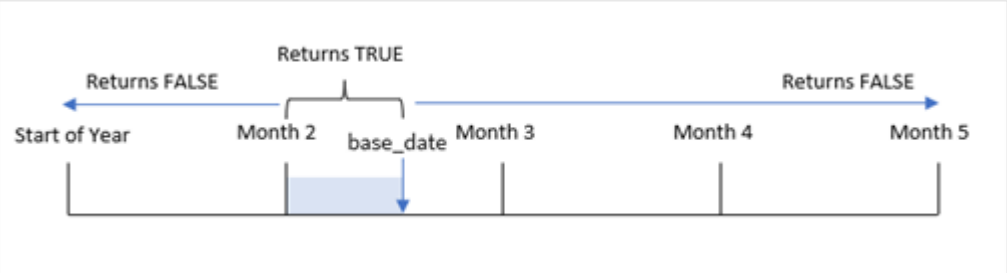
Retourneert True als **date** binnen het gedeelte van de maand ligt dat **basedate** bevat tot en met de laatste milliseconde van **basedate**.

**Syntaxis:**

```
InMonthToDate (timestamp, base_date, period_no)
```

**Retourgegevenstypen:** Booleaanse waarde

*Diagram van `inmonthtodate`-functie.*




De `inmonthtodate()`-functie identificeert een geselecteerde maand als een segment. De begingrenswaarde is het begin van de maand. De eindgrens kan worden ingesteld als een latere datum in de maand. Dit bepaalt vervolgens of een reeks datums al dan niet in dit segment vallen en retourneert een booleaanse waarde voor TRUE of FALSE.

Argumenten

| Argument | Beschrijving |
|---|---|
| **timestamp** | De datum die u wilt vergelijken met **base_date**. |
| **base_date** | Datum op basis waarvan de maand wordt geëvalueerd. |

| Argument | Beschrijving |
|---|---|
| **period_no** | De maand kan worden verschoven met **period_no**. **period_no** is een geheel getal, waarbij de waarde 0 de maand aangeeft die **base_date** bevat. Negatieve waarden in **period_no** geven voorafgaande maanden aan en positieve waarden geven volgende maanden aan. |

#### Wanneer gebruiken

De functie `inmonthtodate()` retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als voorwaarde in een `if expression`. De `inmonthtodate()`-functie retourneert een aggregatie of berekening die afhankelijk is van of een datum plaatsvond in de maand tot en met de desbetreffende datum.

De functie `inmonthtodate()` kan bijvoorbeeld worden gebruikt om alle materieel te identificeren die in een maand tot en met een specifieke datum is geproduceerd.

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `inmonthtodate ('01/25/2013', '25/01/2013', 0)` | Retourneert True |
| `inmonthtodate ('01/25/2013', '24/01/2013', 0)` | Retourneert False |
| `inmonthtodate ('01/25/2013', '28/02/2013', -1)` | Retourneert True |

## Voorbeeld 1 – geen aanvullende argumenten

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties voor 2022 wordt in de tabel `Transactions` geladen.
- Een datumveld dat wordt weergegeven in de `DateFormat`-systeemvariabele (met de notatie `MM/DD/YYYY`).
- Een voorgaande load-instructie met:
    - De `inmonthtodate()`-functie die is ingesteld als het veld `in_month_to_date`. Dit bepaalt welke transacties plaatsvonden tussen 1 juli en 26 juli 2022.

### Load-script

```
SET DateFormat='MM/DD/YYYY';

Transactions:
    Load
    *,
    inmonthtodate(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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- in_month_to_date

Resultatentabel

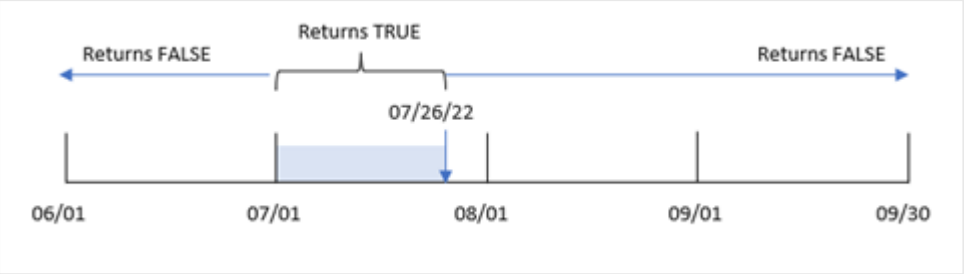
| date | in_month_to_date |
|---|---|
| 1/7/2022 | 0 |
| 1/19/2022 | 0 |
| 2/5/2022 | 0 |
| 2/28/2022 | 0 |
| 3/16/2022 | 0 |
| 4/1/2022 | 0 |
| 5/7/2022 | 0 |
| 5/16/2022 | 0 |
| 6/15/2022 | 0 |
| 6/26/2022 | 0 |
| 7/9/2022 | -1 |
| 7/22/2022 | -1 |
| 7/23/2022 | -1 |
| 7/27/2022 | 0 |
| 8/2/2022 | 0 |
| 8/8/2022 | 0 |
| 8/19/2022 | 0 |
| 9/26/2022 | 0 |
| 10/14/2022 | 0 |
| 10/29/2022 | 0 |

Het veld `in_month_to_date` wordt gemaakt in de vorige load-instructie met behulp van de functie `inmonthtodate()`.

Het eerste argument identificeert welk veld wordt geëvalueerd. Het tweede argument is een vastgestelde datum, 26 juli, die de `base_date` is. Dit `base_date`-argument identificeert welke maand is gesegmenteerd en wat de eindgrens van dat segment is.

Een `period_no` van `0` is het laatste argument, wat betekent dat de functie geen maanden voor of na de gesegmenteerde maand vergelijkt.

*Diagram van `inmonthtodate`-functie zonder aanvullende argumenten.*




Elke transactie die tussen 1 juli en 26 juli plaatsvindt, retourneert een booleaans resultaat TRUE. Alle transacties die in juli na 26 juli plaatsvinden, retourneren een booleaans resultaat FALSE. Hetzelfde geldt voor transacties in een andere maand van het jaar.

## Voorbeeld 2 – period_no

Load-script en resultaten

### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld wordt een veld gemaakt, `six_months_prior`, dat bepaalt of de transacties zes hele maanden vóór 1 juli en 26 juli plaatsvonden.

### Load-script

```
SET DateFormat='MM/DD/YYYY';

Transactions:
    Load
    *,
    inmonthtodate(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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

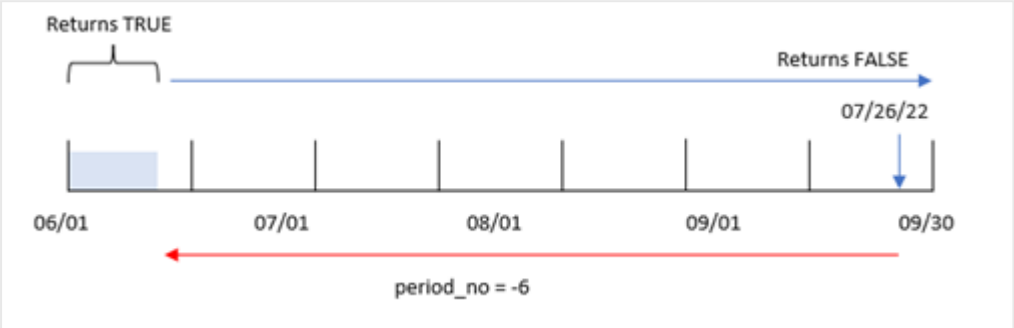
- `date`
- `six_months_prior`

Resultatentabel

| date | six_months_prior |
|---|---|
| 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 |

Door -6 als het period_no-argument te gebruiken in de inmonthtodate()-functie, verschuiven de grenzen van het maandsegment in de vergelijking met zes maanden. In eerste instantie is het maandsegment gelijk aan de periode tussen 1 juli en 26 juli. De period_no verschuift dit segment vervolgens naar zes maanden eerder en de datumgrenzen worden verschoven en vallen tussen 1 januari en 26 januari.

*Diagram van functie inmonthtodate waarbij period_no is ingesteld op -6.*




Elke transactie die tussen 1 januari en 26 januari plaatsvindt, retourneert een booleaans resultaat TRUE.

## Voorbeeld 3 – Diagramvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is de gegevensverzameling niet gewijzigd en wordt in de app geladen. Er moet een berekening worden gemaakt die bepaalt of transacties plaatsvonden tussen 1 juli en 26 juli is gemaakt als meting in een diagram van de app.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

```
date
```

Om te berekenen of transacties plaatsgevonden tussen 1 juli en 26 juli, maakt u de volgende meting:

```
=inmonthtodate(date,'07/26/2022', 0)
```

Resultatentabel

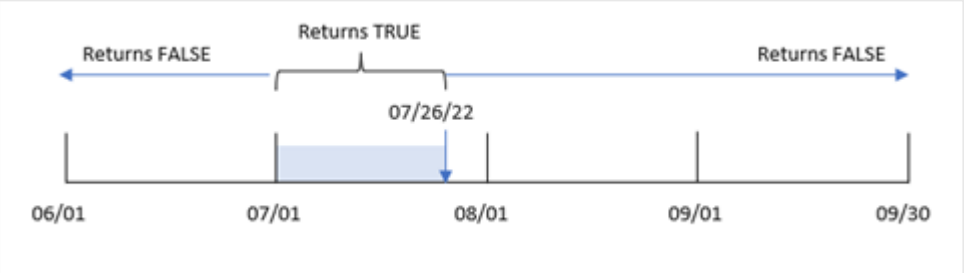
| date | =inmonthtodate(date,'07/26/2022', 0) |
|---|---|
| 1/7/2022 | 0 |
| 1/19/2022 | 0 |
| 2/5/2022 | 0 |
| 2/28/2022 | 0 |
| 3/16/2022 | 0 |
| 4/1/2022 | 0 |
| 5/7/2022 | 0 |
| 5/16/2022 | 0 |
| 6/15/2022 | 0 |
| 6/26/2022 | 0 |
| 7/9/2022 | -1 |
| 7/22/2022 | -1 |
| 7/23/2022 | -1 |
| 7/27/2022 | 0 |
| 8/2/2022 | 0 |
| 8/8/2022 | 0 |
| 8/19/2022 | 0 |

| date | =inmonthtodate(date,'07/26/2022', 0) |
|---|---|
| 9/26/2022 | 0 |
| 10/14/2022 | 0 |
| 10/29/2022 | 0 |

De veldmeting `in_month_to_date` wordt in een diagram gemaakt met behulp van de functie `inmonthtodate ()`.

Het eerste argument identificeert welk veld wordt geëvalueerd. Het tweede argument is een vastgestelde datum, 26 juli, die de `base_date` is. Dit base_date-argument identificeert welke maand is gesegmenteerd en wat de eindgrens van dat segment is. Een `period_no` van 0 is het laatste argument. Dit betekent dat de functie geen maanden voor of na de gesegmenteerde maand vergelijkt.

*Diagram van `inmonthtodate`-functie zonder aanvullende argumenten.*




Elke transactie die tussen 1 juli en 26 juli plaatsvindt, retourneert een booleaans resultaat TRUE. Alle transacties die in juli na 26 juli plaatsvinden, retourneren een booleaans resultaat FALSE. Hetzelfde geldt voor transacties in een andere maand van het jaar.

## Voorbeeld 4 – Scenario

Load-script en resultaten

**Overzicht**

In dit voorbeeld wordt een gegevensset geladen in de tabel '`Products`'. De tabel bevat de volgende velden:

- Product-id
- Productiedatum
- Kostprijs

Door een fout in de apparatuur waren producten die in de maand juli 2022 zijn geproduceerd defect. Het probleem is op 27 juli 2022 opgelost.

De eindgebruiker wil graag een diagram dat op maand de status toont van producten die zijn geproduceerd als 'defect' (booleaans WAAR) of 'foutloos' (booleaans NIET WAAR) en de kosten van de producten die in die maand zijn geproduceerd.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `=monthname(manufacture_date)`
- `=if(Inmonthtodate(manufacture_date,makedate(2022,07,26),0),'Defective','Faultless')`

Maak deze meting om de som van de kosten van de producten te berekenen:

```
=sum(cost_price)
```

Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

| monthname (manufacture_date) | if(Inmonthtodate(manufacture_date,makedate (2022,07,26),0),'Defect','Foutloos') | Sum(cost_ price) |
|---|---|---|
| januari 2022 | Foutloos | $54.40 |
| februari 2022 | Foutloos | $145.69 |
| maart 2022 | Foutloos | $53.80 |

| monthname (manufacture_date) | if(Inmonthtodate(manufacture_date,makedate (2022,07,26),0),'Defect','Foutloos') | Sum(cost_ price) |
|---|---|---|
| april 2022 | Foutloos | $82.06 |
| mei 2022 | Foutloos | $127.60 |
| juni 2022 | Foutloos | $141.82 |
| juli 2022 | Defect | $144.66 |
| juli 2022 | Foutloos | $69.98 |
| augustus 2022 | Foutloos | $147.46 |
| september 2022 | Foutloos | $84.21 |
| oktober 2022 | Foutloos | $163.91 |

De functie `inmonthtodate()` retourneert een booleaanse waarde bij het evalueren van de productiedatums van elk product.

Voor de datums die een booleaanse waarde voor TRUE retourneren, wordt het product gemarkeerd als Defect. Voor elk product dat de waarde voor FALSE retourneert, en dus niet op in de maand tot en met 26 juli is geproduceerd, wordt het product als Foutloos gemarkeerd.

## inquarter

Deze functie retourneert True als **timestamp** binnen het kwartaal valt dat **base_date** bevat.

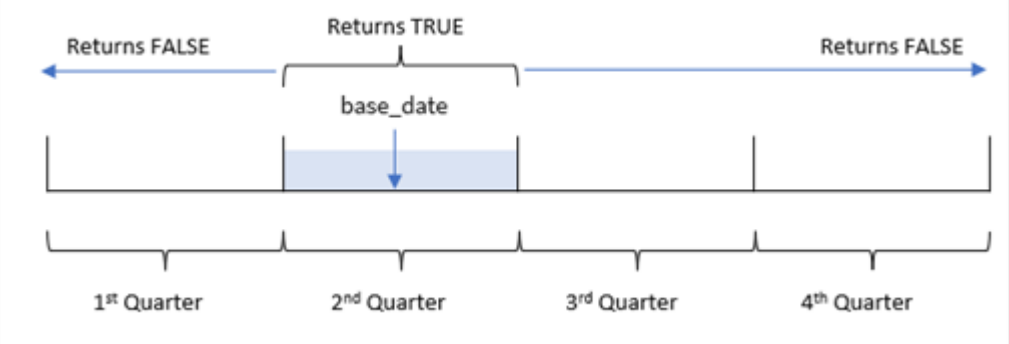
**Syntaxis:**

```
InQuarter (timestamp, base_date, period_no[, first_month_of_year])
```

**Retourgegevenstypen:** Booleaanse waarde

In Qlik Sense wordt de booleaanse waarde Waar vertegenwoordigd door -1 en de waarde Onwaar door 0.

*Diagram van het bereik van de `inquarter()`-functie*




Met andere woorden: de `inquarter()`-functie verdeelt het jaar in vier gelijke kwartalen tussen 1 januari en 31 december. U kunt het `first_month_of_year`-argument gebruiken om te veranderen welke maand als de eerste maand wordt gezien in uw app. Op basis van dat argument veranderen ook de kwartalen. De `base_`

`date`-functie identificeert welk kwartaal moet worden gebruikt als de vergelijkingsdatum voor de functie. Ten slotte retourneert de functie een booleaans resultaat wanneer de datumwaarden met dat segment worden vergeleken.

### Wanneer gebruiken

De functie `inquarter()` retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als voorwaarde in een `if expression`. Hierdoor wordt een aggregatie of berekening geretourneerd, afhankelijk van of een datum al dan niet in het geselecteerde kwartaal valt.

De `inquarter()`-functie kan bijvoorbeeld worden gebruikt om alle apparatuur te identificeren die in een kwartaalsegment is geproduceerd op basis van de datums waarop de apparatuur is geproduceerd.

Argumenten

| Argument | Beschrijving |
|---|---|
| **timestamp** | De datum die u wilt vergelijken met **base_date**. |
| **base_date** | Datum op basis waarvan het kwartaal wordt geëvalueerd. |
| **period_no** | Het kwartaal kan worden verschoven met **period_no**. **period_no** is een geheel getal, waarbij de waarde 0 het kwartaal aangeeft dat **base_date** bevat. Negatieve waarden in **period_no** geven voorafgaande kwartalen aan en positieve waarden geven volgende kwartalen aan. |
| **first_month_of_year** | Als u wilt werken met (boek)jaren die niet in januari beginnen, geeft u een waarde op tussen 2 en 12 in **first_month_of_year**. |

U kunt de volgende waarden gebruiken om de eerste maand van het jaar in te stellen in het argument `first_month_of_year`:

waarden first_month_of_year

| Maand | Waarde |
|---|---|
| Februari | 2 |
| Maart | 3 |
| April | 4 |
| Mei | 5 |
| Juni | 6 |
| Juli | 7 |
| Augustus | 8 |
| September | 9 |
| Oktober | 10 |
| November | 11 |
| December | 12 |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| inquarter ('01/25/2013', '01/01/2013', 0) | Retourneert TRUE |
| inquarter ('01/25/2013', '04/01/2013', 0) | Retourneert FALSE |
| inquarter ('01/25/2013', '01/01/2013', -1) | Retourneert FALSE |
| inquarter ('12/25/2012', '01/01/2013', -1) | Retourneert TRUE |
| inquarter ('01/25/2013', '03/01/2013', 0, 3) | Retourneert FALSE |
| inquarter ('03/25/2013', '03/01/2013', 0, 3) | Retourneert TRUE |

## Voorbeeld 1 – Geen aanvullende argumenten

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een reeks transacties in 2022 die wordt geladen in de tabel `Transactions`.
- Een voorafgaande lading met de `inquarter()`-functie die is ingesteld als het `in_quarter`-veld en die bepaalt welke transacties in hetzelfde kwartaal hebben plaatsgevonden als het kwartaal waarin 15 mei 2022 valt.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `in_quarter`

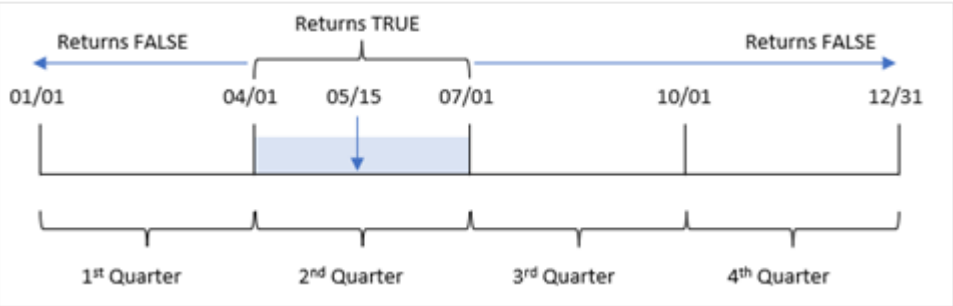
Resultatentabel

| 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 |

Het veld `in_quarter` wordt gemaakt in de vorige load-instructie met behulp van de functie `inquarter()`. Het eerste argument identificeert welk veld wordt geëvalueerd. Het tweede argument is een vastgestelde datum voor 15 mei die identificeert welk kwartaal als de vergelijker moet worden gedefinieerd. Een `period_no` van `0` is het laatste argument en zorgt ervoor dat de `inquarter()`-functie geen kwartalen voor of na het gesegmenteerde kwartaal vergelijkt.

*Diagram van de `inquarter()`-functie met 15 mei als de basisdatum*




Elke transactie die tussen 1 april en het eind van 30 juni plaatsvindt, retourneert een booleaans resultaat TRUE.

## Voorbeeld 2 – period_no

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met een reeks transacties in 2022 die wordt geladen in de tabel `Transactions`.
- Een voorafgaande lading met de `inquarter()`-functie die is ingesteld als het `previous_quarter`-veld en die bepaalt welke transacties in het kwartaal hebben plaatsgevonden voorafgaand aan het kwartaal waarin 15 mei 2022 valt.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

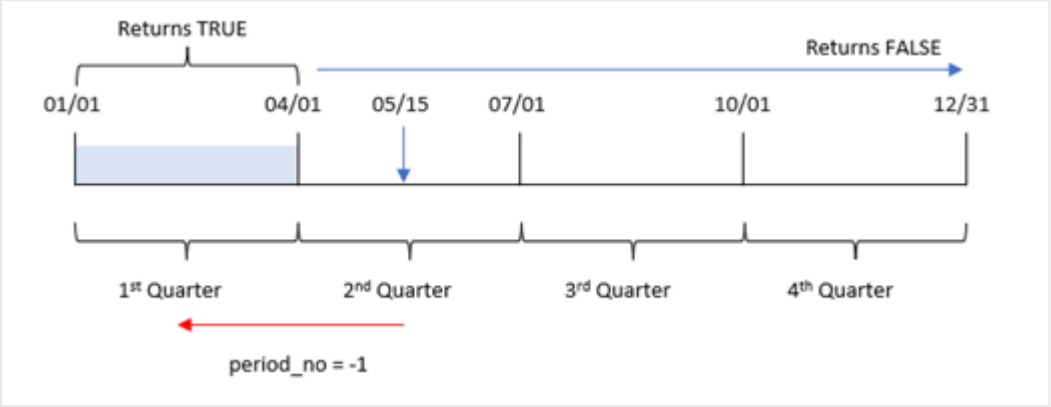
- `date`
- `previous_qtr`

Resultatentabel

| 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 |

Door `-1` als het `period_no`-argument in de `inquarter()`-functie te gebruiken, verschuift de functie de grenzen van het vergelijkingskwartaal terug met een heel een kwartaal. 15 Mei valt in het tweede kwartaal van het jaar en dus komt het segment aanvankelijk overeen met de ingestelde periode tussen 1 april en 30 juni. De `period_no` compenseert dit segment vervolgens drie maanden eerder, waardoor de datumgrenzen van 1 januari tot 30 maart worden.

*Diagram van de `inquarter()`-functie met 15 mei als de basisdatum*




Daarom zal elke transactie die plaatsvindt tussen 1 januari en 30 maart een booleaans resultaat voor TRUE retourneren.

## Voorbeeld 3 – first_month_of_year

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met een reeks transacties in 2022 die wordt geladen in de tabel `Transactions`.
- Een voorafgaande lading met de `inquarter()`-functie die is ingesteld als het `in_quarter`-veld en die bepaalt welke transacties in hetzelfde kwartaal hebben plaatsgevonden als het kwartaal waarin 15 mei 2022 valt.

In dit voorbeeld is maart volgens het organisatiebeleid echter de eerste maand van het boekjaar.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `previous_qtr`

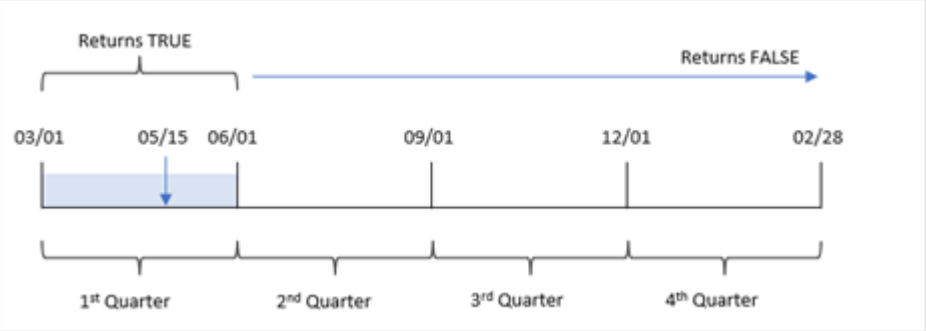
Resultatentabel

| 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 |
| 7/27/2022 | 0 |

| date | previous_qtr |
|---|---|
| 8/2/2022 | 0 |
| 8/8/2022 | 0 |
| 8/19/2022 | 0 |
| 9/26/2022 | 0 |
| 10/14/2022 | 0 |
| 10/29/2022 | 0 |

Door `3` te gebruiken als het `first_month_of_year`-argument in de `inquarter()`-functie, wordt 1 maart ingesteld als het begin van het jaar en wordt het jaar vervolgens in kwartalen verdeeld. Daardoor zijn de kwartaalsegmenten maa-mei, jun-aug, sep-nov en dec-feb. De `base_date` van 15 mei stelt het maa-mei-kwartaal in als het vergelijkingskwartaal voor de functie.

*Diagram van functie `inquarter()` met maart ingesteld als de eerste maand van het jaar*




Daarom zal elke transactie die plaatsvindt tussen 1 maart en 31 mei een booleaans resultaat voor TRUE retourneren.

## Voorbeeld 4 – Diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met een reeks transacties in 2022 die wordt geladen in de tabel `Transactions`.
- Een voorafgaande lading met de `inquarter()`-functie die is ingesteld als het `in_quarter`-veld en die bepaalt welke transacties in hetzelfde kwartaal hebben plaatsgevonden als het kwartaal waarin 15 mei 2022 valt.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `date`

Om te berekenen of transacties in hetzelfde kwartaal hebben plaatsgevonden als het kwartaal waarin 15 mei valt, maakt u de volgende meting:

```
=inquarter(date,'05/15/2022', 0)
```

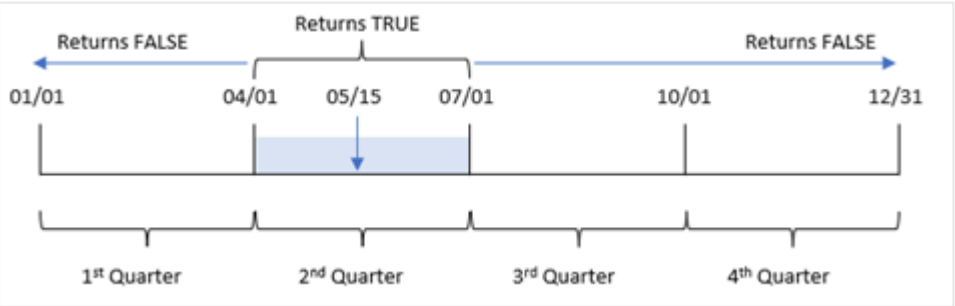
Resultatentabel

| 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 |

De meting `in_quarter` wordt in het diagram gemaakt met behulp van de functie `inquarter()`. Het eerste argument identificeert welk veld wordt geëvalueerd. Het tweede argument is een vastgestelde datum voor 15 mei die identificeert welk kwartaal als de vergelijker moet worden gedefinieerd. Een `period_no` van `0` is het laatste argument en zorgt ervoor dat de `inquarter()`-functie geen kwartalen voor of na het gesegmenteerde kwartaal vergelijkt.

*Diagram van de `inquarter()`-functie met 15 mei als de basisdatum*




Elke transactie die tussen 1 april en het eind van 30 juni plaatsvindt, retourneert een booleaans resultaat TRUE.

## Voorbeeld 5 – Scenario

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset die in de tabel `Products` wordt geladen.
- De tabel bevat de volgende velden:
    - product-id
    - producttype
    - productiedatum
    - kostprijs

Er is vastgesteld dat producten die in het kwartaal van 15 mei 2022 zijn geproduceerd, defect zijn vanwege een materieelfout. De eindgebruiker wil graag een diagram dat op kwartaalnaam de status toont van welke geproduceerde producten 'defect' of 'foutloos' zijn en de kosten van de producten die in die maand zijn geproduceerd.

### Load-script

```
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
];
```

**Resultaten**

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

```
=quartername(manufacture_date)
```

Maak de volgende metingen:

- `=if(only(InQuarter(manufacture_date,makedate(2022,05,15),0)),'Defective','Faultless')`, om te identificeren welke van de producten defect zijn en welke foutloos zijn met behulp van de `inquarter()`-functie.
- `=sum(cost_price)`, om de som van de kosten van elk product te tonen.

**Doe het volgende:**

1. Stel de **Getalnotatie** van de meting in op **Geld**.
2. Onder **Uiterlijk** schakelt u **Totaal** uit.

Resultatentabel

| quartername (manufacture_date) | =if(only(InQuarter(manufacture_date,makedate (2022,05,15),0)),'Defective','Faultless') | Sum(cost_ price) |
|---|---|---|
| jan-mrt 2022 | Foutloos | 253.89 |
| april-jun 2022 | Defect | 351.48 |
| juli-sept 2022 | Foutloos | 446.31 |
| okt-dec 2022 | Foutloos | 163.91 |

De functie `inquarter()` retourneert een booleaanse waarde bij het evalueren van de productiedatums van elk product. Voor elk product dat in het kwartaal waarin 15 mei valt, wordt geproduceerd, retourneert de `inquarter()`-functie een booleaanse waarde TRUE en markeert het de producten als Defect. Voor elk product dat de waarde FALSE retourneert, en dus niet in dat kwartaal is geproduceerd, wordt het product als Foutloos gemarkeerd.

## inquartertodate

Deze functie retourneert True als **timestamp** binnen het gedeelte van het kwartaal ligt dat **base_date** bevat tot en met de laatste milliseconde van **base_date**.

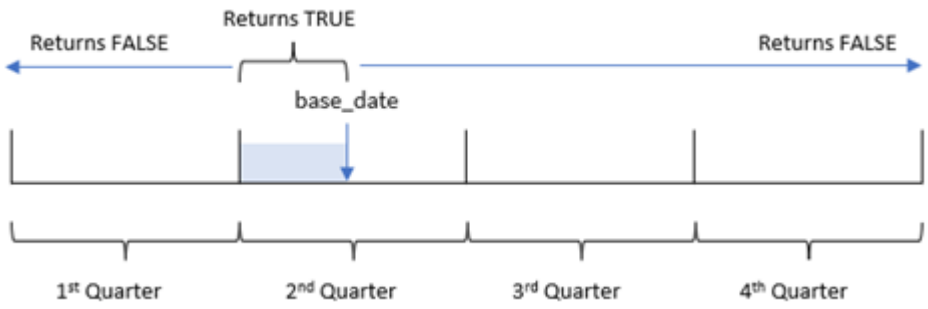
**Syntaxis:**

```
InQuarterToDate (timestamp, base_date, period_no [, first_month_of_year])
```

**Retourgegevenstypen:** Booleaanse waarde

> *In Qlik Sense wordt de booleaanse waarde Waar vertegenwoordigd door -1 en de waarde Onwaar door 0.*

*Diagram van functie inquartertodate*




De functie `inquartertodate()` verdeelt het jaar in vier gelijke kwartalen tussen 1 januari en 31 december (of het door de gebruiker gedefinieerde begin van het jaar en de bijbehorende einddatum). Met de `base_date` zal de functie dan een bepaald kwartaal segmenteren, waarbij de `base_date` zowel welk kwartaal als de maximaal toegestane datum voor dat kwartaalsegment aangeeft. Ten slotte retourneert de functie een booleaans resultaat bij het vergelijken van de voorgeschreven datumwaarden met dat segment.

Argumenten

| Argument | Beschrijving |
|---|---|
| **timestamp** | De datum die u wilt vergelijken met **base_date**. |
| **base_date** | Datum op basis waarvan het kwartaal wordt geëvalueerd. |
| **period_no** | Het kwartaal kan worden verschoven met **period_no**. **period_no** is een geheel getal, waarbij de waarde 0 het kwartaal aangeeft dat **base_date** bevat. Negatieve waarden in **period_no** geven voorafgaande kwartalen aan en positieve waarden geven volgende kwartalen aan. |
| **first_month_of_year** | Als u wilt werken met (boek)jaren die niet in januari beginnen, geeft u een waarde op tussen 2 en 12 in **first_month_of_year**. |

## Wanneer gebruiken

De functie `inquartertodate()` retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als voorwaarde in een `if`-uitdrukking. De `inquartertodate()`-functie wordt gebruikt om een aggregatie of berekening te retourneren afhankelijk van of een geëvalueerde datum plaatsvond in het kwartaal tot en met de betreffende datum.

De functie `inquartertodate()` bijvoorbeeld kan worden gebruikt om alle materieel te identificeren die in een kwartaal tot en met een specifieke datum is geproduceerd.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `inquartertodate ('01/25/2013', '03/25/2013', 0)` | Retourneert`TRUE`, aangezien de waarde van `timestamp`, 01/25/2013, binnen de periode ligt van drie maanden vanaf 01/01/2013 tot 03/25/2013, waarin de waarde van `base_date`, 03/25/2013, ligt. |

| Voorbeeld | Resultaat |
|---|---|
| `inquartertodate ('04/26/2013', '03/25/2013', 0)` | Retourneert `FALSE`, aangezien 04/26/2013 buiten dezelfde periode als het vorige voorbeeld ligt. |
| `inquartertodate ('02/25/2013', '06/09/2013', -1)` | Retourneert `TRUE`, aangezien de waarde van `period_no`, -1, de zoekperiode één periode van drie maanden terugschuift (een kwartaal van het jaar). Dit maakt de zoekperiode 01/01/2013 tot 03/09/2013. |
| `inquartertodate ('03/25/2006', '04/15/2006', 0, 2)` | Retourneert `TRUE`, aangezien de waarde van `first_month_of_year` is ingesteld op 2, waardoor de zoekperiode 02/01/2006 tot 04/15/2006 wordt in plaats van 04/01/2006 tot 04/15/2006. |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – geen aanvullende argumenten

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties voor 2022 die wordt geladen in de tabel `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.
- Het aanmaken van een veld, `in_quarter_to_date`, dat bepaalt welke transacties hebben plaatsgevonden in het kwartaal tot 15 mei 2022.

### Load-script

```
SET DateFormat='MM/DD/YYYY';

Transactions:
    Load
        *,
        inquartertodate(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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `in_quarter_to_date`

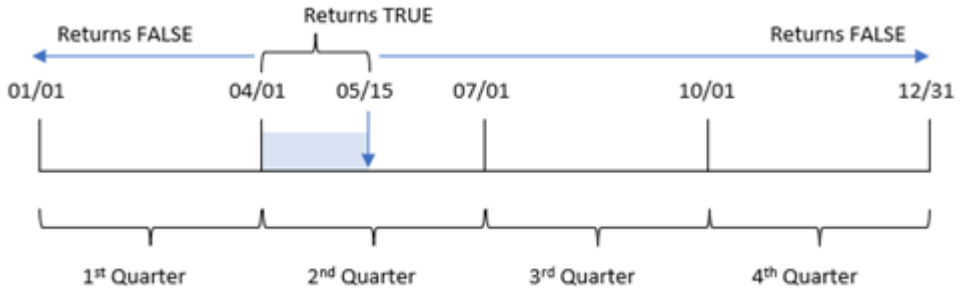
Resultatentabel

| date | in_quarter_to_date |
|---|---|
| 1/7/2022 | 0 |
| 1/19/2022 | 0 |
| 2/5/2022 | 0 |
| 2/28/2022 | 0 |

| date | in_quarter_to_date |
|---|---|
| 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 |

Het veld `in_quarter_to_date` wordt gemaakt in de vorige load-instructie met behulp van de functie `inquartertodate()`. Het eerste verstrekte argument geeft aan welk veld wordt geëvalueerd. Het tweede argument is een vastgestelde datum voor 15 mei, de `base_date` die aangeeft welk kwartaal moet worden gesegmenteerd en de eindgrens van dat segment definieert. Een `period_no` van 0 is het laatste argument, wat betekent dat de functie geen kwartalen voor of na het gesegmenteerde kwartaal vergelijkt.

*Diagram van functie inquartertodate, geen aanvullende argumenten*




Elke transactie die tussen 1 april en 15 mei plaatsvindt, retourneert een booleaans resultaat van TRUE. Transactiedatums van 16 mei en later zullen FALSE retourneren, net als alle transacties vóór 1 april.

## Voorbeeld 2 – period_no

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het aanmaken van een veld, `previous_qtr_to_date`, dat bepaalt welke transacties een volledig kwartaal voor het einde van het kwartaalsegment op 15 mei 2022 hebben plaatsgevonden.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

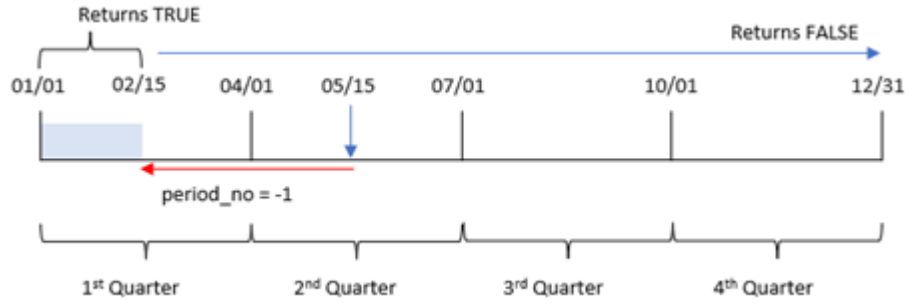
- `date`
- `previous_qtr_to_date`

Resultatentabel

| date | previous_qtr_to_date |
|---|---|
| 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 |

Een `period_no`-waarde van -1 geeft aan dat de functie `inquartertodate ()` het invoerkwartaalsegment vergelijkt met het voorgaande kwartaal. 15 mei valt in het tweede kwartaal van het jaar, dus het segment komt aanvankelijk overeen met tussen 1 april en 15 mei. De `period_no` compenseert dit segment vervolgens drie maanden eerder, waardoor de datumgrenzen van 1 januari tot 15 februari worden.

*Diagram van functie inquartertodate, period_no example*




Daarom zal elke transactie die plaatsvindt tussen 1 januari en 15 februari een booleaans resultaat van TRUE retourneren.

## Voorbeeld 3 – first_month_of_year

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het aanmaken van een veld, `in_quarter_to_date`, dat bepaalt welke transacties hebben plaatsgevonden in hetzelfde kwartaal tot en met 15 mei 2022.

In dit voorbeeld stellen we maart in als de eerste maand van het fiscale jaar.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `in_quarter_to_date`

Resultatentabel

| date | in_quarter_to_date |
|---|---|
| 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 |

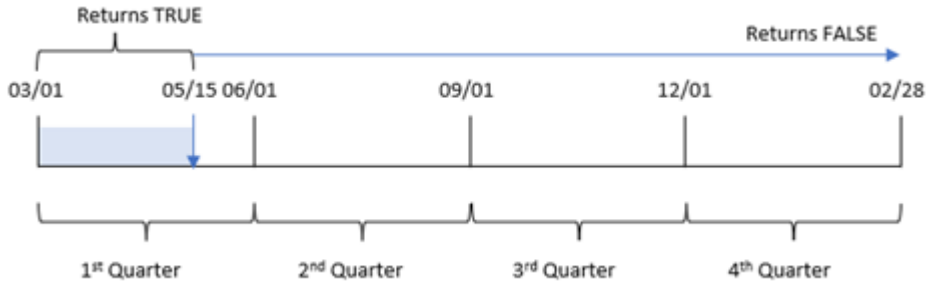
| date | in_quarter_to_date |
|---|---|
| 8/19/2022 | 0 |
| 9/26/2022 | 0 |
| 10/14/2022 | 0 |
| 10/29/2022 | 0 |

Door 3 te gebruiken als het `first_month_of_year`-argument in de `inquartertodate()`-functie, begint de functie het jaar op 1 maart en verdeelt het jaar vervolgens in kwartalen. Daarom zijn de kwartaalsegmenten:

- maart tot mei
- juni tot augustus
- september tot november
- December tot februari

De `base_date` van 15 mei segmenteert vervolgens het kwartaal van maart tot mei door de eindgrens in te stellen op 15 mei.

*Diagram van functie inquartertodate, first_month_of_year example*




Daarom zal elke transactie die plaatsvindt tussen 1 maart en 15 mei een booleaans resultaat van TRUEretourneren, terwijl transacties met datums buiten deze grenzen een waarde van FALSE zullen retourneren.

## Voorbeeld 4 – diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld. In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die bepaalt welke transacties plaatsvonden in hetzelfde kwartaal als 15 mei is gemaakt als een meting in het diagramobject.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:`date`.

Maak de volgende meting:

```
=inquartertodate(date,'05/15/2022', 0)
```

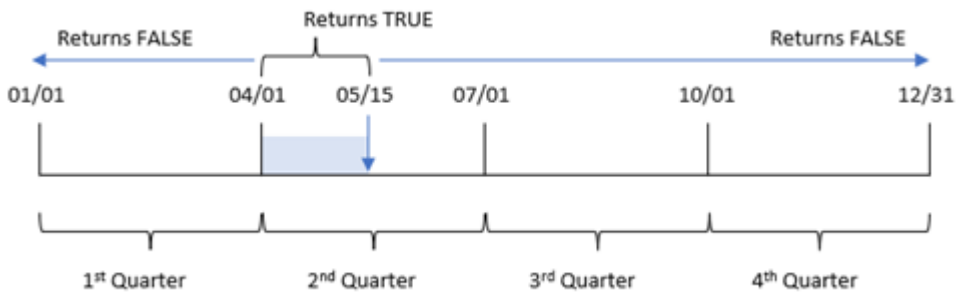
Resultatentabel

| date | =inquartertodate(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 |

| date | =inquartertodate(date,'05/15/2022', 0) |
|---|---|
| 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 |

De meting `in_quarter_to_date` wordt in een diagramobject gemaakt met behulp van de functie `inquartertodate()`. Het eerste argument is het datumveld dat wordt geëvalueerd. Het tweede argument is een vastgestelde datum voor 15 mei, de `base_date` die aangeeft welk kwartaal moet worden gesegmenteerd en de eindgrens van dat segment definieert. Een `period_no` van 0 is het laatste argument, wat betekent dat de functie geen kwartalen voor of na het gesegmenteerde kwartaal vergelijkt.

*Diagram van functie inquartertodate, diagramobjectvoorbeeld*




Elke transactie die tussen 1 april en 15 mei plaatsvindt, retourneert een booleaans resultaat van `TRUE`. Transacties op 16 mei en later zullen `FALSE` retourneren, net als alle transacties vóór 1 april.

## Voorbeeld 5 – Scenario

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset die wordt geladen in een tabel met de naam `Products`.
- Informatie over product-id, productiedatum en kostprijs.

Op 15 mei 2022 werd tijdens het productieproces een fout in de apparatuur vastgesteld en opgelost. Producten die tot die datum in dat kwartaal zijn geproduceerd, zullen defect zijn. De eindgebruiker wil graag een diagramobject dat op kwartaalnaam de status toont of het product 'defect' of 'foutloos' is en de kosten van producten die zijn geproduceerd in dat kwartaal.

### Load-script

```
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
];
```

**Resultaten**

**Doe het volgende:**

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel. Maak een dimensie om de kwartaalnamen weer te geven:
   `=quartername(manufacture_date)`
2. Maak vervolgens een dimensie om te identificeren welke van de producten defect zijn en welke foutloos zijn:
   `=if(inquartertodate(manufacture_date,makedate(2022,05,15),0),'Defective','Faultless')`
3. Maak een meting om de `cost_price` van de producten op te tellen:
   `=sum(cost_price)`
4. Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

| quartername (manufacture_date) | if(inquartertodate(manufacture_date,makedate (2022,05,15),0),'Defective','Faultless') | Sum(cost_ price) |
|---|---|---|
| jan-mrt 2022 | Foutloos | $253.89 |
| april-jun 2022 | Foutloos | $229.03 |
| april-jun 2022 | Defect | $122.45 |
| juli-sept 2022 | Foutloos | $446.31 |
| okt-dec 2022 | Foutloos | $163.91 |

De functie `inquartertodate()` retourneert een booleaanse waarde bij het evalueren van de productiedatums van elk product. Voor degenen die een booleaanse waarde van `TRUE` retourneren, markeert het de producten als `'Defective'`. Voor elk product dat een waarde retourneert van `FALSE`, en dus niet gemaakt in het kwartaal tot en met 15 mei, markeert het de producten als `'Faultless'`.

## inweek

Deze functie retourneert True als **timestamp** binnen de week valt die **base_date** bevat.

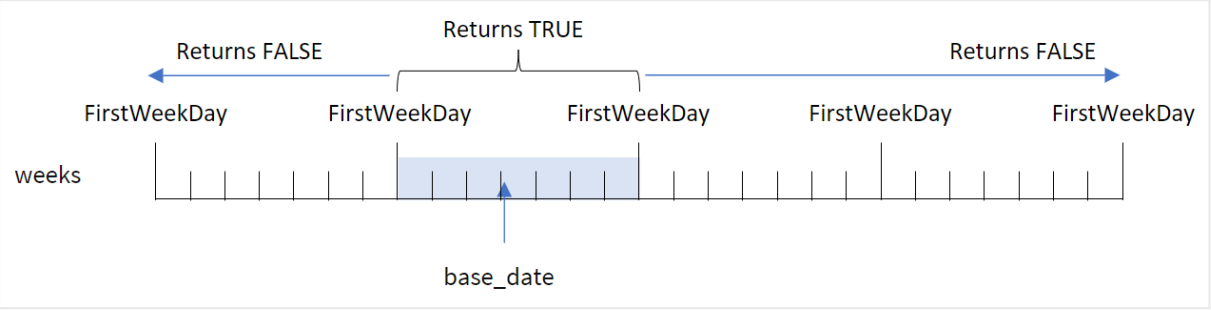
**Syntaxis:**

```
InWeek (timestamp, base_date, period_no[, first_week_day])
```

**Retourgegevenstypen:** Booleaanse waarde

In Qlik Sense wordt de booleaanse waarde Waar vertegenwoordigd door -1 en de waarde Onwaar door 0.

*Diagram van het bereik van de `inweek()`-functie*




De `inweek()`-functie gebruikt het argument `base_date` om te identificeren in welke zevendaagse periode dag valt. De begindag van de week wordt gebaseerd op de systeemvariabele `FirstWeekDay`. U kunt echter ook de eerste dag van de week wijzigen door het `first_week_day`-argument in de `inweek()`-functie te gebruiken.

Nadat de geselecteerde week is gedefinieerd, retourneert de functie booleaanse resultaten bij het vergelijken van de voorgeschreven datumwaarden met dat weeksegment.

### Wanneer gebruiken

De functie `InWeek` retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als voorwaarde in een `if expression`. De `InWeek`-functie retourneert een aggregatie of berekening die afhankelijk is van of een geëvalueerde datum in de week valt met de geselecteerde datum van het `base_date`-argument.

De `InWeek`-functie kan bijvoorbeeld worden gebruikt om alle materieel te identificeren dat in een specifieke week is geproduceerd.

Argumenten

| Argument | Beschrijving |
|---|---|
| **timestamp** | De datum die u wilt vergelijken met **base_date**. |
| **base_date** | Datum op basis waarvan de week wordt geëvalueerd. |
| **period_no** | De week kan worden verschoven met **period_no**. **period_no** is een geheel getal, waarbij de waarde 0 de week aangeeft die **base_date** bevat. Negatieve waarden in **period_no** geven voorgaande weken aan en positieve waarden geven volgende weken aan. |
| **first_week_day** | Standaard is de eerste dag van de week zondag (zoals bepaald door de systeemvariabele FirstWeekDay), beginnend om middernacht tussen zaterdag en zondag. De parameter **first_week_day** vervangt de variabele **FirstWeekDay**. Geef aan dat de week op een andere dag begint door een vlag tussen 0 en 6 op te geven. |

waarden first_week_day

| Dag | Waarde |
|---|---|
| Maandag | 0 |
| Dinsdag | 1 |
| Woensdag | 2 |
| Donderdag | 3 |
| Vrijdag | 4 |
| Zaterdag | 5 |
| Zondag | 6 |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `inweek ('01/12/2006', '01/14/2006', 0)` | Retourneert TRUE |
| `inweek ('01/12/2006', '01/20/2006', 0 )` | Retourneert FALSE |
| `inweek ('01/12/2006', '01/14/2006', -1 )` | Retourneert FALSE |
| `inweek ('01/07/2006', '01/14/2006', -1)` | Retourneert TRUE |

| Voorbeeld | Resultaat |
|---|---|
| `inweek ('01/12/2006', '01/09/2006', 0, 3)` | Retourneert FALSE omdat `first_week_day` is gespecificeerd als 3 (donderdag), waardoor 01/12/2006 de eerste dag van de week is na de week waarin 01/09/2006 valt. |

Deze onderwerpen kunnen u helpen bij het werken met deze functie:

Gerelateerde onderwerpen

| Onderwerp | Standaardvlag/-waarde | Beschrijving |
|---|---|---|
| *FirstWeekDay (page 244)* | 6 / zondag | Definieert de begindag van elke week. |

## Voorbeeld 1 – Geen aanvullende argumenten

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties voor de maand januari 2022 die wordt geladen in de tabel `Transactions`.
- De systeemvariabele `FirstWeekDay` die is ingesteld op `6` (zondag).
- Een voorgaande lading die het volgende bevat:
  - De `inweek()`-functie die is ingesteld als het `in_week`-veld dat bepaalt welke transacties hebben plaatsgevonden in de week tot en met 14 januari 2022.
  - De `weekday()`-functie, ingesteld als het veld `week_day`, toont welke dag van de week overeenkomt met iedere datum.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `week_day`
- `in_week`

Resultatentabel

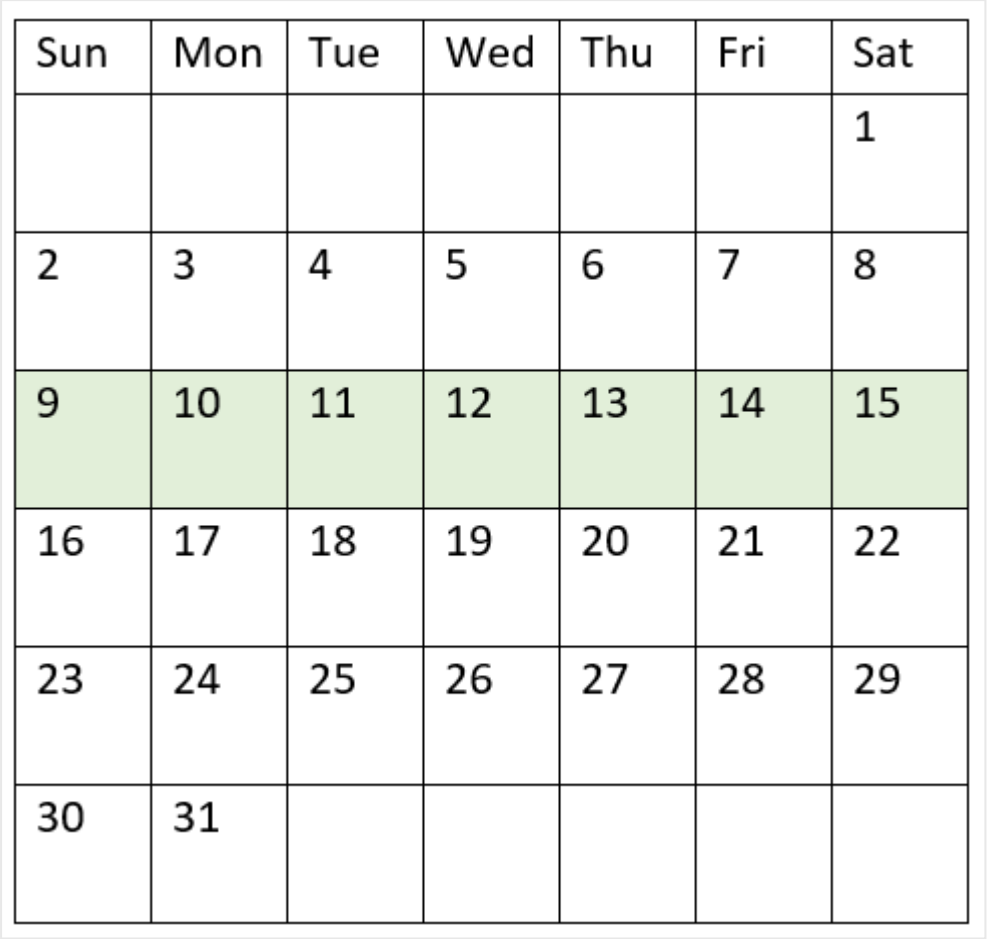
| date | week_day | in_week |
|---|---|---|
| 01/02/2022 | Zo | 0 |
| 01/05/2022 | Wo | 0 |
| 01/06/2022 | Do | 0 |
| 01/08/2022 | Za | 0 |
| 01/09/2022 | Zo | -1 |
| 01/10/2022 | Ma | -1 |
| 01/11/2022 | Di | -1 |
| 01/12/2022 | Wo | -1 |
| 01/13/2022 | Do | -1 |
| 01/14/2022 | Vr | -1 |
| 01/15/2022 | Za | -1 |
| 01/16/2022 | Zo | 0 |

| date | week_day | in_week |
|---|---|---|
| 01/17/2022 | Ma | 0 |
| 01/18/2022 | Di | 0 |
| 01/26/2022 | Wo | 0 |
| 01/27/2022 | Do | 0 |
| 01/28/2022 | Vr | 0 |
| 01/29/2022 | Za | 0 |
| 01/30/2022 | Zo | 0 |
| 01/31/2022 | Ma | 0 |

Het veld `in_week` wordt gemaakt in de vorige load-instructie met behulp van de functie `inweek()`. Het eerste argument identificeert welk veld wordt geëvalueerd. Het tweede argument is een vastgestelde datum, 14 januari, wat de `base_date` is. Het `base_date`-argument werkt met de `FirstWeekDay`-systeemvariabele om de week te bepalen waarmee moet worden vergeleken. Een `period_no` van `0` is het laatste argument, wat betekent dat de functie geen kwartalen voor of na het gesegmenteerde kwartaal vergelijkt.

De systeemvariabele `FirstWeekDay` bepaalt dat weken beginnen op een zondag en eindigen op een zaterdag. Daarom zou januari worden opgedeeld in weken volgens het onderstaande diagram, waarbij de datums tussen 9 en 15 januari de geldige periode vormen voor de `inweek()` berekening:

*Diagram van kalender met het bereik van de `inweek()`-functie gemarkeerd*



| 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 | | | | | |

Elke transactie die tussen 9 januari en 15 januari plaatsvindt, retourneert een booleaans resultaat voor `TRUE`.

## Voorbeeld 2 – period_no

Load-script en resultaten

**Overzicht**

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset met een set transacties voor 2022 wordt in de tabel `Transactions` geladen.
- De systeemvariabele `FirstWeekDay` die is ingesteld op `6` (zondag).
- Een voorgaande lading die het volgende bevat:
    - De `inweek ()`-functie die is ingesteld als het `prev_week`-veld dat bepaalt welke transacties hebben plaatsgevonden in de week voorafgaand aan de week waarin 14 januari 2022 valt.

- De `weekday()`-functie, ingesteld als het veld `week_day`, toont welke dag van de week overeenkomt met iedere datum.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

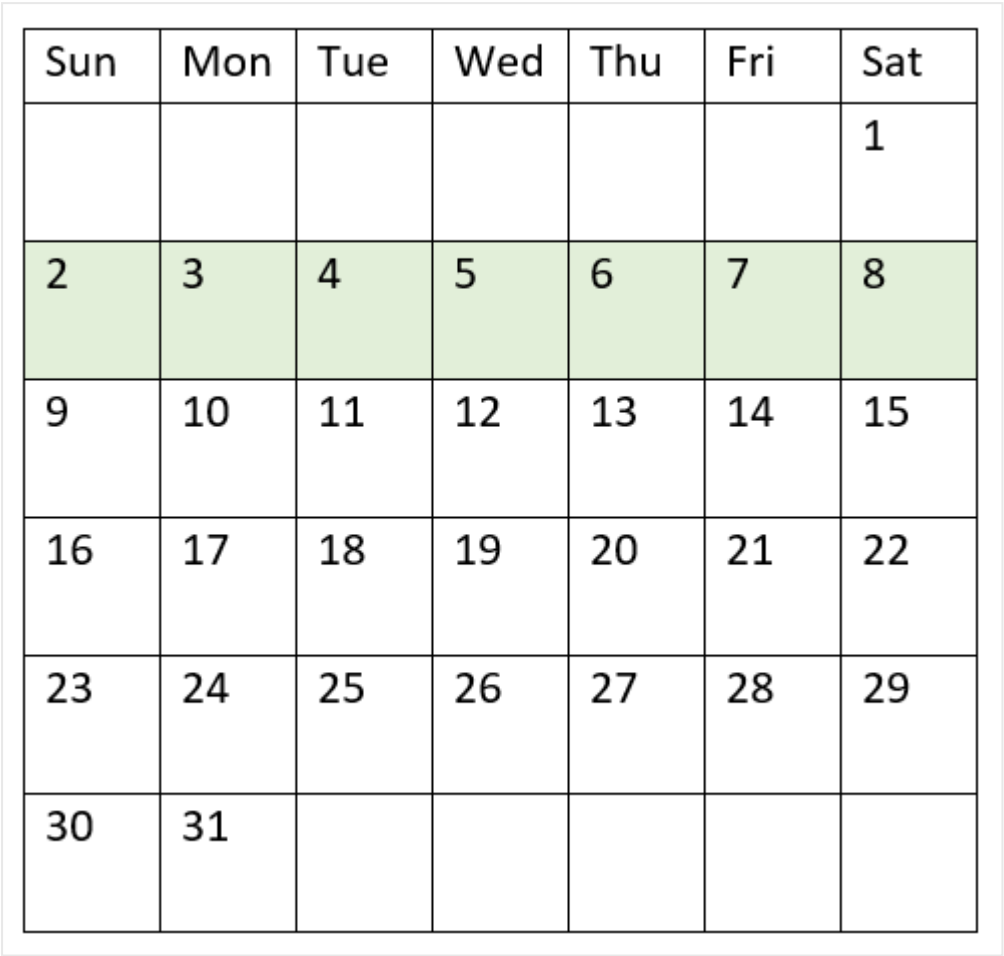
- `date`
- `week_day`
- `prev_week`

Resultatentabel

| date | week_day | prev_week |
|---|---|---|
| 01/02/2022 | Zo | -1 |
| 01/05/2022 | Wo | -1 |
| 01/06/2022 | Do | -1 |
| 01/08/2022 | Za | -1 |
| 01/09/2022 | Zo | 0 |
| 01/10/2022 | Ma | 0 |
| 01/11/2022 | Di | 0 |
| 01/12/2022 | Wo | 0 |
| 01/13/2022 | Do | 0 |
| 01/14/2022 | Vr | 0 |
| 01/15/2022 | Za | 0 |
| 01/16/2022 | Zo | 0 |
| 01/17/2022 | Ma | 0 |
| 01/18/2022 | Di | 0 |
| 01/26/2022 | Wo | 0 |
| 01/27/2022 | Do | 0 |
| 01/28/2022 | Vr | 0 |
| 01/29/2022 | Za | 0 |
| 01/30/2022 | Zo | 0 |
| 01/31/2022 | Ma | 0 |

Door `-1` als het `period_no`-argument in de `inweek()`-functie te gebruiken, verschuift de functie de grenzen van de vergelijkingsweek terug met zeven dagen. Met een `period_no` van `0`, zou de week tussen 9 en 15 januari vallen. Maar in dit voorbeeld verschuift de `period_no` van `-1` de begin- en eindgrens van dit segment terug met één week. De datumgrenzen worden 2 t/m 8 januari.

*Diagram van kalender met het bereik van de `inweek()`-functie gemarkeerd*



| 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 | | | | | |

Daarom zal elke transactie die plaatsvindt tussen 2 januari en 8 januari een booleaans resultaat voor TRUE retourneren.

## Voorbeeld 3 – first_week_day

### Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset met een set transacties voor 2022 wordt in de tabel `Transactions` geladen.
- De systeemvariabele `FirstWeekDay` die is ingesteld op `6` (zondag).
- Een voorgaande lading die het volgende bevat:
    - De `inweek()`-functie die is ingesteld als het `in_week`-veld dat bepaalt welke transacties hebben plaatsgevonden in de week tot en met 14 januari 2022.

- De `weekday()`-functie, ingesteld als het veld `week_day`, toont welke dag van de week overeenkomt met iedere datum.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `week_day`
- `in_week`

Resultatentabel

| date | week_day | in_week |
|---|---|---|
| 01/02/2022 | Zo | 0 |
| 01/05/2022 | Wo | 0 |
| 01/06/2022 | Do | 0 |
| 01/08/2022 | Za | 0 |
| 01/09/2022 | Zo | 0 |
| 01/10/2022 | Ma | -1 |
| 01/11/2022 | Di | -1 |
| 01/12/2022 | Wo | -1 |
| 01/13/2022 | Do | -1 |
| 01/14/2022 | Vr | -1 |
| 01/15/2022 | Za | -1 |
| 01/16/2022 | Zo | -1 |
| 01/17/2022 | Ma | 0 |
| 01/18/2022 | Di | 0 |
| 01/26/2022 | Wo | 0 |
| 01/27/2022 | Do | 0 |
| 01/28/2022 | Vr | 0 |
| 01/29/2022 | Za | 0 |
| 01/30/2022 | Zo | 0 |
| 01/31/2022 | Ma | 0 |

Door `0` te gebruiken als het `first_week_day`-argument in de `inweek()`-functie, wordt de `FirstWeekDay`-systeemvariabele vervangen en maandag ingesteld als de eerste dag van de week.

*Diagram van kalender met het bereik van de `inweek()`-functie gemarkeerd*

| 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 | | | | | |

Daarom zal elke transactie die plaatsvindt tussen 10 en 16 januari een booleaans resultaat voor TRUE retourneren.

## Voorbeeld 4 – Diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is de gegevensverzameling echter niet gewijzigd en wordt in de applicatie geladen. Maak een meting in de resultatentabel om te bepalen welke transacties plaatsvonden in de week van 14 januari 2022.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `date`

Maak de volgende metingen:

- `=inweek (date,'01/14/2022',0)` om te bepalen of transacties in de week van 14 januari plaatsvonden.
- `=weekday(date)` om aan te geven welke dag van de week overeenkomt met elke datum.

Resultatentabel

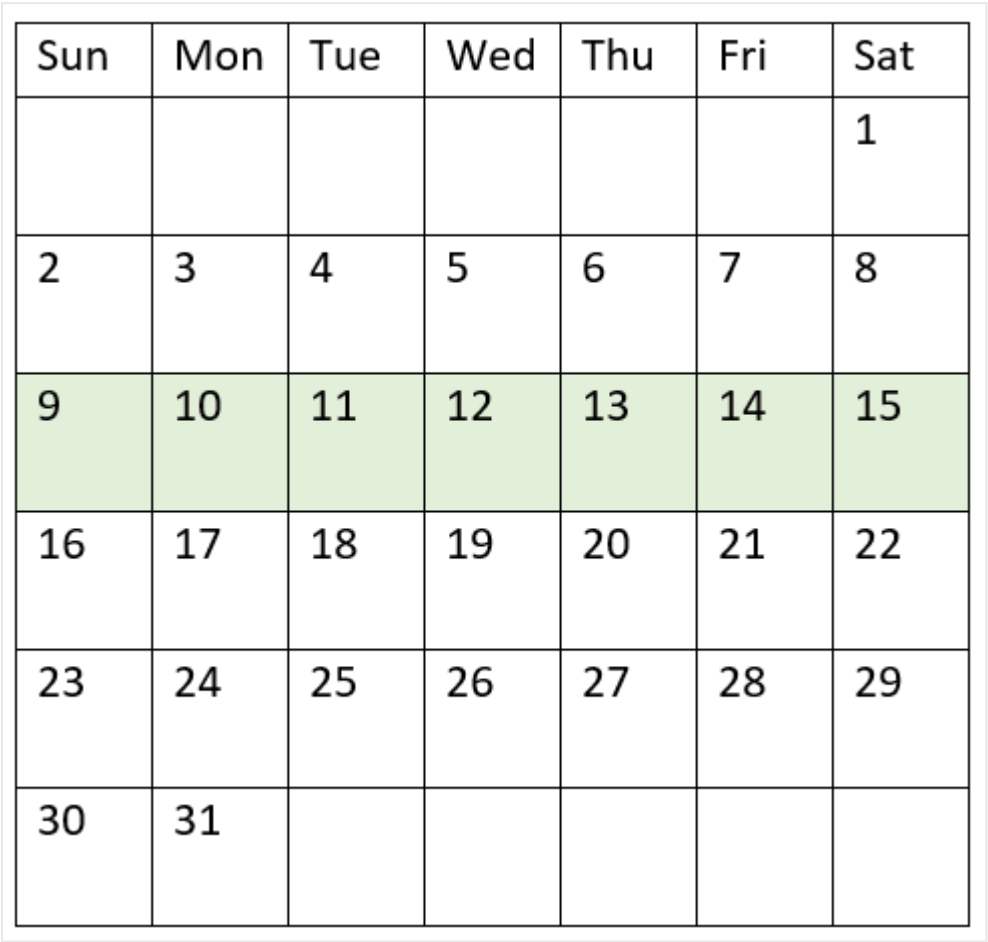
| date | week_day | =inweek (date,'01/14/2022',0) |
|---|---|---|
| 01/02/2022 | Zo | 0 |
| 01/05/2022 | Wo | 0 |
| 01/06/2022 | Do | 0 |
| 01/08/2022 | Za | 0 |
| 01/09/2022 | Zo | -1 |
| 01/10/2022 | Ma | -1 |

| date | week_day | =inweek (date,'01/14/2022',0) |
|---|---|---|
| 01/11/2022 | Di | -1 |
| 01/12/2022 | Wo | -1 |
| 01/13/2022 | Do | -1 |
| 01/14/2022 | Vr | -1 |
| 01/15/2022 | Za | -1 |
| 01/16/2022 | Zo | 0 |
| 01/17/2022 | Ma | 0 |
| 01/18/2022 | Di | 0 |
| 01/26/2022 | Wo | 0 |
| 01/27/2022 | Do | 0 |
| 01/28/2022 | Vr | 0 |
| 01/29/2022 | Za | 0 |
| 01/30/2022 | Zo | 0 |
| 01/31/2022 | Ma | 0 |

De meting `in_week` wordt in het diagram gemaakt met behulp van de `inweek()`-functie. Het eerste argument identificeert welk veld wordt geëvalueerd. Het tweede argument is een vastgestelde datum, 14 januari, wat de `base_date` is. Het `base_date`-argument werkt met de `FirstWeekDay`-systeemvariabele om de week te bepalen waarmee moet worden vergeleken. Een `period_no` van `0` is het laatste argument.

De systeemvariabele `FirstWeekDay` bepaalt dat weken beginnen op een zondag en eindigen op een zaterdag. Daarom zou januari worden opgedeeld in weken volgens het onderstaande diagram, waarbij de datums tussen 9 en 15 januari de geldige periode vormen voor de `inweek()` berekening:

*Diagram van kalender met het bereik van de `inweek()`-functie gemarkeerd*



| 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 | | | | | |

Elke transactie die tussen 9 januari en 15 januari plaatsvindt, retourneert een booleaans resultaat voor TRUE.

## Voorbeeld 5 – Scenario

Load-script en resultaten

**Overzicht**

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset die in de tabel `Products` wordt geladen.
- De tabel bevat de volgende velden:
  - product-id
  - producttype
  - productiedatum
  - kostprijs

Er is vastgesteld dat producten die in de week van 12 januari zijn geproduceerd defect zijn vanwege een materieelfout. De eindgebruiker wil graag een diagram dat op week de status toont welke geproduceerde producten de status 'defect' of 'foutloos' hadden en de kosten van de producten die in die week zijn geproduceerd.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `=weekname(manufacture_date)`

Maak de volgende metingen:

- `=if(only(inweek(manufacture_date,makedate(2022,01,12),0)),'Defective','Faultless')`, om te identificeren welke van de producten defect zijn en welke foutloos zijn met behulp van de `inweek()`-functie.
- `=sum(cost_price)`, om de som van de kosten van elk product te tonen.

**Doe het volgende:**

1. Stel de **Getalnotatie** van de meting in op **Geld**.
2. Onder **Uiterlijk** schakelt u **Totaal** uit.

Resultatentabel

| weekname (manufacture_date) | =if(only(inweek(manufacture_date,makedate (2022,01,12),0)), 'Defect','Foutloos') | Sum(cost_ price) |
|---|---|---|
| 2022/02 | Foutloos | 200.09 |
| 2022/03 | Defect | 441.51 |
| 2022/04 | Foutloos | 178.41 |
| 2022/05 | Foutloos | 231.67 |
| 2022/06 | Foutloos | 163.91 |

De functie `inweek()` retourneert een booleaanse waarde bij het evalueren van de productiedatums van elk product. Voor elk product dat op 12 januari is geproduceerd, retourneert de `inweek()`-functie een booleaanse waarde voor TRUE en markeert de producten als 'Defect'. Voor elk product dat de waarde voor FALSE retourneert, en dus niet in die week is geproduceerd, wordt het product als 'Foutloos' gemarkeerd.

## inweektodate

Deze functie retourneert True als **timestamp** binnen het gedeelte van de week ligt dat **base_ date** bevat tot en met de laatste milliseconde van **base_date**.

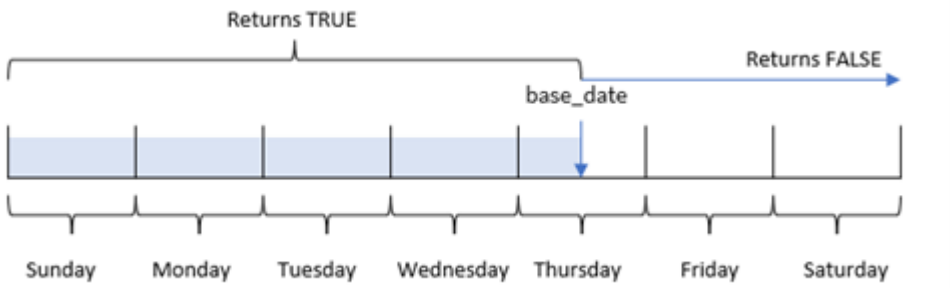
**Syntaxis:**

```
InWeekToDate (timestamp, base_date, period_no [, first_week_day])
```

**Retourgegevenstypen:** Booleaanse waarde

> *In Qlik Sense wordt de booleaanse waarde Waar vertegenwoordigd door -1 en de waarde Onwaar door 0.*

*Diagram van functie inweektodate*




De functie `inweektodate()` gebruikt de parameter `base_date` om een maximale grensdatum van een weeksegment te identificeren, evenals de bijbehorende datum voor het begin van de week, die is gebaseerd op de systeemvariabele `FirstWeekDay` (of door de gebruiker gedefinieerde parameter `first_week_day`). Zodra dit weeksegment is gedefinieerd, retourneert de functie booleaanse resultaten bij het vergelijken van de voorgeschreven datumwaarden met dat segment.

## Wanneer gebruiken

De functie `inweektodate()` retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als voorwaarde in een `if`-uitdrukking. Dit retourneert een aggregatie of berekening afhankelijk van of een geëvalueerde datum plaatsvond in de week tot en met een bepaalde datum.

De functie `inweektodate()` kan bijvoorbeeld worden gebruikt om alle verkopen te berekenen die in een bepaalde week tot een bepaalde datum zijn gedaan.

Argumenten

| Argument | Beschrijving |
|---|---|
| **timestamp** | De datum die u wilt vergelijken met **base_date**. |
| **base_date** | Datum op basis waarvan de week wordt geëvalueerd. |
| **period_no** | De week kan worden verschoven met **period_no**. **period_no** is een geheel getal, waarbij de waarde 0 de week aangeeft die **base_date** bevat. Negatieve waarden in **period_no** geven voorgaande weken aan en positieve waarden geven volgende weken aan. |
| **first_week_day** | Standaard is de eerste dag van de week zondag (zoals bepaald door de systeemvariabele FirstWeekDay), beginnend om middernacht tussen zaterdag en zondag. De parameter **first_week_day** vervangt de variabele **FirstWeekDay**. Geef aan dat de week op een andere dag begint door een vlag tussen 0 en 6 op te geven.<br><br>Gebruik voor een week die begint op maandag en eindigt op zondag een vlag van 0 voor maandag, 1 voor dinsdag, 2 voor woensdag, 3 voor donderdag, 4 voor vrijdag, 5 voor zaterdag en 6 voor zondag. |

Voorbeelden van functies

| Voorbeeld | Interactie |
|---|---|
| `inweektodate ('01/12/2006', '01/12/2006', 0)` | Retourneert `TRUE`. |
| `inweektodate ('01/12/2006', '01/11/2006', 0)` | Retourneert `FALSE`. |
| `inweektodate ('01/12/2006', '01/18/2006', -1)` | Retourneert `FALSE`.<br>omdat `period_no` is opgegeven als -1, vormt 01/11/2006 de daadwerkelijke datum waartegen `timestamp` wordt afgezet. |
| `inweektodate ('01/11/2006', '01/12/2006', 0, 3 )` | Retourneert `FALSE`, omdat `first_week_day` is opgegeven als `3` (donderdag), waardoor 01/12/2006 de eerste dag van de week wordt na de week die 01/12/2006 bevat. |

Deze onderwerpen kunnen u helpen bij het werken met deze functie:

Gerelateerde onderwerpen

| Onderwerp | Standaardvlag/-waarde | Beschrijving |
|---|---|---|
| *FirstWeekDay (page 244)* | 6 / zondag | Definieert de begindag van elke week. |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – geen aanvullende argumenten

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties voor de maand januari 2022 die wordt geladen in de tabel `Transactions`.
- Het gegevensveld in de `TimestampFormat='M/D/YYYY h:mm:ss[.fff]'`-indeling.
- Het aanmaken van een veld, `in_week_to_date`, dat bepaalt welke transacties hebben plaatsgevonden in de week tot en met 14 januari 2022.
- Het maken van een extra veld met de naam `weekday` met behulp van de functie `weekday()`. Dit nieuwe veld is gemaakt om aan te geven welke dag van de week overeenkomt met elke datum.

### Load-script

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff]';
SET FirstWeekDay=6;
Transactions:
    Load
        *,
        weekday(date) as week_day,
        inweektodate(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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- week_day
- in_week_to_date

Resultatentabel

| date | week_day | in_week_to_date |
|---|---|---|
| 2022-01-02 12:22:06 | Zo | 0 |
| 2022-01-05 01:02:30 | Wo | 0 |
| 2022-01-06 15:36:20 | Do | 0 |
| 2022-01-08 10:58:35 | Za | 0 |
| 2022-01-09 08:53:32 | Zo | -1 |
| 2022-01-10 21:13:01 | Ma | -1 |
| 2022-01-11 00:57:13 | Di | -1 |
| 2022-01-12 09:26:02 | Wo | -1 |
| 2022-01-13 15:05:09 | Do | -1 |
| 2022-01-14 18:44:57 | Vr | -1 |
| 2022-01-15 06:10:46 | Za | 0 |
| 2022-01-16 06:39:27 | Zo | 0 |
| 2022-01-17 10:44:16 | Ma | 0 |
| 2022-01-18 18:48:17 | Di | 0 |
| 2022-01-26 04:36:03 | Wo | 0 |
| 2022-01-27 08:07:49 | Do | 0 |
| 2022-01-28 12:24:29 | Vr | 0 |
| 2022-01-30 11:56:56 | Zo | 0 |
| 2022-01-30 14:40:19 | Zo | 0 |
| 2022-01-31 05:28:21 | Ma | 0 |

Het veld `in_week_to_date` wordt gemaakt in de vorige load-instructie met behulp van de functie `inweektodate()`. Het eerste verstrekte argument geeft aan welk veld wordt geëvalueerd. Het tweede argument is een vastgestelde datum voor 14 januari, de `base_date` die aangeeft welke week moet worden gesegmenteerd en de eindgrens van dat segment definieert. Een `period_no` van 0 is het laatste argument, wat betekent dat de functie geen kwartalen voor of na het gesegmenteerde kwartaal vergelijkt.

De systeemvariabele `FirstWeekDay` bepaalt dat weken beginnen op een zondag en eindigen op een zaterdag. Daarom zou januari worden opgedeeld in weken volgens het onderstaande diagram, waarbij de datums tussen 9 en 14 januari de geldige periode vormen voor de `inweekdodate()` berekening:

*Kalenderdiagram met transactiedatums die een Booleaans resultaat van WAAR zouden opleveren*

| 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 | | | | | |

Elke transactie die tussen 9 januari en 14 januari plaatsvindt, retourneert een booleaans resultaat van TRUE. Transacties voor en na de datums retourneren een booleaans resultaat van FALSE.

## Voorbeeld 2 – period_no

### Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het aanmaken van een veld, `prev_week_to_date`, dat bepaalt welke transacties een volledige week voor het einde van het weeksegment op 14 januari 2022 hebben plaatsgevonden.
- Het maken van een extra veld met de naam `weekday` met behulp van de functie `weekday()`. Dit is om aan te geven welke dag van de week overeenkomt met elke datum.

#### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `week_day`
- `prev_week_to_date`

Resultatentabel

| date | week_day | prev_week_to_date |
|---|---|---|
| 2022-01-02 12:22:06 | Zo | -1 |
| 2022-01-05 01:02:30 | Wo | -1 |
| 2022-01-06 15:36:20 | Do | -1 |
| 2022-01-08 10:58:35 | Za | 0 |
| 2022-01-09 08:53:32 | Zo | 0 |
| 2022-01-10 21:13:01 | Ma | 0 |
| 2022-01-11 00:57:13 | Di | 0 |
| 2022-01-12 09:26:02 | Wo | 0 |
| 2022-01-13 15:05:09 | Do | 0 |
| 2022-01-14 18:44:57 | Vr | 0 |
| 2022-01-15 06:10:46 | Za | 0 |

| date | week_day | prev_week_to_date |
|---|---|---|
| 2022-01-16 06:39:27 | Zo | 0 |
| 2022-01-17 10:44:16 | Ma | 0 |
| 2022-01-18 18:48:17 | Di | 0 |
| 2022-01-26 04:36:03 | Wo | 0 |
| 2022-01-27 08:07:49 | Do | 0 |
| 2022-01-28 12:24:29 | Vr | 0 |
| 2022-01-30 11:56:56 | Zo | 0 |
| 2022-01-30 14:40:19 | Zo | 0 |
| 2022-01-31 05:28:21 | Ma | 0 |

Een `period_no`-waarde van -1 geeft aan dat de functie `inweektodate ()` het invoerkwartaalsegment vergelijkt met de voorgaande week. Het weeksegment komt aanvankelijk overeen met tussen 9 januari en 14 januari. De `period_no` verschuift vervolgens zowel de begin- als de eindgrens van dit segment naar een week eerder, waardoor de datumgrenzen van 2 januari tot 7 januari worden.

*Kalenderdiagram met transactiedatums die een Booleaans resultaat van WAAR zouden opleveren*

| 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 | | | | | |

Daarom zal elke transactie die plaatsvindt tussen 2 januari en 8 januari (niet inclusief 8 januari zelf) een booleaans resultaat van `TRUE` retourneren.

## Voorbeeld 3 – first_week_day

### Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het aanmaken van een veld, `in_week_to_date`, dat bepaalt welke transacties hebben plaatsgevonden in de week tot en met 14 januari 2022.
- Het maken van een extra veld met de naam `weekday` met behulp van de functie `weekday()`. Dit is om aan te geven welke dag van de week overeenkomt met elke datum.

In dit voorbeeld beschouwen we maandag als de eerste dag van de week.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `week_day`

- `in_week_to_date`

Resultatentabel

| date | week_day | in_week_to_date |
|---|---|---|
| 2022-01-02 12:22:06 | Zo | 0 |
| 2022-01-05 01:02:30 | Wo | 0 |
| 2022-01-06 15:36:20 | Do | 0 |
| 2022-01-08 10:58:35 | Za | 0 |
| 2022-01-09 08:53:32 | Zo | 0 |
| 2022-01-10 21:13:01 | Ma | -1 |
| 2022-01-11 00:57:13 | Di | -1 |
| 2022-01-12 09:26:02 | Wo | -1 |
| 2022-01-13 15:05:09 | Do | -1 |
| 2022-01-14 18:44:57 | Vr | -1 |
| 2022-01-15 06:10:46 | Za | 0 |
| 2022-01-16 06:39:27 | Zo | 0 |
| 2022-01-17 10:44:16 | Ma | 0 |
| 2022-01-18 18:48:17 | Di | 0 |
| 2022-01-26 04:36:03 | Wo | 0 |
| 2022-01-27 08:07:49 | Do | 0 |
| 2022-01-28 12:24:29 | Vr | 0 |
| 2022-01-30 11:56:56 | Zo | 0 |
| 2022-01-30 14:40:19 | Zo | 0 |
| 2022-01-31 05:28:21 | Ma | 0 |

Door 0 te gebruiken als het `first_week_day`-argument in de `inweektodate()`-functie, vervangt het functie-argument de `FirstWeekDay`-systeemvariabele en stelt het maandag in als de eerste dag van de week.

*Kalenderdiagram met transactiedatums die een Booleaans resultaat van WAAR zouden opleveren*

| 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 | 17 |
| 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 31 | | | | | | |

Daarom zal elke transactie die plaatsvindt tussen 10 en 14 januari een booleaans resultaat van TRUE retourneren, terwijl transacties met datums buiten deze grenzen een waarde van FALSE zullen retourneren.

## Voorbeeld 4 – diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld. In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die bepaalt welke transacties plaatsvonden in de week tot en met 14 januari 2022 is gemaakt als een meting in het diagramobject.

### Load-script

```
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
];
```

**Resultaten**

**Doe het volgende:**

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `date`.
2. Om te berekenen of transacties in dezelfde week tot 14 januari hebben plaatsgevonden, maakt u de volgende meting aan:
   `=inweektodate(date,'01/14/2022',0)`
3. Om aan te geven welke dag van de week overeenkomt met elke datum, maakt u een aanvullende meting:
   `=weekday(date)`

Resultatentabel

| date | week_day | in_week_to_date |
|---|---|---|
| 2022-01-02 12:22:06 | Zo | 0 |
| 2022-01-05 01:02:30 | Wo | 0 |
| 2022-01-06 15:36:20 | Do | 0 |
| 2022-01-08 10:58:35 | Za | 0 |
| 2022-01-09 08:53:32 | Zo | -1 |
| 2022-01-10 21:13:01 | Ma | -1 |
| 2022-01-11 00:57:13 | Di | -1 |
| 2022-01-12 09:26:02 | Wo | -1 |
| 2022-01-13 15:05:09 | Do | -1 |
| 2022-01-14 18:44:57 | Vr | -1 |
| 2022-01-15 06:10:46 | Za | 0 |
| 2022-01-16 06:39:27 | Zo | 0 |

| date | week_day | in_week_to_date |
|---|---|---|
| 2022-01-17 10:44:16 | Ma | 0 |
| 2022-01-18 18:48:17 | Di | 0 |
| 2022-01-26 04:36:03 | Wo | 0 |
| 2022-01-27 08:07:49 | Do | 0 |
| 2022-01-28 12:24:29 | Vr | 0 |
| 2022-01-30 11:56:56 | Zo | 0 |
| 2022-01-30 14:40:19 | Zo | 0 |
| 2022-01-31 05:28:21 | Ma | 0 |

Het veld `in_week_to_date` wordt in een diagramobject gemaakt als een meting met behulp van de functie `inweektodate()`. Het eerste verstrekte argument geeft aan welk veld wordt geëvalueerd. Het tweede argument is een vastgestelde datum voor 14 januari, de `base_date` die aangeeft welke week moet worden gesegmenteerd en de eindgrens van dat segment definieert. Een `period_no` van 0 is het laatste argument, wat betekent dat de functie geen kwartalen voor of na het gesegmenteerde kwartaal vergelijkt.

De systeemvariabele `FirstWeekDay` bepaalt dat weken beginnen op een zondag en eindigen op een zaterdag. Daarom zou januari worden opgedeeld in weken volgens het onderstaande diagram, waarbij de datums tussen 9 en 14 januari de geldige periode vormen voor de `inweekdodate()` berekening:

*Kalenderdiagram met transactiedatums die een Booleaans resultaat van WAAR zouden opleveren*

| 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 | | | | | |

Elke transactie die tussen 9 januari en 14 januari plaatsvindt, retourneert een booleaans resultaat van `TRUE`. Transacties voor en na de datums retourneren een booleaans resultaat van `FALSE`.

## Voorbeeld 5 – Scenario

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset die wordt geladen in een tabel met de naam `Products`.
- Informatie over product-id, productiedatum en kostprijs.

Er is vastgesteld dat producten die in de week van 12 januari zijn geproduceerd defect zijn vanwege een materieelfout. Op 13 januari is het probleem opgelost. De eindgebruiker wil graag een diagramobject dat op week de status toont of de geproduceerde producten 'defect' of 'foutloos' zijn en de kosten van de producten die in die week zijn geproduceerd.

### Load-script

```
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
];
```

**Resultaten**

**Doe het volgende:**

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel. Maak een dimensie om de weeknamen weer te geven:
   `=weekname(manufacture_date)`
2. Maak vervolgens een dimensie om te identificeren welke van de producten defect zijn en welke foutloos zijn:
   `=if(inWeektodate(manufacture_date,makedate(2022,01,12),0),'Defective','Faultless')`
3. Maak een meting om de `cost_price` van de producten op te tellen:
   `=sum(cost_price)`
4. Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

| weekname(manufacture_date) | if(inweektodate(manufacture_date,makedate (2022,01,12),0),'Defective','Faultless') | Sum(cost_price) |
|---|---|---|
| 2022/02 | Foutloos | $200.09 |
| 2022/03 | Defect | $263.46 |
| 2022/03 | Foutloos | $178.05 |
| 2022/04 | Foutloos | $178.41 |
| 2022/05 | Foutloos | $147.46 |
| 2022/06 | Foutloos | $248.12 |

De functie `inweektodate()` retourneert een booleaanse waarde bij het evalueren van de productiedatums van elk product. Voor degenen die een booleaanse waarde van `TRUE` retourneren, markeert het de producten als `'Defective'`. Voor elk product dat een waarde retourneert van `FALSE`, en dus niet gemaakt in de week tot en met 12 januari, markeert het de producten als `'Faultless'`.

## inyear

Deze functie retourneert True als **timestamp** binnen het jaar valt dat **base_date** bevat.

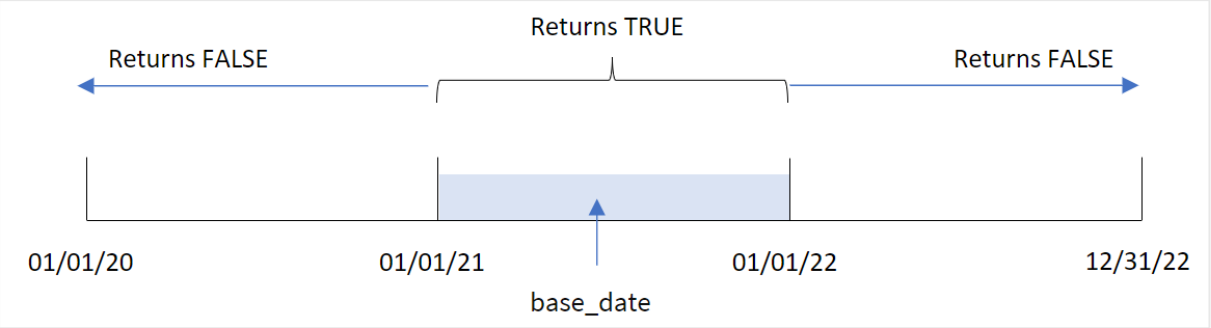
**Syntaxis:**

```
InYear (timestamp, base_date, period_no [, first_month_of_year])
```

**Retourgegevenstypen:** Booleaanse waarde

In Qlik Sense wordt de booleaanse waarde Waar vertegenwoordigd door -1 en de waarde Onwaar door 0.

*Diagram van het bereik van de `inyear()`-functie*




De `inyear()`-functie retourneert een booleaans resultaat wanneer de geselecteerde datumwaarde worden vergeleken met een jaar dat is gedefinieerd door de `base_date`.

### Wanneer gebruiken

De functie `inyear()` retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als voorwaarde in een `if expression`. Hierdoor wordt een aggregatie of berekening geretourneerd, afhankelijk van het feit of een geëvalueerde datum in het desbetreffende jaar plaatsvond. De `inyear()`-functie kan bijvoorbeeld worden gebruikt om alle verkoop te identificeren die in een bepaald jaar heeft plaatsgevonden.

Argumenten

| Argument | Beschrijving |
|---|---|
| **timestamp** | De datum die u wilt vergelijken met **base_date**. |
| **base_date** | Datum op basis waarvan het jaar wordt geëvalueerd. |
| **period_no** | Het jaar kan worden verschoven met **period_no**. **period_no** is een geheel getal, waarbij de waarde 0 het jaar aangeeft dat **base_date** bevat. Negatieve waarden in **period_no** geven voorafgaande jaren aan en positieve waarden geven volgende jaren aan. |
| **first_month_of_year** | Als u wilt werken met (boek)jaren die niet in januari beginnen, geeft u een waarde op tussen 2 en 12 in **first_month_of_year**. |

U kunt de volgende waarden gebruiken om de eerste maand van het jaar in te stellen in het argument `first_month_of_year`:

waarden first_month_of_year

| Maand | Waarde |
|---|---|
| Februari | 2 |
| Maart | 3 |
| April | 4 |
| Mei | 5 |

| Maand | Waarde |
|---|---|
| Juni | 6 |
| Juli | 7 |
| Augustus | 8 |
| September | 9 |
| Oktober | 10 |
| November | 11 |
| December | 12 |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `inyear ('01/25/2013', '01/01/2013', 0 )` | Retourneert TRUE |
| `inyear ('01/25/2012', '01/01/2013', 0)` | Retourneert FALSE |
| `inyear ('01/25/2013', '01/01/2013', -1)` | Retourneert FALSE |
| `inyear ('01/25/2012', '01/01/2013', -1 )` | Retourneert TRUE |
| `inyear ('01/25/2013', '01/01/2013', 0, 3)` | Retourneert TRUE<br><br>De waarde van de `base_date` en de `first_month_of_year` specificeren dat een tijdstempel tussen 01/03/2012 en 02/28/2013 moet vallen |
| `inyear ('03/25/2013', '07/01/2013', 0, 3 )` | Retourneert TRUE |

## Voorbeeld 1 – Basisvoorbeeld

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties tussen 2020 en 2022 die wordt geladen in de tabel `Transactions`.
- Een voorafgaande lading met de `inyear()`-functie die is ingesteld als het `in_year`-veld en die bepaalt welke transacties in hetzelfde jaar hebben plaatsgevonden als het jaar waarin 26 juli 2021 valt.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

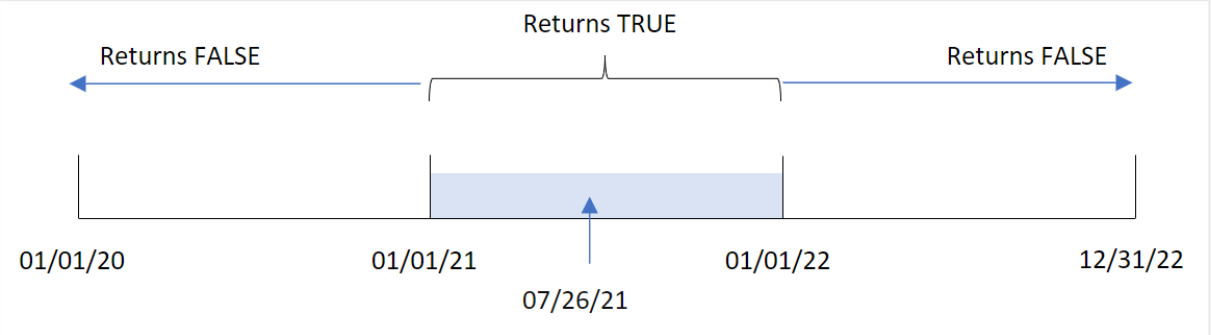
- `date`
- `in_year`

Resultatentabel

| 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 |

Het veld `in_year` wordt gemaakt in de vorige load-instructie met behulp van de functie `inyear()`. Het eerste argument identificeert welk veld wordt geëvalueerd. Het tweede argument is een vastgestelde datum voor 26 juli 2021. Dit is de `base_date` die het jaar bepaalt waarmee de vergelijking moet worden gedaan. Een `period_no` van `0` is het laatste argument, wat betekent dat de `inyear()`-functie geen jaren voor of na het jaar vergelijkt.

*Diagram van het bereik van de `inyear()`-functie met 26 juli als de basisdatum*




Een transactie die in 2021 plaatsvindt, retourneert een booleaans resultaat voor TRUE.

## Voorbeeld 2 – period_no

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met een set transacties tussen 2020 en 2022 die wordt geladen in de tabel `Transactions`.
- Een voorafgaande lading met de `inyear()`-functie die is ingesteld als het `previous_year`-veld en die bepaalt welke transacties in hetzelfde jaar hebben plaatsgevonden als het jaar waarin 26 juli 2021 valt.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `previous_year`

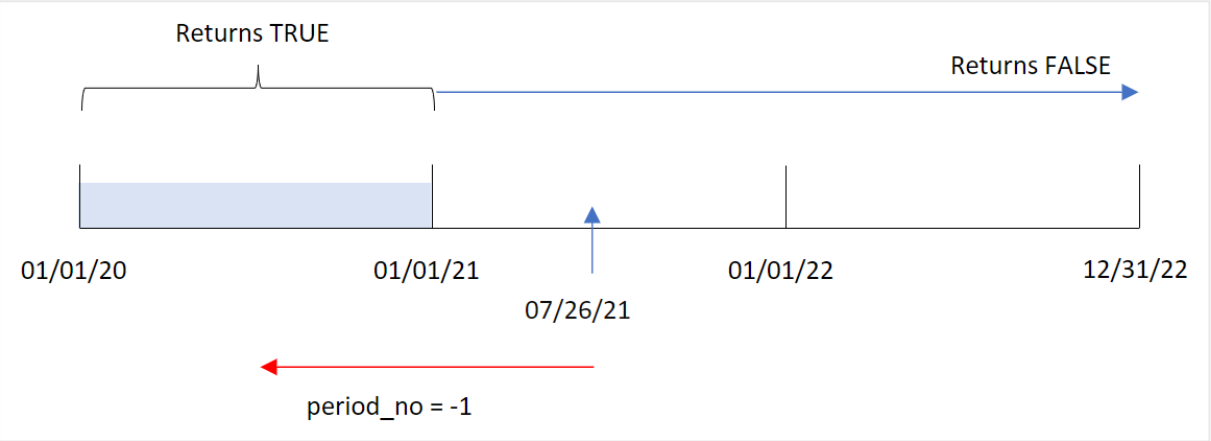
Resultatentabel

| 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 |

| date | previous_year |
|---|---|
| 07/18/2022 | 0 |
| 11/14/2022 | 0 |
| 12/12/2022 | 0 |

Met `-1` als het `period_no`-argument in de `inyear()`-functie verschuiven de grenzen van het jaar waarmee de vergelijking moet worden gedaan met een heel jaar. 2021 wordt aanvankelijk geïdentificeerd als het jaar voor de vergelijking. De `period_no` verschuift het jaar waarmee moet worden vergeleken met één jaar waardoor 2020 het jaar is voor de vergelijking.

*Diagram van het bereik van de `inyear()`-functie met het argument voor de period_no ingesteld op `-1`*




Daarom zal een transactie die in 2020 plaatsvindt, een booleaans resultaat voor TRUE retourneren.

## Voorbeeld 3 – first_month_of_year

### Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met een set transacties tussen 2020 en 2022 die wordt geladen in de tabel `Transactions`.
- Een voorafgaande lading met de `inyear()`-functie die is ingesteld als het `in_year`-veld en die bepaalt welke transacties in hetzelfde jaar hebben plaatsgevonden als het jaar waarin 26 juli 2021 valt.

In dit voorbeeld is maart volgens het organisatiebeleid echter de eerste maand van het boekjaar.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `in_year`

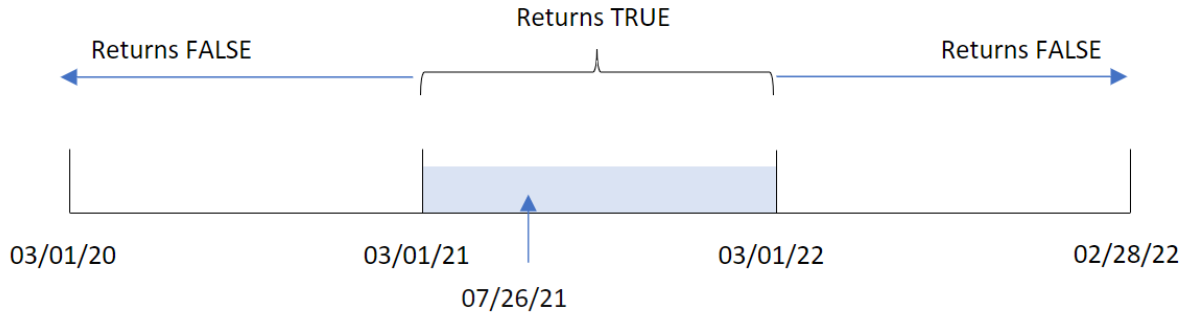
Resultatentabel

| date | in_year |
|---|---|
| 01/13/2020 | 0 |
| 02/26/2020 | 0 |
| 03/27/2020 | 0 |
| 04/16/2020 | 0 |

| date | in_year |
|---|---|
| 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 | -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 |

Met `3` als het `first_month_of_year`-argument in de `inyear()`-functie begint het jaar op 1 maart en eindigt het jaar aan het eind van februari.

*Diagram van het tijdsbereik van de `inyear()`-functie met maart ingesteld als de eerste maand van het jaar*




Daarom zal elke transactie die plaatsvindt tussen 1 maart 2021 en 1 maart 2022 een booleaans resultaat voor TRUE retourneren.

## Voorbeeld 4 – Diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is de gegevensverzameling echter niet gewijzigd en wordt in de applicatie geladen. De berekening die bepaalt of transacties al dan niet hebben plaatsgevonden in hetzelfde jaar als 26 juli 2021, wordt als meting in een diagramobject van de applicatie gemaakt.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `date`

Om te berekenen of transacties in hetzelfde jaar hebben plaatsgevonden als het jaar waarin 26 juli 2021 valt, maakt u de volgende meting:

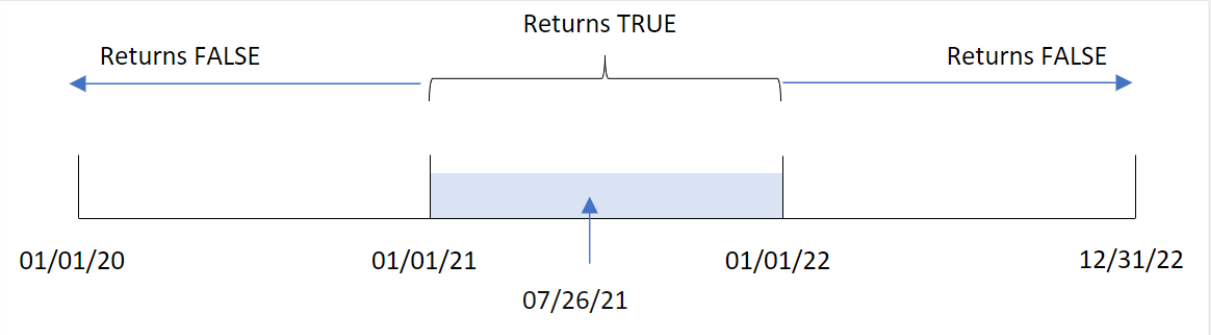
- `=inyear(date,'07/26/2021', 0)`

Resultatentabel

| date | =inyear(date,'07/26/2021',0) |
|---|---|
| 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 |

Het veld `in_year` wordt in een diagram gemaakt met behulp van de functie `inyear()`. Het eerste argument identificeert welk veld wordt geëvalueerd. Het tweede argument is een vastgestelde datum voor 26 juli 2021. Dit is de `base_date` die het jaar bepaalt waarmee de vergelijking moet worden gedaan. Een `period_no` van `0` is het laatste argument, wat betekent dat de `inyear()`-functie geen jaren voor of na het jaar vergelijkt.

*Diagram van het bereik van de `inyear()`-functie met 27 juli als de basisdatum*




Een transactie die in 2021 plaatsvindt, retourneert een booleaans resultaat voor TRUE.

## Voorbeeld 5 – Scenario

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset die in de tabel `Products` wordt geladen.
- De tabel bevat de volgende velden:
  - product-id
  - producttype
  - productiedatum
  - kostprijs

De eindgebruiker wil een diagramobject dat, per producttype, de kosten weergeeft van de producten die in 2021 zijn geproduceerd.

### Load-script

```
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
];
```

**Resultaten**

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `product_type`

Om de som van elk product dat in 2021 is geproduceerd te berekenen, maakt u de volgende meting:

- `=sum(if(InYear(manufacture_date,makedate(2021,01,01),0),cost_price,0))`

**Doe het volgende:**

1. Stel de **Getalnotatie** van de meting in op **Geld**.
2. Onder **Uiterlijk** schakelt u **Totaal** uit.

Resultatentabel

| producttype | =sum(if(InYear(manufacture_date,makedate(2021,01,01),0),cost_price,0)) |
|---|---|
| product A | $95.93 |
| product B | $128.66 |
| product C | $61.89 |
| product D | $171.21 |

De functie `inyear()` retourneert een booleaanse waarde bij het evalueren van de productiedatums van elk product. Voor elk product dat in 2021 is geproduceerd, retourneert de `inyear()`-functie een booleaanse waarde voor TRUE en wordt de som van de `cost_price`weergegeven.

## inyeartodate

Deze functie retourneert True als **timestamp** binnen het gedeelte van het jaar ligt dat **base_date** bevat tot en met de laatste milliseconde van **base_date**.

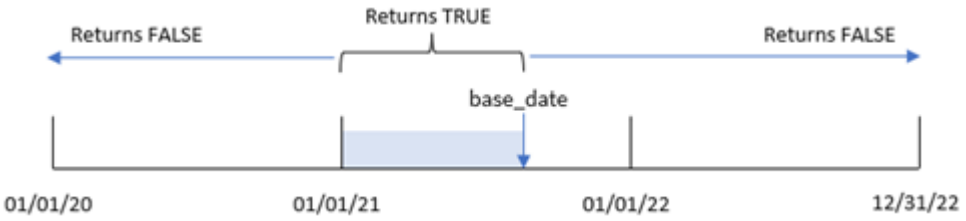
**Syntaxis:**

```
InYearToDate (timestamp, base_date, period_no[, first_month_of_year])
```

**Retourgegevenstypen:** Booleaanse waarde

> *In Qlik Sense wordt de booleaanse waarde Waar vertegenwoordigd door -1 en de waarde Onwaar door 0.*

*Diagram van functie inyeartodate*




De functie `inyeartodate()` segmenteert een bepaald gedeelte van het jaar met de `base_date`, waarbij de maximaal toegestane datum voor dat jaarsegment wordt aangegeven. De functie evalueert vervolgens of een datumveld of waarde in dit segment valt en retourneert een booleaans resultaat.

Argumenten

| Argument | Beschrijving |
|---|---|
| **timestamp** | De datum die u wilt vergelijken met **base_date**. |
| **base_date** | Datum op basis waarvan het jaar wordt geëvalueerd. |
| **period_no** | Het jaar kan worden verschoven met **period_no**. **period_no** is een geheel getal, waarbij de waarde 0 het jaar aangeeft dat **base_date** bevat. Negatieve waarden in **period_no** geven voorafgaande jaren aan en positieve waarden geven volgende jaren aan. |
| **first_month_of_year** | Als u wilt werken met (boek)jaren die niet in januari beginnen, geeft u een waarde op tussen 2 en 12 in **first_month_of_year**. |

## Wanneer gebruiken

De functie `inyeartodate()` retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als voorwaarde in een `if`-uitdrukking. Dit retourneert een aggregatie of berekening afhankelijk van of een geëvalueerde datum plaatsvond in het jaar tot en met de betreffende datum.

De functie `inyeartodate()` bijvoorbeeld kan worden gebruikt om alle materieel te identificeren die in een jaar tot en met een specifieke datum is geproduceerd.

Deze voorbeelden gebruiken de datumopmaak MM/DD/YYYY. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` boven aan uw load-script voor gegevens. Wijzig de opmaak in de voorbeelden zodat deze in uw behoeften voorziet.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `inyeartodate ('01/25/2013', '02/01/2013', 0)` | Retourneert `TRUE`. |
| `inyeartodate ('01/25/2012', '01/01/2013', 0)` | Retourneert `FALSE`. |
| `inyeartodate ('01/25/2012', '02/01/2013', -1)` | Retourneert `TRUE`. |
| `inyeartodate ('11/25/2012', '01/31/2013', 0, 4)` | Retourneert `TRUE`.<br>De waarde van `timestamp` valt binnen het fiscale jaar dat begint in de vierde maand en vóór de waarde van `base_date`. |
| `inyeartodate ('3/31/2013', '01/31/2013', 0, 4 )` | Retourneert `FALSE`.<br>Retourneert Vergeleken met het voorafgaande voorbeeld ligt de waarde van nog steeds binnen het fiscale jaar, maar valt deze na de waarde van , dus valt deze buiten het gedeelte van het jaar.timestampbase_date |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – geen aanvullende argumenten

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties tussen 2020 en 2022 die wordt geladen in de tabel `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.
- Het aanmaken van een veld, `in_year_to_date`, dat bepaalt welke transacties hebben plaatsgevonden in het jaar tot en met 26 juli 2021.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `in_year_to_date`

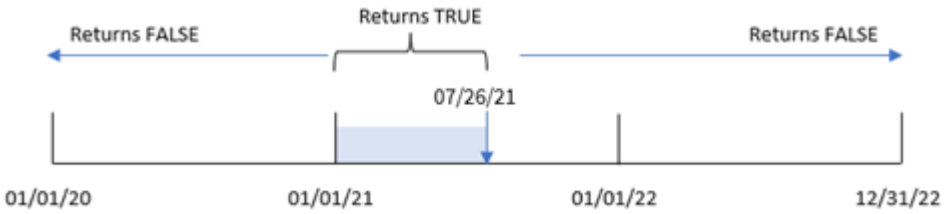
Resultatentabel

| date | jaar tot heden |
|---|---|
| 01/13/2020 | 0 |
| 02/26/2020 | 0 |
| 03/27/2020 | 0 |
| 04/16/2020 | 0 |
| 05/21/2020 | 0 |
| 06/14/2020 | 0 |
| 08/07/2020 | 0 |
| 09/05/2020 | 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 |
| 07/27/2021 | 0 |
| 06/06/2022 | 0 |
| 07/18/2022 | 0 |
| 11/14/2022 | 0 |
| 12/12/2022 | 0 |

Het veld `in_year_to_date` wordt gemaakt in de vorige load-instructie met behulp van de functie `inyeartodate()`. Het eerste verstrekte argument geeft aan welk veld wordt geëvalueerd.

Het tweede argument is een vastgestelde datum voor 26 juli 2021, de `base_date` die de eindgrens van het jaarsegment aangeeft. Een `period_no` van 0 is het laatste argument, wat betekent dat de functie geen jaren voor of na het gesegmenteerde jaar vergelijkt.

*Diagram van functie inyeartodate, geen aanvullende argumenten*




Elke transactie die tussen 1 januari en 26 juli plaatsvindt, retourneert een booleaans resultaat van TRUE. Transactiedatums vóór 2021 en na 26 juli 2021 retourneren FALSE.

## Voorbeeld 2 – period_no

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het aanmaken van een veld, `previous_year_to_date`, dat bepaalt welke transacties een volledig jaar voor het einde van het jaarsegment dat eindigt op 26 juli 2021 hebben plaatsgevonden.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

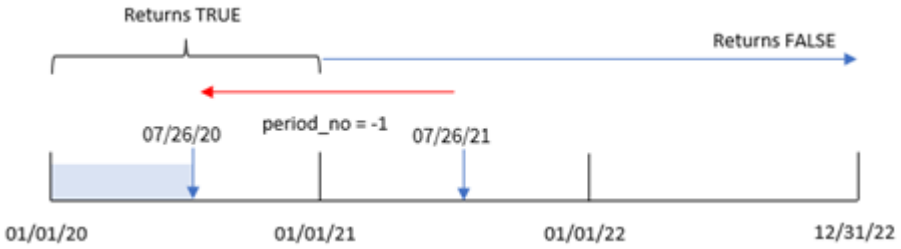
- `date`
- `previous_year_to_date`

Resultatentabel

| date | vorig jaar tot heden |
|---|---|
| 01/13/2020 | -1 |
| 02/26/2020 | -1 |
| 03/27/2020 | -1 |
| 04/16/2020 | -1 |
| 05/21/2020 | -1 |
| 06/14/2020 | -1 |
| 08/07/2020 | 0 |
| 09/05/2020 | 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 |
| 07/27/2021 | 0 |
| 06/06/2022 | 0 |
| 07/18/2022 | 0 |
| 11/14/2022 | 0 |
| 12/12/2022 | 0 |

Een `period_no`-waarde van -1 geeft aan dat de functie `inyeartodate ()` het invoerkwartaalsegment vergelijkt met het voorgaande jaar. Met een invoerdatum van 26 juli 2021, werd het segment van 1 januari 2021 tot 26 juli 2021 aanvankelijk geïdentificeerd als het jaar tot heden. De `period_no` compenseert dit segment vervolgens een volledig jaar eerder, waardoor de datumgrenzen van 1 januari tot 26 juli 2020 worden.

*Diagram van functie inyeartodate, period_no example*




Daarom zal elke transactie die plaatsvindt tussen 1 januari en 26 juli 2020 een booleaans resultaat van `TRUE` retourneren.

## Voorbeeld 3 – first_month_of_year

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het aanmaken van een veld, `in_year_to_date`, dat bepaalt welke transacties hebben plaatsgevonden in hetzelfde jaar tot 26 juli 2021.

In dit voorbeeld stellen we maart in als de eerste maand van het fiscale jaar.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `in_year_to_date`

Resultatentabel

| date | jaar tot heden |
|---|---|
| 01/13/2020 | 0 |
| 02/26/2020 | 0 |
| 03/27/2020 | 0 |
| 04/16/2020 | 0 |
| 05/21/2020 | 0 |
| 06/14/2020 | 0 |
| 08/07/2020 | 0 |
| 09/05/2020 | 0 |
| 01/22/2021 | 0 |
| 02/03/2021 | 0 |
| 03/17/2021 | -1 |
| 04/23/2021 | -1 |
| 05/04/2021 | -1 |

| date | jaar tot heden |
|---|---|
| 06/30/2021 | -1 |
| 07/26/2021 | -1 |
| 07/27/2021 | 0 |
| 06/06/2022 | 0 |
| 07/18/2022 | 0 |
| 11/14/2022 | 0 |
| 12/12/2022 | 0 |

Door 3 te gebruiken als het `first_month_of_year`-argument in de `inyeartodate()`-functie, begint de functie het jaar op 1 maart. De `base_date` van 26 juli 2021 stelt dan de einddatum in voor dat jaarsegment.

*Diagram van functie inyeartodate, first_month_of_year example*

Daarom zal elke transactie die plaatsvindt tussen 1 maart en 26 juli 2021 een booleaans resultaat van TRUE retourneren, terwijl transacties met datums buiten deze grenzen een waarde van FALSE zullen retourneren.

## Voorbeeld 4 – diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld. In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die bepaalt welke transacties plaatsvonden in hetzelfde jaar tot 26 juli 2021 is gemaakt als meting in een diagramobject van de toepassing.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:`date`.

Maak de volgende meting:

```
=inyeartodate(date,'07/26/2021', 0)
```

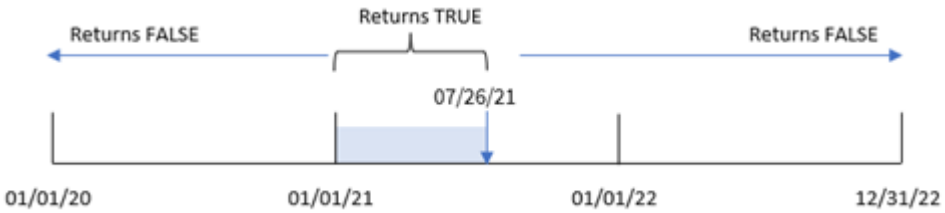
Resultatentabel

| date | =inyeartodate(date,'07/26/2021', 0) |
|---|---|
| 01/13/2020 | 0 |
| 02/26/2020 | 0 |
| 03/27/2020 | 0 |
| 04/16/2020 | 0 |
| 05/21/2020 | 0 |
| 06/14/2020 | 0 |
| 08/07/2020 | 0 |
| 09/05/2020 | 0 |
| 01/22/2021 | -1 |
| 02/03/2021 | -1 |
| 03/17/2021 | -1 |

| date | =inyeartodate(date,'07/26/2021', 0) |
|---|---|
| 04/23/2021 | -1 |
| 05/04/2021 | -1 |
| 06/30/2021 | -1 |
| 07/26/2021 | -1 |
| 07/27/2021 | 0 |
| 06/06/2022 | 0 |
| 07/18/2022 | 0 |
| 11/14/2022 | 0 |
| 12/12/2022 | 0 |

De meting `in_year_to_date` wordt in het diagramobject gemaakt met behulp van de functie `inyeartodate ()`. Het eerste verstrekte argument geeft aan welk veld wordt geëvalueerd. Het tweede argument is een vastgestelde datum voor 26 juli 2021, de `base_date` die de eindgrens van het vergelijkende jaarsegment aangeeft. Een `period_no` van 0 is het laatste argument, wat betekent dat de functie geen jaren voor of na het gesegmenteerde jaar vergelijkt.

*Diagram van functie inyeartodate, diagramobjectvoorbeeld*




Elke transactie die tussen 1 januari en 26 juli 2021 plaatsvindt, retourneert een booleaans resultaat van `TRUE`. Transactiedatums vóór 2021 en na 26 juli 2021 retourneren `FALSE`.

## Voorbeeld 5 – Scenario

Load-script en diagramuitdrukking

**Overzicht**

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset die wordt geladen in een tabel met de naam `Products`.
- Informatie over product-id, producttype, productiedatum en kostprijs.

De eindgebruiker wil een diagramobject dat, per producttype, de kosten weergeeft van de producten die in 2021 tot 26 juli zijn geproduceerd.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:`product_type`.

Maak een meting die de som berekent van elk product dat vóór 27 juli in 2021 is geproduceerd:

```
=sum(if(inyeartodate(manufacture_date,makedate(2021,07,26),0),cost_price,0))
```

Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

| producttype | =sum(if(inyeartodate(manufacture_date,makedate (2021,07,26),0),cost_price,0)) |
|---|---|
| product A | $95.93 |
| product B | $128.66 |

| producttype | =sum(if(inyeartodate(manufacture_date,makedate (2021,07,26),0),cost_price,0)) |
|---|---|
| product C | $61.89 |
| product D | $146.09 |

De functie `inyeartodate()` retourneert een booleaanse waarde bij het evalueren van de productiedatums van elk product. Voor elk product dat in 2021 vóór 27 juli is geproduceerd, retourneert de functie `inyeartodate()` een booleaanse waarde van `TRUE` en telt de `cost_price` op.

Product D is het enige product dat in 2021 ook na 26 juli is geproduceerd. De vermelding met `product_ID` 8203 is geproduceerd op 27 december en kost $25.12. Daarom zijn deze kosten niet opgenomen in het totaal voor Product D in het diagramobject.

## lastworkdate

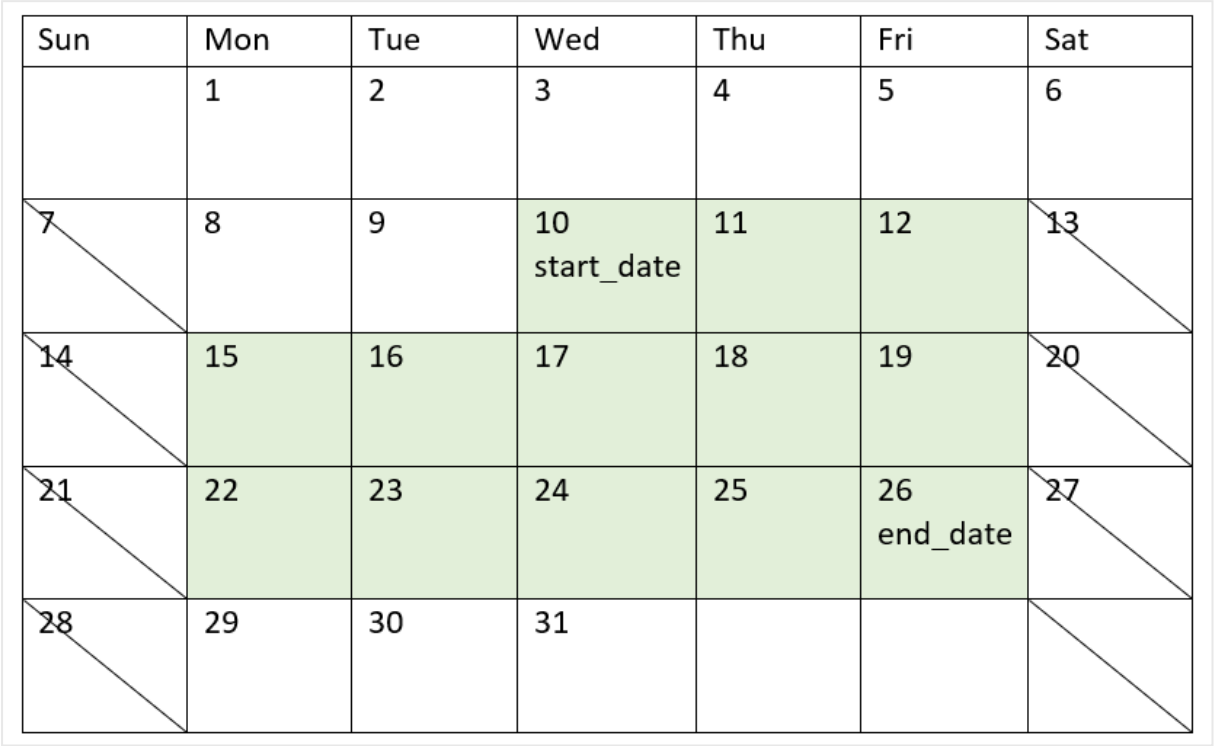
De functie **lastworkdate** retourneert de vroegste einddatum waarbij **no_of_workdays** (maandag-vrijdag) kan worden gehaald als wordt begonnen op **start_date** , waarbij rekening wordt gehouden met een eventueel opgegeven **holiday**. **start_date** en **holiday** moeten geldige datums of tijdstempels zijn.

**Syntaxis:**

```
lastworkdate(start_date, no_of_workdays {, holiday})
```

**Retourgegevenstypen:** geheel getal

*Een kalender die laat zien hoe de `lastworkdate()`-functie wordt gebruikt*



| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| 7 | 8 | 9 | 10 start_date | 11 | 12 | 13 |
| 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 end_date | 27 |
| 28 | 29 | 30 | 31 | | | |

**Beperkingen**

Er is geen methode om de `lastworkdate()`-functie aan te passen voor regio's of scenario's waar sprake is van iets anders dan een werkweek die op maandag begint en op vrijdag eindigt.

De vakantieparameter moet een tekenreeksconstante zijn. Hier wordt geen uitdrukking geaccepteerd.

**Wanneer gebruiken**

De `lastworkdate()`-functie wordt meestal gebruikt als onderdeel van een uitdrukking wanneer de gebruiker de voorgestelde einddatum van een project of opdracht wil berekenen op basis van wanneer het project begint en de vakantiedagen die in die periode plaatsvinden.

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

Argumenten

| Argument | Beschrijving |
|---|---|
| **start_date** | De te evalueren begindatum. |
| **no_of_ workdays** | Het aantal te bereiken werkdagen. |
| **holiday** | Vakantieperioden die moeten worden uitgesloten van werkdagen. Een vakantie wordt aangegeven als een tekenreeks met constante datum. U kunt meerdere vakantiedatums opgegeven, gescheiden door komma's.<br><br>**Voorbeeld:** `'12/25/2013', '12/26/2013', '12/31/2013', '01/01/2014'` |

## Voorbeeld 1 – Basisvoorbeeld

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met project-id's, projectbegindatums en de geschatte inspanning, in dagen, die vereist is voor de projecten. De gegevensset wordt geladen in de tabel `Projects`.
- Een voorafgaande lading met de functie `lastworkdate()` die is ingesteld als het veld `end_date` en bepaalt wat de geplande einddatum is per project.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `start_date`
- `effort`
- `end_date`

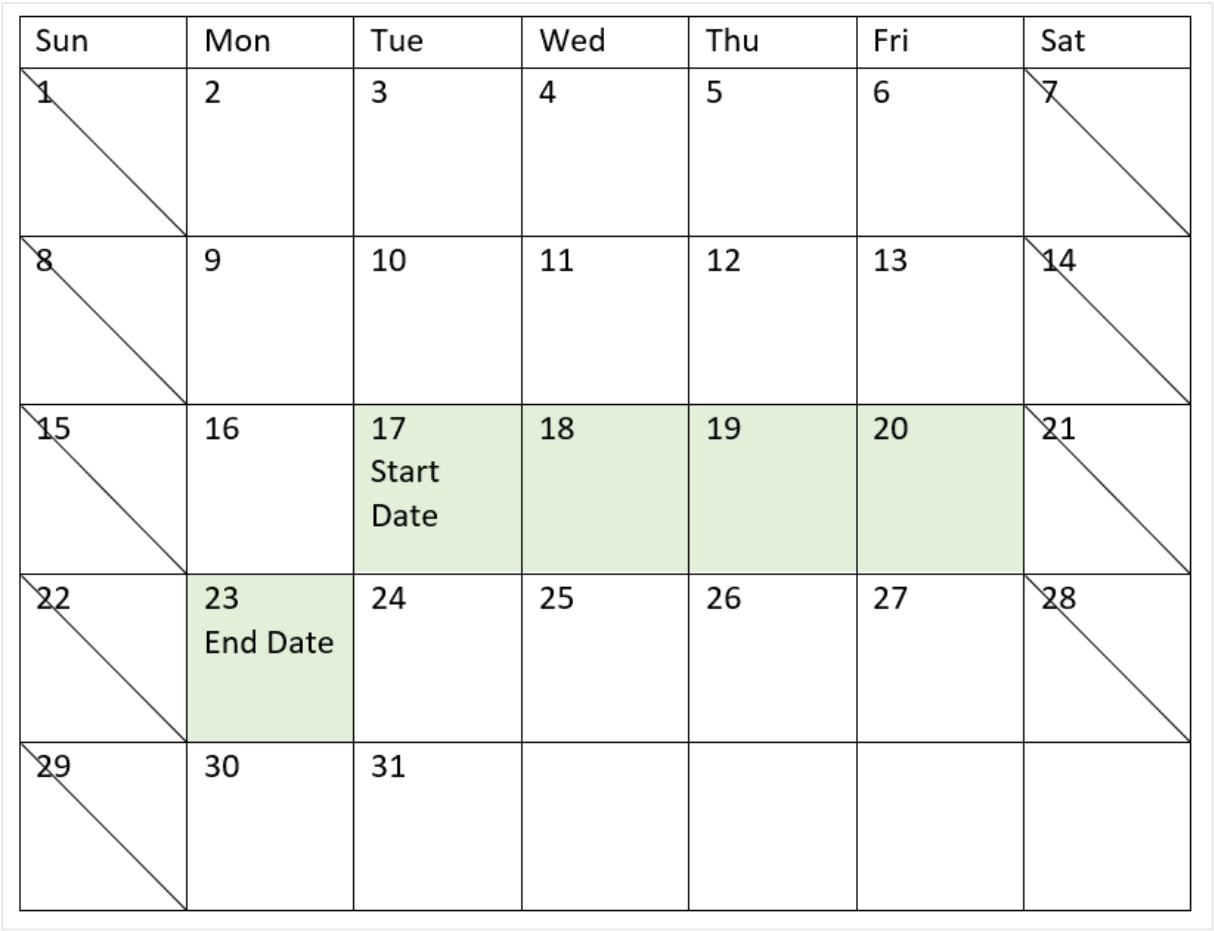
Resultatentabel

| id | start_date | inspanning | end_date |
|---|---|---|---|
| 1 | 01/01/2022 | 14 | 01/20/2022 |
| 2 | 02/10/2022 | 17 | 03/04/2022 |
| 3 | 05/17/2022 | 5 | 05/23/2022 |
| 4 | 06/01/2022 | 12 | 06/16/2022 |
| 5 | 08/10/2022 | 26 | 09/14/2022 |

Omdat er geen vakantiedagen zijn gepland, telt de functie het gedefinieerde aantal werkdagen, van maandag t/m vrijdag, bij de begindatum op om de eerst mogelijke einddatum te vinden.

De volgende kalender toont de begin- en einddatum voor project 3, waarbij de werkdagen groen zijn gemarkeerd.

*Een kalender die de begin- en einddatum van project 3 laat zien*



| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 15 | 16 | 17 Start Date | 18 | 19 | 20 | 21 |
| 22 | 23 End Date | 24 | 25 | 26 | 27 | 28 |
| 29 | 30 | 31 | | | | |

## Voorbeeld 2 - Enkele vakantiedag

Load-script en resultaten

**Overzicht**

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met project-id's, projectbegindatums en de geschatte inspanning, in dagen, die vereist is voor de projecten. De gegevensset wordt geladen in de tabel `Projects`.
- Een voorafgaande lading met de functie `lastworkdate()` die is ingesteld als het veld `end_date` en bepaalt wat de geplande einddatum is per project.

Maar er is één vakantiedag gepland op 18 mei 2022. De `lastworkdate()`-functie in de voorafgaande lading omvat de vakantiedag in het derde argument om te bepalen wanneer het einde van elk project gepland staat.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- start_date
- effort
- end_date

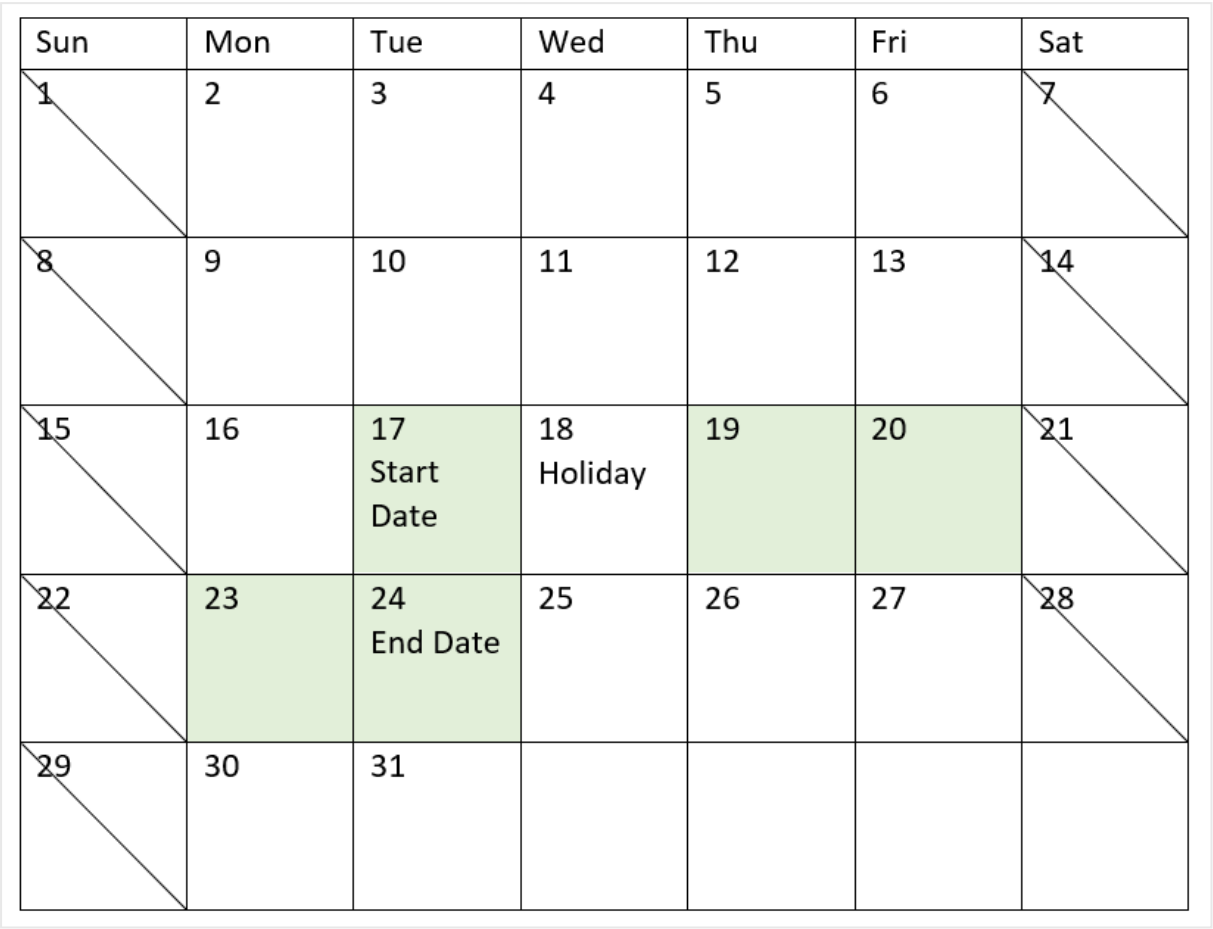
Resultatentabel

| id | start_date | inspanning | end_date |
|---|---|---|---|
| 1 | 01/01/2022 | 14 | 01/20/2022 |
| 2 | 02/10/2022 | 17 | 03/04/2022 |
| 3 | 05/17/2022 | 5 | 05/24/2022 |
| 4 | 06/01/2022 | 12 | 06/16/2022 |
| 5 | 08/10/2022 | 26 | 09/14/2022 |

De enkele, geplande vakantiedag is ingevoerd als het derde argument in de `lastworkdate()`-functie. Het resultaat is dat de einddatum voor project 3 één dag later valt omdat de vakantiedag plaatsvindt op een van de werkdagen vóór de einddatum.

De volgende kalender toont de begin- en einddatum voor project 3 en laat zien dat de vakantiedag de einddatum van het project met één dag verandert.

*Een kalender die de begin- en einddatum van project 3 laat zien met een vakantiedag op 18 mei*



| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 15 | 16 | 17 Start Date | 18 Holiday | 19 | 20 | 21 |
| 22 | 23 | 24 End Date | 25 | 26 | 27 | 28 |
| 29 | 30 | 31 | | | | |

## Voorbeeld 3 - Meerdere vakantiedagen

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met project-id's, projectbegindatums en de geschatte inspanning, in dagen, die vereist is voor de projecten. De gegevensset wordt geladen in de tabel `Projects`.
- Een voorafgaande lading met de functie `lastworkdate()` die is ingesteld als het veld `end_date` en bepaalt wat de geplande einddatum is per project.

Maar er zijn drie vakantiedagen gepland voor 19, 20, 21 en 22 mei. De `lastworkdate()`-functie in de voorafgaande lading omvat elk van de vakantiedagen in het derde argument om te bepalen wanneer het einde van elk project gepland staat.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `start_date`
- `effort`
- `end_date`

Resultatentabel

| id | start_date | inspanning | end_date |
|---|---|---|---|
| 1 | 01/01/2022 | 14 | 01/20/2022 |
| 2 | 02/10/2022 | 17 | 03/04/2022 |
| 3 | 05/17/2022 | 5 | 05/25/2022 |
| 4 | 06/01/2022 | 12 | 06/16/2022 |
| 5 | 08/10/2022 | 26 | 09/14/2022 |

De vier vakantiedagen worden ingevoerd als een lijst met argumenten in de `lastworkdate()`-functie na de begindatum en het aantal werkdagen.

De volgende kalender toont de begin- en einddatum voor project 3 en laat zien dat de vakantiedag de einddatum van het project met drie dagen verandert.

*Een kalender die de begin- en einddatum van project 3 laat zien met een vakantiedag op 19 t/m 22 mei*

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 15 | 16 | 17 Start Date | 18 | 19 Holiday | 20 Holiday | 21 Holiday |
| 22 Holiday | 23 | 24 | 25 End Date | 26 | 27 | 28 |
| 29 | 30 | 31 | | | | |

## Voorbeeld 4 - Enkele vakantiedag (diagram)

Load-script en diagramuitdrukking

### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is de gegevensverzameling echter niet gewijzigd en wordt in de app geladen. Het `end_date`-veld wordt berekend als een meting in een diagram.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `start_date`
- `effort`

Maak de volgende meting om de end_date te berekenen:

- `=LastWorkDate(start_date,effort,'05/18/2022')`

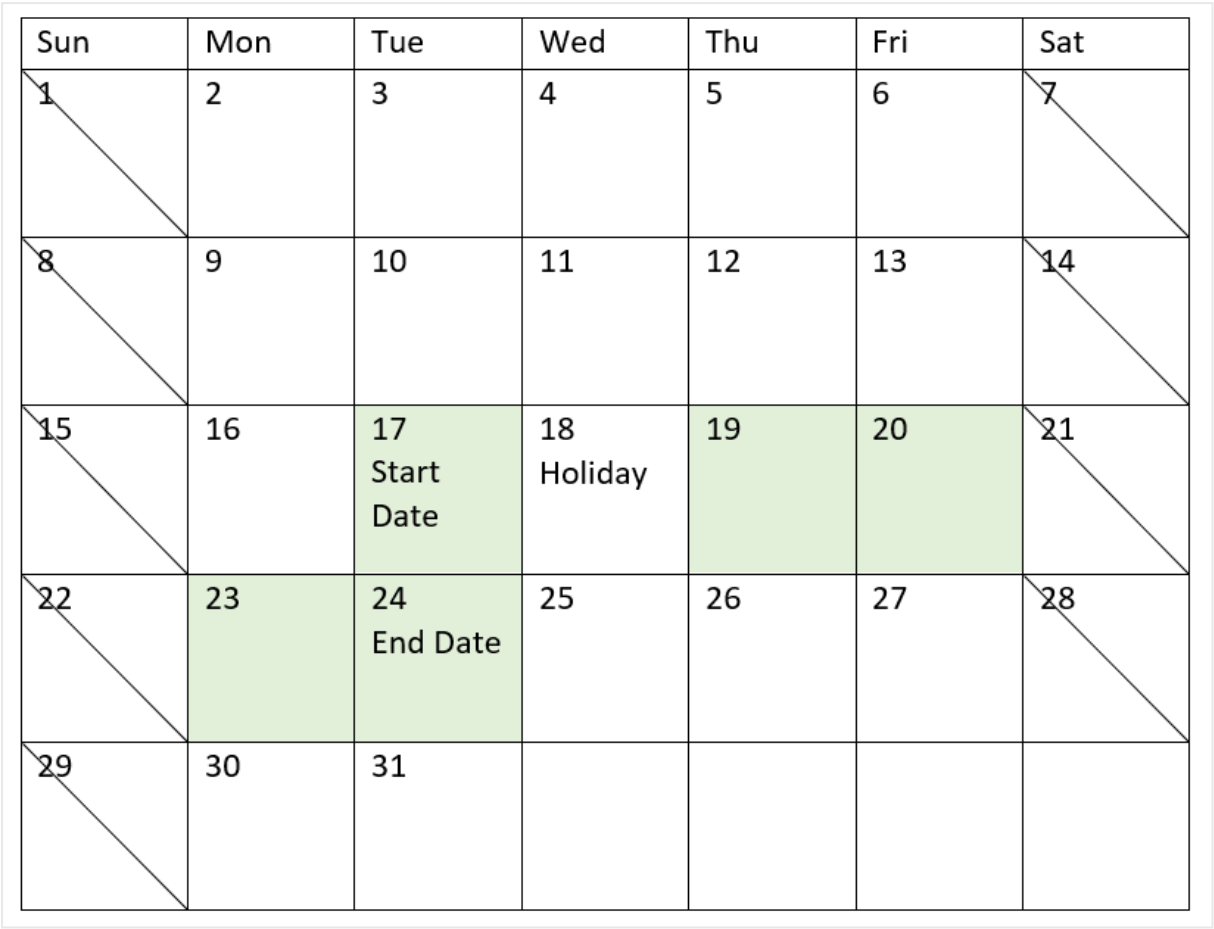
Resultatentabel

| id | start_date | inspanning | =LastWorkDate(start_date,effort,'05/18/2022') |
|---|---|---|---|
| 1 | 01/01/2022 | 14 | 01/20/2022 |
| 2 | 02/10/2022 | 17 | 03/04/2022 |
| 3 | 05/17/2022 | 5 | 05/23/2022 |
| 4 | 06/01/2022 | 12 | 06/16/2022 |
| 5 | 08/10/2022 | 26 | 09/14/2022 |

De enkele, geplande vakantiedag wordt ingevoerd als een meting in het diagram. Het resultaat is dat de einddatum voor project 3 één dag later valt omdat de vakantiedag plaatsvindt op een van de werkdagen vóór de einddatum.

De volgende kalender toont de begin- en einddatum voor project 3 en laat zien dat de vakantiedag de einddatum van het project met één dag verandert.

*Een kalender die de begin- en einddatum van project 3 laat zien met een vakantiedag op 18 mei*



| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 15 | 16 | 17 Start Date | 18 Holiday | 19 | 20 | 21 |
| 22 | 23 | 24 End Date | 25 | 26 | 27 | 28 |
| 29 | 30 | 31 | | | | |

## localtime

Deze functie retourneert een tijdstempel met de huidige tijd voor een opgegeven tijdzone.

**Syntaxis:**

```
LocalTime([timezone [, ignoreDST ]])
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **timezone** | De **timezone** wordt opgegeven als een tekenreeks die een van de geografische plaatsen bevat die worden vermeld onder **Time Zone** in de Windows Control Panel voor Date and Time of als tekenreeks in de vorm 'GMT+hh:mm'.timezone In de onderstaande tabel staat tevens een lijst met acceptabele plaatsen en tijdzones.<br><br>Als geen tijdzone is opgegeven, wordt de lokale tijd geretourneerd.<br><br>*Als u een zomertijdoffset (DST) (d.w.z. u specificeert een **ignoreDST**-argumentwaarde die evalueert naar False) gebruikt, moet u een plek specificeren in plaats van een GMT-offset, in het **place**-argument. De reden hiervoor is dat het aanpassen van de zomertijd breedtegegevens vereist naast de lengtegegevens die door een GMT-offset worden verstrekt. Ga voor meer informatie naar GMT-offsets gebruiken in combinatie met zomertijd (page 971).* |
| **ignoreDST** | Als dit argument evalueert in True, wordt DST (zomertijd) genegeerd. Geldige argumentwaarden die in True evalueren, zijn onder meer `-1` en `True()`.<br><br>Als dit argument evalueert in False, wordt het tijdstempel aangepast voor de zomertijd. Geldige argumentwaarden die in False evalueren, zijn onder meer `0` en `False()`.<br><br>Als de **ignoreDST**-argumentwaarde ongeldig is, evalueert de functie de uitdrukking alsof de **ignore_dst**-waarde evalueert in True. Als de **ignoreDST**-argumentwaarde niet is gespecificeerd, evalueert de functie de uitdrukking alsof de **ignore_dst**-waarde evalueert in False. |

Geldige plaatsen en tijdzones

| 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 |

| A-C | D-K | L-R | S-Z |
|---|---|---|---|
| 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 |
| 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 | - |

| A-C | D-K | L-R | S-Z |
|---|---|---|---|
| Chihuahua | - | Riyadh | - |
| Chongqing | - | Rome | - |
| Copenhagen | - | - | - |

**Voorbeelden en resultaten:**

De onderstaande voorbeelden zijn gebaseerd op dat de functie wordt aangeroepen op 2023-08-14 om 08:39:47 plaatselijke tijd waarbij de plaatselijke tijdzone van de server of desktopomgeving GMT-05:00 is en zich in een regio bevindt waarbij zomertijd is geïmplementeerd vanaf deze vermelde datum.

Scriptvoorbeelden

| Voorbeeld | Resultaat |
|---|---|
| `localtime ()` | Retourneert de lokale tijd, 2023-08-14 08:39:47. |
| `localtime ('London')` | Retourneert de lokale tijd in Londen, 2023-08-14 13:39:47. |
| `localtime ('GMT+02:00')` | Retourneert de lokale tijd in de tijdzone van GMT+02:00, 2023-08-14 14:39:47. Er wordt geen aanpassing gedaan voor zomertijd omdat er een GMT-offset is opgegeven in plaats van een plaats. |
| `localtime ('Paris',-1)` | Retourneert de lokale tijd in Parijs waarbij de zomertijd wordt genegeerd, 2023-08-14 13:39:47. |
| `localtime ('Paris',True())` | Retourneert de lokale tijd in Parijs waarbij de zomertijd wordt genegeerd, 2023-08-14 13:39:47. |
| `localtime ('Paris',0)` | Retourneert de lokale tijd in Parijs waarbij de zomertijd wordt genegeerd, 2023-08-14 14:39:47. |
| `localtime ('Paris',False ())` | Retourneert de lokale tijd in Parijs waarbij de zomertijd wordt genegeerd, 2023-08-14 14:39:47. |

## GMT-offsets gebruiken in combinatie met zomertijd

Sinds de implementatie van ICU-bibliotheken (International Components for Unicode) in Qlik Sense, vereist het gebruik van of GMT-offsets (Greenwich Mean Time) in combinatie met zomertijd (DST) aanvullende breedtegegevens.

GMT is een lengte-offset (oost-west) en zomertijd (DST) is a een breedte-offset (noord-zuid). Zo delen Helsinki (Finland) en Johannesburg (Zuid-Afrika) dezelfde GMT+02:00 offset, maar ze delen niet dezelfde zomertijdoffset. Dit betekent dat iedere zomertijdoffset naast de GMT-offset informatie vereist over de breedtepositie van de plaatselijke tijdzone (geografische tijdzoneinvoer) voor volledige informatie over plaatselijke zomertijdomstandigheden.

## lunarweekend

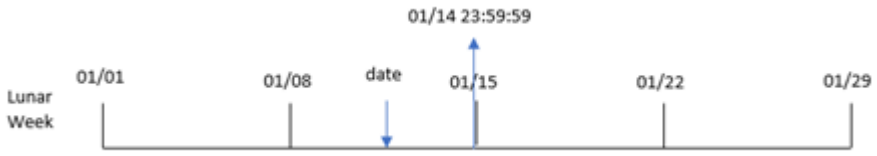
Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de laatste milliseconde van de laatste dag van de maanweek die **date** bevat. Maanweken in Qlik Sense worden gedefinieerd door het feit dat 1 januari wordt gezien als de eerste dag van de week en, met uitzondering van de laatste week van het jaar, bevat elke week precies zeven dagen.

**Syntaxis:**

```
LunarweekEnd(date[, period_no[, first_week_day]])
```

**Retourgegevenstypen:** dubbele waarde

*Voorbeelddiagram van de `lunarweekend()`-functie*




De `lunarweekend()`-functie bepaalt welke maanweek de `date` in valt. Het retourneert vervolgens een tijdstempel, in datumnotatie, voor de laatste milliseconde van die week.

Argumenten

| Argument | Beschrijving |
|---|---|
| **date** | De te evalueren datum of tijdstempel. |
| **period_no** | **period_no** is een geheel getal, of een uitdrukking die wordt herleid tot een geheel getal, waarbij de waarde 0 de maanweek aangeeft die **date** bevat. Negatieve waarden in **period_no** geven voorafgaande maanweken aan en positieve waarden geven volgende maanweken aan. |
| **first_week_day** | Een verschuiving kan groter of kleiner zijn dan nul. Hiermee wordt het begin van het jaar gewijzigd met het opgegeven aantal dagen en/of fracties van een dag. |

### Wanneer gebruiken

De `lunarweekend()`-functie wordt doorgaans gebruikt als onderdeel van een uitdrukking als de gebruiker wil dat in de berekening een deel van de week wordt gebruikt dat nog niet heeft plaatsgevonden. Anders dan de `weekend()`-functie, eindigt de laatste maanweek van het kalenderjaar op 31 december. De `lunarweekend()`-functie zou bijvoorbeeld kunnen worden gebruikt om rente te berekenen die tijdens de week nog niet is opgebouwd.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `lunarweekend('01/12/2013')` | Retourneert 01/14/2013 23:59:59. |
| `lunarweekend('01/12/2013', -1)` | Retourneert 01/07/2013 23:59:59. |
| `lunarweekend('01/12/2013', 0, 1)` | Retourneert 01/15/2013 23:59:59. |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – geen aanvullende argumenten

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties voor 2022 die wordt geladen in de tabel `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.
- Het maken van een veld, `end_of_week`, dat een tijdstempel voor het einde van de maanweek retourneert waarin de transactie plaatsvond.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `end_of_week`
- `end_of_week_timestamp`

Resultatentabel

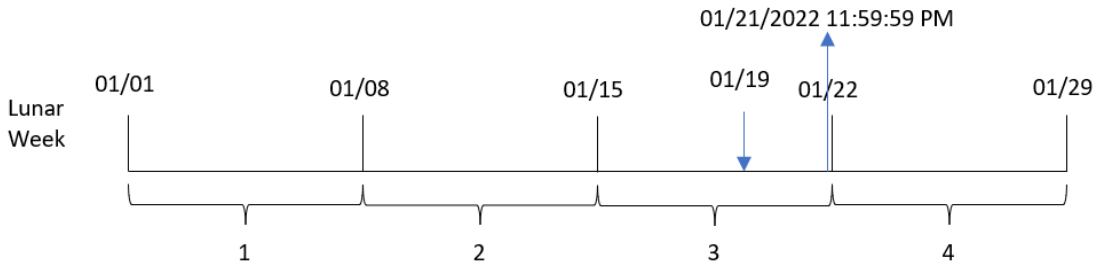
| date | end_of_week | end_of_week_timestamp |
|---|---|---|
| 1/7/2022 | 01/07/2022 | 1/7/2022 11:59:59 PM |
| 1/19/2022 | 01/21/2022 | 1/21/2022 11:59:59 PM |
| 2/5/2022 | 02/11/2022 | 2/11/2022 11:59:59 PM |
| 2/28/2022 | 03/04/2022 | 3/4/2022 11:59:59 PM |
| 3/16/2022 | 03/18/2022 | 3/18/2022 11:59:59 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 |

| date | end_of_week | end_of_week_timestamp |
|---|---|---|
| 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 |
| 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 |

Het `end_of_week`-veld wordt gemaakt in de voorafgaande load-instructie met behulp van de `lunarweekend()`-functie en geeft het `date`-veld door als het argument van de functie.

De `lunarweekend()`-functie identificeert in welke maanweek de datumwaarde valt en retourneert een tijdstempel voor de laatste milliseconde van die week.

*Diagram van `lunarweekend()`-functie, voorbeeld zonder aanvullende argumenten*




Transactie 8189 vond plaats op 19 januari. De `lunarweekend()`-functie identificeert dat de maanweek op 15 januari begint. De `end_of_week`-waarde voor die transactie retourneert daarom de laatste milliseconde van die dag, 21 januari om 11:59:59 PM.

## Voorbeeld 2 – period_no

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het maken van een veld, `previous_lunar_week_end`, dat de tijdstempel voor het einde van de maanweek retourneert voordat de transactie plaatsvond.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

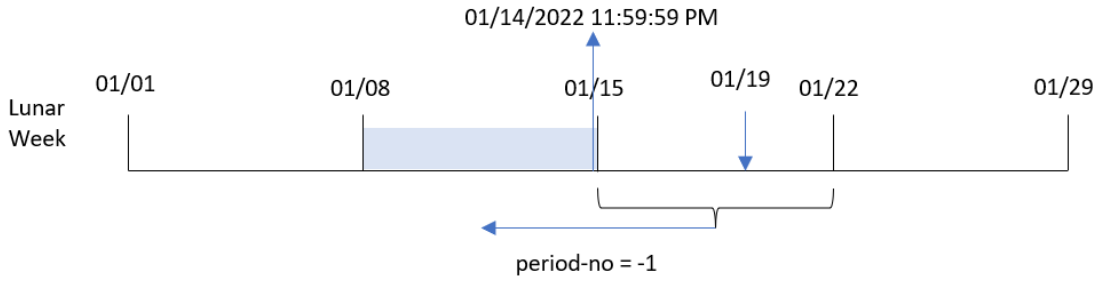
- `date`
- `previous_lunar_week_end`
- `previous_lunar_week_end_timestamp`

Resultatentabel

| date | vorige_maanweekeinde | vorige_maanweekeinde_tijdstempel |
|---|---|---|
| 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 dit geval, omdat een `period_no` van -1 werd gebruikt als het offset-argument in de `lunarweekend()`-functie, identificeert de functie eerst de maanweek waarin de transacties plaatsvonden. Het verschuift dan naar een week eerder en identificeert de laatste milliseconde van die maanweek.

*Diagram van `lunarweekend()`-functie, voorbeeld period_no*




Transactie 8189 vond plaats op 19 januari. De `lunarweekend()`-functie identificeert dat de maanweek op 15 januari begint. Daarom begon de vorige maanweek op 8 januari en eindigde hij op 14 januari om 11:59:59 PM; dit is de waarde die voor het `previous_lunar_week_end`-veld wordt geretourneerd.

## Voorbeeld 3 – first_week_day

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld. In dit voorbeeld stellen we het begin van de maanweek in op 5 januari.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `end_of_week`
- `end_of_week_timestamp`

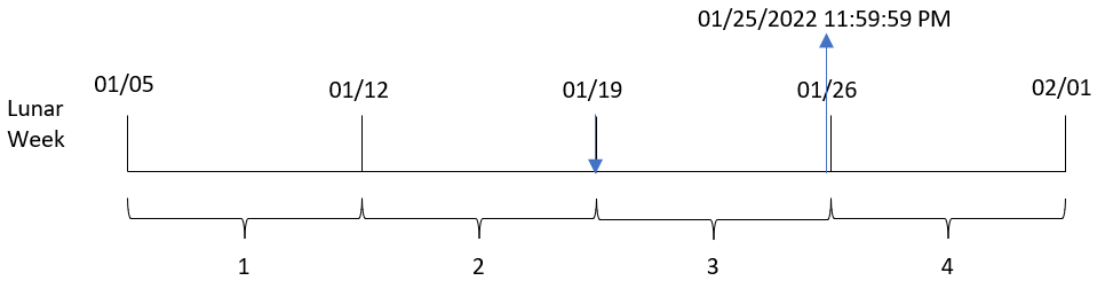
Resultatentabel

| 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 dit geval verschuift het begin van het jaar van 1 januari naar 5 januari omdat het `first_week_date`-argument 4 in de `lunarweekend()`-functie is gebruikt.

*Diagram van de `lunarweekend()`-functie, voorbeeld met first_week_day*




Transactie 8189 vond plaats op 19 januari. Omdat maanweken op 5 januari beginnen, identificeert de `lunarweekend()`-functie dat de maanweek waarin 19 januari valt, tevens begint met 19 januari. Daarom valt het einde van die maanweek op 25 januari om 11:59:59 PM. Dit is de waarde die is geretourneerd voor het `end_of_week`-veld.

## Voorbeeld 4 – diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die een tijdstempel retourneert voor het einde van de maanweek waarin de transacties plaatsvonden is gemaakt als meting in een diagramobject van de applicatie.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `date`.

Voeg de volgende metingen toe:

```
=lunarweekend(date)
```

```
=timestamp(lunarweekend(date))
```

Resultatentabel

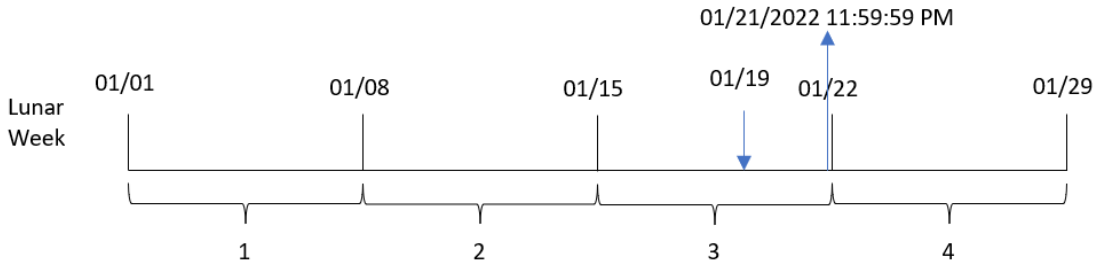
| date | =lunarweekend(date) | =tijdstempel(lunarweekend(date)) |
|---|---|---|
| 1/7/2022 | 01/07/2022 | 1/7/2022 11:59:59 PM |
| 1/19/2022 | 01/21/2022 | 1/21/2022 11:59:59 PM |
| 2/5/2022 | 02/11/2022 | 2/11/2022 11:59:59 PM |
| 2/28/2022 | 03/04/2022 | 3/4/2022 11:59:59 PM |
| 3/16/2022 | 03/18/2022 | 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) | =tijdstempel(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 |

De meting `end_of_week` wordt gemaakt in het diagramobject met behulp van de `lunarweekend()`-functie en geeft het veld `date` door als het argument van de functie.

De `lunarweekend()`-functie identificeert in welke maanweek de datumwaarde valt en retourneert een tijdstempel voor de laatste milliseconde van die week.

*Diagram van `lunarweekend()`-functie, diagramobjectvoorbeeld*




Transactie 8189 vond plaats op 19 januari. De `lunarweekend()`-functie identificeert dat de maanweek op 15 januari begint. De `end_of_week`-waarde voor die transactie retourneert daarom de laatste milliseconde van die dag, 21 januari om 11:59:59 PM.

## Voorbeeld 5 – Scenario

Load-script en diagramuitdrukking

**Overzicht**

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset die wordt geladen in een tabel met de naam `Employee_Expenses`.
- De werknemers-id's, werknemersnaam en de gemiddelde dagelijkse onkostendeclaraties van elke werknemer.

De eindgebruiker wil graag een diagramobject dat per werknemer-id en werknemersnaam de geschatte nog te maken onkostendeclaraties voor de rest van de maanweek weergeeft.

### Load-script

```
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
];
```

### Resultaten

### Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel.
2. Voeg de volgende velden als dimensies toe.
    - `employee_id`
    - `employee_name`
3. Maak nu de volgende meting om de opgebouwde rente te berekenen:
   `=(lunarweekend(today(1))-today(1))*avg_daily_claim`
4. Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

| employee_id | employee_name | =(lunarweekend(today(1))-today(1))*avg_daily_claim |
|---|---|---|
| 182 | Mark | $75.00 |
| 183 | Deryck | $62.50 |
| 184 | Dexter | $62.50 |
| 185 | Sydney | $135.00 |
| 186 | Agatha | $90.00 |

Door de datum van vandaag als enig argument te gebruiken, retourneert de `lunarkweekend()`-functie de einddatum van de huidige maanweek. Door vervolgens de datum van vandaag af te trekken van de einddatum van de maanweek, retourneert de uitdrukking het aantal resterende dagen in deze week.

Deze waarde wordt vervolgens vermenigvuldigd met de gemiddelde dagelijkse onkostendeclaratie van elke werknemer om de geschatte waarde te berekenen van de declaraties die elke werknemer naar verwachting in de resterende maanweek zal indienen.

## lunarweekname

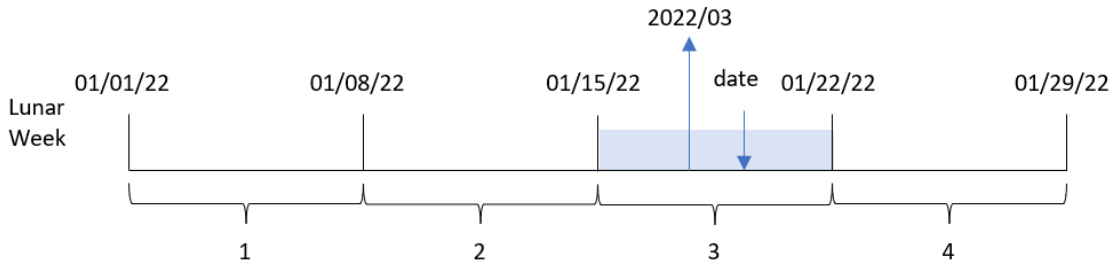
Deze functie retourneert een weergavewaarde voor het jaar en maanweeknummer dat overeenkomt met een tijdstempel van de eerste milliseconde van de eerste dag van de week die **date** bevat. Maanweken in Qlik Sense worden gedefinieerd door het feit dat 1 januari wordt gezien als de eerste dag van de week en, met uitzondering van de laatste week van het jaar, bevat elke week precies zeven dagen.

**Syntaxis:**

```
LunarWeekName(date [, period_no[, first_week_day]])
```

**Retourgegevenstypen:** dubbele waarde

*Voorbeelddiagram van de `lunarweekname()`-functie*




De `lunarweekname()`-functie bepaalt in welke maanweek de datum valt, waarbij de weektelling begint op 1 januari. Er wordt vervolgens een waarde geretourneerd die bestaat uit de `year/weekcount`.

Argumenten

| Argument | Beschrijving |
|---|---|
| **date** | De te evalueren datum of tijdstempel. |
| **period_no** | **period_no** is een geheel getal, of een uitdrukking die wordt herleid tot een geheel getal, waarbij de waarde 0 de maanweek aangeeft die **date** bevat. Negatieve waarden in **period_no** geven voorafgaande maanweken aan en positieve waarden geven volgende maanweken aan. |
| **first_week_day** | Een verschuiving kan groter of kleiner zijn dan nul. Hiermee wordt het begin van het jaar gewijzigd met het opgegeven aantal dagen en/of fracties van een dag. |

## Wanneer gebruiken

De `lunarweekname()`-functie is handig als u aggregaties per maanweek wilt vergelijken. Bijvoorbeeld: de functie kan worden gebruikt om de totale verkoop van producten per maanweek te bepalen. Maanweken zijn handig wanneer u er zeker van wilt zijn dat alle waarden uit de eerste week van het jaar, alleen de waarden vanaf 1 januari omvatten.

Deze dimensies kunnen in het load-script worden gemaakt door de functie te gebruiken om een veld in een masterkalendertabel te maken. De functie kan ook rechtstreeks in een diagram worden gebruikt als een berekende dimensie.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `lunarweekname('01/12/2013')` | Retourneert 2006/02. |
| `lunarweekname('01/12/2013', -1)` | Retourneert 2006/01. |
| `lunarweekname('01/12/2013', 0, 1)` | Retourneert 2006/02. |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – Datum zonder aanvullende argumenten

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties voor 2022 die wordt geladen in de tabel `Transactions`.

- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.
- Het veld `lunar_week_name` wordt gemaakt en retourneert het jaar- en weeknummer voor de maanweek waarin de transacties plaatsvonden.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `lunar_week_name`

Resultatentabel

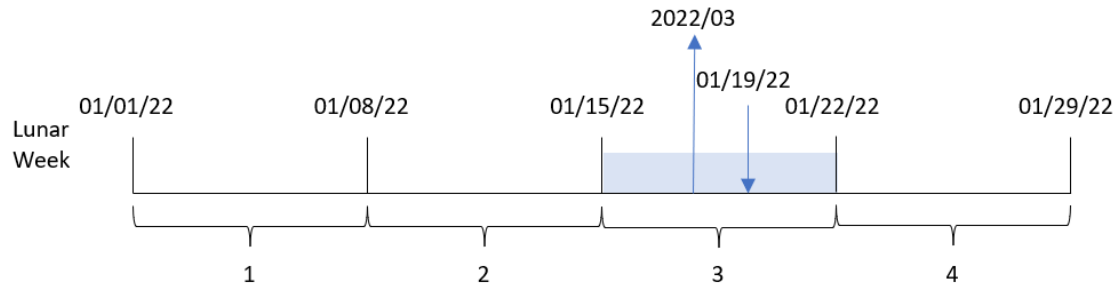
| date | maanweeknaam |
|---|---|
| 1/7/2022 | 2022/01 |

| date | maanweeknaam |
|---|---|
| 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 |
| 8/19/2022 | 2022/33 |
| 9/26/2022 | 2022/39 |
| 10/14/2022 | 2022/41 |
| 10/29/2022 | 2022/44 |

Het `lunar_week_name`-veld wordt gemaakt in de voorafgaande load-instructie met behulp van de `lunarweekname()`-functie en geeft het `date`-veld door als het argument van de functie.

De `lunarweekname()`-functie identificeert in welke maanweek de datumwaarde valt en retourneert het jaar- en weeknummer van die datum.

*Diagram van `lunarweekname()`-functie, voorbeeld zonder aanvullende argumenten*



Transactie 8189 vond plaats op 19 januari. De `lunarweekname()`-functie identificeert dat deze datum in de maanweek valt die op 15 januari begint. Dit is de derde maanweek van het jaar. Daarom is de `lunar_week_name`-waarde die voor deze transactie wordt geretourneerd 2022/03.

## Voorbeeld 2 – Datum met period_no argument

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het veld `previous_lunar_week_name` wordt gemaakt en retourneert het jaar- en weeknummer voor de maanweek voorafgaand aan de week waarin de transacties plaatsvonden.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

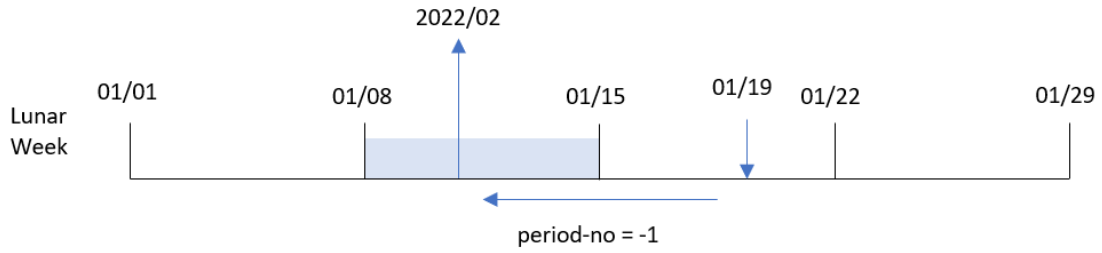
- `date`
- `previous_lunar_week_name`

Resultatentabel

| date | vorige_maanweeknaam |
|---|---|
| 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 dit geval, omdat een `period_no` van -1 werd gebruikt als het offset-argument in de `lunarweekname()`-functie, identificeert de functie eerst de maanweek waarin de transacties plaatsvonden. De functie retourneert vervolgens het jaar en het nummer van één week eerder.

*Diagram van `lunarweekname()`-functie, voorbeeld period_no*




Transactie 8189 vond plaats op 19 januari. De `lunarweekname()`-functie identificeert dat deze transactie in de derde maanweek van het jaar plaatsvond en retourneert daarom het jaar en de waarde voor één week eerder, 2022/02, voor het `previous_lunar_week_name`-veld.

## Voorbeeld 3 – Datum met first_week_day-argument

### Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld. In dit voorbeeld stellen we het begin van de maanweek in op 5 januari.

#### Load-script

```
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
];
```

### Resultaten

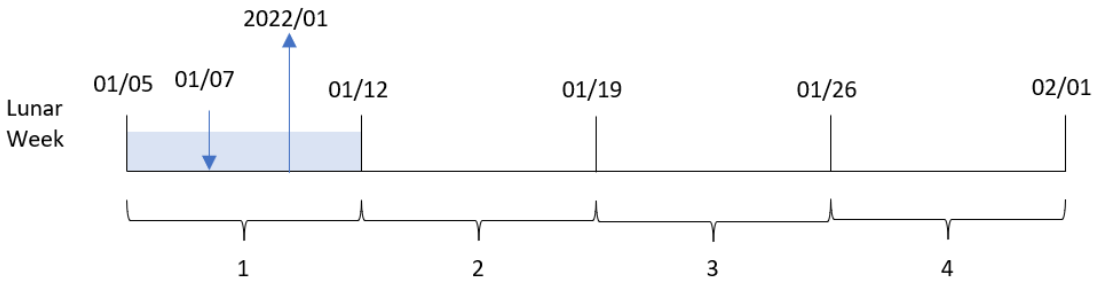
Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `lunar_week_name`

Resultatentabel

| date | maanweeknaam |
|---|---|
| 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 |

*Diagram van de `lunarweekname()`-functie, voorbeeld met first_week_day*




In dit geval verschuift het begin van de maanweken van 1 januari naar 5 januari omdat het `first_week_date`-argument 4 in de `lunarweekname()`-functie is gebruikt.

Transactie 8188 vond plaats op 7 januari. Omdat maanweken op 5 januari beginnen, identificeert de `lunarweekname()`-functie dat de maanweek waarin 7 januari valt, de eerste maanweek van het jaar is. Daarom is de `lunar_week_name`-waarde die voor deze transactie wordt geretourneerd 2022/01.

## Voorbeeld 4 – diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die het maanweeknunmer en het jaar retourneert waarin de transacties plaatsvonden, is gemaakt als meting in een diagramobject van de applicatie.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `date`.

Om de begindatum van de maanweek te berekenen waarin een transactie heeft plaatsgevonden, maakt u de volgende meting:

```
=lunarweekname(date)
```

Resultatentabel

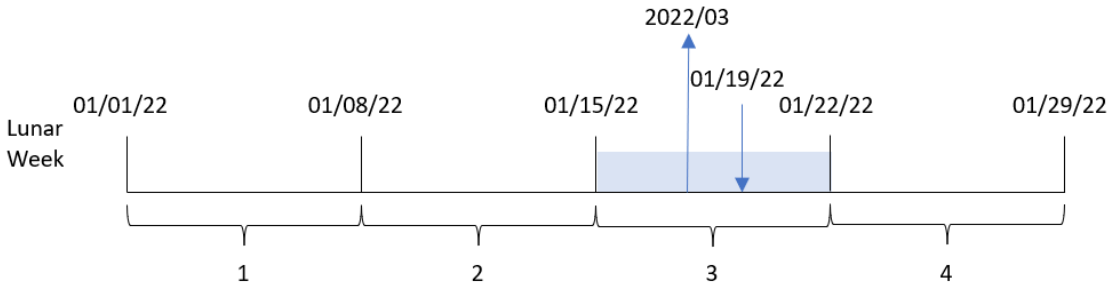
| date | =lunarweekname(datum) |
|---|---|
| 1/7/2022 | 2022/01 |
| 1/19/2022 | 2022/03 |
| 2/5/2022 | 2022/06 |
| 2/28/2022 | 2022/09 |
| 3/16/2022 | 2022/11 |
| 4/1/2022 | 2022/13 |
| 5/7/2022 | 2022/19 |
| 5/16/2022 | 2022/20 |
| 6/15/2022 | 2022/24 |
| 6/26/2022 | 2022/26 |
| 7/9/2022 | 2022/28 |
| 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 |
| 8/19/2022 | 2022/33 |

| date | =lunarweekname(datum) |
|---|---|
| 9/26/2022 | 2022/39 |
| 10/14/2022 | 2022/41 |
| 10/29/2022 | 2022/44 |

De meting `lunar_week_name` wordt gemaakt in het diagramobject met behulp van de functie `lunarweekname()` en geeft het veld `date` door als het argument van de functie.

De `lunarweekname()`-functie identificeert in welke maanweek de datumwaarde valt en retourneert het jaar- en weeknummer van die datum.

*Diagram van `lunarweekname()`-functie, diagramobjectvoorbeeld*




Transactie 8189 vond plaats op 19 januari. De `lunarweekname()`-functie identificeert dat deze datum in de maanweek valt die op 15 januari begint. Dit is de derde maanweek van het jaar. Daarom is de `lunar_week_name`-waarde voor deze transactie 2022/03.

## Voorbeeld 5 – Scenario

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2022 die wordt geladen in een tabel met de naam `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.

De eindgebruiker wil graag een diagramobject dat de totale verkoop per week voor het huidige jaar weergeeft. Week 1, met een lengte van zeven dagen, zou op 1 januari moeten beginnen. Dit kan zelfs worden bereikt als deze dimensie niet beschikbaar is in het gegevensmodel, door de functie `lunarweekname()` te gebruiken als een berekende dimensie in het diagram.

### Load-script

```
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
];
```

### Resultaten

### Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel.
2. Maak een berekende dimensie met behulp van de volgende uitdrukking:
   `=lunarweekname(date)`
3. Bereken de totale verkoop met de volgende aggregatiemeting:
   `=sum(amount)`
4. Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

| =lunarweekname(datum) | =sum(amount) |
|---|---|
| 2022/01 | $17.17 |
| 2022/03 | $37.23 |
| 2022/06 | $57.42 |

| =lunarweekname(datum) | =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

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de eerste milliseconde van de eerste dag van de maanweek die **date** bevat. Maanweken in Qlik Sense worden gedefinieerd door het feit dat 1 januari wordt gezien als de eerste dag van de week en, met uitzondering van de laatste week van het jaar, bevat elke week precies zeven dagen.

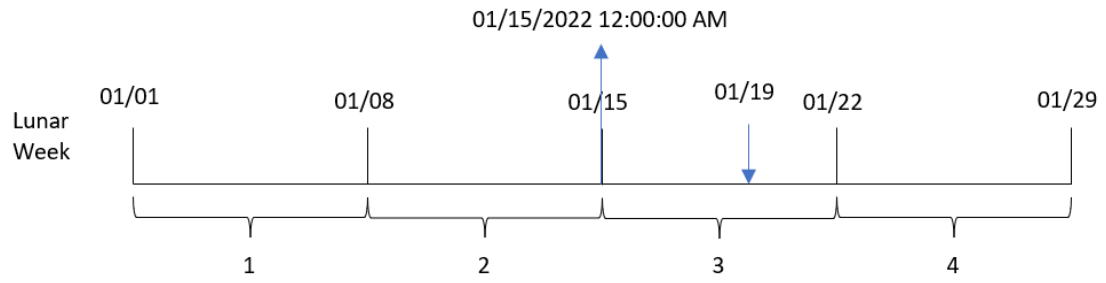
**Syntaxis:**

```
LunarweekStart(date[, period_no[, first_week_day]])
```

**Retourgegevenstypen:** dubbele waarde

De `lunarweekstart()`-functie bepaalt welke maanweek de `date` in valt. Het retourneert vervolgens een tijdstempel, in datumnotatie, voor de eerste milliseconde van die week.

*Voorbeelddiagram van de `lunarweekstart()`-functie*




Argumenten

| Argument | Beschrijving |
|---|---|
| **date** | De te evalueren datum of tijdstempel. |
| **period_no** | **period_no** is een geheel getal, of een uitdrukking die wordt herleid tot een geheel getal, waarbij de waarde 0 de maanweek aangeeft die **date** bevat. Negatieve waarden in **period_no** geven voorafgaande maanweken aan en positieve waarden geven volgende maanweken aan. |
| **first_week_day** | Een verschuiving kan groter of kleiner zijn dan nul. Hiermee wordt het begin van het jaar gewijzigd met het opgegeven aantal dagen en/of fracties van een dag. |

## Wanneer gebruiken

De `lunarweekstart()`-functie wordt doorgaans gebruikt als onderdeel van een uitdrukking als de gebruiker wil dat in de berekening het deel van de week wordt gebruikt dat al is geweest. Anders dan de `weekstart()`-functie beginnen weken aan het begin van ieder nieuw kalenderjaar, op 1 januari en alle daaropvolgende weken beginnen zeven dagen later. De `lunarweekstart()`-functie wordt niet beïnvloed door de systeemvariabele `FirstWeekDay`.

De `lunarweekstart()` kan bijvoorbeeld worden gebruikt om de rente te berekenen die in een week tot nu toe is opgebouwd.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `lunarweekstart ('01/12/2013')` | Retourneert 01/08/2013. |
| `lunarweekstart ('01/12/2013', -1)` | Retourneert 01/01/2013. |
| `lunarweekstart ('01/12/2013', 0, 1 )` | Retourneert 01/09/2013, want als first_week_day wordt ingesteld op 1 wordt het begin van het jaar veranderd in 01/02/2013. |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – Geen aanvullende argumenten

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties voor 2022 die wordt geladen in de tabel `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.
- Het maken van een veld, `start_of_week`, dat een tijdstempel voor het begin van de maanweek retourneert waarin de transactie plaatsvond.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `start_of_week`
- `start_of_week_timestamp`

Resultatentabel

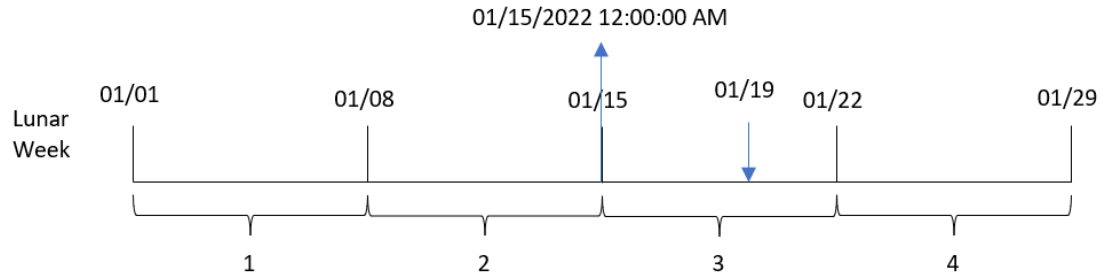
| date | start_of_week | start_of_week_timestamp |
|---|---|---|
| 1/7/2022 | 01/01/2022 | 1/1/2022 12:00:00 AM |
| 1/19/2022 | 01/15/2022 | 1/15/2022 12:00:00 AM |
| 2/5/2022 | 02/05/2022 | 2/5/2022 12:00:00 AM |
| 2/28/2022 | 02/26/2022 | 2/26/2022 12:00:00 AM |
| 3/16/2022 | 03/12/2022 | 3/12/2022 12:00:00 AM |
| 4/1/2022 | 03/26/2022 | 3/26/2022 12:00:00 AM |
| 5/7/2022 | 05/07/2022 | 5/7/2022 12:00:00 AM |
| 5/16/2022 | 05/14/2022 | 5/14/2022 12:00:00 AM |
| 6/15/2022 | 06/11/2022 | 6/11/2022 12:00:00 AM |
| 6/26/2022 | 06/25/2022 | 6/25/2022 12:00:00 AM |
| 7/9/2022 | 07/09/2022 | 7/9/2022 12:00:00 AM |
| 7/22/2022 | 07/16/2022 | 7/16/2022 12:00:00 AM |

| date | start_of_week | start_of_week_timestamp |
|---|---|---|
| 7/23/2022 | 07/23/2022 | 7/23/2022 12:00:00 AM |
| 7/27/2022 | 07/23/2022 | 7/23/2022 12:00:00 AM |
| 8/2/2022 | 07/30/2022 | 7/30/2022 12:00:00 AM |
| 8/8/2022 | 08/06/2022 | 8/6/2022 12:00:00 AM |
| 8/19/2022 | 08/13/2022 | 8/13/2022 12:00:00 AM |
| 9/26/2022 | 09/24/2022 | 9/24/2022 12:00:00 AM |
| 10/14/2022 | 10/08/2022 | 10/8/2022 12:00:00 AM |
| 10/29/2022 | 10/29/2022 | 10/29/2022 12:00:00 AM |

Het `start_of_week`-veld wordt gemaakt in de voorafgaande load-instructie met behulp van de `lunarweekstart()`-functie en geeft het `date`-veld door als het argument van de functie.

De `lunarweekstart()`-functie identificeert in welke maanweek de datum valt en retourneert een tijdstempel voor de eerste milliseconde van die week.

*Diagram van `lunarweekstart()`-functie, voorbeeld zonder aanvullende argumenten*




Transactie 8189 vond plaats op 19 januari. De `lunarweekstart()`-functie identificeert dat de maanweek op 15 januari begint. De `start_of_week`-waarde voor die transactie retourneert daarom de eerste milliseconde van die dag, 15 januari om 12:00:00 AM.

## Voorbeeld 2 – period_no

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het maken van een veld, `previous_lunar_week_start`, dat de tijdstempel voor het begin van de maanweek retourneert voordat de transactie plaatsvond.

### Load-script

```
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
];
```

### Resultaten

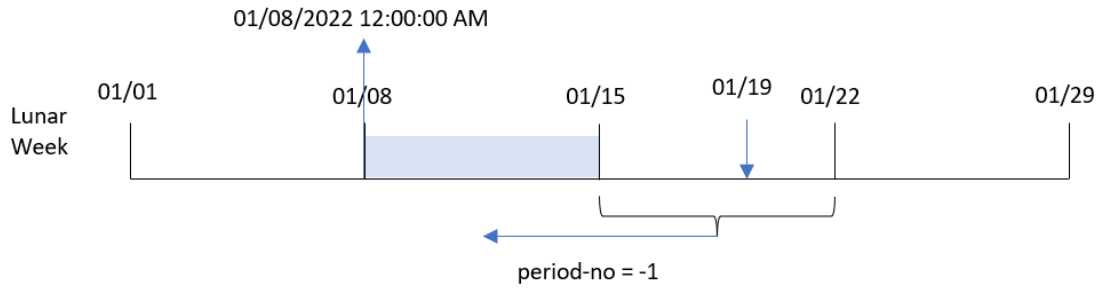
Resultatentabel

| date | previous_lunar_week_start | previous_lunar_week_start_timestamp |
|---|---|---|
| 1/7/2022 | 12/24/2021 | 12/24/2021 12:00:00 AM |
| 1/19/2022 | 01/08/2022 | 1/8/2022 12:00:00 AM |
| 2/5/2022 | 01/29/2022 | 1/29/2022 12:00:00 AM |

| date | previous_lunar_week_start | previous_lunar_week_start_timestamp |
|---|---|---|
| 2/28/2022 | 02/19/2022 | 2/19/2022 12:00:00 AM |
| 3/16/2022 | 03/05/2022 | 3/5/2022 12:00:00 AM |
| 4/1/2022 | 03/19/2022 | 3/19/2022 12:00:00 AM |
| 5/7/2022 | 04/30/2022 | 4/30/2022 12:00:00 AM |
| 5/16/2022 | 05/07/2022 | 5/7/2022 12:00:00 AM |
| 6/15/2022 | 06/04/2022 | 6/4/2022 12:00:00 AM |
| 6/26/2022 | 06/18/2022 | 6/18/2022 12:00:00 AM |
| 7/9/2022 | 07/02/2022 | 7/2/2022 12:00:00 AM |
| 7/22/2022 | 07/09/2022 | 7/9/2022 12:00:00 AM |
| 7/23/2022 | 07/16/2022 | 7/16/2022 12:00:00 AM |
| 7/27/2022 | 07/16/2022 | 7/16/2022 12:00:00 AM |
| 8/2/2022 | 07/23/2022 | 7/23/2022 12:00:00 AM |
| 8/8/2022 | 07/30/2022 | 7/30/2022 12:00:00 AM |
| 8/19/2022 | 08/06/2022 | 8/6/2022 12:00:00 AM |
| 9/26/2022 | 09/17/2022 | 9/17/2022 12:00:00 AM |
| 10/14/2022 | 10/01/2022 | 10/1/2022 12:00:00 AM |
| 10/29/2022 | 10/22/2022 | 10/22/2022 12:00:00 AM |

In dit geval, omdat een `period_no` van -1 werd gebruikt als het offset-argument in de `lunarweekstart()`-functie, identificeert de functie eerst de maanweek waarin de transacties plaatsvinden. Het verschuift dan naar een week eerder en identificeert de eerste milliseconde van die maanweek.

*Diagram van `lunarweekstart()`-functie, voorbeeld period_no*




Transactie 8189 vond plaats op 19 januari. De `lunarweekstart()`-functie identificeert dat de maanweek op 15 januari begint. Daarom begon de vorige maanweek op 8 januari om 12:00:00 AM; dit is de waarde die voor het `previous_lunar_week_start`-veld wordt geretourneerd.

## Voorbeeld 3 – first_week_day

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld. In dit voorbeeld stellen we het begin van de maanweek in op 5 januari.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

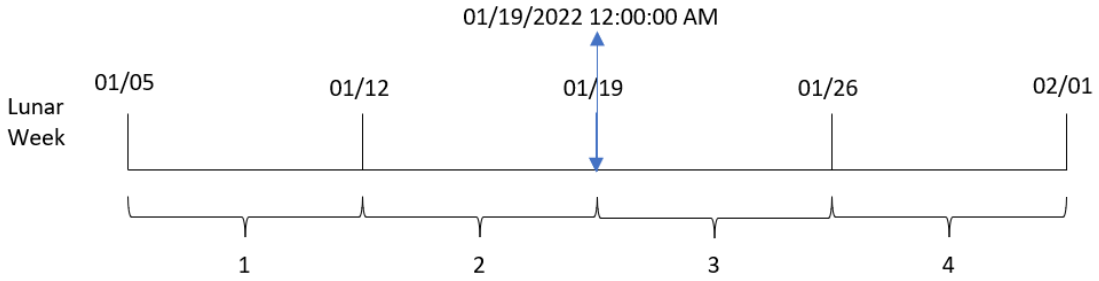
- `date`
- `start_of_week`
- `start_of_week_timestamp`

Resultatentabel

| date | start_of_week | start_of_week_timestamp |
|---|---|---|
| 1/7/2022 | 01/05/2022 | 1/5/2022 12:00:00 AM |
| 1/19/2022 | 01/19/2022 | 1/19/2022 12:00:00 AM |
| 2/5/2022 | 02/02/2022 | 2/2/2022 12:00:00 AM |
| 2/28/2022 | 02/23/2022 | 2/23/2022 12:00:00 AM |
| 3/16/2022 | 03/16/2022 | 3/16/2022 12:00:00 AM |
| 4/1/2022 | 03/30/2022 | 3/30/2022 12:00:00 AM |
| 5/7/2022 | 05/04/2022 | 5/4/2022 12:00:00 AM |
| 5/16/2022 | 05/11/2022 | 5/11/2022 12:00:00 AM |
| 6/15/2022 | 06/15/2022 | 6/15/2022 12:00:00 AM |
| 6/26/2022 | 06/22/2022 | 6/22/2022 12:00:00 AM |
| 7/9/2022 | 07/06/2022 | 7/6/2022 12:00:00 AM |
| 7/22/2022 | 07/20/2022 | 7/20/2022 12:00:00 AM |
| 7/23/2022 | 07/20/2022 | 7/20/2022 12:00:00 AM |
| 7/27/2022 | 07/27/2022 | 7/27/2022 12:00:00 AM |
| 8/2/2022 | 07/27/2022 | 7/27/2022 12:00:00 AM |
| 8/8/2022 | 08/03/2022 | 8/3/2022 12:00:00 AM |
| 8/19/2022 | 08/17/2022 | 8/17/2022 12:00:00 AM |
| 9/26/2022 | 09/21/2022 | 9/21/2022 12:00:00 AM |
| 10/14/2022 | 10/12/2022 | 10/12/2022 12:00:00 AM |
| 10/29/2022 | 10/26/2022 | 10/26/2022 12:00:00 AM |

In dit geval verschuift het begin van het jaar van 1 januari naar 5 januari omdat het `first_week_date`-argument 4 in de `lunarweekstart()`-functie is gebruikt.

*Diagram van de `lunarweekstart()`-functie, voorbeeld met first_week_day*




Transactie 8189 vond plaats op 19 januari. Omdat maanweken op 5 januari beginnen, identificeert de `lunarweekstart()`-functie dat de maanweek waarin 19 januari valt, ook met 19 januari om 12:00:00 AM begint. Daarom is dat de waarde die voor het `start_of_week`-veld wordt geretourneerd.

## Voorbeeld 4 – diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die een tijdstempel retourneert voor het begin van de maanweek waarin de transacties plaatsvonden is gemaakt als meting in een diagramobject van de applicatie.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `date`.

Voeg de volgende metingen toe:

```
=lunarweekstart(date)
```

```
=timestamp(lunarweekstart(date))
```

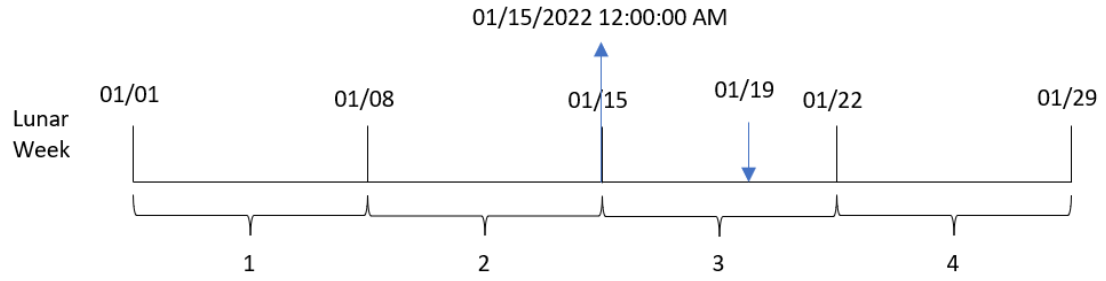
Resultatentabel

| date | =lunarweekstart(date) | =timestamp(lunarweekstart(date)) |
|---|---|---|
| 1/7/2022 | 01/01/2022 | 1/1/2022 12:00:00 AM |
| 1/19/2022 | 01/15/2022 | 1/15/2022 12:00:00 AM |
| 2/5/2022 | 02/05/2022 | 2/5/2022 12:00:00 AM |
| 2/28/2022 | 02/26/2022 | 2/26/2022 12:00:00 AM |
| 3/16/2022 | 03/12/2022 | 3/12/2022 12:00:00 AM |
| 4/1/2022 | 03/26/2022 | 3/26/2022 12:00:00 AM |
| 5/7/2022 | 05/07/2022 | 5/7/2022 12:00:00 AM |
| 5/16/2022 | 05/14/2022 | 5/14/2022 12:00:00 AM |
| 6/15/2022 | 06/11/2022 | 6/11/2022 12:00:00 AM |
| 6/26/2022 | 06/25/2022 | 6/25/2022 12:00:00 AM |
| 7/9/2022 | 07/09/2022 | 7/9/2022 12:00:00 AM |
| 7/22/2022 | 07/16/2022 | 7/16/2022 12:00:00 AM |
| 7/23/2022 | 07/23/2022 | 7/23/2022 12:00:00 AM |
| 7/27/2022 | 07/23/2022 | 7/23/2022 12:00:00 AM |
| 8/2/2022 | 07/30/2022 | 7/30/2022 12:00:00 AM |
| 8/8/2022 | 08/06/2022 | 8/6/2022 12:00:00 AM |
| 8/19/2022 | 08/13/2022 | 8/13/2022 12:00:00 AM |
| 9/26/2022 | 09/24/2022 | 9/24/2022 12:00:00 AM |
| 10/14/2022 | 10/08/2022 | 10/8/2022 12:00:00 AM |
| 10/29/2022 | 10/29/2022 | 10/29/2022 12:00:00 AM |

De meting `start_of_week` wordt gemaakt in het diagramobject met behulp van de functie `lunarweekstart()` en geeft het veld door als het argument van de functie.

De `lunarweekstart()`-functie identificeert in welke maanweek de datumwaarde valt en retourneert een tijdstempel voor de laatste milliseconde van die week.

*Diagram van `lunarweekstart()`-functie, diagramobjectvoorbeeld*




Transactie 8189 vond plaats op 19 januari. De `lunarweekstart()`-functie identificeert dat de maanweek op 15 januari begint. De `start_of_week`-waarde voor die transactie is daarom de eerste milliseconde van die dag, en dat is 15 januari om 12:00:00 AM.

## Voorbeeld 5 – Scenario

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met een set leningsaldo's die wordt geladen in de tabel `Loans`.
- Gegevens bestaan uit lening-id's, het saldo aan het begin van de week en de enkelvoudige rente die op iedere lening per jaar wordt berekend.

De eindgebruiker wil een diagramobject dat per lening-id de huidige rente weergeeft die is opgebouwd op elke lening in de week tot heden.

### Load-script

```
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
];
```

**Resultaten**

**Doe het volgende:**

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel.
2. Voeg de volgende velden als dimensies toe.
   - `loan_id`
   - `start_balance`
3. Maak nu de volgende meting om de opgebouwde rente te berekenen:
   `=start_balance*(rate*(today(1)-lunarweekstart(today(1)))/365)`
4. Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

| loan_id | start_balance | =start_balance*(rate*(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 |

De functie `lunarweekstart()` gebruikt de datum van vandaag als enige argument en retourneert de begindatum van het huidige jaar. Door dat resultaat af te trekken van de huidige datum, retourneert de uitdrukking het aantal dagen dat in deze week tot nu toe is verstreken.

Deze waarde wordt vervolgens vermenigvuldigd met het rentepercentage en gedeeld door 365 om het effectieve rentepercentage voor deze periode te retourneren. Het resultaat wordt vervolgens vermenigvuldigd met het beginsaldo van de lening om de rente terug te betalen die deze week tot nu toe is opgebouwd.

## makedate

Deze functie retourneert een datum die is berekend voor het jaar **YYYY**, de maand **MM** en de dag **DD**.

**Syntaxis:**

```
MakeDate(YYYY [ , MM [ , DD ] ])
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| YYYY | Het jaartal als een geheel getal. |
| MM | De maand als een geheel getal. Als geen maand is vermeld, wordt 1 (januari) verondersteld. |
| DD | De dag als een geheel getal. Als geen dag is vermeld, wordt 1 (de 1ste) verondersteld. |

## Wanneer gebruiken

De `makedate()`-functie wordt normaal gesproken in het script voor gegevensgenerering om een kalender te genereren. Dit kan ook worden gebruikt wanneer het gegevensveld niet rechtstreeks als datum beschikbaar is, maar wat transformaties behoeft om jaar-, maand- en dagcomponenten te extraheren.

Deze voorbeelden gebruiken de datumopmaak MM/DD/YYYY. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` boven aan uw load-script voor gegevens. Wijzig de opmaak in de voorbeelden zodat deze in uw behoeften voorziet.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `makedate(2012)` | Retourneert 01/01/2012. |
| `makedate(12)` | Retourneert 01/01/2012. |
| `makedate(2012,12)` | Retourneert 12/01/2012. |
| `makedate(2012,2,14)` | Retourneert 02/14/2012. |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – Basisvoorbeeld

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties voor 2018 die wordt geladen in een tabel met de naam `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.
- Het veld `transaction_date` wordt gemaakt dat een datum retourneert met de notatie mm/dd/jjjj.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `transaction_year`
- `transaction_month`
- `transaction_day`
- `transaction_date`

Resultatentabel

| transaction_year | transaction_month | transaction_day | transaction_date |
|---|---|---|---|
| 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 |

Het veld `transaction_date` wordt gemaakt in de voorafgaande load-instructie met behulp van de `makedate()`-functie en geeft de jaar-, maand- en dagvelden door als de argumenten van de functie.

De functie combineert en converteert deze waarden in een datumveld en retourneert de resultaten met de notatie van de `DateFormat`-systeemvariabele.

## Voorbeeld 2 – Aangepaste DateFormat

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het veld `transaction_date` wordt gemaakt met de notatie dd/mm/jjjj zonder de `DateFormat`-systeemvariabele aan te passen.

### Load-script

```
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
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `transaction_year`
- `transaction_month`
- `transaction_day`
- `transaction_date`

Resultatentabel

| 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 dit geval is de `makedate()`-functie in de `date()`-functie genest. Het tweede argument van de `date()`-functie stelt de datumnotatie van de `makedate()`-functieresultaten in op de vereiste dd/mm/jjjj.

## Voorbeeld 3 – diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2018 die wordt geladen in een tabel met de naam `Transactions`.
- De transactiedatums worden ingevoerd in twee velden: `year` en `month`.

Maak een diagramobjectmeting, `transaction_date`, die een datum retourneert met de notatie mm/dd/jjjj.

#### Load-script

```
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
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `year`
- `month`

Om de `transaction_date` vast te stellen, maakt u deze meting:

```
=makedate(transaction_year,transaction_month)
```

Resultatentabel

| transaction_year | transaction_month | transaction_date |
|---|---|---|
| 2018 | 08 | 08/01/2018 |
| 2018 | 09 | 09/01/2018 |

De meting `transaction_date` wordt gemaakt in het diagramobject met behulp van de `makedate()`-functie en passeert de jaar- en maandvelden als functieargumenten.

De functie combineert deze waarden vervolgens, evenals de aangenomen dagwaarde van 01. Deze waarden worden geconverteerd in een datumveld en retourneren de resultaten met de indeling van de `DateFormat`-systeemvariabele.

## Voorbeeld 4 – Scenario

Load-script en diagramuitdrukking

#### Overzicht

Maak een kalendergegevensset voor het kalenderjaar 2022.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';

Calendar:
       load
               *
    where year(date)=2022;
load
       date(recno()+makedate(2021,12,31)) as date
AutoGenerate 400;
```

**Resultaten**

Resultatentabel

| 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 |
| Nog 340 rijen |

De `makedate()`-functie maakt een datumwaarde voor 31 december 2021. De `recno()`-functie levert het recordnummer van de huidige record die in de tabel wordt geladen, beginnend bij 1. Daarom heeft de eerste record de datum 1 januari 2022. Iedere achtereenvolgende `recno()` verhoogt deze datum met 1. Deze uitdrukking is verpakt in een `date()`-functie om de waarde om te zetten in een datum. Dit proces

wordt 400 keer herhaald door de `autogenerate`-functie. Als laatste kan een `where`-voorwaarde worden gebruikt om alleen datums uit het jaar 2022. Dit kan door het gebruik van een voorafgaande lading. Dit script genereert een kalender met iedere datum in 2022.

## maketime

Deze functie retourneert een tijd die is berekend voor het uur **hh**, de minuut **mm** en de seconde **ss**.

**Syntaxis:**

```
MakeTime(hh [ , mm [ , ss ] ])
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| hh | Het uur als een geheel getal. |
| mm | De minuut als een geheel getal.<br><br>Als geen minuut is aangegeven, wordt 00 verondersteld. |
| ss | De seconde als een geheel getal.<br><br>Als geen seconde is aangegeven, wordt 00 verondersteld. |

### Wanneer gebruiken

De `maketime()`-functie wordt normaal gesproken in het script voor het genereren van gegevens gebruikt om een tijdveld te genereren. Soms, als het tijdveld wordt afgeleid van invoertekst, kan deze functie worden gebruikt om de tijd op te bouwen met de verschillende componenten.

Deze voorbeelden gebruiken de tijdnotatie `h:mm:ss`. De tijdopmaak wordt opgegeven in de opdracht `SET TimeFormat` boven aan uw load-script voor gegevens. Wijzig de opmaak in de voorbeelden zodat deze in uw behoeften voorziet.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `maketime(22)` | Retourneert 22:00:00. |
| `maketime(22, 17)` | Retourneert 22:17:00. |
| `maketime(22,17,52 )` | Retourneert 22:17:52. |

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege

landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – maketime()

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties die wordt geladen in de tabel `Transactions`.
- De transactietijden worden ingevoerd in drie velden: `hours`, `minutes` en `seconds`.
- Het veld `transaction_time` wordt gemaakt dat de tijd retourneert met de notatie van de `TimeFormat`-systeemvariabele.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- transaction_hour
- transaction_minute
- transaction_second
- transaction_time

Resultatentabel

| transaction_hour | transaction_minute | transaction_second | transaction_time |
|---|---|---|---|
| 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 |

Het veld `transaction_time` wordt gemaakt in de voorafgaande load-instructie met behulp van de `maketime ()`-functie en geeft de uur-, minuut- en secondevelden door als de argumenten van de functie.

De functie combineert en converteert deze waarden in een tijdveld en retourneert de resultaten met de tijdnotatie van de `TimeFormat`-systeemvariabele.

## Voorbeeld 2 – time()-functie

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het veld `transaction_time` wordt gemaakt waarmee we de resultaten kunnen bekijken in een 24-uurs tijdnotatie zonder de `TimeFormat`-systeemvariabele aan te passen.

### Load-script

```
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
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `transaction_hour`
- `transaction_minute`
- `transaction_second`
- `transaction_time`

Resultatentabel

| transaction_hour | transaction_minute | transaction_second | transaction_time |
|---|---|---|---|
| 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 dit geval is de `maketime()`-functie in de `time()`-functie genest. Het tweede argument van de `time()`-functie stelt de notatie van de `maketime()`-functieresultaten in op de vereiste `h:mm:ss`.

## Voorbeeld 3 – diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met een set transacties die wordt geladen in de tabel `Transactions`.
- De transactietijden worden ingevoerd in twee velden: `hours` en `minutes`.
- Het veld `transaction_time` wordt gemaakt dat de tijd retourneert met de notatie van de `TimeFormat`-systeemvariabele.

Maak een diagramobjectmeting, `transaction_time`, die een tijd retourneert met de notatie `h:mm:ss TT`.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `transaction_hour`
- `transaction_minute`

Om de `transaction_time` te berekenen, maakt u deze meting:

```
=maketime(transaction_hour,transaction_minute)
```

Resultatentabel

| transaction_hour | transaction_minute | =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 | 6:43:00 PM |
| 21 | 43 | 9:43:00 PM |

De meting `transaction_time` wordt gemaakt in het diagramobject met behulp van de `maketime()`-functie en passeert de uur- en minuutvelden als functieargumenten.

De functie combineert deze waarden vervolgens en de secondeteller wordt op 00 gezet. Deze waarden worden geconverteerd in een tijdveld en retourneren de resultaten met de notatie van de `TimeFormat`-systeemvariabele.

## Voorbeeld 4 – Scenario

Load-script en diagramuitdrukking

### Overzicht

Maak een kalendergegevensset voor januari 2022, opgesplitst in delen van acht uur.

### Load-script

```
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;
```

### Resultaten

Resultatentabel

| tijdstempel |
| --- |
| 1/1/2022 12:00:00 AM |
| 1/1/2022 8:00:00 AM |
| 1/1/2022 4:00:00 PM |
| 1/2/2022 12:00:00 AM |

| tijdstempel |
| --- |
| 1/2/2022 8:00:00 AM |
| 1/2/2022 4:00:00 PM |
| 1/3/2022 12:00:00 AM |
| 1/3/2022 8:00:00 AM |
| 1/3/2022 4:00:00 PM |
| 1/4/2022 12:00:00 AM |
| 1/4/2022 8:00:00 AM |
| 1/4/2022 4:00:00 PM |
| 1/5/2022 12:00:00 AM |
| 1/5/2022 8:00:00 AM |
| 1/5/2022 4:00:00 PM |
| 1/6/2022 12:00:00 AM |
| 1/6/2022 8:00:00 AM |
| 1/6/2022 4:00:00 PM |
| 1/7/2022 12:00:00 AM |
| 1/7/2022 8:00:00 AM |
| 1/7/2022 4:00:00 PM |
| 1/8/2022 12:00:00 AM |
| 1/8/2022 8:00:00 AM |
| 1/8/2022 4:00:00 PM |
| 1/9/2022 12:00:00 AM |
| Nog 68 rijen |

De aanvankelijke `autogenerate`-functie maakt een kalender met alle datums in januari in de tabel `tmpCalendar`.

Er wordt een tweede tabel gemaakt met drie records. Voor iedere record, wordt `recno() - 1` afgetrokken (waarden 0, 1, 2) en het resultaat wordt vermenigvuldigd met 8. Het resultaat is dat dit de waarden 0, 8 16 genereert. Deze waarden worden gebruikt als de uurparameter in een `maketime()`-functie met minuut- en secondewaarden van 0. Het resultaat is dat de tabel drie tijdvelden bevat: 12:00:00 AM, 8:00:00 AM en 4:00:00 PM.

Deze tabel wordt samengevoegd met de `tmpCalendar`-tabel. Omdat de twee tabellen geen overeenkomende velden bevatten voor de samenvoeging, worden er aan iedere datumrij tijdrijen toegevoegd. Daardoor wordt iedere datumrij nu drie keer met iedere tijdwaarde herhaald.

Als laatste wordt de tabel Kalender gemaakt vanuit een resident-load van de `tmpCalendar`-tabel. De datum- en tijdvelden worden aaneengeschakeld en in de `timestamp()`-functie verpakt om het tijdstempelveld te maken.

De `tmpCalendar`-tabel wordt dan verwijderd.

## makeweekdate

Deze functie retourneert een datum die is berekend voor het jaar, het weeknummer en de dag van de week.

**Syntaxis:**

```
MakeWeekDate(weekyear [, week [, weekday [, first_week_day [, broken_weeks [, reference_day]]]]])
```

**Retourgegevenstypen:** dubbele waarde

De `makeweekdate()`-functie is beschikbaar zowel als een script- als een diagramfunctie. De functie berekent de datum op basis van de parameters die in de functie worden gepasseerd.

Argumenten

| Argument | Beschrijving |
|---|---|
| **weekyear** | Het jaar dat is gedefinieerd door de functie `WeekYear()` voor de specifieke datum is het jaar waartoe het weeknummer behoort.<br><br>*Het weekjaar kan in sommige gevallen afwijken van het kalenderjaar, bijvoorbeeld als week 1 start in december van het voorgaande jaar.* |
| **week** | Het weeknummer dat is gedefinieerd door de functie `Week()` voor de specifieke datum.<br><br>Als er geen weeknummer is aangegeven, wordt 1 verondersteld. |

| Argument | Beschrijving |
|---|---|
| **weekday** | De dag van de week die is gedefinieerd door de functie `WeekDay()` voor de desbetreffende datum. 0 is de eerste dag van de week en 6 is de laatste dag van de week.<br><br>Als er geen dag van de week is aangegeven, wordt 0 verondersteld.<br><br>*Hoewel met 0 altijd de eerste dag van de week en met 6 altijd de laatste dag van de week wordt bedoeld, wordt de overeenkomende weekdag bepaald door de parameter **first_week_day**. Bij weglating wordt de waarde van de variabele **FirstWeekDay** gebruikt.*<br><br>Als er gebroken weken worden gebruikt, samen met een onmogelijk combinatie van parameters, leidt dit mogelijk tot een resultaat dat niet tot het gekozen jaar behoort.<br><br>**Voorbeeld:**<br><br>`MakeWeekDate(2021,1,0,6,1)`<br>Retourneert 'Dec 27 2020' omdat deze dag de eerste dag (zondag) van de opgegeven week is. Jan 1 2021 was een vrijdag. |
| **first_week_day** | Geeft de dag op waarop de week begint. Bij weglating wordt de waarde van de variabele **FirstWeekDay** gebruikt.<br><br>De mogelijke **first_week_day**-waarden zijn 0 voor maandag, 1 voor dinsdag, 2 voor woensdag, 3 voor donderdag, 4 voor vrijdag, 5 voor zaterdag en 6 voor zondag.<br><br>Zie *FirstWeekDay (page 244)* voor meer informatie over de systeemvariabele. |
| **broken_weeks** | Als u **broken_weeks** (gebroken weken) niet specificeert, dan wordt de waarde van variabele **BrokenWeeks** gebruikt om te definiëren of weken al dan niet gebroken zijn. |
| **reference_day** | Als u **reference_day** niet opgeeft, wordt de waarde van variabele **ReferenceDay** gebruikt om te definiëren welke dag in januari wordt ingesteld als referentiedag voor het definiëren van week 1. |

## Wanneer gebruiken

De `makeweekdate()`-functie wordt normaliter gebruikt in het script voor het genereren van gegevens om een lijst met datums te genereren of om datums op te bouwen wanneer het jaar, de week of dag van de week in de invoergegevens zijn verstrekt.

De volgende voorbeelden gaan uit van het volgende:

```
SET FirstWeekDay=0;
SET BrokenWeeks=0;
SET ReferenceDay=4;
```

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `makeweekdate(2014,6,6)` | retourneert 02/09/2014 |
| `makeweekdate(2014,6,1)` | retourneert 02/04/2014 |
| `makeweekdate(2014,6)` | retourneert 02/03/2014 (er wordt uitgegaan van weekdag 0) |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – Inclusief dag

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een totale verkoop per week voor 2022 in de tabel `Sales`.
- De transactiedatums worden ingevoerd in drie velden: `year`, `week` en `sales`.
- Een voorafgaande lading die wordt gebruikt om een meting te maken, `end_of_week`, met de `makeweekdate()`-functie om de datum voor de vrijdag van die week te retourneren met de notitie mm/dd/jjjj.

Om aan te tonen dat de geretourneerde datum een vrijdag is, wordt de `end_of_week`-uitdrukking ook verpakt in de `weekday()`-functie om de dag van de week te laten zien.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `transaction_year`
- `transaction_week`
- `end_of_week`
- `week_day`

Resultatentabel

| transaction_year | transaction_week | end_of_week | week_day |
|---|---|---|---|
| 2022 | 01 | 01/07/2022 | Vr |
| 2022 | 02 | 01/14/2022 | Vr |
| 2022 | 03 | 01/21/2022 | Vr |
| 2022 | 04 | 01/28/2022 | Vr |
| 2022 | 05 | 02/04/2022 | Vr |
| 2022 | 06 | 02/11/2022 | Vr |
| 2022 | 07 | 02/18/2022 | Vr |

Het veld `end_of_week` wordt gemaakt in de vorige load-instructie met behulp van de functie `makeweekdate()`. De velden `transaction_year` en `transaction_week` worden als jaar- en weekargumenten in de functie gepasseerd. Er wordt een waarde van 4 gebruikt voor het dagargument.

De functie combineert en converteert deze waarden in een datumveld en retourneert de resultaten met de notatie van de `DateFormat`-systeemvariabele.

De `makeweekdate()`-functie en de bijbehorende argumenten worden ook verpakt in een `weekday()`-functie om het `week_day`-veld te retourneren, en zoals in de bovenstaande tabel te zien is, toont het `week_day`-veld dat deze datums op een vrijdag vallen.

## Voorbeeld 2 – Exclusief dag

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met een totale verkoop per week voor 2022 in de tabel `Sales`.
- De transactiedatums worden ingevoerd in drie velden: `year`, `week` en `sales`.
- Een voorafgaande lading die wordt gebruikt om een meting te maken, `first_day_of_week`, met behulp van de `makeweekdate()`-functie. Dit retourneert de datum voor de maandag van die week met de notatie mm/dd/jjjj.

Om aan te tonen dat de geretourneerde datum een maandag is, wordt de `first_day_of_week`-uitdrukking ook verpakt in de `weekday()`-functie om de dag van de week te laten zien.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- transaction_year
- transaction_week
- first_day_of_week
- week_day

Resultatentabel

| transaction_year | transaction_week | first_day_of_week | week_day |
|---|---|---|---|
| 2022 | 01 | 01/03/2022 | Ma |
| 2022 | 02 | 01/10/2022 | Ma |
| 2022 | 03 | 01/17/2022 | Ma |
| 2022 | 04 | 01/24/2022 | Ma |
| 2022 | 05 | 01/31/2022 | Ma |
| 2022 | 06 | 02/07/2022 | Ma |
| 2022 | 07 | 02/14/2022 | Ma |

Het veld `first_day_of_week` wordt gemaakt in de vorige load-instructie met behulp van de functie `makeweekdate()`. De parameters `transaction_year` en `transaction_week` worden als functie-argumenten gepasseerd en de dagparameter blijft leeg.

De functie combineert en converteert deze waarden in een datumveld en retourneert de resultaten met de notatie van de `DateFormat`-systeemvariabele.

De `makeweekdate()`-functie en de bijbehorende argumenten worden ook verpakt in een `weekday()`-functie om het `week_day`-veld te retourneren. Zoals te zien is in de bovenstaande tabel, retourneert het veld `week_day` in alle gevallen maandag omdat deze parameter leeg is in de functie `makeweekdate()`, waardoor de standaardwaarde `0` (eerste dag van de week) wordt en de eerste dag van de week door de systeemvariabele `FirstWeekDay` wordt ingesteld op maandag.

## Voorbeeld 3 – diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met een totale verkoop per week voor 2022 in de tabel `Sales`.
- De transactiedatums worden ingevoerd in drie velden: `year`, `week` en `sales`.

In dit voorbeeld wordt een diagramobject gebruikt om een meting te maken die equivalent is aan de `end_of_week`-berekening uit het eerste voorbeeld. Deze meting gebruikt de `makeweekdate()`-functie om de datum te retourneren voor de vrijdag van die week met de notatie mm/dd/jjjj.

Om te bewijzen dat de geretourneerde datum een vrijdag is, wordt er een tweede meting gemaakt om de dag van de week te retourneren.

### Load-script

```
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
];
```

### Resultaten

### Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:
    - `transaction_year`
    - `transaction_week`
2. Om de berekening uit te voeren die equivalent is aan die uit het `end_of_week`-veld uit het eerste voorbeeld, maakt u de volgende meting:
   `=makeweekdate(transaction_year,transaction_week,4)`
3. Maak de volgende meting om de dag van de week per transactie te berekenen:
   `=weekday(makeweekdate(transaction_year,transaction_week,4))`

Resultatentabel

| transaction_ year | transaction_ week | =makeweekdate (transaction_ year,transaction_week,4) | =weekday(makeweekdate (transaction_year,transaction_ week,4)) |
|---|---|---|---|
| 2022 | 01 | 01/07/2022 | Vr |
| 2022 | 02 | 01/14/2022 | Vr |
| 2022 | 03 | 01/21/2022 | Vr |
| 2022 | 04 | 01/28/2022 | Vr |
| 2022 | 05 | 02/04/2022 | Vr |

| transaction_ year | transaction_ week | =makeweekdate (transaction_ year,transaction_week,4) | =weekday(makeweekdate (transaction_year,transaction_ week,4)) |
|---|---|---|---|
| 2022 | 06 | 02/11/2022 | Vr |
| 2022 | 07 | 02/18/2022 | Vr |

Er wordt een equivalent veld voor `end_of_week` in het diagramobject gemaakt als een meting met behulp van de functie `makeweekdate()`. De velden `transaction_year` en `transaction_week` worden als jaar- en weekargumenten gepasseerd. Er wordt een waarde van 4 gebruikt voor het dagargument.

De functie combineert en converteert deze waarden in een datumveld en retourneert de resultaten met de notatie van de `DateFormat`-systeemvariabele.

De `makeweekdate()`-functie en de bijbehorende argumenten worden ook verpakt in een `weekday()`-functie om een berekening te retourneren die equivalent is aan die van het `week_day`-veld uit het eerste voorbeeld. Zoals te zien in de bovenstaande tabel, toont de laatste kolom aan de rechterkant dat deze datums op een vrijdag vallen.

## Voorbeeld 4 – Scenario

Load-script en diagramuitdrukking

### Overzicht

In dit voorbeeld wordt een lijst met datums gemaakt met alle vrijdagen in 2022.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

### Load-script

```
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;
```

### Resultaten

Resultatentabel

| date | weekday |
|---|---|
| 01/07/2022 | Vr |
| 01/14/2022 | Vr |
| 01/21/2022 | Vr |
| 01/28/2022 | Vr |
| 02/04/2022 | Vr |
| 02/11/2022 | Vr |
| 02/18/2022 | Vr |
| 02/25/2022 | Vr |
| 03/04/2022 | Vr |
| 03/11/2022 | Vr |
| 03/18/2022 | Vr |
| 03/25/2022 | Vr |
| 04/01/2022 | Vr |
| 04/08/2022 | Vr |
| 04/15/2022 | Vr |
| 04/22/2022 | Vr |
| 04/29/2022 | Vr |
| 05/06/2022 | Vr |
| 05/13/2022 | Vr |
| 05/20/2022 | Vr |
| 05/27/2022 | Vr |
| 06/03/2022 | Vr |
| 06/10/2022 | Vr |
| 06/17/2022 | Vr |
| Nog 27 rijen | |

De `makeweekdate()`-functie vindt iedere vrijdag in 2022. Door gebruik van een weekparameter van -2 worden geen datums overgeslagen. Als laatste maakt een voorafgaande lading een extra `weekday`-veld voor duidelijkheid waarmee wordt aangetoond dat iedere `date`-waarde een vrijdag is.

## minute

Deze functie retourneert een geheel getal dat de minuut representeert als de breuk van de **expression** wordt geïnterpreteerd als een datum volgens de standaardinterpretatie van getallen.

**Syntaxis:**

```
minute(expression)
```

**Retourgegevenstypen:** geheel getal

### Wanneer gebruiken

De `minute()`-functie is handig als u aggregaties per minuut wilt vergelijken. U kunt de functie bijvoorbeeld gebruiken als u de verdeling van het aantal activiteiten per minuut wilt bekijken.

Deze dimensies kunnen in het load-script worden gemaakt door de functie te gebruiken om een veld in een masterkalendertabel te maken. Ze zouden ook rechtstreeks in een diagram kunnen worden gebruikt als een berekende dimensie.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `minute ( '09:14:36' )` | Retourneert 14. |
| `minute ( '0.5555' )` | Retourneert 19 ( omdat 0,5555 = 13:19:55 ) |

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – Variabele (script)

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met transacties per tijdstempel die wordt geladen in de tabel `Transactions`.
- De standaard `TimeStamp`-systeemvariabele (`M/D/YYYY h:mm:ss[.fff] TT`) wordt gebruikt.
- Maak een veld, `minute` dat berekent wanneer de transacties hebben plaatsgevonden.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `timestamp`
- `minute`

Resultatentabel

| tijdstempel | 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 |

De waarden in het `minute`-veld zijn gemaakt met de `minute()`-functie en geven de `timestamp` door als de uitdrukking in het voorafgaande load-instructie.

## Voorbeeld 2 – Diagramobject (diagram)

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- De standaard `TimeStamp`-systeemvariabele (`M/D/YYYY h:mm:ss[.fff] TT`) wordt gebruikt.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De `minute`-waarden worden berekend via een meting in een diagramobject.

### Load-script

```
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
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `timestamp`.

Maak de volgende meting:

```
=minute(timestamp)
```

Resultatentabel

| tijdstempel | 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 |

De waarden voor `minute` worden gemaakt met behulp van de `minute()`-functie en geven de `timestamp` door als de uitdrukking in een meting voor het diagramobject.

## Voorbeeld 3 – Scenario

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met tijdstempels die wordt gegenereerd om invoeren in een ticketbarrière te vertegenwoordigen.
- Informatie bij iedere `timestamp` en de bijbehorende `id`, die wordt geladen in de tabel `Ticket_Barrier_Tracker`.
- De standaard `TimeStamp`-systeemvariabele (`M/D/YYYY h:mm:ss[.fff] TT`) wordt gebruikt.

De gebruiker wil een diagramobject dat het aantal barrière-invoeren per minuut weergeeft.

### Load-script

```
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;
```

### Resultaten

### Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel.
2. Maak een berekende dimensie met behulp van de volgende uitdrukking:
   `=minute(timestamp)`
3. Voeg de volgende aggregatiemeting toe om het totaalaantal invoeren te berekenen:
   `=count(id)`
4. Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

| minute(timestamp) | =count(id) |
|---|---|
| 0 | 174 |
| 1 | 171 |

| minute(timestamp) | =count(id) |
|---|---|
| 2 | 175 |
| 3 | 165 |
| 4 | 188 |
| 5 | 176 |
| 6 | 158 |
| 7 | 187 |
| 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 |
| Nog 35 rijen | |

## month

Deze functie retourneert een duale waarde: een maandnaam zoals vastgelegd in de omgevingsvariabele **MonthNames**, en een geheel getal tussen 1-12. De maand wordt berekend volgens de datuminterpretatie van de uitdrukking, in overeenstemming met de standaardinterpretatie van getallen.

De functie retourneert de naam van de maand in de indeling van de systeemvariabele `MonthName` voor een specifieke datum. Dit wordt vaak gebruikt om een dagveld te maken als een dimensie in een hoofdagenda.

**Syntaxis:**

```
month(expression)
```

**Retourgegevenstypen:** geheel getal

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `month( 2012-10-12 )` | Retourneert Okt |
| `month( 35648 )` | retourneert Aug, omdat 35648 = 1997-08-06 |

## Voorbeeld 1 - gegevensverzameling DateFormat (script)

Load-script en resultaten

### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensverzameling met de naam `Master_Calendar`. De `DateFormat` systeemvariabele is ingesteld op DD/MM/JJJJ.
- Een voorafgaande lading maakt een extra veld met de naam `month_name` met behulp van de `month()`functie.
- Een extra veld met de naam `long_date` met behulp van de `date()` functie voor het weergeven van de volledige datum.

### Load-script

```
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
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `long_date`
- `month_name`

Resultatentabel

| date | 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 | Okt |
| 03/11/2022 | 03-November- 2022 | Nov |

De naam van de maand wordt juist geëvalueerd door de `month()` functie in het script.

## Voorbeeld 2 – ANSI-datums (script)

Load-script en resultaten

#### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensverzameling met de naam `Master_Calendar`. De DateFormat systeemvariabele DD/MM/JJJJ wordt gebruikt. De datums die zijn opgenomen in de gegevensverzameling hebben echter een standaard ANSI-datumnotatie.

- Een voorafgaande lading maakt een extra veld met de naam `month_name` met behulp van de month() functie.
- Een extra veld met de naam `long_date` met behulp van de date() functie voor het weergeven van de volledige datum.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `long_date`
- `month_name`

Resultatentabel

| date | long_date | month_name |
|---|---|---|
| 03/11/2022 | 11-March- 2022 | 11 |
| 03/12/2022 | 11-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 |

| date | long_date | month_name |
|---|---|---|
| 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 |

De naam van de maand wordt juist geëvalueerd door de `month()` functie in het script.

## Voorbeeld 3 – Datums zonder notatie (script)

Load-script en resultaten

### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensverzameling met de naam `Master_Calendar`. De `DateFormat` systeemvariabele DD/MM/JJJJ wordt gebruikt.
- Een voorafgaande lading maakt een extra veld met de naam `month_name` met behulp van de `month()` functie.
- De originele datum zonder notatie, met de naam `unformatted_date`.
- Een extra veld met de naam `long_date` met behulp van de `date()` functie voor het weergeven van de volledige datum.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `unformatted_date`
- `long_date`
- `month_name`

Resultatentabel

| 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 | Okt |
| 45068 | 03-November- 2022 | Nov |

De naam van de maand wordt juist geëvalueerd door de `month()` functie in het script.

## Voorbeeld 4 – Vervalmaand berekenen

Load-script en diagramuitdrukking

### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensverzameling met bestellingen die in maart zijn geplaatst, met de naam `Subscriptions`. De tabel bevat drie velden.

- id
- order_date
- amount

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `order_date`.

Maak deze meting om de maand te berekenen waarin een bestelling verloopt: `=month(order_date+180)`.

Resultatentabel

| 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 | Okt |
| 03/06/2022 | Nov |
| 03/07/2022 | Dec |
| 03/08/2022 | Jan |

| order_date | =month(order_date+180) |
|---|---|
| 03/09/2022 | Mar |
| 03/10/2022 | Apr |
| 03/11/2022 | May |

De `month()` functie bepaalt op correcte wijze dat een bestelling die op 11 maart is geplaatst in juli vervalt.

## monthend

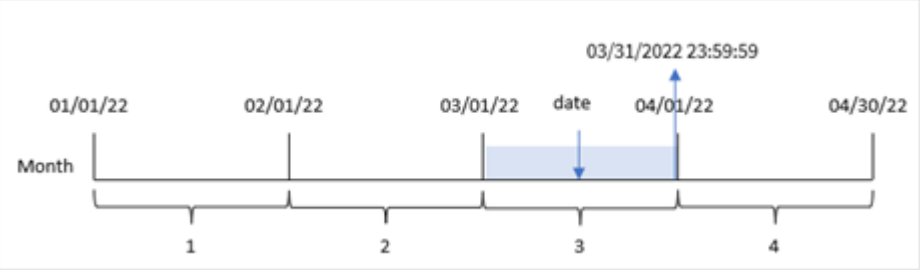
Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de laatste milliseconde van de laatste dag van de maand die `date` bevat. De standaarduitvoernotatie is de `DateFormat` die is ingesteld in het script.

**Syntaxis:**

```
MonthEnd(date[, period_no])
```

Met andere woorden, de functie `monthend()` bepaalt in welke maand de datum valt. Hij retourneert vervolgens een tijdstempel, in datumnotatie, voor de laatste milliseconde van die maand.

*Diagram van `monthend`-functie.*




**Wanneer gebruiken**

De functie `monthend()` wordt gebruikt als onderdeel van een uitdrukking als u wilt dat in de berekening een deel van de maand wordt gebruikt dat nog niet heeft plaatsgevonden. Bijvoorbeeld als u de totale rente wilt berekenen die nog niet is berekend gedurende de maand.

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **date** | De te evalueren datum of tijdstempel. |
| **period_no** | **period_no** is een geheel getal dat, indien het 0 is of wordt weggelaten, de maand aangeeft die **date** bevat. Negatieve waarden in **period_no** geven voorafgaande maanden aan en positieve waarden geven volgende maanden aan. |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `monthend('02/19/2012')` | Retourneert 02/29/2012 23:59:59. |
| `monthend('02/19/2001', -1)` | Retourneert 01/31/2001 23:59:59. |

## Voorbeeld 1 – Basisvoorbeeld

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties voor 2022 wordt in de tabel `Transactions` geladen.
- Een datumveld dat wordt weergegeven in de `DateFormat`-systeemvariabele met de notatie `MM/DD/YYYY`.
- Een voorgaande load-instructie met:
  - De `monthend()`-functie die is ingesteld als het veld `end_of_month`.
  - De `timestamp`-functie die is ingesteld als het veld `end_of_month_timestamp`.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date
- end_of_month
- end_of_month_timestamp

Resultatentabel

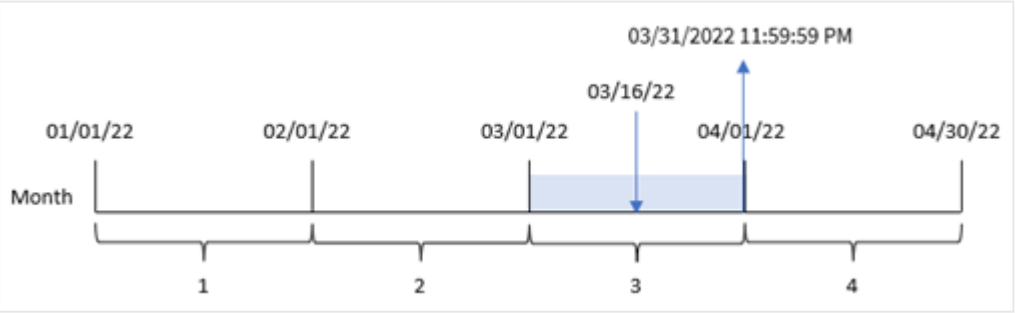
| id | date | end_of_month | end_of_month_timestamp |
|---|---|---|---|
| 8188 | 1/7/2022 | 01/31/2022 | 1/31/2022 11:59:59 PM |
| 8189 | 1/19/2022 | 01/31/2022 | 1/31/2022 11:59:59 PM |
| 8190 | 2/5/2022 | 02/28/2022 | 2/28/2022 11:59:59 PM |
| 8191 | 2/28/2022 | 02/28/2022 | 2/28/2022 11:59:59 PM |
| 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 |

| id | date | end_of_month | end_of_month_timestamp |
|---|---|---|---|
| 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 | 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 |

Het veld '`end_of_month`' wordt gemaakt in de voorafgaande load-instructie met behulp van de functie `monthend()` en geeft het datumveld door als het argument van de functie.

De functie `monthend()` identificeert in welke maand de datumwaarde valt en retourneert een tijdstempel voor de laatste milliseconde van die maand.

*Diagram van de `monthend`-functie waarbij maart de geselecteerde maand is.*




Transactie 8192 vond plaats op 16 maart. De functie `monthend()` retourneert de laatste milliseconde van die maand, namelijk 31 maart om 11:59:59 PM.

## Voorbeeld 2 – period_no

Load-script en resultaten

### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld wordt een veld gemaakt, `previous_month_end`, dat de tijdstempel voor het eind van de maand retourneert voordat de transactie plaatsvond.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

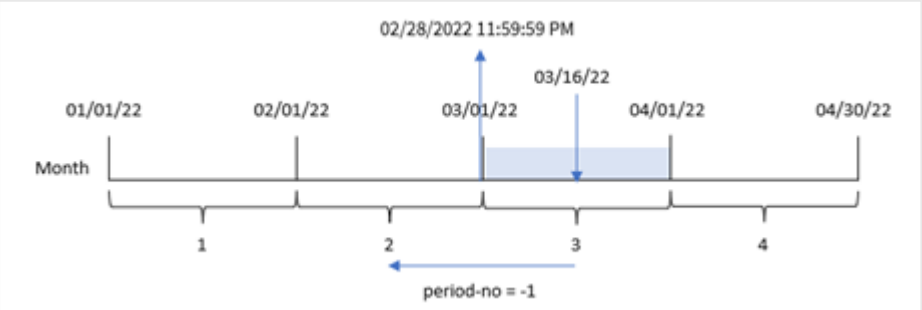
- `id`
- `date`
- `previous_month_end`
- `previous_month_end_timestamp`

Resultatentabel

| 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 |

De `monthend()`-functie identificeert eerst de maand waarin de transacties plaatsvinden als een `period_no` van `-1` wordt gebruikt als het offset-argument. Het verschuift dan naar een maand eerder en identificeert de laatste milliseconde van die maand.

*Diagram van functie `monthend` met de variabele period_no.*



Transactie 8192 vond plaats op 16 maart. De `monthend()`-functie stelt vast dat de maand voor de transactie februari was. Vervolgens wordt de laatste milliseconde van die maand geretourneerd: 28 februari om 11:59:59 PM.

## Voorbeeld 3 – Diagramvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is de gegevensverzameling niet gewijzigd en wordt in de app geladen. De berekening die een tijdstempel retourneert voor het einde van de maand waarin de transacties plaatsvonden is gemaakt als meting in een diagramobject van de app.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `id`

Maak de volgende metingen om de einddatum van de maand waarin een transactie plaatsvindt te berekenen:

- `=monthend(date)`
- `=timestamp(monthend(date))`

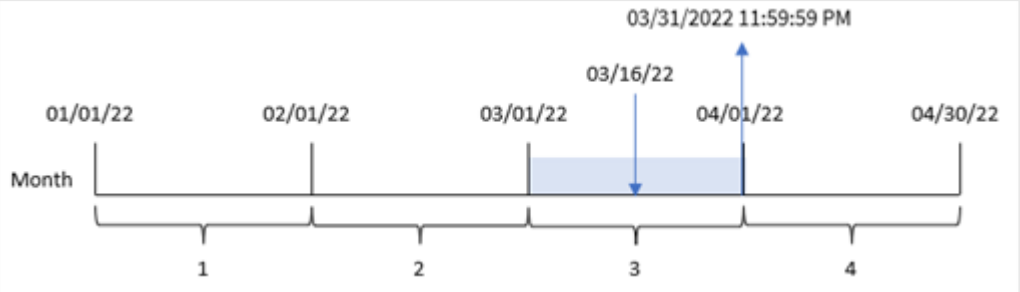
Resultatentabel

| 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 |

De meting `end_of_month` wordt gemaakt in het diagram met behulp van de functie `monthend()` en geeft het veld door als het argument van de functie.

De functie `monthend()` identificeert in welke maand de datumwaarde valt en retourneert een tijdstempel voor de laatste milliseconde van die maand.

*Diagram van functie `monthend` met de variabele period_no.*




Transactie 8192 vond plaats op 16 maart. De functie `monthend()` retourneert de laatste milliseconde van die maand, namelijk 31 maart om 11:59:59 PM.

## Voorbeeld 4 – Scenario

Load-script en resultaten

### Overzicht

In dit voorbeeld wordt een gegevensset geladen in de tabel '`Employee_Expenses`'. De tabel bevat de volgende velden:

- Werknemers-id's
- Werknemersnamen
- De gemiddelde dagelijkse onkostendeclaraties van elke werknemer.

De eindgebruiker wil graag een diagram dat per werknemer-id en werknemersnaam de geschatte onkostendeclaraties voor de rest van de maand weergeeft.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- employee_id
- employee_name

Maak deze meting om de opgebouwde rente te berekenen:

```
=floor(monthend(today(1),0)-today(1))*avg_daily_claim
```

*Deze meting is dynamisch en produceert verschillende tabelresultaten, afhankelijk van de datum waarop u de gegevens laadt.*

Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

| employee_id | employee_name | =floor(monthend(today(1),0)-today(1))*avg_daily_claim |
|---|---|---|
| 182 | Mark | $30.00 |
| 183 | Deryck | $25.00 |
| 184 | Dexter | $25.00 |
| 185 | Sydney | $54.00 |
| 186 | Agatha | $36.00 |

De `monthend()`-functie retourneert de einddatum van de huidige maand door de datum van vandaag als enige argument te gebruiken. Door vervolgens de datum van vandaag af te trekken van de einddatum van de maand, retourneert de uitdrukking het aantal resterende dagen in deze maand.

Deze waarde wordt vervolgens vermenigvuldigd met de gemiddelde dagelijkse onkostendeclaratie van elke werknemer om de geschatte waarde te berekenen van de declaraties die elke werknemer naar verwachting in de resterende maand zal indienen.

## monthname

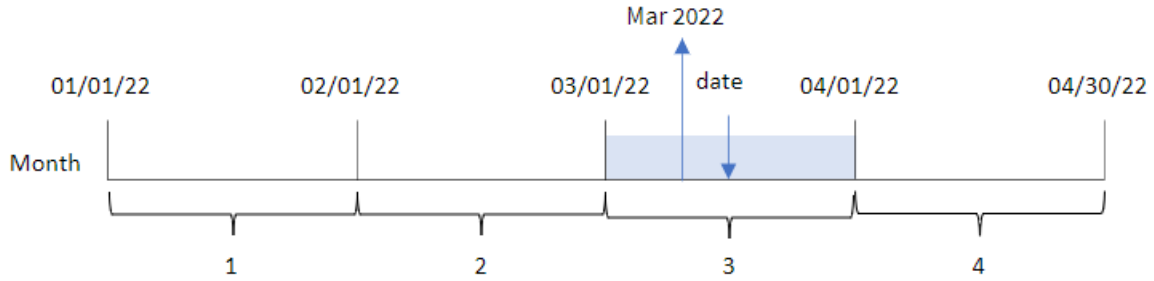
Deze functie retourneert de maand (opgemaakt volgens de scriptvariabele **MonthNames**) en het jaar met een onderliggende numerieke waarde die overeenkomt met een tijdstempel van de eerste milliseconde van de eerste dag van de maand.

**Syntaxis:**

```
MonthName(date[, period_no])
```

**Retourgegevenstypen:** dubbele waarde

*Diagram van functie monthname*




Argumenten

| Argument | Beschrijving |
|---|---|
| **date** | De te evalueren datum of tijdstempel. |
| **period_no** | **period_no** is een geheel getal dat, indien het 0 is of wordt weggelaten, de maand aangeeft die **date** bevat. Negatieve waarden in **period_no** geven voorafgaande maanden aan en positieve waarden geven volgende maanden aan. |

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `monthname('10/19/2013')` | Retourneert Oct 2013 |
| `monthname('10/19/2013', -1)` | Retourneert Sep 2013 |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – Basisvoorbeeld

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties voor 2022 die wordt geladen in de tabel `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.
- Het maken van een veld, `transaction_month`, dat de maand retourneert waarin de transacties plaatsvonden.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

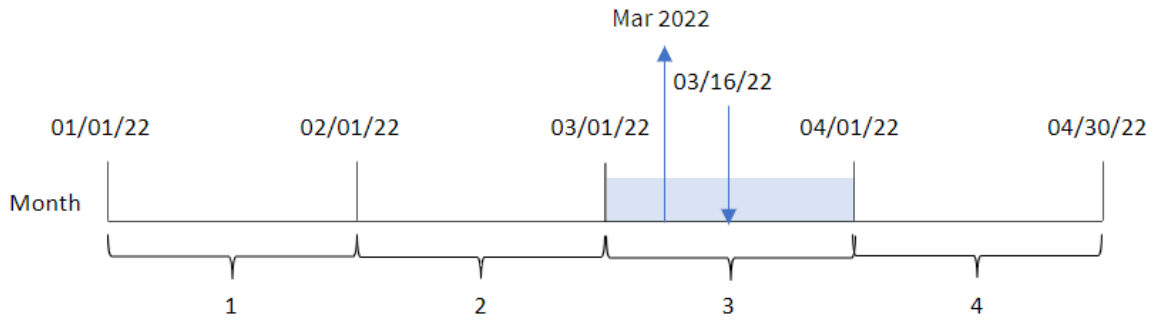
- `date`
- `transaction_month`

Resultatentabel

| date | transaction_month |
|---|---|
| 1/7/2022 | januari 2022 |
| 1/19/2022 | januari 2022 |
| 2/5/2022 | februari 2022 |
| 2/28/2022 | februari 2022 |
| 3/16/2022 | maart 2022 |
| 4/1/2022 | april 2022 |
| 5/7/2022 | mei 2022 |
| 5/16/2022 | mei 2022 |
| 6/15/2022 | juni 2022 |
| 6/26/2022 | juni 2022 |
| 7/9/2022 | juli 2022 |
| 7/22/2022 | juli 2022 |
| 7/23/2022 | juli 2022 |
| 7/27/2022 | juli 2022 |
| 8/2/2022 | augustus 2022 |
| 8/8/2022 | augustus 2022 |
| 8/19/2022 | augustus 2022 |
| 9/26/2022 | september 2022 |
| 10/14/2022 | oktober 2022 |
| 10/29/2022 | oktober 2022 |

Het `transaction_month`-veld wordt gemaakt in de voorafgaande load-instructie met behulp van de `monthname()`-functie en geeft het `date`-veld door als het argument van de functie.

*Diagram van functie monthname, basisvoorbeeld*




De functie `monthname()` identificeert dat transactie 8192 plaatsvond in maart 2022 en retourneert deze waarde met behulp van de systeemvariabele `MonthNames`.

## Voorbeeld 2 – period_no

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde inline gegevensset en het scenario uit het eerste voorbeeld.
- Het maken van een veld, `transaction_previous_month`, dat de tijdstempel voor het einde van de maand retourneert voordat de transactie plaatsvond.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `transaction_previous_month`

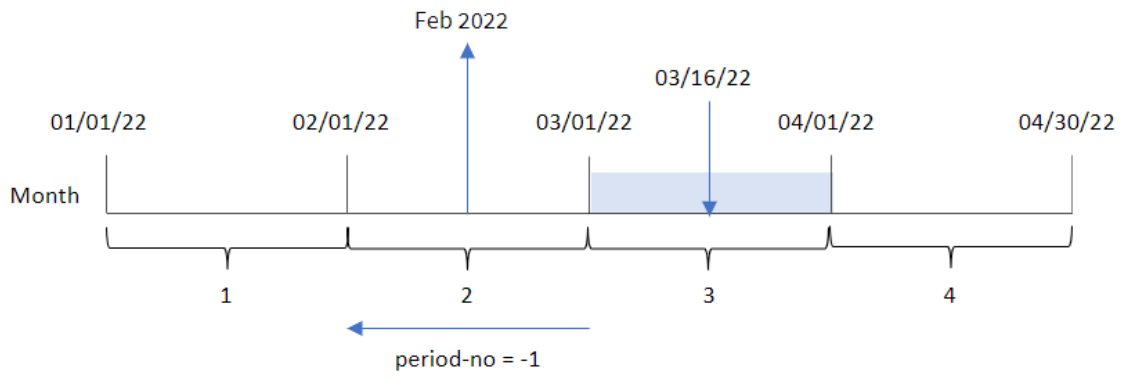
Resultatentabel

| date | transactie_vorige_maand |
|---|---|
| 1/7/2022 | december 2021 |
| 1/19/2022 | december 2021 |
| 2/5/2022 | januari 2022 |
| 2/28/2022 | januari 2022 |
| 3/16/2022 | februari 2022 |
| 4/1/2022 | maart 2022 |
| 5/7/2022 | april 2022 |
| 5/16/2022 | april 2022 |
| 6/15/2022 | mei 2022 |
| 6/26/2022 | mei 2022 |
| 7/9/2022 | juni 2022 |
| 7/22/2022 | juni 2022 |
| 7/23/2022 | juni 2022 |
| 7/27/2022 | juni 2022 |
| 8/2/2022 | juli 2022 |
| 8/8/2022 | juli 2022 |
| 8/19/2022 | juli 2022 |

| date | transactie_vorige_maand |
|---|---|
| 9/26/2022 | augustus 2022 |
| 10/14/2022 | september 2022 |
| 10/29/2022 | september 2022 |

In dit geval, omdat een `period_no` van -1 werd gebruikt als het offset-argument in de `monthname()`-functie, identificeert de functie eerst de maand waarin de transacties plaatsvinden. Het verschuift dan naar een maand eerder en retourneert de naam van de maand en het jaar.

*Diagram van functie monthname, voorbeeld period_no*




Transactie 8192 vond plaats op 16 maart. De functie `monthname()` identificeert dat de maand voordat de transactie plaatsvond februari was en retourneert de maand, in de systeemvariabele-indeling `MonthNames`, samen met het jaar 2022.

## Voorbeeld 3 – diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde inline gegevensset en scenario als het eerste voorbeeld. In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die een tijdstempel retourneert voor het einde van de maand waarin de transacties plaatsvonden is gemaakt als meting in een diagramobject van de toepassing.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:`date`.

Maak de volgende meting:

```
=monthname(date)
```

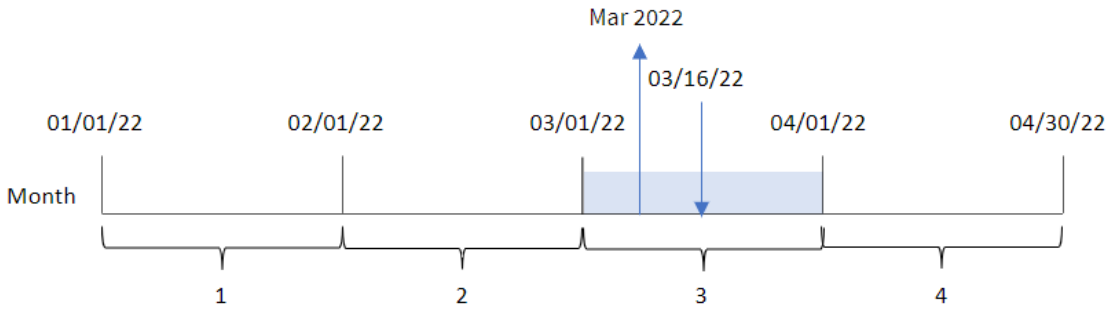
Resultatentabel

| date | =maandnaam(datum) |
|---|---|
| 1/7/2022 | januari 2022 |
| 1/19/2022 | januari 2022 |
| 2/5/2022 | februari 2022 |
| 2/28/2022 | februari 2022 |
| 3/16/2022 | maart 2022 |
| 4/1/2022 | april 2022 |
| 5/7/2022 | mei 2022 |
| 5/16/2022 | mei 2022 |
| 6/15/2022 | juni 2022 |

| date | =maandnaam(datum) |
|---|---|
| 6/26/2022 | juni 2022 |
| 7/9/2022 | juli 2022 |
| 7/22/2022 | juli 2022 |
| 7/23/2022 | juli 2022 |
| 7/27/2022 | juli 2022 |
| 8/2/2022 | augustus 2022 |
| 8/8/2022 | augustus 2022 |
| 8/19/2022 | augustus 2022 |
| 9/26/2022 | september 2022 |
| 10/14/2022 | oktober 2022 |
| 10/29/2022 | oktober 2022 |

De meting `month_name` wordt gemaakt in het diagramobject met behulp van de functie `monthname()` en geeft het veld `date` door als het argument van de functie.

*Diagram van functie monthname, diagramobjectvoorbeeld*




De functie `monthname()` identificeert dat transactie 8192 plaatsvond in maart 2022 en retourneert deze waarde met behulp van de systeemvariabele `MonthNames`.

## monthsend

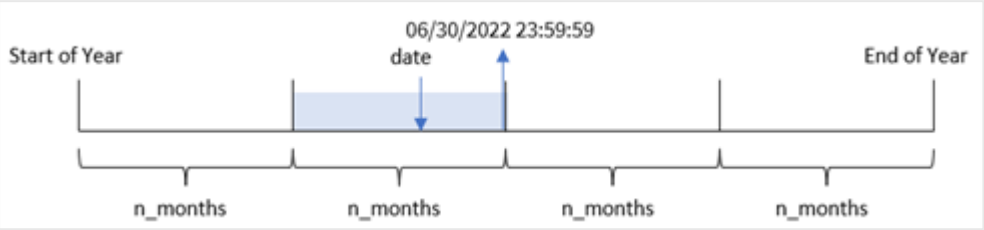
Deze functie retourneert een waarde die overeenkomt met de tijdstempel van de laatste milliseconde van de periode van een maand, twee maanden, kwartaal, vier maanden of half jaar die een basisdatum bevat. Het is tevens mogelijk om de tijdstempel voor het eind van een voorafgaande of volgende tijdsperiode te bepalen. De standaarduitvoernotatie is de `DateFormat` die is ingesteld in het script.

**Syntaxis:**

```
MonthsEnd(n_months, date[, period_no [, first_month_of_year]])
```

**Retourgegevenstypen:** dubbele waarde

*Diagram van `monthsend`-functie.*




Argumenten

| Argument | Beschrijving |
|---|---|
| **n_months** | Het aantal maanden dat de periode definieert. Een geheel getal of uitdrukking die wordt herleid tot een geheel getal dat een van de volgende waarden moet hebben: 1 (equivalent aan de functie inmonth()), 2 (2 maanden), 3 (equivalent aan de functie inquarter()), 4 (tertiaal) of 6 (half jaar). |
| **date** | De te evalueren datum of tijdstempel. |
| **period_no** | De periode kan worden verschoven met **period_no**, een geheel getal of een uitdrukking die wordt herleid tot een geheel getal, waarbij de waarde 0 de periode aangeeft die **base_date** bevat. Negatieve waarden in **period_no** geven voorafgaande perioden aan en positieve waarden geven volgende perioden aan. |
| **first_month_of_year** | Als u wilt werken met (boek)jaren die niet in januari beginnen, geeft u een waarde op tussen 2 en 12 in **first_month_of_year**. |

De functie `monthsend()` verdeelt het jaar in segmenten op basis van het opgegeven `n_months`-argument. De functie bekijkt vervolgens in welk segment elke opgegeven datum valt en retourneert de laatste milliseconde van dat segment in datumnotitie. De functie kan de eindtijdstempel retourneren van voorgaande of volgende segmenten en kan de eerste maand van het jaar opnieuw instellen.

De volgende segmenten van het jaar zijn beschikbaar in de functie als `n_month`-argumenten:

`n_month`-argumenten

| Periode | Aantal maanden |
|---|---|
| maandelijks | 1 |
| tweemaandelijks | 2 |
| kwartaal | 3 |
| tertiaal | 4 |
| halfjaarlijks | 6 |

#### Wanneer gebruiken

De `monthsend()`-functie wordt gebruikt als onderdeel van een uitdrukking als de gebruiker wil dat in de berekening het deel van de maand wordt gebruikt dat al is geweest. De gebruiker heeft de mogelijkheid om met gebruik van een variabele, de gewenste periode te selecteren. De `monthsend()`-functie kan bijvoorbeeld een invoervariabele geven waarmee de gebruiker de totale hoeveelheid rente kan berekenen die nog niet is opgebouwd gedurende de maand, het kwartaal of het halve jaar.

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `monthsend(4, '07/19/2013')` | Retourneert 08/31/2013. |
| `monthsend(4, '10/19/2013', -1)` | Retourneert 08/31/2013. |
| `monthsend(4, '10/19/2013', 0, 2)` | Retourneert 01/31/2014.<br>Omdat het begin van het jaar maand 2 wordt. |

## Voorbeeld 1 – Basisvoorbeeld

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties voor 2022 wordt in de tabel `Transactions` geladen.
- Een datumveld dat wordt weergegeven in de `DateFormat`-systeemvariabele met de datumnotitie `(MM/DD/YYYY)`.

- Een voorgaande load-instructie met:
    - De `monthsend`-functie die is ingesteld als het veld `bi_monthly_end`. De transacties van deze groep in tweemaandelijkse segmenten.
    - De `timestamp`-functie die de begintijdstempel van het segment voor iedere transactie retourneert.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

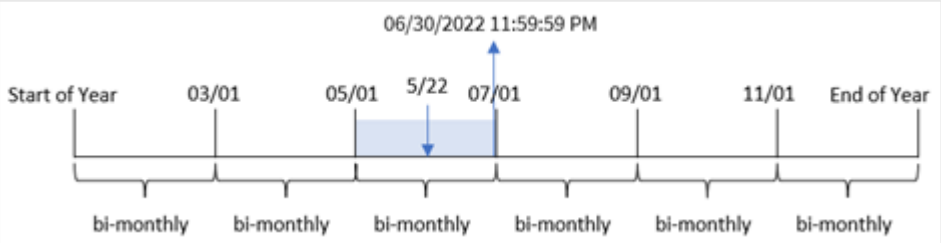
- `id`
- `date`
- `bi_monthly_end`
- `bi_monthly_end_timestamp`

Resultatentabel

| 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 |

Het veld `bi_monthly_end` wordt gemaakt in de vorige load-instructie met behulp van de functie `monthsend()`. Het eerste opgegeven argument is `2`, waarbij het jaar wordt verdeeld in tweemaandelijkse segmenten. Het tweede argument geeft aan welk veld wordt geëvalueerd.

*Diagram van de `monthsend`-functie met tweemaandelijkse segmenten.*



Transactie 8195 vindt plaats op 22 mei. De functie `monthsend()` verdeelt het jaar in eerste instantie in tweemaandelijkse segmenten. Transactie 8195 valt in het segment tussen 1 mei en 30 juni. Het resultaat is dat de functie de laatste milliseconde van dit segment retourneert: 06/30/2022 11:59:59 PM.

## Voorbeeld 2 – period_no

Load-script en resultaten

### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld wordt een veld gemaakt, `prev_bi_monthly_end`, dat de eerste milliseconde retourneert van het tweemaandelijkse segment voordat de transactie plaatsvond.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

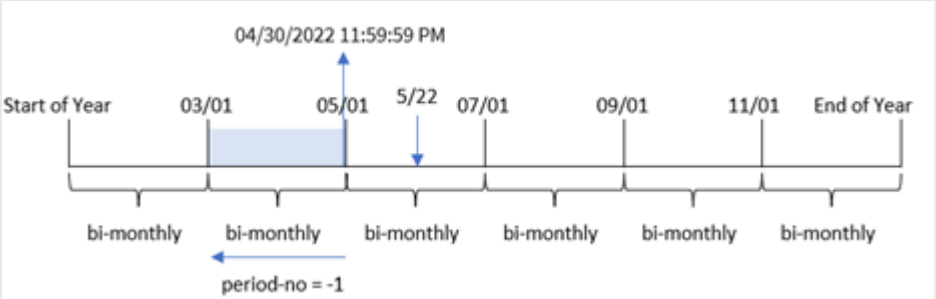
- `id`
- `date`
- `prev_bi_monthly_end`
- `prev_bi_monthly_end_timestamp`

Resultatentabel

| 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 |

Door `-1` als het `period_no`-argument in de `monthsend()`-functie te gebruiken, nadat een jaar eerst in tweemaandelijkse segmenten is verdeeld, retourneert de functie de laatste milliseconde van het vorige tweemaandelijkse segment ten opzichte van wanneer een transactie plaatsvindt.

*Diagram van de monthsend-functie die het vorige tweemaandelijkse segment retourneert.*




Transactie 8195 vindt plaats in het segment tussen mei en juni. Het resultaat is dat het vorige tweemaandelijkse segment tussen 1 maart en 30 april ligt en de functie daarom de laatste milliseconde van dit segment retourneert: 04/30/2022 11:59:59 PM.

## Voorbeeld 3 – first_month_of_year

Load-script en resultaten

### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is april volgens het organisatiebeleid de eerste maand van het financiële boekjaar.

Maak een veld, bi_monthly_end, dat transacties groepeert in tweemaandelijkse segmenten en de laatste milliseconde-tijdstempel van het segment voor elke transactie retourneert.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `date`
- `bi_monthly_end`
- `bi_monthly_end_timestamp`

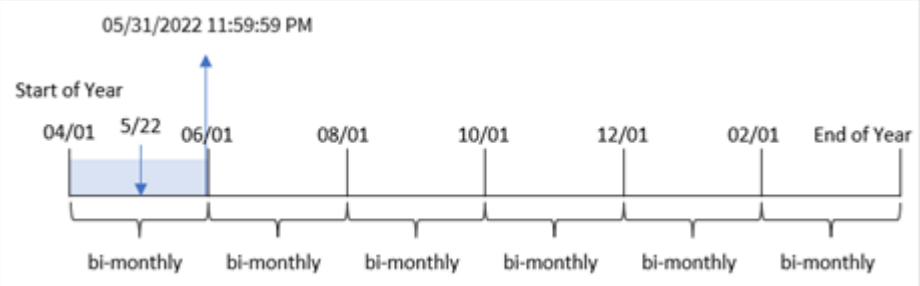
Resultatentabel

| 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 |

Door `4` te gebruiken als het `first_month_of_year`-argument in de `monthsend()`-functie, begint de functie het jaar op 1 april. Het verdeelt vervolgens het jaar in tweemaandelijkse segmenten: Apr-Mei, Jun-Jul, Aug-Sep, Okt-Nov, Dec-Jan, Feb-Mrt.

*Diagram van functie `monthsend` met april ingesteld als de eerste maand van het jaar.*




Transactie 8195 vond plaats op 22 mei en valt in het segment tussen 1 april en 31 mei. Het resultaat is dat de functie de laatste milliseconde van dit segment retourneert: 05/31/2022 11:59:59 PM.

## Voorbeeld 4 – Diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt. In dit voorbeeld is de gegevensverzameling echter niet gewijzigd en wordt in de app geladen.

In dit voorbeeld is de taak om een berekening te maken die transacties groepeert in tweemaandelijkse segmenten en de laatste milliseconde-tijdstempel van het segment voor iedere transactie retourneert als een meting in een diagramobject van een app.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

```
date
```

Maak de volgende metingen om de laatste milliseconde-tijdstempel van het tweemaandelijkse segment waarin de transactie plaatsvond op te halen:

- `=monthsEnd(2,date)`
- `=timestamp(monthsend(2,date))`

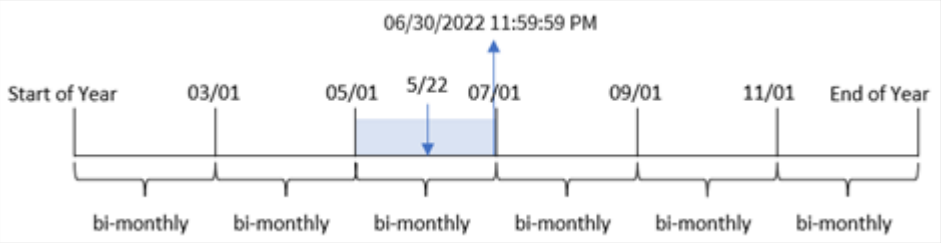
Resultatentabel

| 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 |

Het veld `bi_monthly_end` wordt in een diagramobject gemaakt als een meting met behulp van de functie `monthsend()`. Het eerste opgegeven argument is `2`, dat het jaar verdeelt in tweemaandelijkse segmenten. Het tweede argument geeft aan welk veld wordt geëvalueerd.

*Diagram van de `monthsend`-functie met tweemaandelijkse segmenten.*




Transactie 8195 vindt plaats op 22 mei. De functie `monthsend()` verdeelt het jaar in eerste instantie in tweemaandelijkse segmenten. Transactie 8195 valt in het segment tussen 1 mei en 30 juni. Het resultaat is dat de functie de eerste milliseconde van dit segment retourneert: 06/30/2022 11:59:59 PM.

## Voorbeeld 5 – Scenario

Load-script en resultaten

**Overzicht**

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

In dit voorbeeld wordt een gegevensset geladen in de tabel '`Employee_Expenses`'. De tabel bevat de volgende velden:

- Werknemers-id's
- Werknemersnamen

- De gemiddelde dagelijkse onkostendeclaraties van elke werknemer.

De eindgebruiker wil graag een diagram dat per werknemer-id en werknemersnaam de geschatte onkostendeclaraties voor de rest van een periode naar keuze weergeeft. Het boekjaar begint in januari.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een nieuw werkblad.

Aan het begin van het load-script is een variabele, `vPeriod`, gemaakt die wordt gekoppeld aan het besturingselement voor variabeleninvoer.

Doe het volgende:

1. Klik in het bedrijfsmiddelenvenster op **Aangepaste objecten**.
2. Selecteer **Qlik Dashboard-bundel** en maak een **Variabeleninvoer**-object.
3. Voer een titel in voor het diagramobject.
4. Selecteer onder **Variabele** de optie **vPeriod** als de naam en stel het object in om als een **Vervolgkeuzelijst** te worden weergegeven.
5. Klik onder **Waarden** op **Dynamische** waarden. Voer het volgende in:
   `='1~month|2~bi-month|3~quarter|4~tertial|6~half-year'.`

Maak een nieuwe tabel en deze velden als dimensies:

- `employee_id`
- `employee_name`

Maak deze meting om de opgebouwde rente te berekenen:

```
=floor(monthsend($(vPeriod),today(1))-today(1))*avg_daily_claim
```

*Deze meting is dynamisch en produceert verschillende tabelresultaten, afhankelijk van de datum waarop u de gegevens laadt.*

Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

| employee_id | employee_ name | =floor(monthsend($(vPeriod),today(1))-today(1))*avg_daily_ claim |
|---|---|---|
| 182 | Mark | $1410.00 |
| 183 | Deryck | $1175.00 |
| 184 | Dexter | $1175.00 |
| 185 | Sydney | $2538.00 |
| 186 | Agatha | $1692.00 |

De `monthsend()`-functie gebruikt de invoer van de gebruiker als het eerste argument en de datum van vandaag als het tweede argument. Dit retourneert de einddatum voor de door de gebruiker geselecteerde tijdperiode. De uitdrukking retourneert vervolgens het aantal dagen dat nog resteert in de geselecteerde periode door de datum van vandaag af te trekken van deze einddatum.

Deze waarde wordt vervolgens vermenigvuldigd met de gemiddelde dagelijkse onkostendeclaratie van elke werknemer om de geschatte waarde te berekenen van de declaraties die elke werknemer naar verwachting in de resterende dagen van deze periode zal indienen.

## monthsname

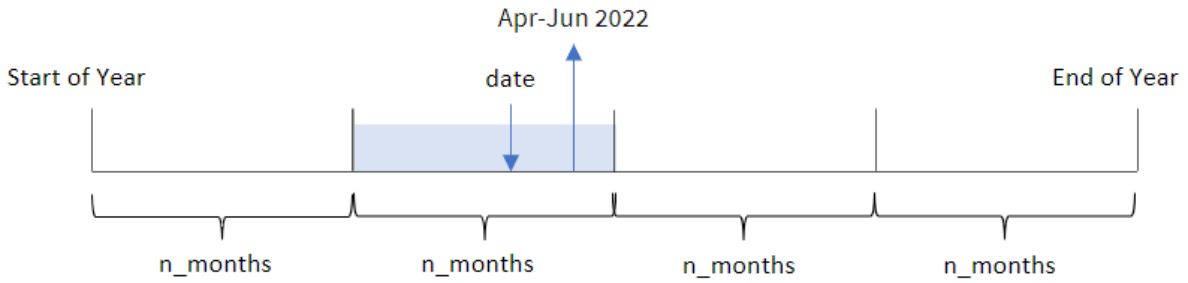
Deze functie retourneert een weergavewaarde die het bereik vertegenwoordigt van de maanden van de periode (geformatteerd volgens de scriptvariabele **MonthNames**) alsmede het jaar. De onderliggende numerieke waarde komt overeen met een tijdstempel van de eerste milliseconde van de periode van een maand, twee maanden, kwartaal, tertiaal of half jaar die een basisdatum bevat.

**Syntaxis:**

```
MonthsName(n_months, date[, period_no[, first_month_of_year]])
```

**Retourgegevenstypen:** dubbele waarde

*Diagram van functie monthsname*




De functie `monthsname()` verdeelt het jaar in segmenten op basis van het opgegeven `n_months`-argument. Het evalueert vervolgens het segment waartoe elke opgegeven `date` behoort, en retourneert de namen van de begin- en eindmaanden van dat segment, evenals het jaar. De functie biedt ook de mogelijkheid om deze grenzen van voorgaande of volgende segmenten te retourneren, en om opnieuw te definiëren wat de eerste maand van het jaar is.

De volgende segmenten van het jaar zijn beschikbaar in de functie als `n_month`-argumenten:

Mogelijke `n_month`-argumenten

| Perioden | Aantal maanden |
|---|---|
| maandelijks | 1 |
| tweemaandelijks | 2 |
| kwartaal | 3 |
| tertiaal | 4 |
| halfjaarlijks | 6 |

Argumenten

| Argument | Beschrijving |
|---|---|
| **n_months** | Het aantal maanden dat de periode definieert. Een geheel getal of uitdrukking die wordt herleid tot een geheel getal dat een van de volgende waarden moet hebben: 1 (equivalent aan de functie inmonth()), 2 (2 maanden), 3 (equivalent aan de functie inquarter()), 4 (tertiaal) of 6 (half jaar). |
| **date** | De te evalueren datum of tijdstempel. |
| **period_no** | De periode kan worden verschoven met **period_no**, een geheel getal of een uitdrukking die wordt herleid tot een geheel getal, waarbij de waarde 0 de periode aangeeft die **base_date** bevat. Negatieve waarden in **period_no** geven voorafgaande perioden aan en positieve waarden geven volgende perioden aan. |

| Argument | Beschrijving |
|---|---|
| **first_month_of_year** | Als u wilt werken met (boek)jaren die niet in januari beginnen, geeft u een waarde op tussen 2 en 12 in **first_month_of_year**. |

## Wanneer gebruiken

De functie `monthsname()` is handig wanneer u de gebruiker de functionaliteit wilt bieden om aggregaties te vergelijken op een periode naar keuze. U kunt bijvoorbeeld een invoervariabele opgeven om de gebruiker de totale verkoop van producten per maand, kwartaal of halfjaar te laten zien.

Deze dimensies kunnen worden gemaakt in het load-script door de functie toe te voegen als een veld in een masterkalendertabel, of door de dimensie rechtstreeks in een diagram te maken als een berekende dimensie.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `monthsname(4, '10/19/2013')` | Retourneert 'Sep-Dec 2013.' In dit en andere voorbeelden is de opdracht **SET Monthnames** ingesteld op Jan;Feb;Mar, enzovoort. |
| `monthsname(4, '10/19/2013', -1)` | Retourneert 'May-Aug 2013'. |
| `monthsname(4, '10/19/2013', 0, 2)` | Retourneert 'Oct-Jan 2014', aangezien het jaar is opgegeven om te beginnen in maand 2. Daarom eindigt de periode van vier maanden in de eerste maand van het volgende jaar. |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – Basisvoorbeeld

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties voor 2022 die wordt geladen in de tabel `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.
- Het maken van een veld, `bi_monthly_range`, dat transacties groepeert in tweemaandelijkse segmenten en de grensnamen van dat segment voor elke transactie retourneert.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

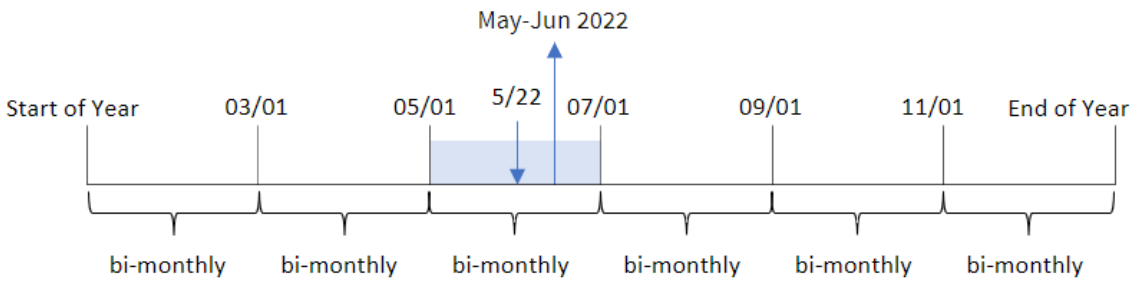
- `date`
- `bi_monthly_range`

Resultatentabel

| date | bi_monthly_range |
|---|---|
| 2/19/2022 | jan-feb 2022 |
| 3/7/2022 | mrt-apr 2022 |
| 3/30/2022 | mrt-apr 2022 |
| 4/5/2022 | mrt-apr 2022 |
| 4/16/2022 | mrt-apr 2022 |
| 5/1/2022 | mei-juni 2022 |
| 5/7/2022 | mei-juni 2022 |
| 5/22/2022 | mei-juni 2022 |
| 6/15/2022 | mei-juni 2022 |
| 6/26/2022 | mei-juni 2022 |
| 7/9/2022 | juli-aug 2022 |
| 7/22/2022 | juli-aug 2022 |
| 7/23/2022 | juli-aug 2022 |
| 7/27/2022 | juli-aug 2022 |
| 8/2/2022 | juli-aug 2022 |
| 8/8/2022 | juli-aug 2022 |
| 8/19/2022 | juli-aug 2022 |
| 9/26/2022 | sep-okt 2022 |
| 10/14/2022 | sep-okt 2022 |
| 10/29/2022 | sep-okt 2022 |

Het veld `bi_monthly_range` wordt gemaakt in de vorige load-instructie met behulp van de functie `monthsname()`. Het eerste opgegeven argument is 2, waarbij het jaar wordt verdeeld in tweemaandelijkse segmenten. Het tweede argument geeft aan welk veld wordt geëvalueerd.

*Diagram van functie monthsname, basisvoorbeeld*




Transactie 8195 vindt plaats op 22 mei. De functie `monthsname()` verdeelt het jaar in eerste instantie in tweemaandelijkse segmenten. Transactie 8195 valt in het segment tussen 1 mei en 30 juni. Daarom retourneert de functie deze maanden in de systeemvariabele-indeling `MonthNames`, evenals het jaar, mei-juni 2022.

## Voorbeeld 2 – period_no

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde inline gegevensset en het scenario uit het eerste voorbeeld.
- Het maken van een veld, `prev_bi_monthly_range`, dat transacties groepeert in tweemaandelijkse segmenten en de grensnamen van het eerdere segment voor elke transactie retourneert.

Voeg hier uw andere tekst toe, indien nodig, met lijsten enz.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `prev_bi_monthly_range`

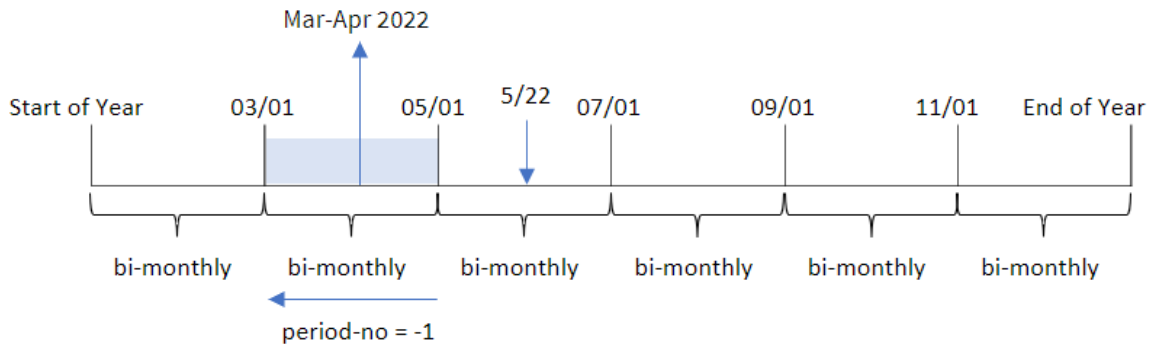
Resultatentabel

| date | prev_bi_monthly_range |
|---|---|
| 2/19/2022 | nov-dec 2021 |
| 3/7/2022 | jan-feb 2022 |
| 3/30/2022 | jan-feb 2022 |
| 4/5/2022 | jan-feb 2022 |
| 4/16/2022 | jan-feb 2022 |
| 5/1/2022 | mrt-apr 2022 |
| 5/7/2022 | mrt-apr 2022 |
| 5/22/2022 | mrt-apr 2022 |
| 6/15/2022 | mrt-apr 2022 |
| 6/26/2022 | mrt-apr 2022 |
| 7/9/2022 | mei-juni 2022 |
| 7/22/2022 | mei-juni 2022 |
| 7/23/2022 | mei-juni 2022 |
| 7/27/2022 | mei-juni 2022 |

| date | prev_bi_monthly_range |
|---|---|
| 8/2/2022 | mei-juni 2022 |
| 8/8/2022 | mei-juni 2022 |
| 8/19/2022 | mei-juni 2022 |
| 9/26/2022 | juli-aug 2022 |
| 10/14/2022 | juli-aug 2022 |
| 10/29/2022 | juli-aug 2022 |

In dit voorbeeld wordt -1 gebruikt als het `period_no`-argument in de `monthsname()`-functie. Na aanvankelijk een jaar in tweemaandelijkse segmenten te hebben verdeeld, retourneert de functie vervolgens de vorige segmentgrenzen voor wanneer een transactie plaatsvindt.

*Diagram van functie monthsname, voorbeeld period_no*




Transactie 8195 vindt plaats in het segment tussen mei en juni. Daarom was het vorige tweemaandelijkse segment tussen 1 maart en 30 april, en daarom retourneert de functie mrt-apr 2022.

## Voorbeeld 3 – first_month_of_year

### Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde inline gegevensset en het scenario uit het eerste voorbeeld.
- Het maken van een ander veld, `bi_monthly_range`, dat transacties groepeert in tweemaandelijkse segmenten en de segmentgrenzen voor elke transactie retourneert.

In dit voorbeeld moeten we echter ook april instellen als de eerste maand van het boekjaar.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `bi_monthly_range`

Resultatentabel

| date | bi_monthly_range |
|---|---|
| 2/19/2022 | feb-mrt 2021 |
| 3/7/2022 | feb-mrt 2021 |
| 3/30/2022 | feb-mrt 2021 |
| 4/5/2022 | apr-mei 2022 |

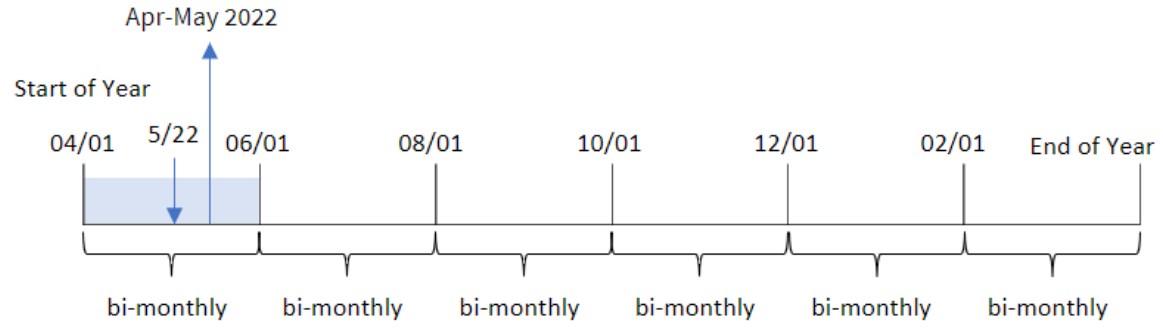
| date | bi_monthly_range |
|---|---|
| 4/16/2022 | apr-mei 2022 |
| 5/1/2022 | apr-mei 2022 |
| 5/7/2022 | apr-mei 2022 |
| 5/22/2022 | apr-mei 2022 |
| 6/15/2022 | jun-jul 2022 |
| 6/26/2022 | jun-jul 2022 |
| 7/9/2022 | jun-jul 2022 |
| 7/22/2022 | jun-jul 2022 |
| 7/23/2022 | jun-jul 2022 |
| 7/27/2022 | jun-jul 2022 |
| 8/2/2022 | aug-sept 2022 |
| 8/8/2022 | aug-sept 2022 |
| 8/19/2022 | aug-sept 2022 |
| 9/26/2022 | aug-sept 2022 |
| 10/14/2022 | okt-nov 2022 |
| 10/29/2022 | okt-nov 2022 |

Door 4 te gebruiken als het `first_month_of_year`-argument in de `monthsname()`-functie, begint de functie het jaar op 1 april. Het verdeelt vervolgens het jaar in tweemaandelijkse segmenten: apr-mei,jun-jul,aug-sep,okt-nov,dec-jan,feb-mrt.

Alineatekst voor Resultaten.

Transactie 8195 vond plaats op 22 mei en valt in het segment tussen 1 april en 31 mei. Daarom retourneert de functie apr-mei 2022.

*Diagram van functie monthsname, voorbeeld first_month_of_year*



## Voorbeeld 4 – diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde inline gegevensset en scenario als het eerste voorbeeld. In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die transacties groepeert in tweemaandelijkse segmenten en de segmentgrenzen retourneert voor elke transactie wordt gemaakt als een meting in een diagramobject van de toepassing.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:`date`.

Maak de volgende meting:

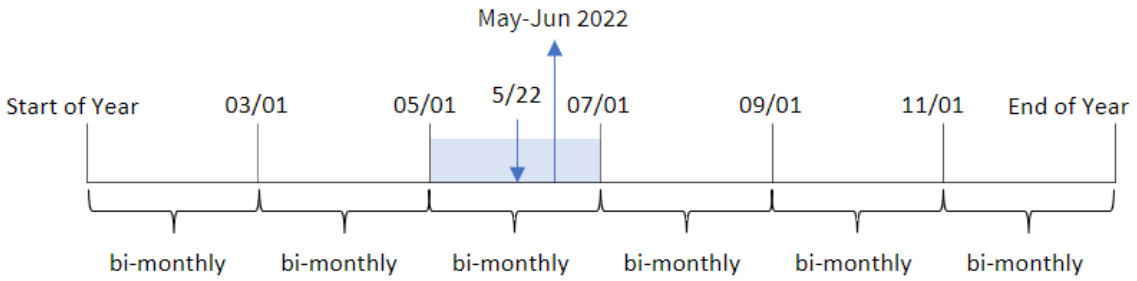
```
=monthsname(2,date)
```

Resultatentabel

| date | =monthsname(2,date) |
|---|---|
| 2/19/2022 | jan-feb 2022 |
| 3/7/2022 | mrt-apr 2022 |
| 3/30/2022 | mrt-apr 2022 |
| 4/5/2022 | mrt-apr 2022 |
| 4/16/2022 | mrt-apr 2022 |
| 5/1/2022 | mei-juni 2022 |
| 5/7/2022 | mei-juni 2022 |
| 5/22/2022 | mei-juni 2022 |
| 6/15/2022 | mei-juni 2022 |
| 6/26/2022 | mei-juni 2022 |
| 7/9/2022 | juli-aug 2022 |
| 7/22/2022 | juli-aug 2022 |
| 7/23/2022 | juli-aug 2022 |
| 7/27/2022 | juli-aug 2022 |
| 8/2/2022 | juli-aug 2022 |
| 8/8/2022 | juli-aug 2022 |
| 8/19/2022 | juli-aug 2022 |
| 9/26/2022 | sep-okt 2022 |
| 10/14/2022 | sep-okt 2022 |
| 10/29/2022 | sep-okt 2022 |

Het veld `bi_monthly_range` wordt in een diagramobject gemaakt als een meting met behulp van de functie `monthsname()`. Het eerste opgegeven argument is 2, waarbij het jaar wordt verdeeld in tweemaandelijkse segmenten. Het tweede argument geeft aan welk veld wordt geëvalueerd.

*Diagram van functie monthsname, diagramobjectvoorbeeld*




Transactie 8195 vindt plaats op 22 mei. De functie `monthsname()` verdeelt het jaar in eerste instantie in tweemaandelijkse segmenten. Transactie 8195 valt in het segment tussen 1 mei en 30 juni. Daarom retourneert de functie deze maanden in de systeemvariabele-indeling `MonthNames`, evenals het jaar, mei-juni 2022.

## Voorbeeld 5 – Scenario

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met transacties voor 2022 die wordt geladen in een tabel met de naam `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.

De eindgebruiker wil graag een diagramobject dat de totale omzet per periode naar eigen keuze weergeeft. Dit kan zelfs worden bereikt als deze dimensie niet beschikbaar is in het gegevensmodel, door de functie `monthsname()` te gebruiken als een berekende dimensie die dynamisch wordt gewijzigd door een besturingselement voor variabeleninvoer.

### Load-script

```
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
];
```

**Resultaten**

Laad de gegevens en open een werkblad.

Aan het begin van het load-script is een variabele (`vPeriod`) gemaakt die wordt gekoppeld aan het besturingselement voor variabeleninvoer. Configureer vervolgens de variabele als een aangepast object in het werkblad.

**Doe het volgende:**

1. Klik in het bedrijfsmiddelenvenster op **Aangepaste objecten**.
2. Selecteer **Qlik Dashboard-bundel** en maak een **Variabeleninvoer**-object.
3. Voer een titel in voor het diagramobject.
4. Selecteer onder **Variabele** de optie **vPeriod** als de naam en stel het object in om als een **Vervolgkeuzelijst** te worden weergegeven.
5. Configureer onder **Waarden** het object om dynamische waarden te gebruiken. Voer het volgende in:
   `='1~month|2~bi-month|3~quarter|4~tertial|6~half-year'`

Maak vervolgens de resultatentabel.

**Doe het volgende:**

1. Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:
   `=monthsname($(vPeriod),date)`
2. Voeg deze meting toe om de totale verkoop te berekenen:
   `=sum(amount)`
3. Stel de **Getalnotatie** van de meting in op **Geld**.Klik op ✔ **Klaar met bewerken**. U kunt nu de gegevens in de tabel wijzigen door het tijdsegment in het variabele-object aan te passen.

Dit is hoe de resultatentabel eruit zal zien als de optie `tertial` is geselecteerd:

Resultatentabel

| monthsname($(vPeriod),date) | =sum(amount) |
|---|---|
| jan-apr 2022 | 253.89 |
| Mei-aug 2022 | 713.58 |
| sep-dec 2022 | 248.12 |

## monthsstart

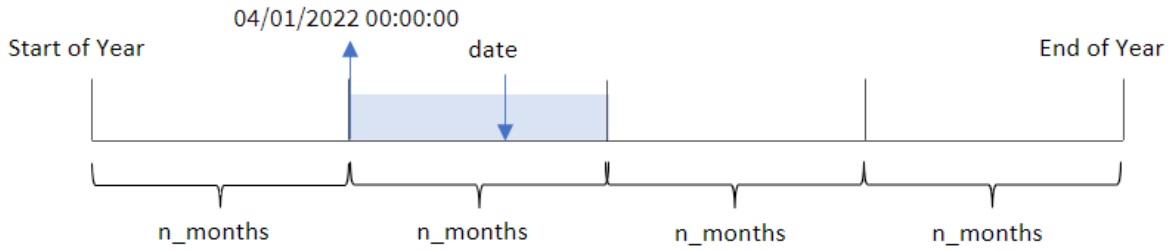
Deze functie retourneert een waarde die overeenkomt met de tijdstempel van de eerste milliseconde van de periode van een maand, twee maanden, kwartaal, tertiaal of half jaar die een basisdatum bevat. Het is tevens mogelijk om de tijdstempel voor een voorafgaande of volgende tijdsperiode te bepalen. De standaarduitvoernotatie is de **DateFormat** die is ingesteld in het script.

**Syntaxis:**

```
MonthsStart(n_months, date[, period_no [, first_month_of_year]])
```

**Retourgegevenstypen:** dubbele waarde

*Diagram van de `monthsstart()`-functie*




De functie `monthsstart()` verdeelt het jaar in segmenten op basis van het opgegeven `n_months`-argument. De functie bekijkt vervolgens in welk segment elke opgegeven datum valt en retourneert de eerste milliseconde van dat segment in datumnotitie. De functie biedt ook de mogelijkheid om de begintijdstempel van voorgaande of volgende segmenten te retourneren, en om opnieuw te definiëren wat de eerste maand van het jaar is.

De volgende segmenten van het jaar zijn beschikbaar in de functie als `n_month`-argumenten:

Mogelijke `n_month`-argumenten

| Perioden | Aantal maanden |
|---|---|
| maandelijks | 1 |
| tweemaandelijks | 2 |

| Perioden | Aantal maanden |
|---|---|
| kwartaal | 3 |
| tertiaal | 4 |
| halfjaarlijks | 6 |

Argumenten

| Argument | Beschrijving |
|---|---|
| **n_months** | Het aantal maanden dat de periode definieert. Een geheel getal of uitdrukking die wordt herleid tot een geheel getal dat een van de volgende waarden moet hebben: 1 (equivalent aan de functie inmonth()), 2 (2 maanden), 3 (equivalent aan de functie inquarter()), 4 (tertiaal) of 6 (half jaar). |
| **date** | De te evalueren datum of tijdstempel. |
| **period_no** | De periode kan worden verschoven met **period_no**, een geheel getal of een uitdrukking die wordt herleid tot een geheel getal, waarbij de waarde 0 de periode aangeeft die **base_date** bevat. Negatieve waarden in **period_no** geven voorafgaande perioden aan en positieve waarden geven volgende perioden aan. |
| **first_month_of_year** | Als u wilt werken met (boek)jaren die niet in januari beginnen, geeft u een waarde op tussen 2 en 12 in **first_month_of_year**. |

## Wanneer gebruiken

De `monthsstart()`-functie wordt doorgaans gebruikt als onderdeel van een uitdrukking als de gebruiker wil dat in de berekening een deel van de periode wordt gebruikt dat nog niet heeft plaatsgevonden. Dit kan bijvoorbeeld worden gebruikt om een invoervariabele te verstrekken waarmee de gebruiker de totale hoeveelheid rente kan berekenen die tot nu toe in de maand het kwartaal of het halve jaar is opgebouwd.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `monthsstart(4, '10/19/2013')` | Retourneert 09/01/2013. |
| `monthsstart(4, '10/19/2013, -1)` | Retourneert 05/01/2013. |
| `monthsstart(4, '10/19/2013', 0, 2 )` | Retourneert 10/01/2013, omdat het begin van het jaar maand 2 wordt. |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – geen aanvullende argumenten

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties voor 2022 die wordt geladen in de tabel `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.
- Het maken van een veld, `bi_monthly_start`, dat transacties groepeert in tweemaandelijkse segmenten en de begintijdstempel van het segment voor elke transactie retourneert.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `bi_monthly_start`
- `bi_monthly_start_timestamp`

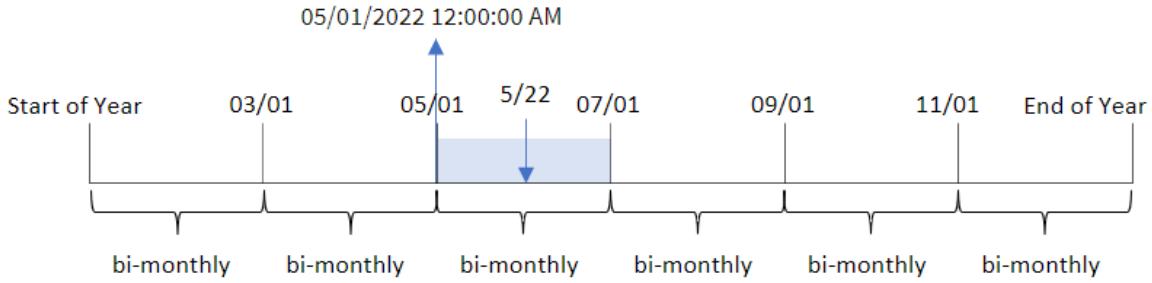
Resultatentabel

| 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 | 05/01/2022 | 5/1/2022 12:00:00 AM |
| 5/22/2022 | 05/01/2022 | 5/1/2022 12:00:00 AM |
| 6/15/2022 | 05/01/2022 | 5/1/2022 12:00:00 AM |
| 6/26/2022 | 05/01/2022 | 5/1/2022 12:00:00 AM |
| 7/9/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |
| 7/22/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |
| 7/23/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |
| 7/27/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |
| 8/2/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |
| 8/8/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |
| 8/19/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |
| 9/26/2022 | 09/01/2022 | 9/1/2022 12:00:00 AM |

| date | bi_monthly_start | bi_monthly_start_timestamp |
|---|---|---|
| 10/14/2022 | 09/01/2022 | 9/1/2022 12:00:00 AM |
| 10/29/2022 | 09/01/2022 | 9/1/2022 12:00:00 AM |

Het veld `bi_monthly_start` wordt gemaakt in de vorige load-instructie met behulp van de functie `monthsstart()`. Het eerste opgegeven argument is 2, waarbij het jaar wordt verdeeld in tweemaandelijkse segmenten. Het tweede argument geeft aan welk veld wordt geëvalueerd.

*Diagram van `monthsstart()`-functie, voorbeeld zonder aanvullende argumenten*




Transactie 8195 vindt plaats op 22 mei. De functie `monthsstart()` verdeelt het jaar in eerste instantie in tweemaandelijkse segmenten. Transactie 8195 valt in het segment tussen 1 mei en 30 juni. Daarom retourneert de functie de eerste milliseconde van dit segment: 1 mei 2022 om 12:00:00 AM.

## Voorbeeld 2 – period_no

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Er wordt een veld gemaakt, `prev_bi_monthly_start`, dat de eerste milliseconde retourneert van het tweemaandelijkse segment voordat de transactie plaatsvond.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `prev_bi_monthly_start`
- `prev_bi_monthly_start_timestamp`

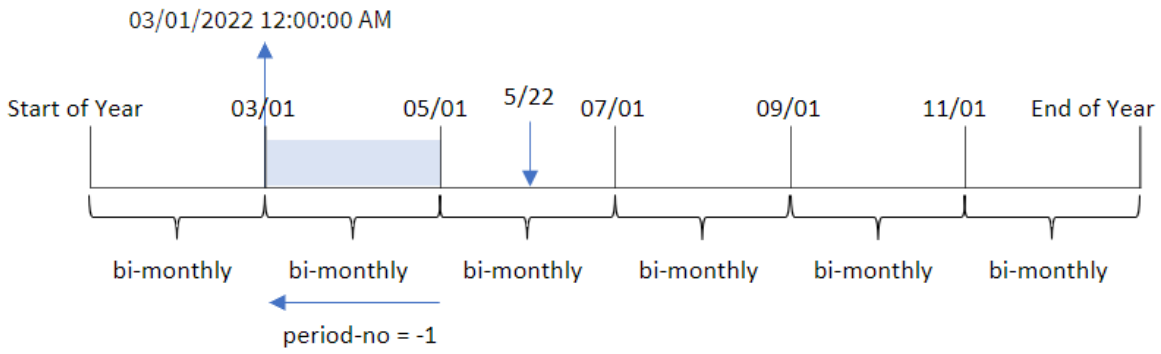
Resultatentabel

| 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 | 3/1/2022 12:00:00 AM |
| 5/7/2022 | 03/01/2022 | 3/1/2022 12:00:00 AM |
| 5/22/2022 | 03/01/2022 | 3/1/2022 12:00:00 AM |

| date | prev_bi_monthly_start | prev_bi_monthly_start_timestamp |
|---|---|---|
| 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 |
| 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 | 7/1/2022 12:00:00 AM |
| 10/29/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |

Door -1 als het `period_no`-argument in de `monthsstart()`-functie te gebruiken, nadat een jaar eerst in tweemaandelijkse segmenten is verdeeld, retourneert de functie de eerste milliseconde van het vorige tweemaandelijkse segment ten opzichte van wanneer een transactie plaatsvindt.

*Diagram van `monthsstart()`-functie, voorbeeld period_no*




Transactie 8195 vindt plaats in het segment tussen mei en juni. Daarom was het vorige tweemaandelijkse segment tussen 1 maart en 30 april waardoor de functie de eerste milliseconde van dit segment retourneert: 1 maart 2022 om 12:00:00 AM.

## Voorbeeld 3 – first_month_of_year

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het maken van een veld, `bi_monthly_start`, dat transacties groepeert in tweemaandelijkse segmenten en de begintijdstempel van de set voor elke transactie retourneert.

In dit voorbeeld moeten we echter ook april instellen als de eerste maand van het boekjaar.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

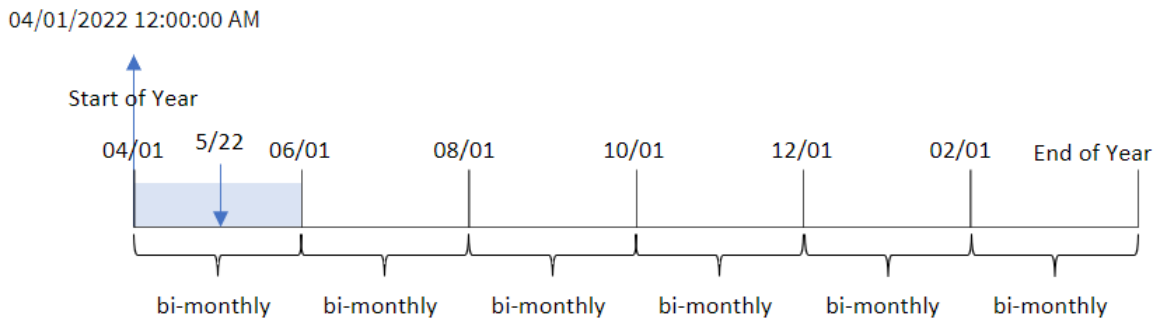
- `date`
- `bi_monthly_start`
- `bi_monthly_start_timestamp`

Resultatentabel

| date | bi_monthly_start | bi_monthly_start_timestamp |
|---|---|---|
| 2/19/2022 | 02/01/2022 | 2/1/2022 12:00:00 AM |
| 3/7/2022 | 02/01/2022 | 2/1/2022 12:00:00 AM |
| 3/30/2022 | 02/01/2022 | 2/1/2022 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 | 4/1/2022 12:00:00 AM |
| 5/7/2022 | 04/01/2022 | 4/1/2022 12:00:00 AM |
| 5/22/2022 | 04/01/2022 | 4/1/2022 12:00:00 AM |
| 6/15/2022 | 06/01/2022 | 6/1/2022 12:00:00 AM |
| 6/26/2022 | 06/01/2022 | 6/1/2022 12:00:00 AM |
| 7/9/2022 | 06/01/2022 | 6/1/2022 12:00:00 AM |
| 7/22/2022 | 06/01/2022 | 6/1/2022 12:00:00 AM |
| 7/23/2022 | 06/01/2022 | 6/1/2022 12:00:00 AM |
| 7/27/2022 | 06/01/2022 | 6/1/2022 12:00:00 AM |
| 8/2/2022 | 08/01/2022 | 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 | 8/1/2022 12:00:00 AM |
| 9/26/2022 | 08/01/2022 | 8/1/2022 12:00:00 AM |
| 10/14/2022 | 10/01/2022 | 10/1/2022 12:00:00 AM |
| 10/29/2022 | 10/01/2022 | 10/1/2022 12:00:00 AM |

Door 4 te gebruiken als het `first_month_of_year`-argument in de `monthsstart()`-functie, begint de functie het jaar op 1 april. Het verdeelt vervolgens het jaar in tweemaandelijkse segmenten: apr-mei,jun-jul,aug-sep,okt-nov,dec-jan,feb-mrt.

*Diagram van de `monthsstart()`-functie, voorbeeld first_month_of_year*




Transactie 8195 vond plaats op 22 mei en valt in het segment tussen 1 april en 31 mei. Daarom retourneert de functie de eerste milliseconde van dit segment: 1 april 2022 om 12:00:00 AM.

## Voorbeeld 4 – diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die transacties groepeert in tweemaandelijkse segmenten en de begintijdstempel van de set retourneert voor elke transactie wordt gemaakt als een meting in een diagramobject van de applicatie.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `date`.

Maak de volgende metingen:

```
=monthsstart(2,date)
```

```
=timestamp(monthsstart(2,date))
```

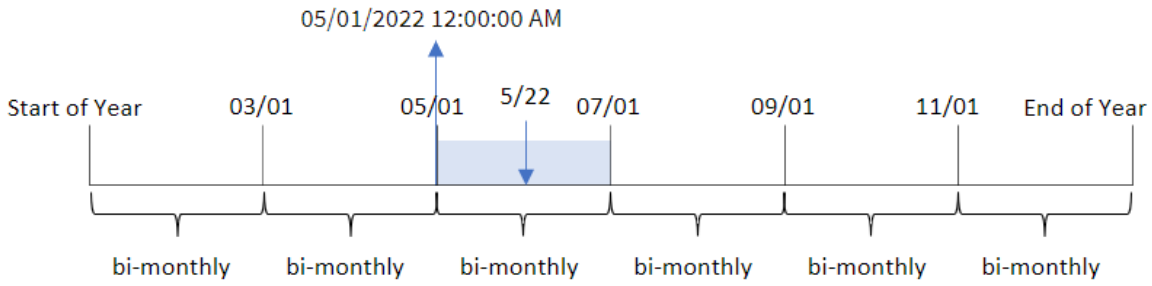
Deze berekeningen halen de begintijdstempel op van het tweemaandelijkse segment waarin iedere transactie plaatsvond.

Resultatentabel

| date | =monthsstart(2,date) | =timestamp(monthsstart(2,date)) |
|---|---|---|
| 9/26/2022 | 09/01/2022 | 9/1/2022 12:00:00 AM |
| 10/14/2022 | 09/01/2022 | 9/1/2022 12:00:00 AM |
| 10/29/2022 | 09/01/2022 | 9/1/2022 12:00:00 AM |
| 7/9/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |
| 7/22/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |
| 7/23/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |
| 7/27/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |
| 8/2/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |
| 8/8/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |
| 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 | 05/01/2022 | 5/1/2022 12:00:00 AM |
| 5/22/2022 | 05/01/2022 | 5/1/2022 12:00:00 AM |
| 6/15/2022 | 05/01/2022 | 5/1/2022 12:00:00 AM |
| 6/26/2022 | 05/01/2022 | 5/1/2022 12:00:00 AM |

| date | =monthsstart(2,date) | =timestamp(monthsstart(2,date)) |
|---|---|---|
| 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 | 1/1/2021 12:00:00 AM |

*Diagram van `monthsstart()`-functie, diagramobjectvoorbeeld*




Transactie 8195 vond plaats op 22 mei. De functie `monthsstart()` verdeelt het jaar in eerste instantie in tweemaandelijkse segmenten. Transactie 8195 valt in het segment tussen 1 mei en 30 juni. Daarom retourneert de functie de eerste milliseconde van dit segment: 05/01/2022 om 12:00:00 AM.

## Voorbeeld 5 – Scenario

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met een set leningsaldo's die wordt geladen in de tabel `Loans`.
- Gegevens bestaan uit lening-id's, het saldo aan het begin van de maand en de enkelvoudige rente die op iedere lening per jaar wordt berekend.

De eindgebruiker wil een diagramobject dat per lening-id de huidige rente weergeeft die is opgebouwd voor de door de gebruiker geselecteerde periode. Het boekjaar begint in januari.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad.

Aan het begin van het load-script is een variabele (`vPeriod`) gemaakt die wordt gekoppeld aan het besturingselement voor variabeleninvoer. Configureer vervolgens de variabele als een aangepast object in het werkblad.

**Doe het volgende:**

1. Klik in het bedrijfsmiddelenvenster op **Aangepaste objecten**.
2. Selecteer **Qlik Dashboard-bundel** en maak een object **Variabeleninvoer**.
3. Voer een titel in voor het diagramobject.
4. Selecteer onder **Variabele** de optie **vPeriod** als de naam en stel het object in om als een **Vervolgkeuzelijst** te worden weergegeven.
5. Configureer onder **Waarden** het object om dynamische waarden te gebruiken. Voer het volgende in:
   `='1~month|2~bi-month|3~quarter|4~tertial|6~half-year'`

Maak vervolgens de resultatentabel.

**Doe het volgende:**

1. Maak een nieuwe tabel. Voeg de volgende velden als dimensies toe.
   - `employee_id`
   - `employee_name`
2. Maak een meting om de opgebouwde rente te berekenen:
   `=start_balance*(rate*(today(1)-monthsstart($(vPeriod),today(1)))/365)`
3. Stel de **Getalnotatie** van de meting in op **Geld**.Klik op ✔ **Klaar met bewerken**. U kunt nu de gegevens in de tabel wijzigen door het tijdsegment in het variabele-object aan te passen.

Dit is hoe de resultatentabel eruit zal zien als de periodeoptie `month` is geselecteerd:

Resultatentabel

| 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 |

De `monthsstart()`-functie maakt gebruik van de invoer van de gebruiker als het eerste argument en de datum van vandaag als het tweede argument om de begindatum van de door gebruiker geselecteerde periode te retourneren. Door dat resultaat af te trekken van de huidige datum, retourneert de uitdrukking het aantal dagen dat in deze periode tot nu toe is verstreken.

Deze waarde wordt vervolgens vermenigvuldigd met het rentepercentage en gedeeld door 365 om het effectieve rentepercentage voor deze periode te retourneren. Het resultaat wordt vervolgens vermenigvuldigd met het beginsaldo van de lening om de rente terug te betalen die deze periode tot nu toe is opgebouwd.

## monthstart

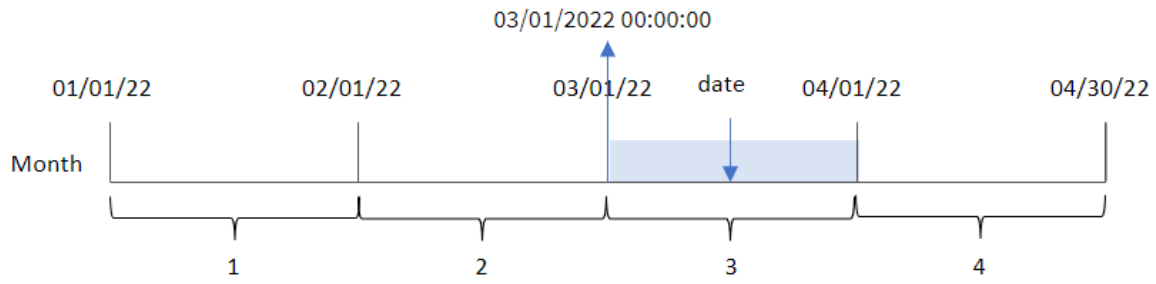
Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de eerste milliseconde van de eerste dag van de maand die **date** bevat. De standaarduitvoernotatie is de **DateFormat** die is ingesteld in het script.

**Syntaxis:**

```
MonthStart(date[, period_no])
```

**Retourgegevenstypen:** dubbele waarde

*Diagram van de `monthstart()`-functie*




De `monthstart()`-functie bepaalt in welke maand de datum valt. Hij retourneert vervolgens een tijdstempel, in datumnotatie, voor de eerste milliseconde van die maand.

Argumenten

| Argument | Beschrijving |
|---|---|
| **date** | De te evalueren datum of tijdstempel. |
| **period_no** | **period_no** is een geheel getal dat, indien het 0 is of wordt weggelaten, de maand aangeeft die **date** bevat. Negatieve waarden in **period_no** geven voorafgaande maanden aan en positieve waarden geven volgende maanden aan. |

## Wanneer gebruiken

De functie `monthstart()` wordt doorgaans gebruikt als onderdeel van een uitdrukking als de gebruiker wil dat in de berekening het deel van de maand wordt gebruikt dat al is geweest. Hij kan bijvoorbeeld worden gebruikt om de rente te berekenen die in een maand tot een bepaalde datum is opgebouwd.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `monthstart('10/19/2001')` | Retourneert 10/01/2001. |
| `monthstart('10/19/2001', -1)` | Retourneert 09/01/2001. |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – geen aanvullende argumenten

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties voor 2022 die wordt geladen in de tabel `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.
- Het maken van een veld, `start_of_month`, dat een tijdstempel voor het begin van de maand retourneert waarin de transactie plaatsvond.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `start_of_month`
- `start_of_month_timestamp`

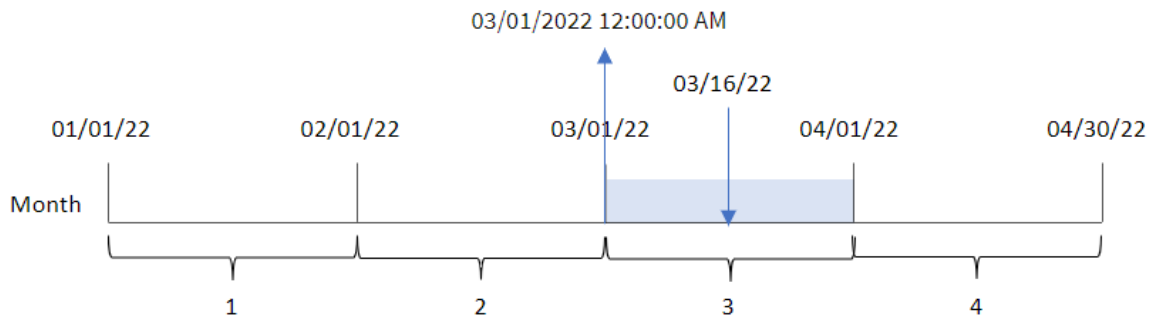
Resultatentabel

| date | start_of_month | start_of_month_timestamp |
|---|---|---|
| 1/7/2022 | 01/01/2022 | 1/1/2022 12:00:00 AM |
| 1/19/2022 | 01/01/2022 | 1/1/2022 12:00:00 AM |
| 2/5/2022 | 02/01/2022 | 2/1/2022 12:00:00 AM |
| 2/28/2022 | 02/01/2022 | 2/1/2022 12:00:00 AM |
| 3/16/2022 | 03/01/2022 | 3/1/2022 12:00:00 AM |
| 4/1/2022 | 04/01/2022 | 4/1/2022 12:00:00 AM |
| 5/7/2022 | 05/01/2022 | 5/1/2022 12:00:00 AM |
| 5/16/2022 | 05/01/2022 | 5/1/2022 12:00:00 AM |
| 6/15/2022 | 06/01/2022 | 6/1/2022 12:00:00 AM |
| 6/26/2022 | 07/01/2022 | 6/1/2022 12:00:00 AM |
| 7/9/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |
| 7/22/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |
| 7/23/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |
| 7/27/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |
| 8/2/2022 | 08/01/2022 | 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 | 8/1/2022 12:00:00 AM |
| 9/26/2022 | 09/01/2022 | 9/1/2022 12:00:00 AM |
| 10/14/2022 | 10/01/2022 | 10/1/2022 12:00:00 AM |
| 10/29/2022 | 10/01/2022 | 10/1/2022 12:00:00 AM |

Het veld '`start_of_month`' wordt gemaakt in de voorafgaande load-instructie met behulp van de functie `monthstart()` en geeft het datumveld door als het argument van de functie.

De functie `monthstart()` identificeert in welke maand de datumwaarde valt en retourneert een tijdstempel voor de eerste milliseconde van die maand.

*Diagram van `monthstart()`-functie, voorbeeld zonder aanvullende argumenten*




Transactie 8192 vond plaats op 16 maart. De functie `monthstart()` retourneert de eerste milliseconde van die maand, 1 maart om 12:00:00 AM.

## Voorbeeld 2 – period_no

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het maken van een veld, `previous_month_start`, dat een tijdstempel voor het begin van de maand retourneert voordat de transactie plaatsvond.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `previous_month_start`
- `previous_month_start_timestamp`

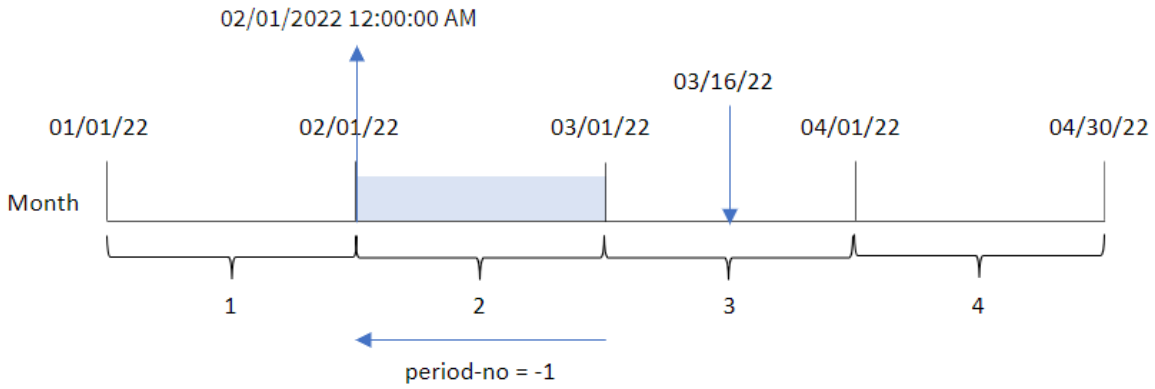
Resultatentabel

| date | previous_month_start | previous_month_start_timestamp |
|---|---|---|
| 1/7/2022 | 12/01/2021 | 12/1/2021 12:00:00 AM |
| 1/19/2022 | 12/01/2021 | 12/1/2021 12:00:00 AM |
| 2/5/2022 | 01/01/2022 | 1/1/2022 12:00:00 AM |
| 2/28/2022 | 01/01/2022 | 1/1/2022 12:00:00 AM |
| 3/16/2022 | 02/01/2022 | 2/1/2022 12:00:00 AM |
| 4/1/2022 | 03/01/2022 | 3/1/2022 12:00:00 AM |
| 5/7/2022 | 04/01/2022 | 4/1/2022 12:00:00 AM |
| 5/16/2022 | 04/01/2022 | 4/1/2022 12:00:00 AM |
| 6/15/2022 | 05/01/2022 | 5/1/2022 12:00:00 AM |
| 6/26/2022 | 05/01/2022 | 5/1/2022 12:00:00 AM |
| 7/9/2022 | 06/01/2022 | 6/1/2022 12:00:00 AM |
| 7/22/2022 | 06/01/2022 | 6/1/2022 12:00:00 AM |
| 7/23/2022 | 06/01/2022 | 6/1/2022 12:00:00 AM |
| 7/27/2022 | 06/01/2022 | 6/1/2022 12:00:00 AM |
| 8/2/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |

| date | previous_month_start | previous_month_start_timestamp |
|---|---|---|
| 8/8/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |
| 8/19/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |
| 9/26/2022 | 08/01/2022 | 8/1/2022 12:00:00 AM |
| 10/14/2022 | 09/01/2022 | 9/1/2022 12:00:00 AM |
| 10/29/2022 | 09/01/2022 | 9/1/2022 12:00:00 AM |

In dit geval, omdat een `period_no` van -1 werd gebruikt als het offset-argument in de `monthstart()`-functie, identificeert de functie eerst de maand waarin de transacties plaatsvinden. Het verschuift dan naar een maand eerder en identificeert de eerste milliseconde van die maand.

*Diagram van `monthstart()`-functie, voorbeeld period_no*




Transactie 8192 vond plaats op 16 maart. De `monthstart()`-functie stelt vast dat de maand voor de transactie februari was. Vervolgens wordt de eerste milliseconde van die maand geretourneerd: 1 februari om 12:00:00 AM.

## Voorbeeld 3 – diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die een tijdstempel retourneert voor het begin van de maand waarin de transacties plaatsvonden is gemaakt als meting in een diagramobject van de applicatie.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `date`.

Maak de volgende metingen om de begindatum van de maand waarin een transactie plaatsvindt te berekenen:

- `=monthstart(date)`
- `=timestamp(monthstart(date))`

Resultatentabel

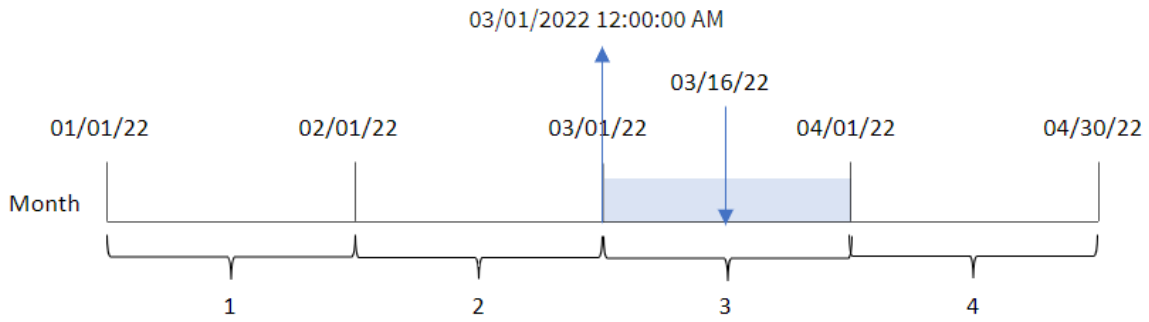
| date | =monthstart(date) | =timestamp(monthstart(date)) |
|---|---|---|
| 10/14/2022 | 10/01/2022 | 10/1/2022 12:00:00 AM |
| 10/29/2022 | 10/01/2022 | 10/1/2022 12:00:00 AM |
| 9/26/2022 | 09/01/2022 | 9/1/2022 12:00:00 AM |
| 8/2/2022 | 08/01/2022 | 8/1/2022 12:00:00 AM |

| date | =monthstart(date) | =timestamp(monthstart(date)) |
|---|---|---|
| 8/8/2022 | 08/01/2022 | 8/1/2022 12:00:00 AM |
| 8/19/2022 | 08/01/2022 | 8/1/2022 12:00:00 AM |
| 7/9/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |
| 7/22/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |
| 7/23/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |
| 7/27/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |
| 6/15/2022 | 06/01/2022 | 6/1/2022 12:00:00 AM |
| 6/26/2022 | 06/01/2022 | 6/1/2022 12:00:00 AM |
| 5/7/2022 | 05/01/2022 | 5/1/2022 12:00:00 AM |
| 5/16/2022 | 05/01/2022 | 5/1/2022 12:00:00 AM |
| 4/1/2022 | 04/01/2022 | 4/1/2022 12:00:00 AM |
| 3/16/2022 | 03/01/2022 | 3/1/2022 12:00:00 AM |
| 2/5/2022 | 02/01/2022 | 2/1/2022 12:00:00 AM |
| 2/28/2022 | 02/01/2022 | 2/1/2022 12:00:00 AM |
| 1/7/2022 | 01/01/2022 | 1/1/2022 12:00:00 AM |
| 1/19/2022 | 01/01/2022 | 1/1/2022 12:00:00 AM |

De meting `start_of_month` wordt gemaakt in het diagramobject met behulp van de functie `monthstart()` en geeft het veld door als het argument van de functie.

De functie `monthstart()` identificeert in welke maand de datumwaarde valt en retourneert een tijdstempel voor de eerste milliseconde van die maand.

*Diagram van `monthstart()`-functie, diagramobjectvoorbeeld*




Transactie 8192 vond plaats op 16 maart. De `monthstart()`-functie identificeert dat de transactie in maart heeft plaatsgevonden retourneert de eerste milliseconde van die maand, 1 maart om 12:00:00 AM.

## Voorbeeld 4 – Scenario

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met een set leningsaldo's die wordt geladen in de tabel `Loans`.
- Gegevens bestaan uit lening-id's, het saldo aan het begin van de maand en de enkelvoudige rente die op iedere lening per jaar wordt berekend.

De eindgebruiker wil een diagramobject dat per lening-id de huidige rente weergeeft die is opgebouwd op elke lening in de maand tot heden.

### Load-script

```
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
];
```

### Resultaten

### Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:
    - `loan_id`
    - `start_balance`
2. Maak nu een meting om de opgebouwde rente te berekenen:
   `=start_balance*(rate*(today(1)-monthstart(today(1)))/365)`
3. Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

| loan_id | start_balance | =start_balance*(rate*(today(1)-monthstart(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 |

De functie `monthstart()` gebruikt de datum van vandaag als enige argument en retourneert de begindatum van de huidige maand. Door dat resultaat af te trekken van de huidige datum, retourneert de uitdrukking het aantal dagen dat in deze maand tot nu toe is verstreken.

Deze waarde wordt vervolgens vermenigvuldigd met het rentepercentage en gedeeld door 365 om het effectieve rentepercentage voor deze periode te retourneren. Het resultaat wordt vervolgens vermenigvuldigd met het beginsaldo van de lening om de rente terug te betalen die deze maand tot nu toe is opgebouwd.

## networkdays

De functie **networkdays** retourneert het aantal werkdagen (maandag-vrijdag) tussen **start_date** en **end_date**, waarbij rekening wordt gehouden met een eventueel opgegeven **holiday**.

**Syntaxis:**

```
networkdays (start_date, end_date [, holiday])
```

**Retourgegevenstypen:** geheel getal

*Kalenderdiagram dat het datumbereik weergeeft dat wordt geretourneerd door de networkdays-functie*

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| 7 | 8 | 9 | 10 start_date | 11 | 12 | 13 |
| 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 end_date | 27 |
| 28 | 29 | 30 | 31 | | | |

De `networkdays`-functie heeft de volgende beperkingen:

- Er is geen methode om werkdagen aan te passen. Met andere woorden: er is geen manier om de functie aan te passen voor regio's of situaties waarbij sprake is van iets anders dan werk van maandag t/m vrijdag.
- De `holiday`-parameter moet een tekenreeksconstante zijn. Uitdrukkingen worden niet geaccepteerd.

Argumenten

| Argument | Beschrijving |
|---|---|
| **start_date** | De te evalueren begindatum. |
| **end_date** | De te evalueren einddatum. |
| **holiday** | Vakantieperioden die moeten worden uitgesloten van werkdagen. Een vakantie wordt aangegeven als een tekenreeks met constante datum. U kunt meerdere vakantiedatums opgegeven, gescheiden door komma's.<br><br>**Voorbeeld:** `'12/25/2013', '12/26/2013', '12/31/2013', '01/01/2014'` |

## Wanneer gebruiken

De `networkdays()`-functie wordt doorgaans gebruikt als onderdeel van een uitdrukking als de gebruiker wil dat in de berekening het aantal werkweekdagen gebruikt dat tussen twee datums valt. Bijvoorbeeld als een gebruiker het totale salaris wil berekenen dat een werknemer met een PAYE-contract (pay-as-you-earn) verdient.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `networkdays ('12/19/2013', '01/07/2014')` | Retourneert 14. In dit voorbeeld wordt geen rekening gehouden met vakanties. |
| `networkdays ('12/19/2013', '01/07/2014', '12/25/2013', '12/26/2013')` | Retourneert 12. In dit voorbeeld wordt rekening gehouden met de vakantie van 12/25/2013 t/m 12/26/2013. |
| `networkdays ('12/19/2013', '01/07/2014', '12/25/2013', '12/26/2013', '12/31/2013', '01/01/2014')` | Retourneert 10. In dit voorbeeld wordt rekening gehouden met twee vakantieperioden. |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – Basisvoorbeeld

### Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met project-id's, hun begindatums en hun einddatums. Deze informatie wordt geladen in de tabel `Projects`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.
- Er wordt een extra veld gemaakt, `net_work_days`, om het aantal werkdagen voor elk project te berekenen.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `start_date`
- `end_date`
- `net_work_days`

Resultatentabel

| id | start_date | end_date | net_work_days |
|---|---|---|---|
| 1 | 01/01/2022 | 01/18/2022 | 12 |
| 2 | 02/10/2022 | 02/17/2022 | 6 |
| 3 | 05/17/2022 | 07/05/2022 | 36 |
| 4 | 06/01/2022 | 06/12/2022 | 8 |
| 5 | 08/10/2022 | 08/26/2022 | 13 |

Omdat er geen geplande vakantiedagen zijn (deze zouden aanwezig zijn geweest in het derde argument van de `networkdays()`-functie), trekt de functie de `start_date` af van de `end_date`, evenals alle weekenddagen, om het aantal werkdagen tussen de twee datums te berekenen.

*Kalenderdiagram dat werkdagen voor project 5 (geen vakantiedagen) markeert*

| 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 | | | |

De bovenstaande kalender markeert het project visueel met de `id` 5. Project 5 begint op woensdag, 10 augustus 2022 en eindigt op 26 augustus 2022. Doordat alle zaterdagen en zondagen worden genegeerd, zitten er 13 werkdagen tussen deze twee datums.

## Voorbeeld 2 - Enkele vakantie

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het vorige voorbeeld.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.
- Er wordt een extra veld gemaakt, `net_work_days`, om het aantal werkdagen voor elk project te berekenen.

In dit voorbeeld staat er één vakantiedag gepland op 19 augustus 2022.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `start_date`
- `end_date`
- `net_work_days`

Resultatentabel

| id | start_date | end_date | net_work_days |
|---|---|---|---|
| 1 | 01/01/2022 | 01/18/2022 | 12 |
| 2 | 02/10/2022 | 02/17/2022 | 6 |
| 3 | 05/17/2022 | 07/05/2022 | 36 |
| 4 | 06/01/2022 | 06/12/2022 | 8 |
| 5 | 08/10/2022 | 08/26/2022 | 12 |

De enkele, geplande vakantiedag is ingevoerd als het derde argument in de `networkdays()`-functie.

*Kalenderdiagram dat werkdagen voor project 5 (één vakantiedag) markeert*

De bovenstaande kalender markeert project 5 visueel en toont deze aanpassing waarbij de vakantiedag is toegevoegd. Deze vakantiedag valt in project 5 op vrijdag, 19 augustus 2022. Het resultaat is dat de totale `net_work_days`-waarde voor project 5 afneemt met één dag, van 13 naar 12 dagen.

## Voorbeeld 3 – Meerdere vakantiedagen

### Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.
- Er wordt een extra veld gemaakt, `net_work_days`, om het aantal werkdagen voor elk project te berekenen.

Maar in dit voorbeeld staan er vier vakantiedagen gepland, van 18 augustus t/m 21 augustus 2022.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `start_date`
- `end_date`
- `net_work_days`

Resultatentabel

| id | start_date | end_date | net_work_days |
|---|---|---|---|
| 1 | 01/01/2022 | 01/18/2022 | 12 |
| 2 | 02/10/2022 | 02/17/2022 | 6 |
| 3 | 05/17/2022 | 07/05/2022 | 36 |
| 4 | 06/01/2022 | 06/12/2022 | 8 |
| 5 | 08/10/2022 | 08/26/2022 | 11 |

De vier geplande vakantiedagen worden ingevoerd als een door komma's gescheiden lijst, vanaf het derde argument tot in de `networkdays()`-functie.

*Kalenderdiagram dat werkdagen voor project 5 (meerdere vakantiedagen) markeert*

| 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 Holiday | 19 Holiday | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| 28 | 29 | 30 | 31 | | | |

De bovenstaande kalender markeert project 5 visueel en toont deze aanpassing waarbij deze vakantiedagen zijn toegevoegd. Deze periode van geplande vakantiedagen vindt plaats tijdens project 5, waarbij twee van de dagen op een donderdag en vrijdag vallen. Het resultaat is dat de totale `net_work_days`-waarde voor project 5 afneemt van 13 naar 11 dagen.

## Voorbeeld 4 - Enkele vakantiedag

### Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.

Er staat één vakantiedag gepland op 19 augustus 2022.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. Het `net_work_days`-veld wordt berekend als een meting in een diagramobject.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `start_date`
- `end_date`

Maak de volgende meting:

```
= networkdays(start_date,end_date,'08/19/2022')
```

Resultatentabel

| id | start_date | end_date | net_work_days |
|---|---|---|---|
| 1 | 01/01/2022 | 01/18/2022 | 12 |
| 2 | 02/10/2022 | 02/17/2022 | 6 |
| 3 | 05/17/2022 | 07/05/2022 | 36 |
| 4 | 06/01/2022 | 06/12/2022 | 8 |
| 5 | 08/10/2022 | 08/26/2022 | 12 |

De enkele, geplande vakantiedag is ingevoerd als het derde argument in de `networkdays()`-functie.

*Het kalenderdiagram toont nettowerkdagen met één vakantiedag (diagramobject)*

| 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 | | | |

De bovenstaande kalender markeert project 5 visueel en toont deze aanpassing waarbij de vakantiedag is toegevoegd. Deze vakantiedag valt in project 5 op vrijdag, 19 augustus 2022. Het resultaat is dat de totale `net_work_days`-waarde voor project 5 afneemt met één dag, van 13 naar 12 dagen.

## now

Deze functie retourneert een tijdstempel van de huidige tijd. Deze functie retourneert waarden in de notatie van de **TimeStamp**-systeemvariabele. De standaardwaarde voor **timer_mode** is 1.

**Syntaxis:**

```
now([ timer_mode])
```

**Retourgegevenstypen:** dubbele waarde

De `now()`-functie kan ofwel in het load-script ofwel in diagramobjecten worden gebruikt.

Argumenten

| Argument | Beschrijving |
|---|---|
| timer_mode | Kan de volgende waarden hebben:<br><br>0 (tijd van de als laatste voltooide laadbewerking voor gegevens)<br>1 (tijd van de functieaanroep)<br>2 (tijd wanneer de app werd geopend)<br><br>*Als u de functie gebruikt in een script voor het laden van gegevens, resulteert **timer_mode=0** in de tijd van de als laatste voltooide laadbewerking voor gegevens, terwijl **timer_mode=1** de tijd van de functieaanroep in de huidige laadbewerking voor gegevens oplevert.* |

*De `now()`-functie heeft een impact met hoge snelheid, wat kan resulteren in scrolproblemen als de functie wordt gebruikt in de uitdrukkingen van tabellen. Wanneer het gebruik niet strikt noodzakelijk is, raden we in plaats hiervan aan de `today()`-functie te gebruiken. Als het gebruik van `now()` vereist is in een lay-out, raden we aan waar mogelijk de niet-standaardinstellingen `now(0)` of `now(2)` te gebruiken omdat deze instellingen niet continue herberekeningen vereisen*

## Wanneer gebruiken

De `now()`-functie wordt meestal gebruikt als een component in een uitdrukking. Hij kan bijvoorbeeld worden gebruikt om de resterende tijd te berekenen in een levenscyclus van een project. De `now()`-functie zou in plaats van de `today()`-functie worden gebruikt wanneer de uitdrukking het gebruik van een deel van een dag vereist.

De volgende tabel geeft een uitleg van het resultaat dat door de `now()`-functie is geretourneerd en geeft verschillende waarden voor het `timer_mode`-argument:

Voorbeelden van functies

| timer_mode-waarde | Resultaat bij gebruik in load-script | Resultaat bij gebruik in diagramobject |
|---|---|---|
| 0 | Retourneert een tijdstempel met de notatie van de `TimeStamp`-systeemvariabele, van de laatste keer dat de gegevens opnieuw zijn geladen voorafgaand aan de laatste keer dat de gegevens opnieuw zijn geladen. | Retourneert een tijdstempel met de notatie van de `TimeStamp`-systeemvariabele voor de laatste keer dat de gegevens opnieuw zijn geladen. |
| 1 | Retourneert een tijdstempel met de notatie van de `TimeStamp`-systeemvariabele voor de laatste keer dat de gegevens opnieuw zijn geladen. | Retourneert een tijdstempel met de notatie van de `TimeStamp`-systeemvariabele van de functieaanroep. |

| timer_mode-waarde | Resultaat bij gebruik in load-script | Resultaat bij gebruik in diagramobject |
|---|---|---|
| 2 | Retourneert een tijdstempel met de notatie van de `TimeStamp`-systeemvariabele voor het begin van de sessie van de gebruiker in de applicatie. Dit wordt niet bijgewerkt tenzij de gebruiker het script opnieuw laadt. | Retourneert een tijdstempel met de notatie van de `TimeStamp`-systeemvariabele voor het begin van de sessie van de gebruiker in de applicatie. Dit wordt vernieuwd zodra een nieuwe sessie begint of de gegevens weer in de applicatie zijn geladen. |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – Genereren van objecten met behulp van het load-script

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Dit voorbeeld creëert die variabelen met behulp van de `now()`-functie. Elke variabele gebruikt een van de `timer_mode`-opties om het effect van de variabele te demonstreren.

Laad het script opnieuw en herlaad het na een korte tijdsperiode zodat de variabelen hun doel kunnen laten zien. De variabelen `now(0)` en `now(1)` zullen nu verschillende waarden laten zien en hun doel demonstreren.

### Load-script

```
LET vPreviousDataLoad = now(0);
LET vCurrentDataLoad = now(1);
LET vApplicationOpened = now(2);
```

### Resultaten

Nadat de gegevens voor de tweede keer zijn geladen, maakt u drie tekstvakken door de onderstaande instructies te volgen.

Maak eerst een tekstvak voor de gegevens die eerder zijn geladen.

**Doe het volgende:**

1. Gebruik het diagramobject **Tekst en afbeelding** om een tekstvak te maken.
2. Voeg de volgende meting toe aan het object:
   `=vPreviousDataLoad`
3. Onder **Uiterlijk** selecteert u **Show titles** en voegt u de titel 'Previous Reload Time' toe aan het object.

Maak vervolgens een tekstvak voor de gegevens die momenteel worden geladen.

**Doe het volgende:**

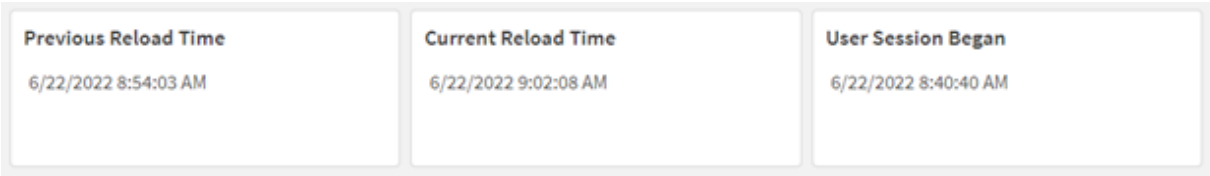
1. Gebruik het diagramobject **Tekst en afbeelding** om een tekstvak te maken.
2. Voeg de volgende meting toe aan het object:
   `=vCurrentDataLoad`
3. Onder **Uiterlijk** selecteert u **Show titles** en voegt u de titel 'Current Reload Time' toe aan het object.

Maak een definitief tekstvak dat wordt getoond als de sessie van de gebruiker in de toepassing wordt gestart.

**Doe het volgende:**

1. Gebruik het diagramobject **Tekst en afbeelding** om een tekstvak te maken.
2. Voeg de volgende meting toe aan het object:
   `=vApplicationOpened`
3. Onder **Uiterlijk** selecteert u **Show titles** en voegt u de titel 'User Session Started' toe aan het object.

*now()-load-scriptvariabelen*




De bovenstaande afbeelding geeft voorbeeldwaarden weer voor elk van de gemaakte variabelen. De waarden zouden bijvoorbeeld als volgt kunnen zijn:

- Vorige herlaadtijd: 6/22/2022 8:54:03 AM
- Huidige herlaadtijd: 6/22/2022 9:02:08 AM
- Start gebruikerssessie: 6/22/2022 8:40:40 AM

## Voorbeeld 2 – Genereren van objecten zonder het load-script

Load-script en diagramuitdrukking

### Overzicht

In dit voorbeeld maakt u drie diagramobjecten met de `now()`-functie zonder dat er variabelen of gegevens in de applicatie worden geladen. Elk diagramobject gebruikt een van de `timer_mode`-opties om het effect te demonstreren.

Er is geen load-script voor dit voorbeeld.

### Doe het volgende:

1. Open de editor voor laden van gegevens.
2. Klik zonder het bestaande load-script te wijzigen op **Gegevens laden**.
3. Laad het script na korte tijd nog een tweede keer.

### Resultaten

Maak drie tekstvakken zodra de gegevens voor een tweede keer zijn geladen.

Maak eerst een tekstvak voor de laatste keer dat de gegevens opnieuw zijn geladen.

### Doe het volgende:

1. Gebruik het diagramobject **Tekst en afbeelding** om een tekstvak te maken.
2. Voeg de volgende meting toe:
   `=now(0)`
3. Onder **Uiterlijk** selecteert u **Titels tonen**en voegt u de titel Laatste herlaadtijd gegevens toe aan het object.

Maak nu een tekstvak om de actuele tijd te tonen.

### Doe het volgende:

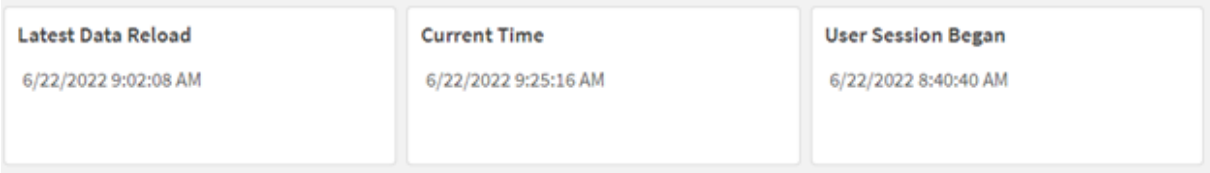
1. Gebruik het diagramobject **Tekst en afbeelding** om een tekstvak te maken.
2. Voeg de volgende meting toe:
   `=now(1)`
3. Onder **Uiterlijk** selecteert u **Titels tonen** en voegt u de titel Actuele tijd toe aan het object.

Maak een definitief tekstvak dat wordt getoond als de sessie van de gebruiker in de toepassing wordt gestart.

**Doe het volgende:**

1. Gebruik het diagramobject **Tekst en afbeelding** om een tekstvak te maken.
2. Voeg de volgende meting toe:
   `=now(2)`
3. Onder **Uiterlijk** selecteert u **Titels tonen** en voegt u de titel Start gebruikerssessie toe aan het object.

*Voorbeelden van `now()`-diagramobject*




De bovenstaande afbeelding geeft voorbeeldwaarden weer voor elk van de gemaakte objecten. De waarden zouden bijvoorbeeld als volgt kunnen zijn:

- Laatste herlaadtijd gegevens: 6/22/2022 9:02:08 AM
- Huidige tijd: 6/22/2022 9:25:16 AM
- Start gebruikerssessie: 6/22/2022 8:40:40 AM

Het diagramobject Laatste herlaadtijd gegevens maakt gebruik van een `timer_mode`-waarde van 0. Dit retourneert de tijdstempel voor de laatste keer dat de gegevens opnieuw zijn geladen.

Het diagramobject Actuele tijd gebruikt een `timer_mode`-waarde van 1. Dit retourneert de actuele tijd volgens de systeemklok. Als het werkblad of object wordt vernieuwd, wordt deze waarde bijgewerkt.

Het diagramobject Start gebruikerssessie gebruikt een `timer_mode`-waarde van 2. Dit retourneert de tijdstempel voor het moment waarop de applicatie werd geopend en waarop de gebruikerssessie is begonnen.

## Voorbeeld 3 – Scenario

Load-script en diagramuitdrukking

**Overzicht**

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die bestaat uit de voorraad voor een cryptovaluta-miningactiviteit die wordt geladen in de tabel `Inventory`.
- Gegevens met de volgende velden: `id`, `purchase_date` en `wph` (watt per uur).

De gebruiker wil een tabel hebben die de totale kosten per `id` weergeeft die iedere mining-rig in de maand tot nu toe heeft gemaakt wat betreft stroomverbruik.

Deze waarde moet worden bijgewerkt wanneer het diagramobject wordt vernieuwd. De huidige prijs voor elektriciteit is USD 0,0678 per kWu.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `id`.

Maak de volgende meting:

```
=(now(1)-monthstart(now(1)))*24*wph/1000*0.0678
```

Als het diagramobject is vernieuwd op 6/22/2022 om 10:39:05 AM, dan zou het de volgende resultaten retourneren:

Resultatentabel

| id | =(now(1)-monthstart(now(1)))*24*wph/1000*0.0678 |
|---|---|
| 8188 | $39.18 |
| 8189 | $49.97 |
| 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 |

De gebruiker wil graag dat de objectresultaten elke keer dat het object wordt vernieuwd, ook worden vernieuwd. Daarom staat het `timer_mode`-argument in instanties van de `now()`-functie in de uitdrukking. De tijdstempel voor het begin van de maand, die is vastgesteld met behulp van de `now()`-functie als het tijdstempelargument in de `monthstart()`-functie, wordt afgetrokken van de huidige tijd die wordt vastgesteld door de `now()`-functie. Dit verstrekt de totale hoeveelheid tijd die tot nu doe in deze maand (in dagen) is verstreken.

Deze waarde wordt vermenigvuldigd met 24 (het aantal uren in een dag) en vervolgens met de waarde in het `wph`-veld.

Voor de conversie van watt per uur naar kilowatt per uur, wordt het resultaat gedeeld door 1000 voordat het uiteindelijk wordt vermenigvuldigd met het verstrekte kWu-tarief.

## quarterend

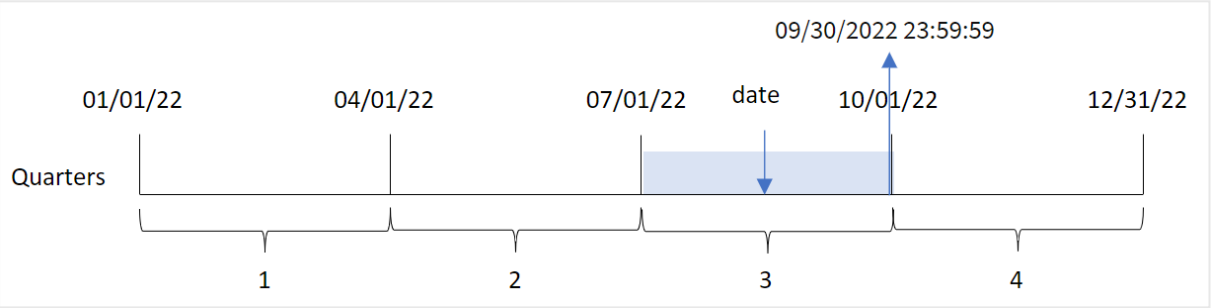
Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de laatste milliseconde van het kwartaal dat **date** bevat. De standaarduitvoernotatie is de **DateFormat** die is ingesteld in het script.

**Syntaxis:**

```
QuarterEnd(date[, period_no[, first_month_of_year]])
```

**Retourgegevenstypen:** dubbele waarde

*Diagram van de `quarterend()`-functie*




De `quarterend()`-functie bepaalt welk kwartaal de datum in valt. Hij retourneert vervolgens een tijdstempel, in datumnotatie, voor de laatste milliseconde van de laatste maand van dat kwartaal. De eerste maand van het jaar is standaard januari. U kunt echter ook wijzigen welke maand als eerste wordt ingesteld met behulp van het argument `first_month_of_year` in de functie `quarterend()` te gebruiken.

> *De `quarterend()`-functie houdt geen rekening met de systeemvariabele `FirstMonthOfYear`. Het jaar begint op 1 januari, tenzij het argument `first_month_of_year` wordt gebruikt om het te wijzigen.*

### Wanneer gebruiken

De `quarterend()`-functie wordt voornamelijk gebruikt als onderdeel van een uitdrukking als u wilt dat in de berekening een deel van het kwartaal wordt gebruikt dat nog niet heeft plaatsgevonden. Bijvoorbeeld als u de totale rente wilt berekenen die nog niet is berekend gedurende het kwartaal.

Argumenten

| Argument | Beschrijving |
|---|---|
| **date** | De te evalueren datum of tijdstempel. |
| **period_no** | **period_no** is een geheel getal, waarbij de waarde 0 het kwartaal aangeeft dat **date** bevat. Negatieve waarden in **period_no** geven voorafgaande kwartalen aan en positieve waarden geven volgende kwartalen aan. |

| Argument | Beschrijving |
|---|---|
| **first_month_of_year** | Als u wilt werken met (boek)jaren die niet in januari beginnen, geeft u een waarde op tussen 2 en 12 in **first_month_of_year**. |

U kunt de volgende waarden gebruiken om de eerste maand van het jaar in te stellen in het argument `first_month_of_year`:

waarden first_month_of_year

| Maand | Waarde |
|---|---|
| Februari | 2 |
| Maart | 3 |
| April | 4 |
| Mei | 5 |
| Juni | 6 |
| Juli | 7 |
| Augustus | 8 |
| September | 9 |
| Oktober | 10 |
| November | 11 |
| December | 12 |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `quarterend('10/29/2005')` | Retourneert 12/31/2005 23:59:59. |
| `quarterend('10/29/2005', -1)` | Retourneert 09/30/2005 23:59:59. |
| `quarterend('10/29/2005', 0, 3)` | Retourneert 11/30/2005 23:59:59. |

## Voorbeeld 1 – Basisvoorbeeld

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een reeks transacties in 2022 die wordt geladen in de tabel `Transactions`.
- Een voorgaande lading die het volgende bevat:
  - De `quarterend()`-functie die is ingesteld als het `end_of_quarter`-veld en een tijdstempel retourneert voor het eind van het kwartaal waarin de transacties plaatsvonden.
  - De `timestamp()`-functie die is ingesteld als het `end_of_quarter_timestamp`-veld en de exacte tijdstempel retourneert voor het eind van het geselecteerde kwartaal.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `date`
- `end_of_quarter`
- `end_of_quarter_timestamp`

Resultatentabel

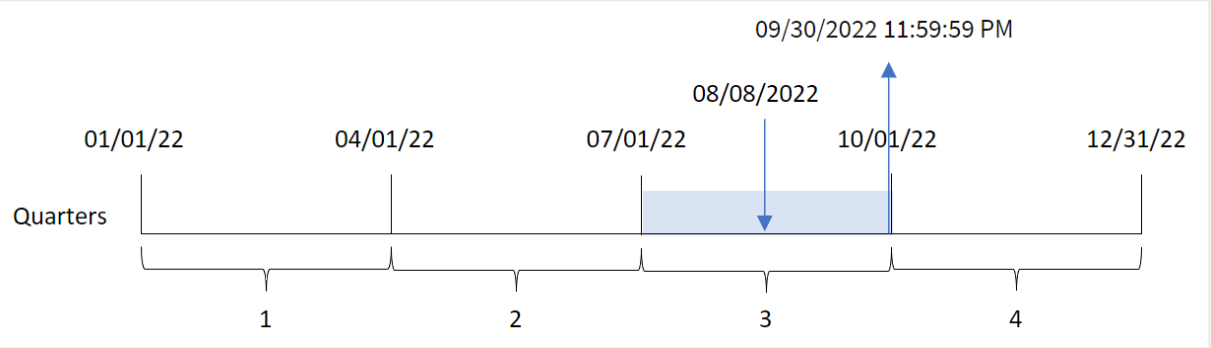
| id | date | end_of_quarter | end_of_quarter_timestamp |
|---|---|---|---|
| 8188 | 1/7/2022 | 03/31/2022 | 3/31/2022 11:59:59 PM |
| 8189 | 1/19/2022 | 03/31/2022 | 3/31/2022 11:59:59 PM |
| 8190 | 2/5/2022 | 03/31/2022 | 3/31/2022 11:59:59 PM |
| 8191 | 2/28/2022 | 03/31/2022 | 3/31/2022 11:59:59 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 |

| id | date | end_of_quarter | end_of_quarter_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 | 12/31/2022 | 12/31/2022 11:59:59 PM |
| 8207 | 10/29/2022 | 12/31/2022 | 12/31/2022 11:59:59 PM |

Het veld '`end_of_quarter`' wordt gemaakt in de voorafgaande load-instructie met behulp van de functie `quarterend()` en geeft het datumveld door als het argument van de functie.

De `quarterend()`-functie identificeert aanvankelijk in welk kwartaal de datumwaarde valt en retourneert een tijdstempel voor de laatste milliseconde van dat kwartaal.

*Diagram van de `quarterend()`-functie waarbij het kwartaaleinde van transactie 8203 is geïdentificeerd*




Transactie 8203 vond plaats op 8 augustus. De `quarterend()`-functie identificeert dat de transactie in het derde kwartaal plaatsvond en retourneert de laatste milliseconde van dat kwartaal. Dat is 30 september om 11:59:59 PM.

## Voorbeeld 2 – period_no

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met een reeks transacties in 2022 die wordt geladen in de tabel `Transactions`.
- Een voorgaande lading die het volgende bevat:
  - De `quarterend()`-functie die is ingesteld als het `previous_quarter_end`-veld en een tijdstempel retourneert voor het eind van het kwartaal vóór het kwartaal waarin de transactie plaatsvond.

- De `timestamp()`-functie die is ingesteld als het `previous_end_of_quarter_timestamp`-veld en de exacte tijdstempel retourneert voor het eind van het kwartaal vóór het kwartaal waarin de transactie plaatsvond.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

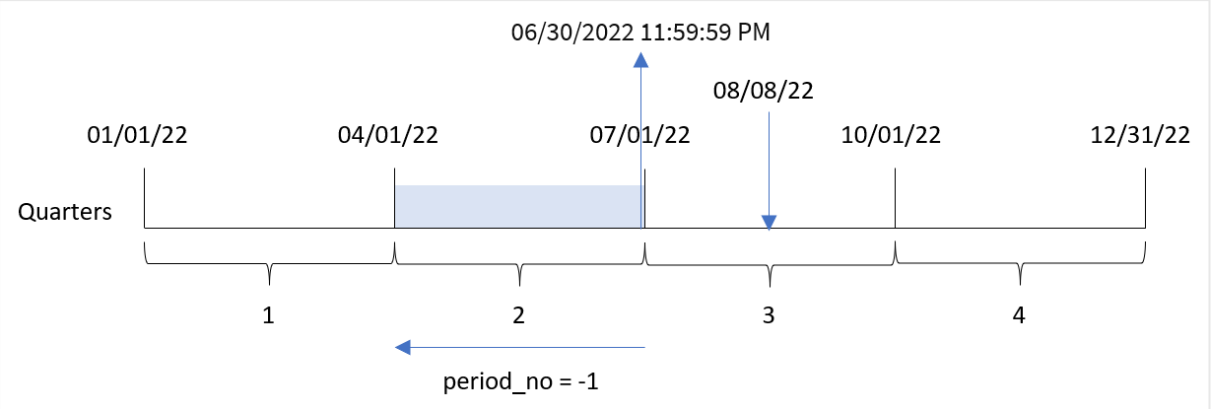
- `id`
- `date`
- `previous_quarter_end`
- `previous_quarter_end_timestamp`

Resultatentabel

| 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 |
| 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 |

Omdat een `period_no` van `-1` wordt gebruikt als het offset-argument in de functie `quarterend()`, identificeert de functie eerst het kwartaal waarin de transacties plaatsvinden. Het verschuift dan naar een kwartaal eerder en identificeert de laatste milliseconde van dat kwartaal.

*Diagram van de quarterend()-functie met een period_no van -1.*




Transactie 8203 vond plaats op 8 augustus. De `quarterend()`-functie identificeert dat het kwartaal vóór het kwartaal waarin de transactie plaatsvond tussen 1 april en 30 juni viel. De functie retourneert vervolgens de laatste milliseconde van dat kwartaal, namelijk 30 juni om 11:59:59 PM.

## Voorbeeld 3 – first_month_of_year

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met een reeks transacties in 2022 die wordt geladen in de tabel `Transactions`.
- Een voorgaande lading die het volgende bevat:
  - De `quarterend()`-functie die is ingesteld als het `end_of_quarter`-veld en een tijdstempel retourneert voor het eind van het kwartaal waarin de transacties plaatsvonden.
  - De `timestamp()`-functie die is ingesteld als het `end_of_quarter_timestamp`-veld en de exacte tijdstempel retourneert voor het eind van het geselecteerde kwartaal.

Maar in dit voorbeeld is het bedrijfsbeleid dat het boekjaar op 1 maart begint.

### Load-script

```
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
];
```

**Resultaten**

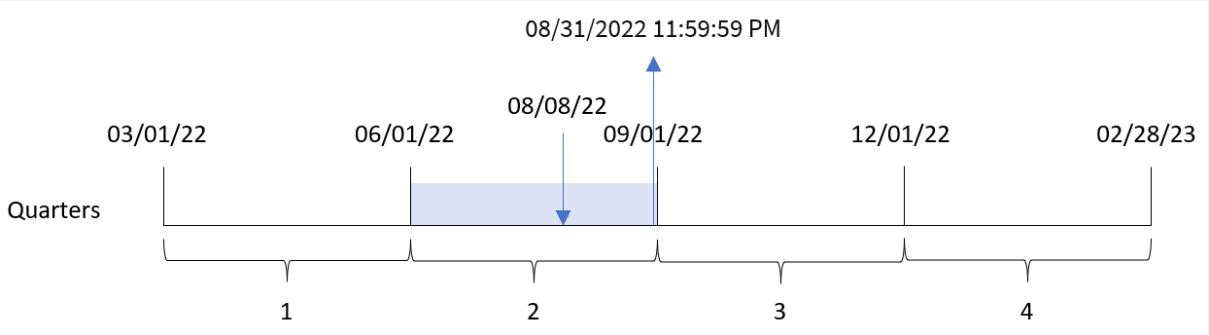
Resultatentabel

| 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 |

| id | date | end_of_quarter | end_of_quarter_timestamp |
|---|---|---|---|
| 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 |

Omdat het `first_month_of_year`-argument van `3` is gebruikt in de `quarterend()`-functie, wordt het begin van het jaar verplaatst van 1 januari naar 1 maart.

*Diagram van de `quarterend()`-functie waarbij maart is ingesteld als de eerste maand van het jaar*




Transactie 8203 vond plaats op 8 augustus. Omdat het begin van het jaar 1 maart is, liggen de kwartalen van het jaar als volgt maa-mei, jun-aug, sep-nov en dec-feb.

De `quarterend()`-functie identificeert dat de transactie in het kwartaal tussen het begin van juni en augustus plaatsvond en retourneert de laatste milliseconde van dat kwartaal. Dat is 31 augustus om 11:59:59 PM.

## Voorbeeld 4 – Diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is de gegevensverzameling echter niet gewijzigd en wordt in de applicatie geladen. De berekening die een tijdstempel retourneert voor het eind van het kwartaal waarin de transacties plaatsvonden is gemaakt als meting in een diagramobject in de app.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `date`

Maak de volgende metingen om de einddatum van het kwartaal waarin een transactie plaatsvindt te berekenen:

- `=quarterend(date)`
- `=timestamp(quarterend(date))`

Resultatentabel

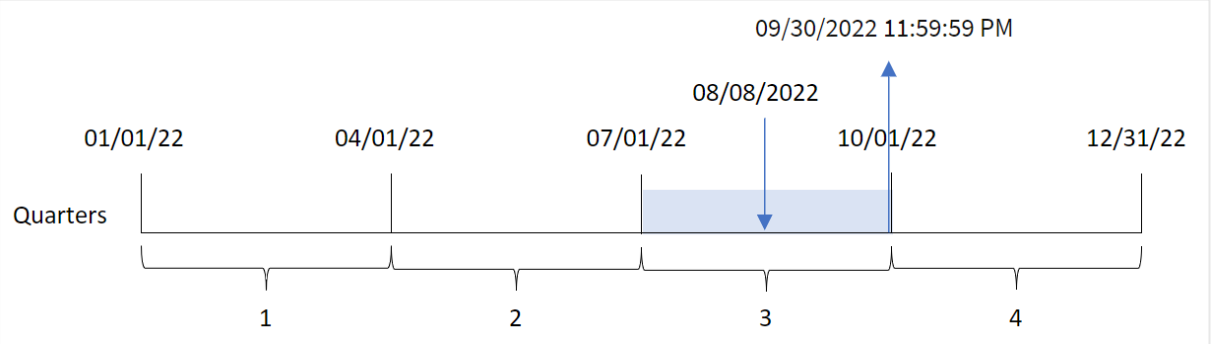
| id | date | =quarterend(date) | =timestamp(quarterend(date)) |
|---|---|---|---|
| 8188 | 1/7/2022 | 03/31/2022 | 3/31/2022 11:59:59 PM |
| 8189 | 1/19/2022 | 03/31/2022 | 3/31/2022 11:59:59 PM |
| 8190 | 2/5/2022 | 03/31/2022 | 3/31/2022 11:59:59 PM |
| 8191 | 2/28/2022 | 03/31/2022 | 3/31/2022 11:59:59 PM |
| 8192 | 3/16/2022 | 03/31/2022 | 3/31/2022 11:59:59 PM |

| id | date | =quarterend(date) | =timestamp(quarterend(date)) |
|---|---|---|---|
| 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 |

Het veld '`end_of_quarter`' wordt gemaakt in de voorafgaande load-instructie met behulp van de functie `quarterend()` en geeft het datumveld door als het argument van de functie.

De `quarterend()`-functie identificeert aanvankelijk in welk kwartaal de datumwaarde valt en retourneert een tijdstempel voor de laatste milliseconde van dat kwartaal.

*Diagram van de `quarterend()`-functie waarbij het kwartaaleinde van transactie 8203 is geïdentificeerd*




Transactie 8203 vond plaats op 8 augustus. De `quarterend()`-functie identificeert dat de transactie in het derde kwartaal plaatsvond en retourneert de laatste milliseconde van dat kwartaal. Dat is 30 september om 11:59:59 PM.

## Voorbeeld 5 – Scenario

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset wordt geladen in een tabel met de naam '`Employee_Expenses`'. De tabel bevat de volgende velden:
    - Werknemers-id's
    - Werknemersnamen
    - De gemiddelde dagelijkse onkostendeclaraties van elke werknemer.

De eindgebruiker wil graag een diagramobject dat per werknemer-id en werknemernaam de geschatte nog te maken onkostendeclaraties voor de rest van het kwartaal weergeeft. Het boekjaar begint in januari.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `employee_id`
- `employee_name`

Maak de volgende meting om de opgebouwde rente te berekenen:

- `=(quarterend(today(1))-today(1))*avg_daily_claim`

Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

| employee_id | employee_name | =(quarterend(today(1))-today(1))*avg_daily_claim |
|---|---|---|
| 182 | Mark | $480.00 |
| 183 | Deryck | $400.00 |
| 184 | Dexter | $400.00 |
| 185 | Sydney | $864.00 |
| 186 | Agatha | $576.00 |

De `quarterend()`-functie gebruikt de datum van vandaag als het enige argument en retourneert de einddatum van de huidige maand. Door vervolgens de datum van vandaag af te trekken van de einddatum van het jaar, retourneert de uitdrukking het aantal resterende dagen in deze maand.

Deze waarde wordt vervolgens vermenigvuldigd met de gemiddelde dagelijkse onkostendeclaratie van elke werknemer om de geschatte waarde te berekenen van de declaraties die elke werknemer naar verwachting in het resterende kwartaal zal indienen.

## quartername

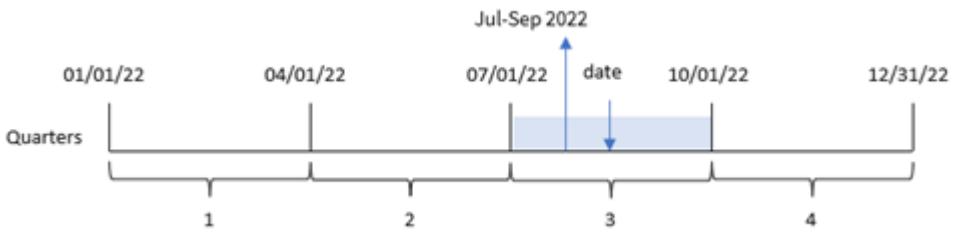
Deze functie retourneert de maanden van het kwartaal (opgemaakt volgens de scriptvariabele **MonthNames**) en het jaar met een onderliggende numerieke waarde die overeenkomt met een tijdstempel van de eerste milliseconde van de eerste dag van het kwartaal.

**Syntaxis:**

```
QuarterName(date[, period_no[, first_month_of_year]])
```

**Retourgegevenstypen:** dubbele waarde

*Diagram van de `quartername()`-functie*




De `quartername()`-functie bepaalt welk kwartaal de datum in valt. Hij retourneert vervolgens een waarde die de begin- en eindmaanden van het kwartaal en van het jaar laten zien. De achterliggende numerieke waarde van dit resultaat is de eerste milliseconde van het kwartaal.

Argumenten

| Argument | Beschrijving |
|---|---|
| **date** | De te evalueren datum of tijdstempel. |
| **period_no** | **period_no** is een geheel getal, waarbij de waarde 0 het kwartaal aangeeft dat **date** bevat. Negatieve waarden in **period_no** geven voorafgaande kwartalen aan en positieve waarden geven volgende kwartalen aan. |
| **first_month_of_year** | Als u wilt werken met (boek)jaren die niet in januari beginnen, geeft u een waarde op tussen 2 en 12 in **first_month_of_year**. |

## Wanneer gebruiken

De `quartername()`-functie is handig als u aggregaties per kwartaal wilt vergelijken. Bijvoorbeeld als u de totale verkoop van producten per kwartaal wilt zien.

Deze functie kan in het load-script worden gebruikt om een veld in een masterkalendertabel te maken. Hij zou ook rechtstreeks in een diagram kunnen worden gebruikt als een berekende dimensie.

Deze voorbeelden gebruiken de datumopmaak MM/DD/YYYY. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` boven aan uw script voor het laden van gegevens. Wijzig de opmaak in de voorbeelden zodat deze in uw behoeften voorziet.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `quartername('10/29/2013')` | Retourneert Oct-Dec 2013. |
| `quartername('10/29/2013', -1)` | Retourneert Jul-Sep 2013. |
| `quartername('10/29/2013', 0, 3)` | Retourneert Sep-Nov 2013. |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – Datum zonder aanvullende argumenten

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties voor 2022 die wordt geladen in de tabel `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.
- Het maken van een veld, `transaction_quarter`, dat het kwartaal retourneert waarin de transacties plaatsvonden.

Voeg hier uw andere tekst toe, indien nodig, met lijsten enz.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `transaction_quarter`

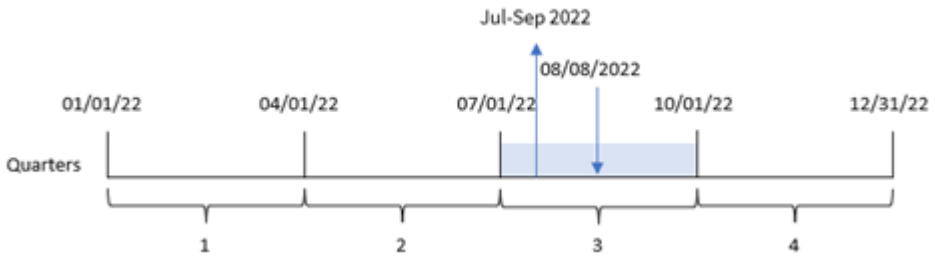
Resultatentabel

| date | transaction_quarter |
|---|---|
| 1/7/2022 | jan-mrt 2022 |
| 1/19/2022 | jan-mrt 2022 |
| 2/5/2022 | jan-mrt 2022 |
| 2/28/2022 | jan-mrt 2022 |
| 3/16/2022 | jan-mrt 2022 |
| 4/1/2022 | april-jun 2022 |
| 5/7/2022 | april-jun 2022 |
| 5/16/2022 | april-jun 2022 |
| 6/15/2022 | april-jun 2022 |
| 6/26/2022 | april-jun 2022 |
| 7/9/2022 | juli-sept 2022 |
| 7/22/2022 | juli-sept 2022 |
| 7/23/2022 | juli-sept 2022 |
| 7/27/2022 | juli-sept 2022 |
| 8/2/2022 | juli-sept 2022 |
| 8/8/2022 | juli-sept 2022 |
| 8/19/2022 | juli-sept 2022 |
| 9/26/2022 | juli-sept 2022 |
| 10/14/2022 | okt-dec 2022 |
| 10/29/2022 | okt-dec 2022 |

Het veld '`transaction_quarter`' wordt gemaakt in de voorafgaande load-instructie met behulp van de functie `quartername()` en geeft het datumveld door als het argument van de functie.

De `quartername()`-functie identificeert eerst het kwartaal waarin de datumwaarde valt. Hij retourneert vervolgens een waarde die de begin- en eindmaanden van het kwartaal en van het jaar laten zien.

*Diagram van `quartername()`-functie, voorbeeld zonder aanvullende argumenten*




Transactie 8203 vond plaats op 8 augustus 2022. De `quartername()`-functie identificeert dat de transactie plaatsvond in het derde kwartaal en retourneert daarom jul-sep 2022. De maanden worden weergegeven met dezelfde notatie als de `MonthNames`-systeemvariabele.

## Voorbeeld 2 – Datum met period_no argument

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het veld `previous_quarter` wordt gemaakt en retourneert het kwartaal dat voor het kwartaal valt waarin de transacties plaatsvonden.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `previous_quarter`

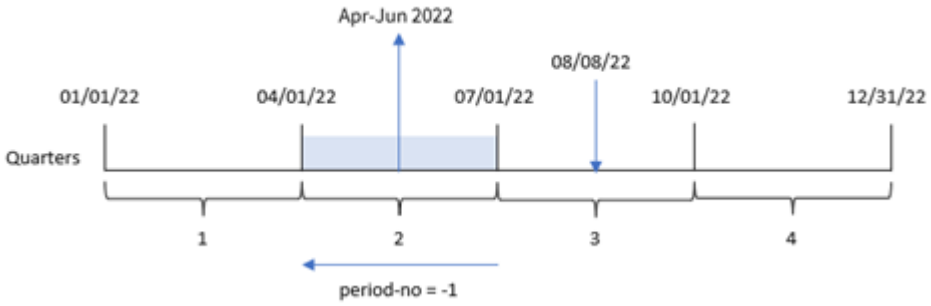
Resultatentabel

| date | previous_quarter |
|---|---|
| 1/7/2022 | okt-dec 2021 |
| 1/19/2022 | okt-dec 2021 |
| 2/5/2022 | okt-dec 2021 |
| 2/28/2022 | okt-dec 2021 |
| 3/16/2022 | okt-dec 2021 |
| 4/1/2022 | jan-mrt 2022 |
| 5/7/2022 | jan-mrt 2022 |
| 5/16/2022 | jan-mrt 2022 |
| 6/15/2022 | jan-mrt 2022 |
| 6/26/2022 | jan-mrt 2022 |
| 7/9/2022 | april-jun 2022 |
| 7/22/2022 | april-jun 2022 |
| 7/23/2022 | april-jun 2022 |
| 7/27/2022 | april-jun 2022 |
| 8/2/2022 | april-jun 2022 |
| 8/8/2022 | april-jun 2022 |
| 8/19/2022 | april-jun 2022 |

| date | previous_quarter |
|---|---|
| 9/26/2022 | april-jun 2022 |
| 10/14/2022 | juli-sept 2022 |
| 10/29/2022 | juli-sept 2022 |

In dit geval, omdat een `period_no` van -1 is gebruikt als het offset-argument in de `quartername()`-functie, identificeert de functie eerst dat de transacties in het derde kwartaal hebben plaatsgevonden. Vervolgens wordt dit naar één kwartaal eerder verschoven en wordt een waarde geretourneerd die de begin- en eindmaanden van het kwartaal en van het jaar laten zien.

*Diagram van `quartername()`-functie, voorbeeld period_no*




Transactie 8203 vond plaats op 8 augustus. De `quartername()`-functie identificeert dat het kwartaal vóór het kwartaal waarin de transactie plaatsvond tussen 1 april en 30 juni viel. Daarom wordt apr-jun 2022 geretourneerd.

## Voorbeeld 3 – Datum met first_week_day-argument

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld. In dit voorbeeld moeten we echter 1 maart instellen als het begin van het boekjaar.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `transaction_quarter`

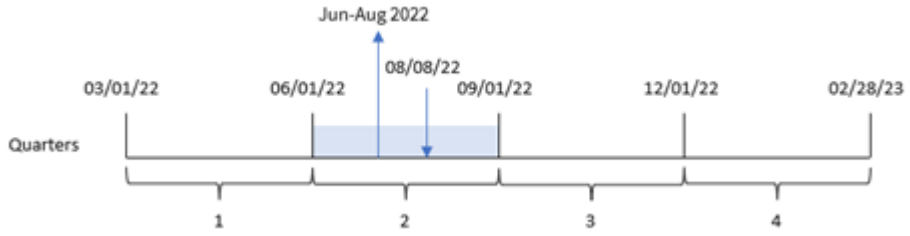
Resultatentabel

| date | transaction_quarter |
|---|---|
| 1/7/2022 | Dec-feb 2021 |
| 1/19/2022 | Dec-feb 2021 |
| 2/5/2022 | Dec-feb 2021 |
| 2/28/2022 | Dec-feb 2021 |
| 3/16/2022 | Maa-mei 2022 |
| 4/1/2022 | Maa-mei 2022 |
| 5/7/2022 | Maa-mei 2022 |
| 5/16/2022 | Maa-mei 2022 |
| 6/15/2022 | Jun-aug 2022 |
| 6/26/2022 | Jun-aug 2022 |

| date | transaction_quarter |
|---|---|
| 7/9/2022 | Jun-aug 2022 |
| 7/22/2022 | Jun-aug 2022 |
| 7/23/2022 | Jun-aug 2022 |
| 7/27/2022 | Jun-aug 2022 |
| 8/2/2022 | Jun-aug 2022 |
| 8/8/2022 | Jun-aug 2022 |
| 8/19/2022 | Jun-aug 2022 |
| 9/26/2022 | Sep-nov 2022 |
| 10/14/2022 | Sep-nov 2022 |
| 10/29/2022 | Sep-nov 2022 |

In deze instantie wordt het begin van het jaar van 1 januari naar 1 maart verplaatst omdat het `first_month_of_year`-argument van 3 is gebruikt in de `quartername()`-functie. Daarom worden de kwartalen van het jaar verdeeld in maart-mei, juni-augustus, september-november en december-februari.

*Diagram van de `quartername()`-functie, voorbeeld met first_week_day*




Transactie 8203 vond plaats op 8 augustus. De `quartername()`-functie identificeert dat de transactie plaatsvond in het tweede kwartaal, tussen het begin van juni en het eind van augustus. Daarom wordt jun-aug 2022 geretourneerd.

## Voorbeeld 4 – diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die een tijdstempel retourneert voor het eind van het kwartaal waarin de transacties plaatsvonden is gemaakt als meting in een diagramobject van de toepassing.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `date`.

Maak de volgende meting:

```
=quartername(date)
```

Resultatentabel

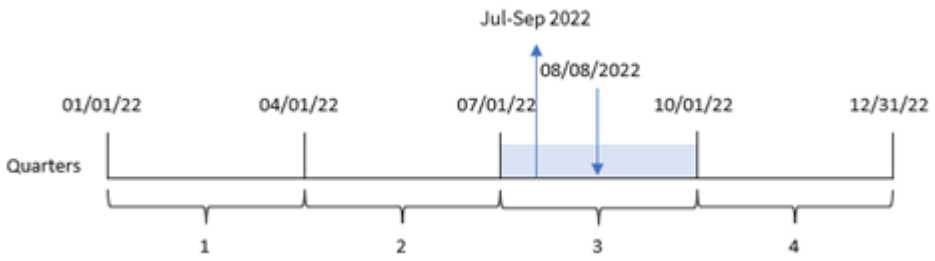
| date | =quartername(date) |
|---|---|
| 1/7/2022 | jan-mrt 2022 |
| 1/19/2022 | jan-mrt 2022 |
| 2/5/2022 | jan-mrt 2022 |
| 2/28/2022 | jan-mrt 2022 |
| 3/16/2022 | jan-mrt 2022 |
| 4/1/2022 | april-jun 2022 |
| 5/7/2022 | april-jun 2022 |

| date | =quartername(date) |
|---|---|
| 5/16/2022 | april-jun 2022 |
| 6/15/2022 | april-jun 2022 |
| 6/26/2022 | april-jun 2022 |
| 7/9/2022 | juli-sept 2022 |
| 7/22/2022 | juli-sept 2022 |
| 7/23/2022 | juli-sept 2022 |
| 7/27/2022 | juli-sept 2022 |
| 8/2/2022 | juli-sept 2022 |
| 8/8/2022 | juli-sept 2022 |
| 8/19/2022 | juli-sept 2022 |
| 9/26/2022 | juli-sept 2022 |
| 10/14/2022 | okt-dec 2022 |
| 10/29/2022 | okt-dec 2022 |

De meting `transaction_quarter` wordt gemaakt in het diagramobject met behulp van de functie `quartername()` en geeft het veld `date` door als het argument van de functie.

De `quartername()`-functie identificeert eerst het kwartaal waarin de datumwaarde valt. Hij retourneert vervolgens een waarde die de begin- en eindmaanden van het kwartaal en van het jaar laten zien.

*Diagram van `quartername()`-functie, diagramobjectvoorbeeld*




Transactie 8203 vond plaats op 8 augustus 2022. De `quartername()`-functie identificeert dat de transactie plaatsvond in het derde kwartaal en retourneert daarom jul-sep 2022. De maanden worden weergegeven met dezelfde notatie als de `MonthNames`-systeemvariabele.

## Voorbeeld 5 – Scenario

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2022 die wordt geladen in een tabel met de naam `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.

De eindgebruiker wil graag een diagramobject dat de totale omzet per kwartaal voor de transacties weergeeft. Dit kan zelfs worden bereikt als deze dimensie niet beschikbaar is in het gegevensmodel, door de functie `quartername()` te gebruiken als een berekende dimensie in het diagram.

### Load-script

```
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
];
```

**Resultaten**

**Doe het volgende:**

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel.
2. Maak een berekende dimensie met behulp van de volgende uitdrukking:
   `=quartername(date)`
3. Bereken vervolgens de totale verkoop met de volgende aggregatiemeting:
   `=sum(amount)`
4. Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

| =quartername(date) | =sum(amount) |
|---|---|
| juli-sept 2022 | $446.31 |
| april-jun 2022 | $351.48 |
| jan-mrt 2022 | $253.89 |
| okt-dec 2022 | $163.91 |

## quarterstart

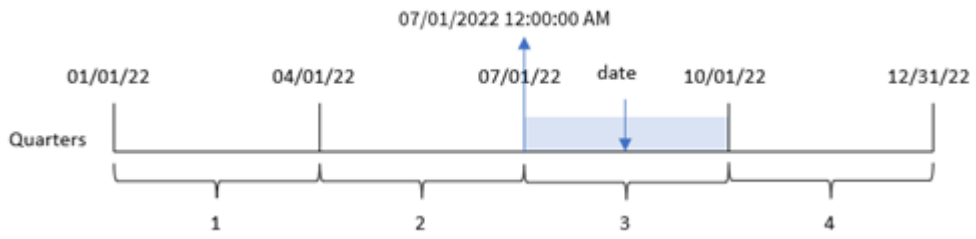
Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de eerste milliseconde van het kwartaal dat **date** bevat. De standaarduitvoernotatie is de **DateFormat** die is ingesteld in het script.

**Syntaxis:**

```
QuarterStart(date[, period_no[, first_month_of_year]])
```

**Retourgegevenstypen:** dubbele waarde

*Diagram van de `quarterstart()`-functie*




De `quarterstart()`-functie bepaalt in welk kwartaal de `date`valt. Hij retourneert vervolgens een tijdstempel, in datumnotatie, voor de laatste milliseconde van de laatste maand van dat kwartaal.

Argumenten

| Argument | Beschrijving |
|---|---|
| **date** | De te evalueren datum of tijdstempel. |
| **period_no** | **period_no** is een geheel getal, waarbij de waarde 0 het kwartaal aangeeft dat **date** bevat. Negatieve waarden in **period_no** geven voorafgaande kwartalen aan en positieve waarden geven volgende kwartalen aan. |
| **first_month_of_year** | Als u wilt werken met (boek)jaren die niet in januari beginnen, geeft u een waarde op tussen 2 en 12 in **first_month_of_year**. |

## Wanneer gebruiken

De `quarterstart()`-functie wordt doorgaans gebruikt als onderdeel van een uitdrukking als de gebruiker wil dat in de berekening het deel van het kwartaal wordt gebruikt dat al is geweest. Hij kan bijvoorbeeld worden gebruikt als een gebruiker de rente wil berekenen die in een kwartaal tot nu toe is opgebouwd.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `quarterstart('10/29/2005')` | Retourneert 10/01/2005. |
| `quarterstart('10/29/2005', -1 )` | Retourneert 07/01/2005. |
| `quarterstart('10/29/2005', 0, 3)` | Retourneert 09/01/2005. |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – geen aanvullende argumenten

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties voor 2022 die wordt geladen in de tabel `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.
- Het maken van een veld, `start_of_quarter`, dat een tijdstempel voor het begin van het kwartaal retourneert waarin de transactie plaatsvond.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

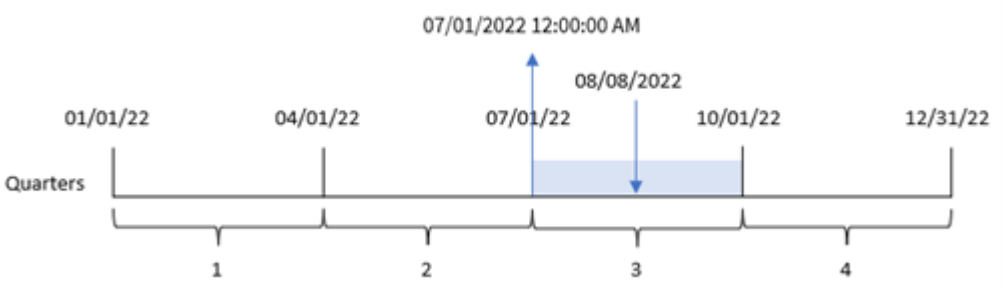
- `date`
- `start_of_quarter`
- `start_of_quarter_timestamp`

Resultatentabel

| date | start_of_quarter | start_of_quarter_timestamp |
|---|---|---|
| 1/7/2022 | 01/01/2022 | 1/1/2022 12:00:00 AM |
| 1/19/2022 | 01/01/2022 | 1/1/2022 12:00:00 AM |
| 2/5/2022 | 01/01/2022 | 1/1/2022 12:00:00 AM |
| 2/28/2022 | 01/01/2022 | 1/1/2022 12:00:00 AM |
| 3/16/2022 | 01/01/2022 | 1/1/2022 12:00:00 AM |
| 4/1/2022 | 04/01/2022 | 4/1/2021 12:00:00 AM |
| 5/7/2022 | 04/01/2022 | 4/1/2021 12:00:00 AM |
| 5/16/2022 | 04/01/2022 | 4/1/2021 12:00:00 AM |
| 6/15/2022 | 04/01/2022 | 4/1/2021 12:00:00 AM |
| 6/26/2022 | 04/01/2022 | 4/1/2021 12:00:00 AM |
| 7/9/2022 | 07/01/2022 | 7/1/2021 12:00:00 AM |
| 7/22/2022 | 07/01/2022 | 7/1/2021 12:00:00 AM |
| 7/23/2022 | 07/01/2022 | 7/1/2021 12:00:00 AM |
| 7/27/2022 | 07/01/2022 | 7/1/2021 12:00:00 AM |
| 8/2/2022 | 07/01/2022 | 7/1/2021 12:00:00 AM |
| 8/8/2022 | 07/01/2022 | 7/1/2021 12:00:00 AM |
| 8/19/2022 | 07/01/2022 | 7/1/2021 12:00:00 AM |
| 9/26/2022 | 07/01/2022 | 7/1/2021 12:00:00 AM |
| 10/14/2022 | 10/01/2022 | 10/1/2022 12:00:00 AM |
| 10/29/2022 | 10/01/2022 | 10/1/2022 12:00:00 AM |

Het veld '`start_of_quarter`' wordt gemaakt in de voorafgaande load-instructie met behulp van de functie `quarterstart()` en geeft het datumveld door als het argument van de functie. De `quarterstart()`-functie identificeert eerst het kwartaal waarin de datumwaarde valt. Hij retourneert vervolgens een tijdstempel voor de eerste milliseconde van dat kwartaal.

*Diagram van `quarterstart()`-functie, voorbeeld zonder aanvullende argumenten*




Transactie 8203 vond plaats op 8 augustus. De `quarterstart()`-functie identificeert dat de transactie in het derde kwartaal plaatsvond en retourneert de eerste milliseconde van dat kwartaal. Dat is 1 juli om 12:00:00 AM.

## Voorbeeld 2 – period_no

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het maken van een veld, `previous_quarter_start`, dat de tijdstempel voor het begin van het kwartaal retourneert voordat de transactie plaatsvond.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `previous_quarter_start`
- `previous_quarter_start_timestamp`

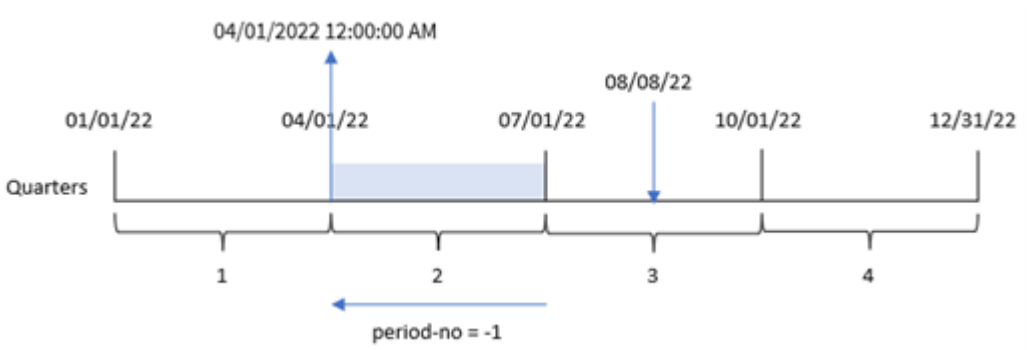
Resultatentabel

| date | previous_quarter_start | previous_quarter_start_timestamp |
|---|---|---|
| 1/7/2022 | 10/01/2021 | 10/1/2021 12:00:00 AM |
| 1/19/2022 | 10/01/2021 | 10/1/2021 12:00:00 AM |
| 2/5/2022 | 10/01/2021 | 10/1/2021 12:00:00 AM |
| 2/28/2022 | 10/01/2021 | 10/1/2021 12:00:00 AM |
| 3/16/2022 | 10/01/2021 | 10/1/2021 12:00:00 AM |
| 4/1/2022 | 01/01/2022 | 1/1/2022 12:00:00 AM |
| 5/7/2022 | 01/01/2022 | 1/1/2022 12:00:00 AM |
| 5/16/2022 | 01/01/2022 | 1/1/2022 12:00:00 AM |
| 6/15/2022 | 01/01/2022 | 1/1/2022 12:00:00 AM |
| 6/26/2022 | 01/01/2022 | 1/1/2022 12:00:00 AM |
| 7/9/2022 | 04/01/2022 | 4/1/2021 12:00:00 AM |
| 7/22/2022 | 04/01/2022 | 4/1/2021 12:00:00 AM |
| 7/23/2022 | 04/01/2022 | 4/1/2021 12:00:00 AM |
| 7/27/2022 | 04/01/2022 | 4/1/2021 12:00:00 AM |
| 8/2/2022 | 04/01/2022 | 4/1/2021 12:00:00 AM |

| date | previous_quarter_start | previous_quarter_start_timestamp |
|---|---|---|
| 8/8/2022 | 04/01/2022 | 4/1/2021 12:00:00 AM |
| 8/19/2022 | 04/01/2022 | 4/1/2021 12:00:00 AM |
| 9/26/2022 | 04/01/2022 | 4/1/2021 12:00:00 AM |
| 10/14/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |
| 10/29/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |

In dit geval, omdat een `period_no` van -1 werd gebruikt als het offset-argument in de `quarterstart()`-functie, identificeert de functie eerst het kwartaal waarin de transacties plaatsvinden. Het verschuift vervolgens naar een kwartaal eerder en identificeert de eerste milliseconde van dat kwartaal.

*Diagram van `quarterstart()`-functie, voorbeeld period_no*




Transactie 8203 vond plaats op 8 augustus. De `quarterstart()`-functie identificeert dat het kwartaal vóór het kwartaal waarin de transactie plaatsvond tussen 1 april en 30 juni viel. Vervolgens wordt de eerste milliseconde van dat kwartaal geretourneerd: 1 april om 12:00:00 AM.

## Voorbeeld 3 – first_month_of_year

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld. In dit voorbeeld moeten we echter 1 maart instellen als het begin van het boekjaar.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `start_of_quarter`
- `start_of_quarter_timestamp`

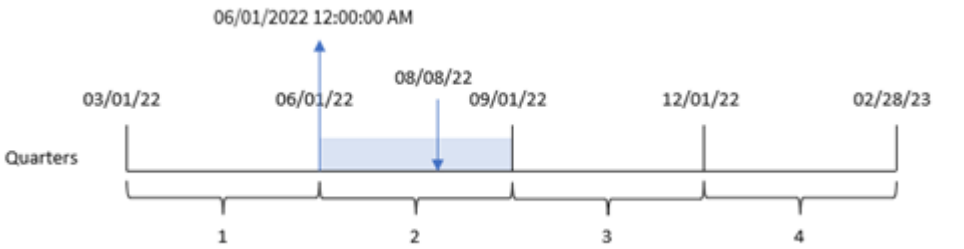
Resultatentabel

| date | start_of_quarter | start_of_quarter_timestamp |
|---|---|---|
| 1/7/2022 | 12/01/2021 | 12/1/2021 12:00:00 AM |
| 1/19/2022 | 12/01/2021 | 12/1/2021 12:00:00 AM |
| 2/5/2022 | 12/01/2021 | 12/1/2021 12:00:00 AM |
| 2/28/2022 | 12/01/2021 | 12/1/2021 12:00:00 AM |
| 3/16/2022 | 03/01/2022 | 3/1/2022 12:00:00 AM |
| 4/1/2022 | 03/01/2022 | 3/1/2022 12:00:00 AM |
| 5/7/2022 | 03/01/2022 | 3/1/2022 12:00:00 AM |

| date | start_of_quarter | start_of_quarter_timestamp |
|---|---|---|
| 5/16/2022 | 03/01/2022 | 3/1/2022 12:00:00 AM |
| 6/15/2022 | 06/01/2022 | 6/1/2022 12:00:00 AM |
| 6/26/2022 | 06/01/2022 | 6/1/2022 12:00:00 AM |
| 7/9/2022 | 06/01/2022 | 6/1/2022 12:00:00 AM |
| 7/22/2022 | 06/01/2022 | 6/1/2022 12:00:00 AM |
| 7/23/2022 | 06/01/2022 | 6/1/2022 12:00:00 AM |
| 7/27/2022 | 06/01/2022 | 6/1/2022 12:00:00 AM |
| 8/2/2022 | 06/01/2022 | 6/1/2022 12:00:00 AM |
| 8/8/2022 | 06/01/2022 | 6/1/2022 12:00:00 AM |
| 8/19/2022 | 06/01/2022 | 6/1/2022 12:00:00 AM |
| 9/26/2022 | 09/01/2022 | 9/1/2022 12:00:00 AM |
| 10/14/2022 | 09/01/2022 | 9/1/2022 12:00:00 AM |
| 10/29/2022 | 09/01/2022 | 9/1/2022 12:00:00 AM |

Omdat in dit geval het `first_month_of_year`-argument van 3 is gebruikt in de `quarterstart()`-functie, wordt het begin van het jaar verplaatst van 1 januari naar 1 maart.

*Diagram van de `quarterstart()`-functie, voorbeeld first_month_of_year*




Transactie 8203 vond plaats op 8 augustus. Omdat het begin van het jaar 1 maart is, vallen de kwartalen van het jaar tussen maart-mei, juni-augustus, september-november en december-februari. De `quarterstart()`-functie identificeert dat de transactie in het kwartaal tussen het begin van juni en augustus plaatsvond en retourneert de eerste milliseconde van dat kwartaal. Dat is 1 juni om 12:00:00 AM.

## Voorbeeld 4 – diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die een tijdstempel retourneert voor het eind van het kwartaal waarin de transacties plaatsvonden is gemaakt als meting in een diagramobject van de toepassing.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `date`.

Voeg de volgende metingen toe:

- `=quarterstart(date)`
- `=timestamp(quarterstart(date))`

Resultatentabel

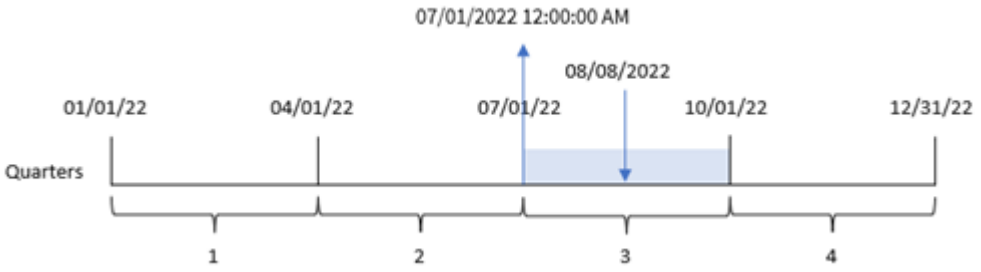
| date | =quarterstart(date) | =timestamp(quarterstart(date)) |
|---|---|---|
| 10/14/2022 | 10/01/2022 | 10/1/2022 12:00:00 AM |
| 10/29/2022 | 10/01/2022 | 10/1/2022 12:00:00 AM |
| 7/9/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |

| date | =quarterstart(date) | =timestamp(quarterstart(date)) |
|---|---|---|
| 7/22/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |
| 7/23/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |
| 7/27/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |
| 8/2/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |
| 8/8/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |
| 8/19/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |
| 9/26/2022 | 07/01/2022 | 7/1/2022 12:00:00 AM |
| 4/1/2022 | 04/01/2022 | 4/1/2022 12:00:00 AM |
| 5/7/2022 | 04/01/2022 | 4/1/2022 12:00:00 AM |
| 5/16/2022 | 04/01/2022 | 4/1/2022 12:00:00 AM |
| 6/15/2022 | 04/01/2022 | 4/1/2022 12:00:00 AM |
| 6/26/2022 | 04/01/2022 | 4/1/2022 12:00:00 AM |
| 1/7/2022 | 01/01/2022 | 1/1/2022 12:00:00 AM |
| 1/19/2022 | 01/01/2022 | 1/1/2022 12:00:00 AM |
| 2/5/2022 | 01/01/2022 | 1/1/2022 12:00:00 AM |
| 2/28/2022 | 01/01/2022 | 1/1/2022 12:00:00 AM |
| 3/16/2022 | 01/01/2022 | 1/1/2022 12:00:00 AM |

De meting `start_of_quarter` wordt gemaakt in het diagramobject met behulp van de functie `quarterstart()` en geeft het veld `date` door als het argument van de functie.

De `quarterstart()`-functie identificeert in welke kwartaal de datumwaarde valt en retourneert een tijdstempel voor de eerste milliseconde van dat kwartaal.

*Diagram van `quarterstart()`-functie, diagramobjectvoorbeeld*




Transactie 8203 vond plaats op 8 augustus. De `quarterstart()`-functie identificeert dat de transactie in het derde kwartaal plaatsvond en retourneert de eerste milliseconde van dat kwartaal. De geretourneerde waarde is 1 juli om 12:00:00 AM.

## Voorbeeld 5 – Scenario

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met een set leningsaldo's die wordt geladen in de tabel `Loans`.
- Gegevens bestaan uit lening-id's, het saldo aan het begin van het kwartaal en de enkelvoudige rente die op iedere lening per jaar wordt berekend.

De eindgebruiker wil een diagramobject dat per lening-id de huidige rente weergeeft die is opgebouwd op elke lening in het kwartaal tot heden.

### Load-script

```
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
];
```

### Resultaten

### Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:
   - `loan_id`
   - `start_balance`
2. Maak nu de volgende meting om de opgebouwde rente te berekenen:
   `=start_balance*(rate*(today(1)-quarterstart(today(1)))/365)`
3. Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

| loan_id | start_balance | =start_balance*(rate*(today(1)-quarterstart(today(1)))/365) |
|---|---|---|
| 8188 | $10000.00 | $15.07 |

| loan_id | start_balance | =start_balance*(rate*(today(1)-quarterstart(today(1)))/365) |
|---|---|---|
| 8189 | $15000.00 | $128.84 |
| 8190 | $17500.00 | $63.29 |
| 8191 | $21000.00 | $107.59 |
| 8192 | $90000.00 | $1139.18 |

De functie `quarterstart()` gebruikt de datum van vandaag als enige argument en retourneert de begindatum van het huidige jaar. Door dat resultaat af te trekken van de huidige datum, retourneert de uitdrukking het aantal dagen dat in dit kwartaal tot nu toe is verstreken.

Deze waarde wordt vervolgens vermenigvuldigd met het rentepercentage en gedeeld door 365 om het effectieve rentepercentage voor deze periode te retourneren. Het resultaat wordt vervolgens vermenigvuldigd met het beginsaldo van de lening om de rente terug te betalen die dit kwartaal tot nu toe is opgebouwd.

## second

Deze functie retourneert een geheel getal dat de seconde representeert als de breuk van de **expression** wordt geïnterpreteerd als een datum volgens de standaardinterpretatie van getallen.

**Syntaxis:**

```
second (expression)
```

**Retourgegevenstypen:** geheel getal

### Wanneer gebruiken

De `second()`-functie is handig als u aggregaties per seconde wilt vergelijken. U kunt de functie bijvoorbeeld gebruiken als u de verdeling van het aantal activiteiten per seconde wilt bekijken.

Deze dimensies kunnen worden gemaakt in het load-script door de functie te gebruiken om een veld te maken in een hoofdagendatabel, of door de dimensie rechtstreeks in een diagram te gebruiken als een berekende dimensie.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `second( '09:14:36' )` | retourneert 36 |
| `second( '0.5555' )` | retourneert 55 ( omdat 0,5555 = 13:19:55 ) |

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege

landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – Variabele

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met transacties per tijdstempel die wordt geladen in de tabel `Transactions`.
- De standaard `TimeStamp`-systeemvariabele (`M/D/YYYY h:mm:ss[.fff] TT`) wordt gebruikt.
- Maak een veld, `second` dat berekent wanneer de aankopen hebben plaatsgevonden.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `second`

Resultatentabel

| date | second |
|---|---|
| 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 |

De waarden in het `second`-veld zijn gemaakt met de `second()`-functie en geven de datum door als de uitdrukking in het voorafgaande load-instructie.

## Voorbeeld 2 – Diagramobject

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld. In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De `second`-waarden worden berekend via een meting in een diagramobject.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:`date`.

Maak de volgende meting:

```
=second(date)
```

Resultatentabel

| date | =second(date) |
|---|---|
| 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 |

De waarden voor `second` worden gemaakt met behulp van de `second()`-functie en geven de datum door als de uitdrukking in een meting voor het diagramobject.

## Voorbeeld 3 – Scenario

Load-script en diagramuitdrukkingen

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset van tijdstempels, die wordt gegenereerd om het verkeer naar de ticketverkoopsite van een specifiek festival te vertegenwoordigen. Deze tijdstempels en een bijbehorende `id` worden in de tabel `Web_Traffic` geladen.
- De `TimeStamp`-systeemvariabele `M/D/YYYY h:mm:ss[.fff] TT` wordt gebruikt.

In dit scenario zijn er 10.000 tickets verkocht die om 9.00 uur op 20 mei 2021 in de verkoop zijn gegaan. Eén minuut later waren de tickets uitverkocht.

De gebruiker wil een diagramobject dat per seconde het aantal bezoeken aan de website laat zien.

### Load-script

```
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;
```

### Resultaten

### Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel.
2. Maak nu een berekende dimensie met behulp van de volgende uitdrukking:
   `=second(timestamp)`
3. Maak een aggregatiemeting om het totaalaantal invoeren te berekenen:
   `=count(id)`

De resultatentabel ziet er vergelijkbaar uit met de onderstaande tabel, maar dan met andere waarden voor de aggregatiemeting:

Resultatentabel

| 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 |
| 16 | 191 |
| 17 | 203 |
| 18 | 158 |
| 19 | 159 |
| 20 | 163 |
| Nog 39 rijen | |

## setdateyear

Deze functie neemt een **timestamp** en een **year** als invoer en werkt de **timestamp** bij met het **year** dat is opgegeven in de invoer.

**Syntaxis:**

```
setdateyear (timestamp, year)
```

**Retourgegevenstypen:** dual

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| **timestamp** | Een standaardtijdstempel in Qlik Sense (vaak gewoon een datum). |
| **year** | Een jaartal in vier cijfers. |

Voorbeelden en resultaten:
Deze voorbeelden gebruiken de datumopmaak **DD/MM/YYYY**. De datumopmaak wordt opgegeven in de opdracht **SET DateFormat** boven aan uw script voor het laden van gegevens. Wijzig de opmaak in de voorbeelden zodat deze in uw behoeften voorziet.

Scriptvoorbeelden

| Voorbeeld | Resultaat |
|---|---|
| `setdateyear ('29/10/2005', 2013)` | Retourneert '29/10/2013 |
| `setdateyear ('29/10/2005 04:26:14', 2013)` | Retourneert '29/10/2013 04:26:14'<br>Als u het tijdsgedeelte van de tijdstempel wilt bekijken in een visualisatie, moet u de getalopmaak instellen op Date en een waarde kiezen voor Formatting die tijdwaarden weergeeft. |

**Voorbeeld:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
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

];
```

De resulterende tabel bevat de oorspronkelijke datums en een kolom waarin het jaar is ingesteld op 2013.

Resultatentabel

| testdates | NewYear |
|---|---|
| 1/11/2012 | 1/11/2013 |
| 10/12/2012 | 10/12/2013 |
| 2/1/2012 | 2/1/2013 |
| 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

Deze functie neemt een **timestamp**, een **month** en een **year** als invoer en werkt de **timestamp** bij met het **year** en de **month** die zijn opgegeven in de invoer. .

**Syntaxis:**

```
SetDateYearMonth (timestamp, year, month)
```

**Retourgegevenstypen:** dual

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| **timestamp** | Een standaardtijdstempel in Qlik Sense (vaak gewoon een datum). |
| **year** | Een jaartal in vier cijfers. |
| **month** | Een maand in één of twee cijfers. |

Voorbeelden en resultaten:
Deze voorbeelden gebruiken de datumopmaak **DD/MM/YYYY**. De datumopmaak wordt opgegeven in de opdracht **SET DateFormat** boven aan uw script voor het laden van gegevens. Wijzig de opmaak in de voorbeelden zodat deze in uw behoeften voorziet.

Scriptvoorbeelden

| Voorbeeld | Resultaat |
|---|---|
| `setdateyearmonth ('29/10/2005', 2013, 3)` | Retourneert '29/03/2013 |
| `setdateyearmonth ('29/10/2005 04:26:14', 2013, 3)` | Retourneert '29/03/2013 04:26:14'<br>Als u het tijdsgedeelte van de tijdstempel wilt bekijken in een visualisatie, moet u de getalopmaak instellen op Date en een waarde kiezen voor Formatting die tijdwaarden weergeeft. |

**Voorbeeld:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
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

];
```

De resulterende tabel bevat de oorspronkelijke datums en een kolom waarin het jaar is ingesteld op 2013.

Resultatentabel

| 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

De functie retourneert de tijdzone zoals gedefinieerd op de computer waarop de Qlik-engine wordt uitgevoerd.

**Syntaxis:**

```
TimeZone( )
```

**Retourgegevenstypen:** dual

**Voorbeeld:**

```
timezone( )
```

Als u verschillende tijdzones in een meting in uw app wilt zien, moet u de functie `localtime()` in een meting gebruiken.

## today

Deze functie retourneert de huidige datum. Deze functie retourneert waarden in de indeling van de systeemvariabele `DateFormat`.

**Syntaxis:**

```
today([ timer_mode])
```

**Retourgegevenstypen:** dubbele waarde

De `today()`-functie kan ofwel in het load-script ofwel in diagramobjecten worden gebruikt.

De standaardwaarde voor `timer_mode` is 1.

Argumenten

| Argument | Beschrijving |
|---|---|
| timer_mode | Kan de volgende waarden hebben:<br><br>0 (dag van de als laatste voltooide laadbewerking voor gegevens)<br>1 (dag van de functieaanroep)<br>2 (dag wanneer de app werd geopend)<br><br>*Als u de functie in een load-script gebruikt, resulteert **timer_mode=0** in de dag van de als laatste voltooide laadbewerking voor gegevens, terwijl **timer_mode=1** de dag van de functieaanroep in de huidige laadbewerking voor gegevens oplevert.* |

Voorbeelden van functies

| timer_mode-waarde | Resultaat bij gebruik in load-script | Resultaat bij gebruik in diagramobject |
|---|---|---|
| 0 | Retourneert een datum met de notatie van de systeemvariabele `DateFormat`, van de laatste keer dat de gegevens opnieuw zijn geladen voorafgaand aan de laatste keer dat de gegevens opnieuw zijn geladen. | Retourneert een datum met de notatie van de systeemvariabele `DateFormat` voor de laatste keer dat de gegevens opnieuw zijn geladen. |
| 1 | Retourneert een datum met de notatie van de systeemvariabele `DateFormat` voor de laatste keer dat de gegevens opnieuw zijn geladen. | Retourneert een datum met de notatie van de systeemvariabele `DateFormat` van de functieaanroep. |

| timer_mode-waarde | Resultaat bij gebruik in load-script | Resultaat bij gebruik in diagramobject |
|---|---|---|
| 2 | Retourneert een datum met de notatie van de systeemvariabele `DateFormat` voor het begin van de sessie van de gebruiker in de applicatie. Dit wordt niet bijgewerkt tenzij de gebruiker het script opnieuw laadt. | Retourneert de datum met de notatie van de systeemvariabele `DateFormat` voor het begin van de sessie van de gebruiker in de applicatie. Dit wordt vernieuwd zodra een nieuwe sessie begint of de gegevens weer in de applicatie zijn geladen. |

## Wanneer gebruiken

De `today()`-functie wordt meestal gebruikt als een component in een uitdrukking. Hij kan bijvoorbeeld worden gebruikt om de rente te berekenen die in een maand tot nu toe is opgebouwd.

De volgende tabel geeft een uitleg van het resultaat dat door de `today()`-functie is geretourneerd en geeft verschillende waarden voor het `timer_mode`-argument:

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – Genereren van objecten met behulp van het load-script

### Load-script en resultaten

#### Overzicht

Het volgende voorbeeld creëert die variabelen met behulp van de `today()`-functie. Elke variabele gebruikt een van de `timer_mode`-opties om het effect van de variabele te demonstreren.

Laad het script opnieuw en herlaad het na 24 nog een keer zodat de variabelen hun doel kunnen laten zien. De variabelen `today(0)` en `today(1)` zullen nu verschillende waarden laten zien en hun doel demonstreren.

### Load-script

```
LET vPreviousDataLoad = today(0);
LET vCurrentDataLoad = today(1);
LET vApplicationOpened = today(2);
```

### Resultaten

Nadat de gegevens voor de tweede keer zijn geladen, maakt u drie tekstvakken door de onderstaande instructies te volgen.

Maak eerst een tekstvak voor de gegevens die eerder zijn geladen.

**Doe het volgende:**

1. Gebruik het diagramobject **Tekst en afbeelding** om een tekstvak te maken.
2. Voeg de volgende meting toe aan het object:
   `=vPreviousDataLoad`
3. Onder **Uiterlijk** selecteert u **Show titles** en voegt u de titel 'Previous Reload Time' toe aan het object.

Maak vervolgens een tekstvak voor de gegevens die momenteel worden geladen.

**Doe het volgende:**

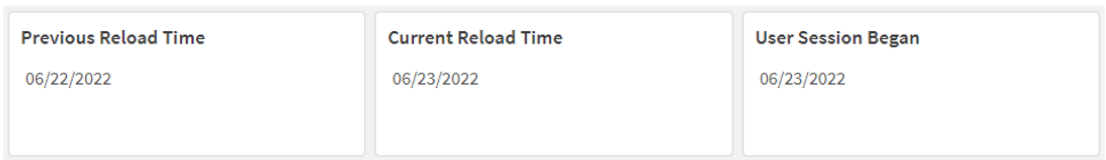
1. Gebruik het diagramobject **Tekst en afbeelding** om een tekstvak te maken.
2. Voeg de volgende meting toe aan het object:
   `=vCurrentDataLoad`
3. Onder **Uiterlijk** selecteert u **Show titles** en voegt u de titel 'Current Reload Time' toe aan het object.

Maak een definitief tekstvak dat wordt getoond als de sessie van de gebruiker in de toepassing wordt gestart.

**Doe het volgende:**

1. Gebruik het diagramobject **Tekst en afbeelding** om een tekstvak te maken.
2. Voeg de volgende meting toe aan het object:
   `=vApplicationOpened`
3. Onder **Uiterlijk** selecteert u **Show titles** en voegt u de titel 'User Session Started' toe aan het object.

*Diagram van variabelen die zijn gemaakt met de `today()`-functie in het load-script*



De bovenstaande afbeelding geeft voorbeeldwaarden weer voor elk van de gemaakte variabelen. De waarden zouden bijvoorbeeld als volgt kunnen zijn:

- Vorige herlaadtijd: 06/22/2022
- Huidige herlaadtijd: 06/23/2022
- Start gebruikerssessie: 06/23/2022

## Voorbeeld 2 – Genereren van objecten zonder het load-script

Load-script en diagramuitdrukking

### Overzicht

Het volgende voorbeeld creëert die variabelen met behulp van de `today()`-functie. Elk diagramobject gebruikt een van de `timer_mode`-opties om het effect te demonstreren.

Er is geen load-script voor dit voorbeeld.

### Resultaten

Maak drie tekstvakken zodra de gegevens voor een tweede keer zijn geladen.

Maak eerst een tekstvak voor de laatste keer dat de gegevens opnieuw zijn geladen.

**Doe het volgende:**

1. Gebruik het diagramobject **Tekst en afbeelding** om een tekstvak te maken.
2. Voeg de volgende meting toe:
   `=today(0)`
3. Onder **Uiterlijk** selecteert u **Titels tonen**en voegt u de titel Laatste herlaadtijd gegevens toe aan het object.

Maak nu een tekstvak om de actuele tijd te tonen.

**Doe het volgende:**

1. Gebruik het diagramobject **Tekst en afbeelding** om een tekstvak te maken.
2. Voeg de volgende meting toe:
   `=today(1)`
3. Onder **Uiterlijk** selecteert u **Titels tonen** en voegt u de titel Actuele tijd toe aan het object.

Maak een definitief tekstvak dat wordt getoond als de sessie van de gebruiker in de toepassing wordt gestart.

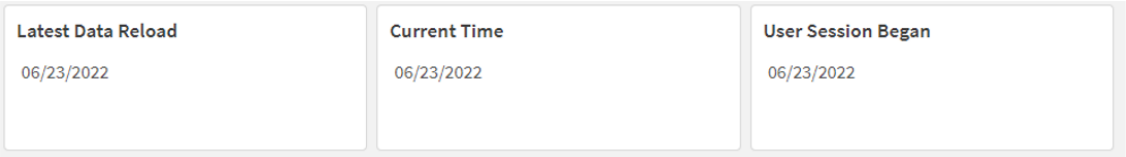
**Doe het volgende:**

1. Gebruik het diagramobject **Tekst en afbeelding** om een tekstvak te maken.
2. Voeg de volgende meting toe:

```
=today(2)
```

3. Onder **Uiterlijk** selecteert u **Titels tonen** en voegt u de titel Start gebruikerssessie toe aan het object.

*Diagram van objecten die zijn gemaakt met de `today()`-functie zonder het load-script*




De bovenstaande afbeelding geeft voorbeeldwaarden weer voor elk van de gemaakte objecten. De waarden zouden bijvoorbeeld als volgt kunnen zijn:

- Laatste herlaadtijd gegevens: 06/23/2022
- Actuele tijd: 06/23/2022
- Start gebruikerssessie: 06/23/2022

Het diagramobject Laatste herlaadtijd gegevens maakt gebruik van een `timer_mode`-waarde van 0. Dit retourneert de tijdstempel voor de laatste keer dat de gegevens opnieuw zijn geladen.

Het diagramobject Actuele tijd gebruikt een `timer_mode`-waarde van 1. Dit retourneert de actuele tijd volgens de systeemklok. Als het werkblad of object wordt vernieuwd, wordt deze waarde bijgewerkt.

Het diagramobject Start gebruikerssessie gebruikt een `timer_mode`-waarde van 2. Dit retourneert de tijdstempel voor het moment waarop de applicatie werd geopend en waarop de gebruikerssessie is begonnen.

## Voorbeeld 3 – Scenario

Load-script en diagramuitdrukking

**Overzicht**

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set leningsaldo's die wordt geladen in de tabel `Loans`.
- Tabelgegevens met velden voor lening-id, saldo aan het begin van de maand en de enkelvoudige rente die op elke lening per jaar wordt berekend.

De eindgebruiker wil een diagramobject dat per lening-id de huidige rente weergeeft die is opgebouwd op elke lening in de maand tot heden. De applicatie wordt slechts eens per week opnieuw geladen, maar de gebruiker wil graag dat de resultaten worden vernieuwd zodra het object of de applicatie wordt vernieuwd.

### Load-script

```
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
];
```

### Resultaten

### Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel.
2. Voeg de volgende velden als dimensies toe.
    - `loan_id`
    - `start_balance`
3. Maak nu een meting om de opgebouwde rente te berekenen:
   `=start_balance*(rate*(today(1)-monthstart(today(1)))/365)`
4. Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

| loan_id | start_balance | =start_balance*(rate*(today(1)-monthstart(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 |

De functie `monthstart()` gebruikt de functie `today()` om de datum van vandaag als enige argument te retourneren en retourneert de begindatum van de huidige maand. Door dat resultaat af te trekken van de huidige datum en de `today()`-functie opnieuw te gebruiken, retourneert de uitdrukking het aantal dagen dat in deze maand tot nu toe is verstreken.

Deze waarde wordt vervolgens vermenigvuldigd met het rentepercentage en gedeeld door 365 om het effectieve rentepercentage voor deze periode te retourneren. Het resultaat wordt vervolgens vermenigvuldigd met het beginsaldo van de lening om de rente terug te betalen die deze maand tot nu toe is opgebouwd.

Omdat de waarde 1 is gebruikt als het `timer_mode`-argument in de `today()`-functies in de uitdrukking, wordt elke keer dat het diagramobject wordt vernieuwd (door de applicatie te openen, de pagina te vernieuwen, tussen werkbladen te wisselen, enzovoort) de huidige datum geretourneerd en worden de resultaten overeenkomstig vernieuwd.

## UTC

Retourneert de huidige Coordinated Universal Time.

**Syntaxis:**

```
UTC( )
```

**Retourgegevenstypen:** dubbele waarde

**Voorbeeld:**

```
utc( )
```

## week

Deze functie retourneert een geheel getal dat het weeknummer representeert die overeenkomt met de opgegeven datum.

**Syntaxis:**

```
week(timestamp [, first_week_day [, broken_weeks [, reference_day]]])
```

**Retourgegevenstypen:** geheel getal

Argumenten

| Argument | Beschrijving |
|---|---|
| **timestamp** | De te evalueren datum of tijdstempel. |
| **first_week_day** | Geeft de dag op waarop de week begint. Bij weglating wordt de waarde van de variabele **FirstWeekDay** gebruikt.<br><br>De mogelijke **first_week_day**-waarden zijn 0 voor maandag, 1 voor dinsdag, 2 voor woensdag, 3 voor donderdag, 4 voor vrijdag, 5 voor zaterdag en 6 voor zondag.<br><br>Zie *FirstWeekDay (page 244)* voor meer informatie over de systeemvariabele. |
| **broken_weeks** | Als u **broken_weeks** (gebroken weken) niet specificeert, dan wordt de waarde van variabele **BrokenWeeks** gebruikt om te definiëren of weken al dan niet gebroken zijn. |
| **reference_day** | Als u **reference_day** (referentiedag) niet specificeert, dan wordt de waarde van variabele **ReferenceDay** gebruikt om te definiëren welke dag in januari wordt ingesteld als referentiedag om week 1 te definiëren. Standaard wordt in Qlik Sense-functies 4 gebruikt als de referentiedag. Dit betekent dat week 1 4 januari moet bevatten of, anders gezegd, dat week 1 altijd ten minste 4 dagen in januari moet hebben. |

De functie `week()` bepaalt in welke week de datum valt en retourneert het weeknummer.

In Qlik Sense worden de landinstellingen opgehaald wanneer de app wordt gemaakt en de bijbehorende instellingen worden als omgevingsvariabelen opgeslagen in het script. Deze worden gebruikt om het weeknummer te bepalen.

Dit betekent dat de meeste Europese app-ontwikkelaars de volgende omgevingsvariabelen krijgen, in overeenstemming met de definitie van 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
```

Een Noord-Amerikaanse app-ontwikkelaar krijgt meestal de volgende omgevingsvariabelen:

```
Set FirstWeekDay =6;    // Sunday as first week day
Set BrokenWeeks  =1;    // Use broken weeks
Set ReferenceDay =1;    // Jan 1st is always in week 1
```

De eerste dag van de week wordt bepaald door de systeemvariabele `FirstWeekDay`. U kunt ook de eerste dag van de week wijzigen door het argument `first_week_day` in de functie `week()` te gebruiken.

Als uw applicatie gebroken weken gebruikt, dan begint het weeknummer op 1 januari en eindigt op de dag voorafgaand aan de systeemvariabele `FirstWeekDay`, ongeacht hoeveel dagen er zijn verstreken.

Als uw toepassing ononderbroken weken gebruikt, kan week 1 in het voorgaande jaar of in de eerste paar dagen van januari beginnen. Dit hangt af van hoe u de omgevingsvariabelen `FirstWeekDay` en `ReferenceDay` gebruikt.

## Wanneer gebruiken

De `The week()`-functie is handig als u aggregaties per week wilt vergelijken. Hij kan bijvoorbeeld worden gebruikt om de totale verkoop van producten per week te bekijken. De `week()`-functie wordt verkozen boven `weekname()` wanneer de gebruiker wil dat de berekening niet per se de systeemvariabelen `BrokenWeeks`, `FirstWeekDay` of `ReferenceDay` gebruikt.

Bijvoorbeeld als u de totale verkoop van producten per week wilt zien.

Als de toepassing ononderbroken weken gebruikt, kan week 1 datums van december van het voorgaande jaar bevatten of datums in januari van het lopende jaar uitsluiten. Als de toepassing gebruikmaakt van gebroken weken, kan week 1 minder dan zeven dagen bevatten.

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze

landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

De onderstaande voorbeelden gaan uit van het volgende:

```
Set DateFormat= 'MM/DD/YYYY';
Set FirstWeekDay=0;
Set BrokenWeeks=0;
Set ReferenceDay=4;
```

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `week('12/28/2021')` | Retourneert 52. |
| `week(44614)` | Retourneert 8, aangezien dit het serienummer is voor 02/22/2022. |
| `week('01/03/2021')` | Retourneert 53. |
| `week('01/03/2021',6)` | Retourneert 1. |

## Voorbeeld 1 – Standaardsysteemvariabelen

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties voor de laatste week van 2021 en de eerste twee weken van 2022 wordt geladen in de tabel `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.
- Het veld `week_number` wordt gemaakt en retourneert het jaar- en weeknummer waarin de transacties plaatsvonden.
- Het veld `week_day` wordt gemaakt om de weekdagwaarde van iedere transactiedatum weer te geven.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `date`
- `week_day`
- `week_number`

Resultatentabel

| id | date | week_day | week_nummer |
|---|---|---|---|
| 8183 | 12/27/2021 | Ma | 53 |
| 8184 | 12/28/2021 | Di | 53 |
| 8185 | 12/29/2021 | Wo | 53 |
| 8186 | 12/30/2021 | Do | 53 |
| 8187 | 12/31/2021 | Vr | 53 |
| 8188 | 01/01/2022 | Za | 1 |
| 8189 | 01/02/2022 | Zo | 2 |

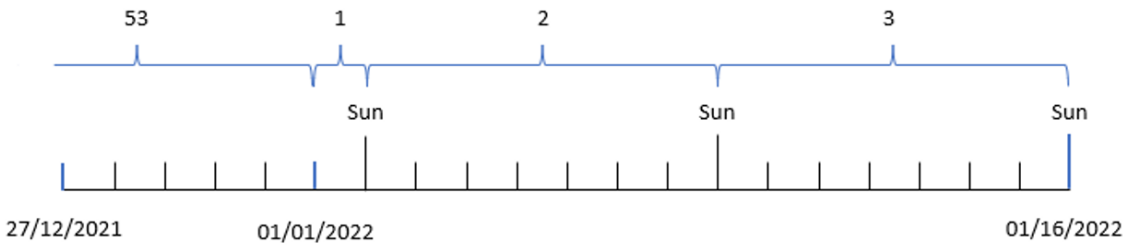
| id | date | week_day | week_nummer |
|---|---|---|---|
| 8190 | 01/03/2022 | Ma | 2 |
| 8191 | 01/04/2022 | Di | 2 |
| 8192 | 01/05/2022 | Wo | 2 |
| 8193 | 01/06/2022 | Do | 2 |
| 8194 | 01/07/2022 | Vr | 2 |
| 8195 | 01/08/2022 | Za | 2 |
| 8196 | 01/09/2022 | Zo | 3 |
| 8197 | 01/10/2022 | Ma | 3 |
| 8198 | 01/11/2022 | Di | 3 |
| 8199 | 01/12/2022 | Wo | 3 |
| 8200 | 01/13/2022 | Do | 3 |
| 8201 | 01/14/2022 | Vr | 3 |

Het `week_number`-veld wordt gemaakt in de voorafgaande load-instructie met behulp van de `week()`-functie en geeft het `date`-veld door als het argument van de functie.

Er worden geen andere parameters doorgegeven in de functie en daarom zijn de volgende standaardvariabelen die invloed hebben op de `week()`-functie van kracht:

- `BrokenWeeks`: dDe weeknummering begint op 1 januari
- `FirstWeekDay`: de eerste dag van de week is zondag

*Diagram van de `week()`-functie met standaardsysteemvariabelen*




Omdat de applicatie de standaardsysteemvariabele `BrokenWeeks` gebruikt, begint week 1 op 1 januari, een zaterdag.

Vanwege de standaardsysteemvariabele `FirstWeekDay` begint de week op een zondag. De eerste zondag na 1 januari valt op 2 januari en daarom begint week 2 op 2 januari.

## Voorbeeld 2 – first_week_day

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Het veld `week_number` wordt gemaakt en retourneert het jaar- en weeknummer waarin de transacties plaatsvonden.
- Het veld `week_day` wordt gemaakt om de weekdagwaarde van iedere transactiedatum weer te geven.

In dit voorbeeld willen we het begin van de werkweek instellen op dinsdag.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

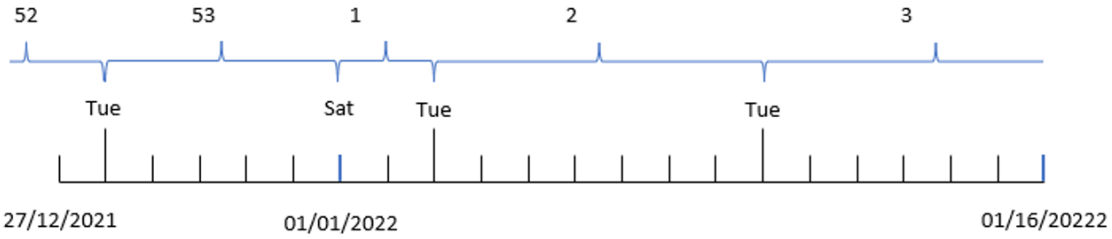
- `id`
- `date`
- `week_day`
- `week_number`

Resultatentabel

| id | date | week_day | week_nummer |
|---|---|---|---|
| 8183 | 12/27/2021 | Ma | 52 |
| 8184 | 12/28/2021 | Di | 53 |
| 8185 | 12/29/2021 | Wo | 53 |
| 8186 | 12/30/2021 | Do | 53 |
| 8187 | 12/31/2021 | Vr | 53 |
| 8188 | 01/01/2022 | Za | 1 |
| 8189 | 01/02/2022 | Zo | 1 |
| 8190 | 01/03/2022 | Ma | 1 |
| 8191 | 01/04/2022 | Di | 2 |
| 8192 | 01/05/2022 | Wo | 2 |
| 8193 | 01/06/2022 | Do | 2 |
| 8194 | 01/07/2022 | Vr | 2 |
| 8195 | 01/08/2022 | Za | 2 |
| 8196 | 01/09/2022 | Zo | 2 |
| 8197 | 01/10/2022 | Ma | 2 |
| 8198 | 01/11/2022 | Di | 3 |
| 8199 | 01/12/2022 | Wo | 3 |
| 8200 | 01/13/2022 | Do | 3 |
| 8201 | 01/14/2022 | Vr | 3 |

De applicatie past nog steeds opgesplitste weken toe. Maar het `first_week_day`-argument is in de `week()`-functie ingesteld op 1. Hierdoor wordt de eerste dag van de week ingesteld op dinsdag.

*Diagram van de week()-functie, voorbeeld met first_week_day*




De applicatie gebruikt de standaardsysteemvariabele `BrokenWeeks` en daarom begint week 1 op 1 januari, een zaterdag.

Het `first_week_day`-argument van de `week()`-functie stelt de eerste weekdag in op een dinsdag. Daarom begint week 53 op 28 december 2021.

Maar omdat de functie nog steeds opgesplitste weken toepast, duurt week 1 slechts twee dagen omdat de eerste dinsdag na 1 januari op 3 januari valt.

## Voorbeeld 3 – unbroken_weeks

### Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld.

In dit voorbeeld gebruiken we niet-opgesplitste weken.

#### Load-script

```
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
```

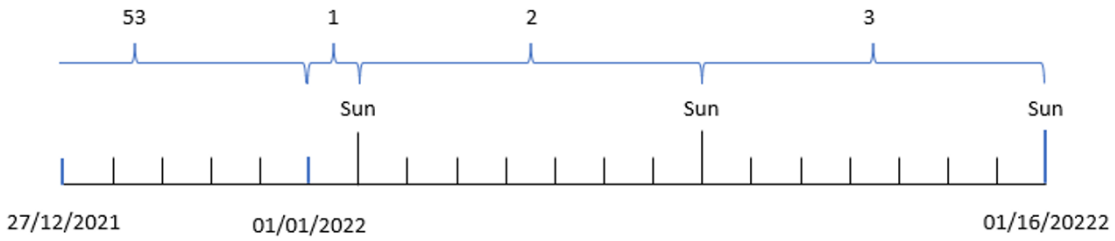
```
8184,12/28/2022,67.42
8185,12/29/2022,23.80
8186,12/30/2022,82.06
8187,12/31/2021,40.56
8188,01/01/2022,37.23
8189,01/02/2022,17.17
8190,01/03/2022,88.27
8191,01/04/2022,57.42
8192,01/05/2022,53.80
8193,01/06/2022,82.06
8194,01/07/2022,40.56
8195,01/08/2022,53.67
8196,01/09/2022,26.63
8197,01/10/2022,72.48
8198,01/11/2022,18.37
8199,01/12/2022,45.26
8200,01/13/2022,58.23
8201,01/14/2022,18.52
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `date`
- `week_day`
- `week_number`

*Diagram van `week()`-functie, diagramobjectvoorbeeld*




Resultatentabel

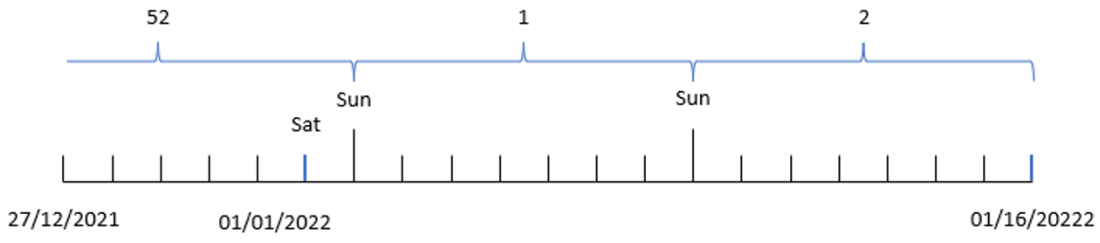
| id | date | week_day | week_nummer |
|---|---|---|---|
| 8183 | 12/27/2021 | Ma | 52 |
| 8184 | 12/28/2021 | Di | 52 |
| 8185 | 12/29/2021 | Wo | 52 |
| 8186 | 12/30/2021 | Do | 52 |

| id | date | week_day | week_nummer |
|---|---|---|---|
| 8187 | 12/31/2021 | Vr | 52 |
| 8188 | 01/01/2022 | Za | 52 |
| 8189 | 01/02/2022 | Zo | 1 |
| 8190 | 01/03/2022 | Ma | 1 |
| 8191 | 01/04/2022 | Di | 1 |
| 8192 | 01/05/2022 | Wo | 1 |
| 8193 | 01/06/2022 | Do | 1 |
| 8194 | 01/07/2022 | Vr | 1 |
| 8195 | 01/08/2022 | Za | 1 |
| 8196 | 01/09/2022 | Zo | 2 |
| 8197 | 01/10/2022 | Ma | 2 |
| 8198 | 01/11/2022 | Di | 2 |
| 8199 | 01/12/2022 | Wo | 2 |
| 8200 | 01/13/2022 | Do | 2 |
| 8201 | 01/14/2022 | Vr | 2 |

De parameter first_week_date is ingesteld op 1, waardoor dinsdag de eerste dag van de week is. De parameter broken_weeks is ingesteld op 0 waardoor de functie wordt gedwongen niet-opgesplitste weken toe te passen. De derde parameter stelt de reference_day in op 2.

De parameter `first_week_date` is ingesteld op 6, waardoor zondag de eerste dag van de week is. De parameter `broken_weeks` is ingesteld op 0 waardoor de functie wordt gedwongen niet-opgesplitste weken toe te passen.

*Diagram van de week()-functie, voorbeeld met niet-opgesplitste weken*




Door niet-opgesplitste weken te gebruiken, begint week 1 niet noodzakelijkerwijs op 1 januari. In plaats daarvan moet de week ten minste vier dagen bevatten. Daarom eindigt week 52 in de gegevensset op een zaterdag, 1 januari 2022. Week 1 begint vervolgens op de `FirstWeekDay`-systeemvariabele. Dit is zaterdag 2 januari. Deze week eindigt op de volgende zaterdag, 8 januari.

## Voorbeeld 4 – reference_day

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het derde voorbeeld.
- Het veld `week_number` wordt gemaakt en retourneert het jaar- en weeknummer waarin de transacties plaatsvonden.
- Het veld `week_day` wordt gemaakt om de weekdagwaarde van iedere transactiedatum weer te geven.

Daarnaast moet worden voldaan aan de volgende voorwaarden:

- De werkweek begint op een dinsdag.
- Het bedrijf gebruikt niet-opgesplitste weken.
- De `reference_day`-waarde is 2. Met andere woorden: het minimumaantal dagen in januari in week 1, wordt 2.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `date`
- `week_day`
- `week_number`

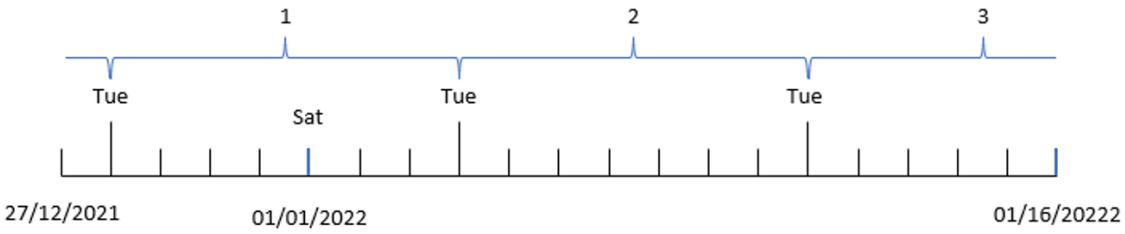
Resultatentabel

| id | date | week_day | week_nummer |
|---|---|---|---|
| 8183 | 12/27/2021 | Ma | 52 |
| 8184 | 12/28/2021 | Di | 1 |
| 8185 | 12/29/2021 | Wo | 1 |
| 8186 | 12/30/2021 | Do | 1 |
| 8187 | 12/31/2021 | Vr | 1 |
| 8188 | 01/01/2022 | Za | 1 |
| 8189 | 01/02/2022 | Zo | 1 |
| 8190 | 01/03/2022 | Ma | 1 |
| 8191 | 01/04/2022 | Di | 2 |
| 8192 | 01/05/2022 | Wo | 2 |
| 8193 | 01/06/2022 | Do | 2 |
| 8194 | 01/07/2022 | Vr | 2 |
| 8195 | 01/08/2022 | Za | 2 |
| 8196 | 01/09/2022 | Zo | 2 |
| 8197 | 01/10/2022 | Ma | 2 |
| 8198 | 01/11/2022 | Di | 3 |
| 8199 | 01/12/2022 | Wo | 3 |

| id | date | week_day | week_nummer |
|---|---|---|---|
| 8200 | 01/13/2022 | Do | 3 |
| 8201 | 01/14/2022 | Vr | 3 |

De `first_week_date`-parameter is ingesteld op 1, waardoor dinsdag de eerste dag van de week is. De `broken_weeks` -parameter is ingesteld op 0 waardoor de functie wordt gedwongen niet-opgesplitste weken toe te passen. De derde parameter stelt de `reference_day`-parameter in op 2.

*Diagram van de `week()`-functie, voorbeeld met reference_day*




Met de functie met niet-opgesplitste weken en een `reference_day`-waarde van 2 als een parameter, hoeft week 1 slechts twee dagen in januari te bevatten. Omdat de eerste dag van de week dinsdag is, begint week 1 op 28 december 2021 en eindigt hij op maandag 3 januari 2022.

## Voorbeeld 5 – Diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die het weeknummer retourneert, wordt gemaakt als een meting in een diagramobject.

### Load-script

```
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
];
```

**Resultaten**

**Doe het volgende:**

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel.
2. Voeg de volgende velden als dimensies toe:
   - `id`
   - `date`
3. Maak nu de volgende meting:
   `=week (date)`
4. Maak een , `week_day`-meting om de weekdagwaarde per transactiedatum weer te geven:
   `=weekday(date)`

Resultatentabel

| id | date | =week(datum) | =weekday(datum) |
|---|---|---|---|
| 8183 | 12/27/2021 | 53 | Ma |
| 8184 | 12/28/2021 | 53 | Di |
| 8185 | 12/29/2021 | 53 | Wo |
| 8186 | 12/30/2021 | 53 | Do |
| 8187 | 12/31/2021 | 53 | Vr |
| 8188 | 01/01/2022 | 1 | Za |
| 8189 | 01/02/2022 | 2 | Zo |
| 8190 | 01/03/2022 | 2 | Ma |
| 8191 | 01/04/2022 | 2 | Di |
| 8192 | 01/05/2022 | 2 | Wo |
| 8193 | 01/06/2022 | 2 | Do |

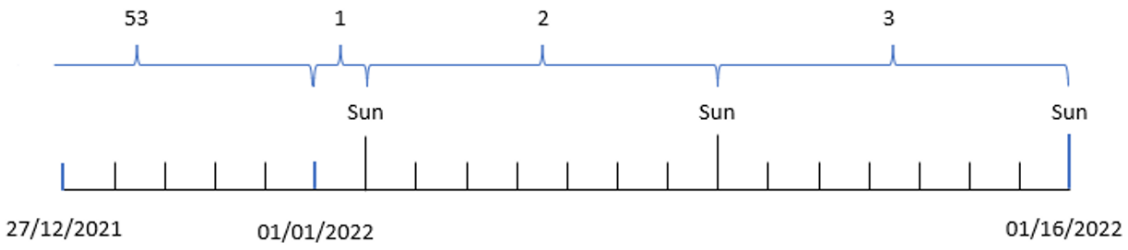
| id | date | =week(datum) | =weekday(datum) |
|---|---|---|---|
| 8194 | 01/07/2022 | 2 | Vr |
| 8195 | 01/08/2022 | 2 | Za |
| 8196 | 01/09/2022 | 3 | Zo |
| 8197 | 01/10/2022 | 3 | Ma |
| 8198 | 01/11/2022 | 3 | Di |
| 8199 | 01/12/2022 | 3 | Wo |
| 8200 | 01/13/2022 | 3 | Do |
| 8201 | 01/14/2022 | 3 | Vr |

Het veld `week_number` wordt gemaakt in de voorafgaande load-instructie met behulp van de functie `week()` en geeft het veld `date` door als het argument van de functie.

Er worden geen andere parameters doorgegeven in de functie en daarom zijn de volgende standaardvariabelen die invloed hebben op de `week()`-functie van kracht:

- `BrokenWeeks`: dDe weeknummering begint op 1 januari
- `FirstWeekDay`: de eerste dag van de week is zondag

*Diagram van `week()`-functie, diagramobjectvoorbeeld*




Omdat de applicatie de standaardsysteemvariabele `BrokenWeeks` gebruikt, begint week 1 op 1 januari, een zaterdag.

Vanwege de standaardsysteemvariabele `FirstWeekDay` begint de week op een zondag. De eerste zondag na 1 januari valt op 2 januari en daarom begint week 2 op 2 januari.

## Voorbeeld 6 – Scenario

Load-script en diagramuitdrukking

**Overzicht**

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met een set transacties voor de laatste week van 2019 en eerste twee weken van 2020 wordt geladen in de tabel `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.

De applicatie gebruikt voornamelijk opgesplitste weken op het dashboard. Maar de eindgebruiker wil een diagramobject dat de totale verkoop per week weergeeft met niet-opgesplitste weken. De referentiedag moet 2 januari zijn en de weken moeten beginnen op een dinsdag. Dit kan zelfs worden bereikt als deze dimensie niet beschikbaar is in het gegevensmodel, door de functie `week()` te gebruiken als een berekende dimensie in het diagram.

### Load-script

```
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
];
```

### Resultaten

### Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel.
2. Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

`=week(date)`

3. Maak nu de volgende aggregatiemeting:
   `=sum(amount)`
4. Stel de **Getalnotatie** van de meting in op **Geld**.
5. Selecteer het menu **Sorteren** en verwijder Aangepaste sortering voor de berekende dimensie.
6. Hef de selecties **Numeriek sorteren** en **Alfabetisch sorteren** op.

Resultatentabel

| week(date) | sum(amount) |
|---|---|
| 52 | $125.69 |
| 53 | $146.42 |
| 1 | $200.09 |
| 2 | $347.57 |
| 3 | $122.01 |

## weekday

Deze functie retourneert een duale waarde met:

- De naam van een dag zoals gedefinieerd in de omgevingsvariabele **DayNames**.
- Een geheel getal tussen 0 en 6 dat overeenkomt met de nominale dag van de week (0-6).

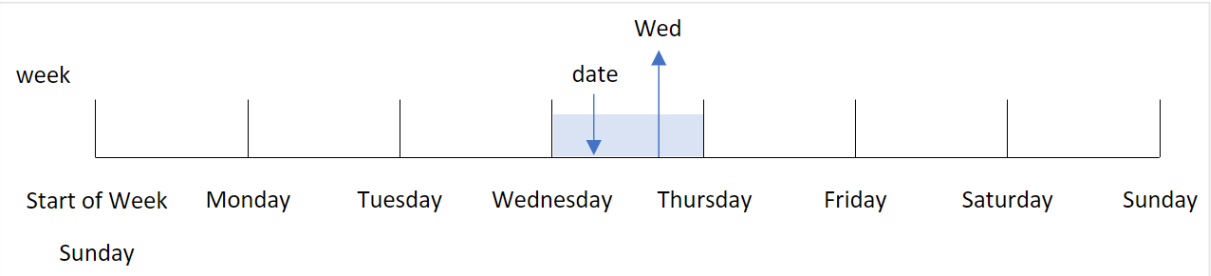
**Syntaxis:**

```
weekday(date [,first_week_day=0])
```

**Retourgegevenstypen:** dubbele waarde

De `weekday()`-functie stelt vast op welke dag van de week een datum valt. Vervolgens wordt een tekenreekswaarde geretourneerd die die dag vertegenwoordigt.

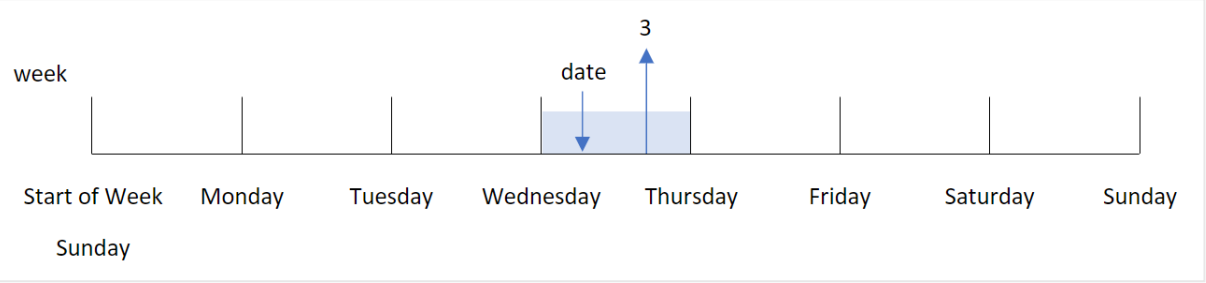
*Diagram van `weekday()`-functie die de naam van de dag retourneert waarop een datum valt*




Het resultaat retourneert de nummerwaarde die bij die dag van de week hoort (0-6), op basis van de begindag van de week. Bijvoorbeeld: als de eerste dag van de week is ingesteld op zondag, dan retourneert woensdag een nummerwaarde van `3`. Deze begindag wordt bepaald door de systeemvariabele `FirstWeekDay` of door de `first_week_day`-functieparameter.

U kunt deze getalwaarde gebruiken als onderdeel van een rekenkundige uitdrukking. Vermenigvuldig deze bijvoorbeeld met `1` om de waarde zelf te retourneren.

*Diagram van `weekday()`-functie waarbij de nummerwaarde van de dag wordt weergegeven in plaats van de naam van de dag*




### Wanneer gebruiken

De `weekday()`-functie is handig als u aggregaties per dag van de week wilt vergelijken. Bijvoorbeeld als u de gemiddelde verkoop van producten per weekdag wilt vergelijken.

Deze dimensies kunnen worden gemaakt in het load-script door de functie te gebruiken om een veld te maken in een **masterkalendertabel**, of rechtstreeks in een diagram worden gemaakt als een berekende meting.

Gerelateerde onderwerpen

| Onderwerpen | Interactie |
|---|---|
| *FirstWeekDay (page 244)* | Definieert de begindag van elke week. |

Argumenten

| Argument | Beschrijving |
|---|---|
| **date** | De te evalueren datum of tijdstempel. |
| **first_week_day** | Geeft de dag op waarop de week begint. Bij weglating wordt de waarde van de variabele **FirstWeekDay** gebruikt.<br><br>*FirstWeekDay (page 244)* |

U kunt de volgende waarden gebruiken om de dag in te stellen waarop de week begint in het argument `first_week_day`:

waarden first_week_day

| Dag | Waarde |
|---|---|
| Maandag | 0 |
| Dinsdag | 1 |

| Dag | Waarde |
|---|---|
| Woensdag | 2 |
| Donderdag | 3 |
| Vrijdag | 4 |
| Zaterdag | 5 |
| Zondag | 6 |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

*Tenzij anders aangegeven, is `FirstWeekDay` in deze voorbeelden ingesteld op `0`.*

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `weekday('10/12/1971')` | Retourneert Tue en 1. |
| `weekday('10/12/1971' , 6)` | Retourneert Tue en 2.<br><br>In dit voorbeeld gebruiken we zondag (`6`) als eerst dag van de week. |
| `SET FirstWeekDay=6;`<br>`...`<br><br>`weekday('10/12/1971')` | Retourneert Tue en 2. |

## Voorbeeld 1 - Weekdagtekenreeks

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties voor 2022 die wordt geladen in de tabel `Transactions`.
- De systeemvariabele `FirstWeekDay` die is ingesteld op 6 (zondag).
- De `DayNames`-variabele die is ingesteld om de standaarddagnamen te gebruiken.
- Een voorafgaande lading die de `weekday()`-functie bevat, die is ingesteld als het `week_day`-veld en de weekdag retourneert waarin de transacties hebben plaatsgevonden.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `date`
- `week_day`

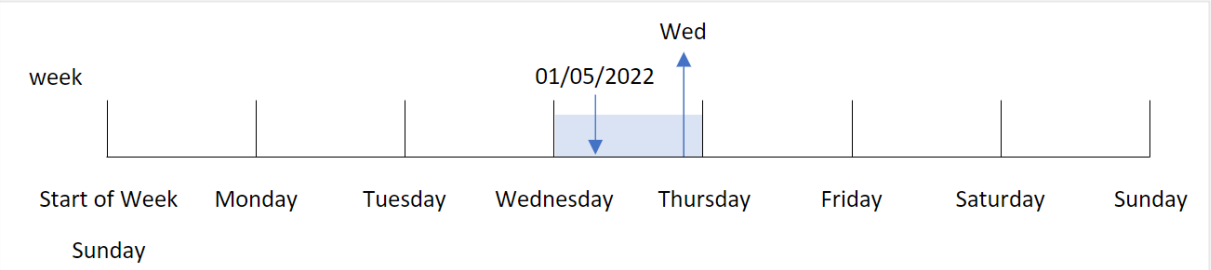
Resultatentabel

| id | date | week_day |
|---|---|---|
| 8188 | 01/01/2022 | Za |
| 8189 | 01/02/2022 | Zo |
| 8190 | 01/03/2022 | Ma |
| 8191 | 01/04/2022 | Di |
| 8192 | 01/05/2022 | Wo |
| 8193 | 01/06/2022 | Do |
| 8194 | 01/07/2022 | Vr |

Het veld '`week_day`' wordt gemaakt in de voorafgaande load-instructie met behulp van de functie `weekday()` en geeft het datumveld door als het argument van de functie.

De `weekday()`-functie retourneert de weekdagtekenreekswaarde; dat wil zeggen, hij retourneert de naam van de weekdag die is ingesteld door de `DayNames`-systeemvariabele.

*Diagram van `weekday()`-functie die woensdag als de weekdag retourneert voor transactie 8192*




Transactie 8192 vond plaats op 5 januari. De systeemvariabele `FirstWeekDay` stelt de eerste dag van de week in op zondag. De `weekday()`-functie bepaalt dat de transactie op een woensdag plaatsvond en retourneert deze waarde in afgekorte vorm van de `DayNames`-systeemvariabele, in het `week_day`-veld.

De waarden in het `week_day`-veld zijn rechts uitgelijnd in de kolom omdat er een dubbel nummer- en tekstresultaat bestaat voor het veld (woensdag 3). Om de veldwaarde om te zetten in de bijbehorende nummerequivalent, kan het veld in de `num()`-functie worden omsloten. Zo kan in transactie 8192 de woensdagwaarde worden omzet in nummer 3.

## Voorbeeld 2 – first_week_day

### Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2022 die wordt geladen in de tabel `Transactions`.
- De systeemvariabele `FirstWeekDay` die is ingesteld op 6 (zondag).
- De `DayNames`-variabele die is ingesteld om de standaarddagnamen te gebruiken.
- Een voorafgaande lading die de `weekday()`-functie bevat, die is ingesteld als het `week_day`-veld en de weekdag retourneert waarin de transacties hebben plaatsgevonden.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

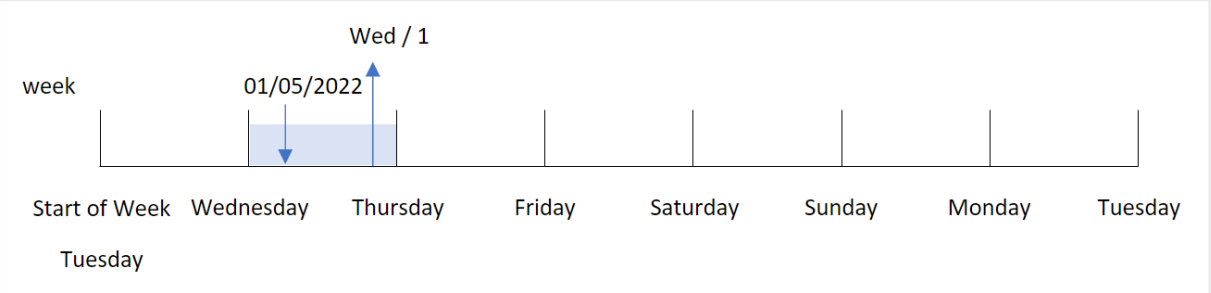
- `id`
- `date`
- `week_day`

Resultatentabel

| id | date | week_day |
|---|---|---|
| 8188 | 01/01/2022 | Za |
| 8189 | 01/02/2022 | Zo |
| 8190 | 01/03/2022 | Ma |
| 8191 | 01/04/2022 | Di |
| 8192 | 01/05/2022 | Wo |

| id | date | week_day |
|---|---|---|
| 8193 | 01/06/2022 | Do |
| 8194 | 01/07/2022 | Vr |

*Diagram van de `weekday()`-functie dat laat zien dat woensdag de dubbele nummerwaarde van 1 heeft*




Omdat het `first_week_day`-argument is ingesteld op `1` in de `weekday()`-functie, wordt dinsdag als eerste dag van de week gebruikt. Daarom krijgen alle transacties die op een dinsdag plaatsvinden een dubbele nummerwaarde van `0`.

Transactie 8192 vond plaats op 5 januari. De `weekday()`-functie identificeert dit als een woensdag en daarom zou de uitdrukking de dubbele nummerwaarde van `1` retourneren.

## Voorbeeld 3 - Diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2022 die wordt geladen in de tabel `Transactions`.
- De systeemvariabele `FirstWeekDay` die is ingesteld op `6` (zondag).
- De `DayNames`-variabele die is ingesteld om de standaarddagnamen te gebruiken.

In dit voorbeeld is de gegevensverzameling echter niet gewijzigd en wordt in de applicatie geladen. De berekening die de weekdagwaarde identificeert, wordt gemaakt als een meting in een diagram in de app.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `date`

Maak de volgende meting om de weekdagwaarde te berekenen:

- `=weekday(date)`

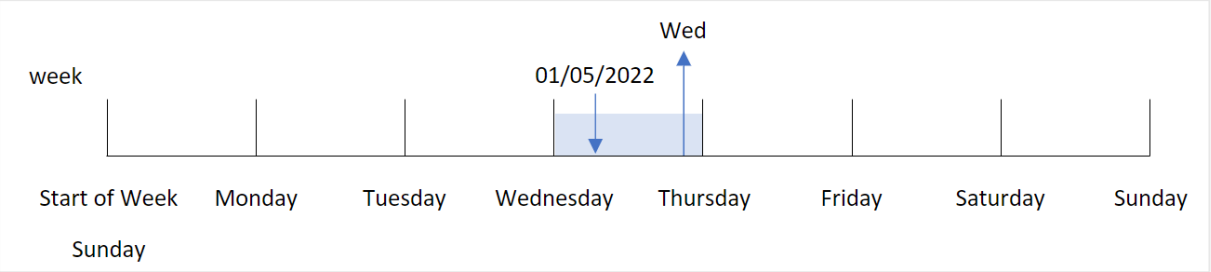
Resultatentabel

| id | date | =weekday(datum) |
|---|---|---|
| 8188 | 01/01/2022 | Za |
| 8189 | 01/02/2022 | Zo |
| 8190 | 01/03/2022 | Ma |
| 8191 | 01/04/2022 | Di |
| 8192 | 01/05/2022 | Wo |
| 8193 | 01/06/2022 | Do |
| 8194 | 01/07/2022 | Vr |

Het veld `=weekday(date)` wordt gemaakt in het diagram door de functie `weekday()` te gebruiken en het datumveld door te geven als het argument van de functie.

De `weekday()`-functie retourneert de weekdagtekenreekswaarde; dat wil zeggen, hij retourneert de naam van de weekdag die is ingesteld door de `DayNames`-systeemvariabele.

*Diagram van `weekday()`-functie die woensdag als de weekdag retourneert voor transactie 8192*




Transactie 8192 vond plaats op 5 januari. De systeemvariabele `FirstWeekDay` stelt de eerste dag van de week in op zondag. De `weekday()`-functie bepaalt dat de transactie op een woensdag plaatsvond en retourneert deze waarde in afgekorte vorm van de `DayNames`-systeemvariabele, in het `=weekday(date)`-veld.

## Voorbeeld 4 - Scenario

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2022 die wordt geladen in de tabel `Transactions`.
- De systeemvariabele `FirstWeekDay` die is ingesteld op `6` (zondag).
- De `DayNames`-variabele die is ingesteld om de standaarddagnamen te gebruiken.

De eindgebruiker wil graag een diagram die de gemiddelde verkoop per weekdag voor de transacties weergeeft.

### Load-script

```
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);
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `=weekday(date)`
- `=avg(amount)`

Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

| weekday(date) | Avg(amount) |
|---|---|
| Zo | $536.96 |
| Ma | $500.80 |
| Di | $515.63 |
| Wo | $509.21 |
| Do | $482.70 |
| Vr | $441.33 |
| Za | $505.22 |

## weekend

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de laatste milliseconde van de laatste dag (zondag) van de kalenderweek die **date** bevat. De standaarduitvoernotatie is de **DateFormat** die is ingesteld in het script.

**Syntaxis:**

```
WeekEnd(timestamp [, period_no [, first_week_day ]])
```

**Retourgegevenstypen:** dubbele waarde

De `weekend()`-functie bepaalt welke week de datum in valt. Het retourneert vervolgens een tijdstempel, in datumnotatie, voor de laatste milliseconde van die week. De eerste dag van de week wordt bepaald door de omgevingsvariabele `FirstWeekDay`. Maar dit kan worden vervangen door het `first_week_day`-argument in de `weekend()`-functie.

Argumenten

| Argument | Beschrijving |
|---|---|
| **timestamp** | De te evalueren datum of tijdstempel. |
| **period_no** | **shift** is een geheel getal, waarbij de waarde 0 de week aangeeft die **date** bevat. Negatieve waarden in verschuiving geven voorgaande weken aan en positieve waarden geven volgende weken aan. |

| Argument | Beschrijving |
|---|---|
| **first_week_day** | Geeft de dag op waarop de week begint. Bij weglating wordt de waarde van de variabele **FirstWeekDay** gebruikt.<br><br>De mogelijke waarden voor **first_week_day** zijn 0 voor maandag, 1 voor dinsdag, 2 voor woensdag, 3 voor donderdag, 4 voor vrijdag, 5 voor zaterdag en 6 voor zondag.<br><br>Zie *FirstWeekDay (page 244)* voor meer informatie over de systeemvariabele |

## Wanneer gebruiken

De `weekend()`-functie wordt doorgaans gebruikt als onderdeel van een uitdrukking als de gebruiker wil dat in de berekening de resterende dagen van de week voor de opgegeven datum wordt gebruikt. Hij zou bijvoorbeeld kunnen worden gebruikt als een gebruiker de totale rente wil berekenen die tijdens de week nog niet is opgebouwd.

De volgende voorbeelden gaan uit van het volgende:

```
SET FirstWeekDay=0;
```

| Voorbeeld | Resultaat |
|---|---|
| `weekend('01/10/2013')` | Retourneert 01/12/2013 23:59:59. |
| `weekend('01/10/2013', -1)` | Retourneert 01/05/2013 23:59:59.. |
| `weekend('01/10/2013', 0, 1)` | Retourneert 01/14/2013 23:59:59. |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

**Voorbeelden:**

Als u ISO-instellingen wilt voor weken en weeknummers, moet u het volgende opnemen in het script:

```
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
```

Als u US-instellingen wilt, moet u het volgende opnemen in het script:

```
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
```

De bovenstaande voorbeelden resulteren in het volgende op basis van de functie `weekend()`:

Voorbeeld van de functie Weekend

| Datum | ISO laatste dag van de week | US laatste dag van de week |
|---|---|---|
| Za 2020 Dec 26 | 2020-12-27 | 12/26/2020 |
| Zo 2020 Dec 27 | 2020-12-27 | 1/2/2021 |
| Ma 2020 Dec 28 | 2021-01-03 | 1/2/2021 |
| Di 2020 Dec 29 | 2021-01-03 | 1/2/2021 |
| Wo 2020 Dec 30 | 2021-01-03 | 1/2/2021 |
| Do 2020 Dec 31 | 2021-01-03 | 1/2/2021 |
| Vr 2021 Jan 1 | 2021-01-03 | 1/2/2021 |
| Za 2021 Jan 2 | 2021-01-03 | 1/2/2021 |
| Zo 2021 Jan 3 | 2021-01-03 | 1/9/2021 |
| Ma 2021 Jan 4 | 2021-01-10 | 1/9/2021 |
| Di 2021 Jan 5 | 2021-01-10 | 1/9/2021 |

*De week eindigt op zondag in de ISO-kolom en op zaterdag in de US-kolom.*

## Voorbeeld 1 – Basisvoorbeeld

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties voor 2022 die wordt geladen in de tabel `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.

- Het maken van een veld, `end_of_week`, dat een tijdstempel voor het einde van de week retourneert waarin de transactie plaatsvond.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `end_of_week`
- `end_of_week_timestamp`

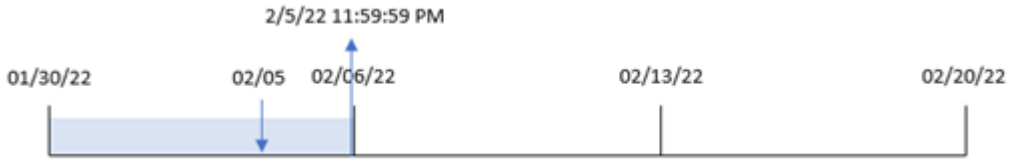
Resultatentabel

| 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 |
| 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 |

Het veld '`end_of_week`' wordt gemaakt in de voorafgaande load-instructie met behulp van de functie `weekend()` en geeft het datumveld door als het argument van de functie.

De functie `weekend()` identificeert in welke week de datumwaarde valt en retourneert een tijdstempel voor de laatste milliseconde van die week.

*Diagram van `weekend()`-functie, basisvoorbeeld*



Transactie 8191 vond plaats op 5 februari. De systeemvariabele `FirstWeekDay` stelt de eerste dag van de week in op een zondag. De `weekend()`-functie identificeert dat de eerste zaterdag na 5 februari, en daarom het einde van de week, op 5 februari viel. Daarom retourneert de `end_of_week`-waarde voor die transactie de laatste milliseconde van die dag, die op 5 februari om 11:59:59 PM was.

## Voorbeeld 2 – period_no

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het maken van een veld, `previous_week_end`, dat de tijdstempel voor het einde van de week retourneert voordat de transactie plaatsvond.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

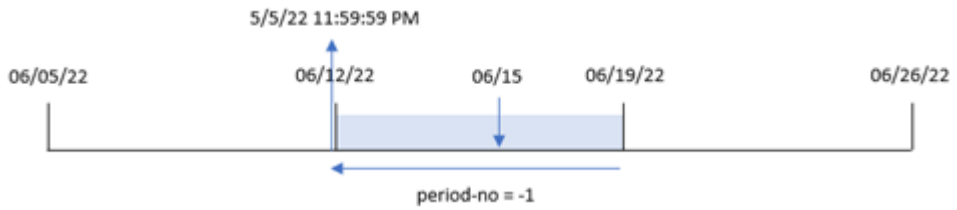
- `date`
- `previous_week_end`
- `previous_week_end_timestamp`

Resultatentabel

| 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 dit geval, omdat een `period_no` van -1 werd gebruikt als het offset-argument in de `weekend()`-functie, identificeert de functie eerst de week waarin de transacties plaatsvinden. Het kijkt dan een week eerder en identificeert de laatste milliseconde van die week.

*Diagram van weekend()-functie, voorbeeld period_no*




Transactie 8196 vond plaats op 15 juni. De `weekend()`-functie identificeert dat de week op 12 juni begint. Daarom eindigt de vorige week op 11 juni om 11:59:59 PM; dit is de waarde die voor het `previous_week_end`-veld wordt geretourneerd.

## Voorbeeld 3 – first_week_day

### Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld. Maar in dit voorbeeld moeten we dinsdag instellen als de eerste dag van de werkweek.

#### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

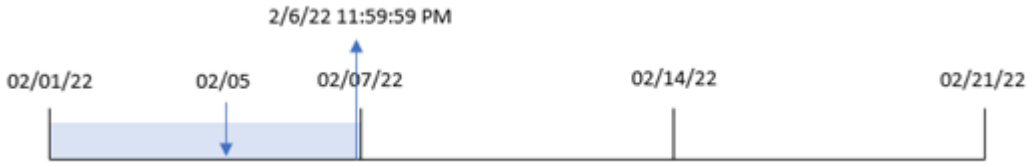
- `date`
- `end_of_week`
- `end_of_week_timestamp`

Resultatentabel

| 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 dit geval geldt dat omdat het `first_week_date`-argument 1 wordt gebruikt in de `weekend()`-functie, dinsdag als eerste dag van de week wordt ingesteld.

*Diagram van de `weekend()`-functie, voorbeeld met first_week_day*




Transactie 8191 vond plaats op 5 februari. De `weekend()`-functie identificeert dat de eerste maandag na deze datum, en daarom het einde van de week en geretourneerde waarde, op 6 februari om 11:59:59 PM viel.

## Voorbeeld 4 – diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld. In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die een tijdstempel retourneert voor het einde van de week waarin de transacties plaatsvonden, wordt gemaakt als meting in een diagramobject van de toepassing.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `date`.

Voeg de volgende metingen toe om de start van de week waarin een transactie plaatsvindt te berekenen:

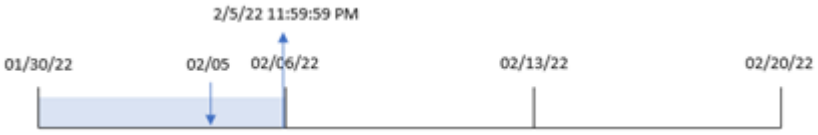
- `=weekend(date)`
- `=timestamp(weekend(date))`

Resultatentabel

| 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 |

De meting `end_of_week` wordt gemaakt in het diagramobject met behulp van de functie `weekend()` en geeft het veld door als het argument van de functie. De `weekend()`-functie identificeert in welke week de datumwaarde valt en retourneert een tijdstempel voor de laatste milliseconde van die week.

*Diagram van `weekend()`-functie, diagramobjectvoorbeeld*




Transactie 8191 vond plaats op 5 februari. De systeemvariabele `FirstWeekDay` stelt de eerste dag van de week in op een zondag. De `weekend()`-functie identificeert dat de eerste zaterdag na 5 februari, en daarom het einde van de week, op 5 februari viel. Daarom retourneert de `end_of_week`-waarde voor die transactie de laatste milliseconde van die dag, die op 5 februari om 11:59:59 PM was.

## Voorbeeld 5 – Scenario

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset die wordt geladen in een tabel met de naam `Employee_Expenses`.
- Gegevens omvatten werknemers-id's, werknemersnamen en de gemiddelde dagelijkse onkostendeclaraties van elke werknemer.

De eindgebruiker wil graag een diagramobject dat per werknemer-id en werknemersnaam de geschatte nog te maken onkostendeclaraties voor de rest van de week weergeeft.

### Load-script

```
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
];
```

**Resultaten**

**Doe het volgende:**

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:
   - `employee_id`
   - `employee_name`
2. Maak nu een meting om de opgebouwde rente te berekenen:
   `=(weekend(today(1))-today(1))*avg_daily_claim`
3. Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

| employee_id | employee_name | =(weekend(today(1))-today(1))*avg_daily_claim |
|---|---|---|
| 182 | Mark | $90.00 |
| 183 | Deryck | $75.00 |
| 184 | Dexter | $75.00 |
| 185 | Sydney | $162.00 |
| 186 | Agatha | $108.00 |

Door de datum van vandaag als het enige argument te gebruiken, retourneert de `weekend()`-functie de einddatum van de huidige week. Door vervolgens de datum van vandaag af te trekken van de einddatum van de week, retourneert de uitdrukking het aantal resterende dagen in deze week.

Deze waarde wordt vervolgens vermenigvuldigd met de gemiddelde dagelijkse onkostendeclaratie van elke werknemer om de geschatte waarde te berekenen van de declaraties die elke werknemer naar verwachting in de resterende week zal indienen.

## weekname

Deze functie retourneert het jaar en weeknummer met een onderliggende numerieke waarde die overeenkomt met een tijdstempel van de eerste milliseconde van de eerste dag van de week die **date** bevat.

**Syntaxis:**

```
WeekName(date[, period_no [, first_week_day [, broken_weeks [, reference_day]]]])
```

De functie `weekname()` bepaalt in welke week de datum valt en retourneert het weeknummer en het jaar van die week. De eerste dag van de week wordt bepaald door de systeemvariabele `FirstWeekDay`. U kunt echter ook de eerste dag van de week wijzigen door het argument `first_week_day` in de functie `weekname()` te gebruiken.

In Qlik Sense worden de landinstellingen opgehaald wanneer de app wordt gemaakt en de bijbehorende instellingen worden als omgevingsvariabelen opgeslagen in het script.

Een Noord-Amerikaanse app-ontwikkelaar krijgt vaak `Set BrokenWeeks=1;` in het script, wat overeenkomt met gebroken weken. Een Europese app-ontwikkelaar krijgt vaak `Set BrokenWeeks=0;` in het script, wat overeenkomt met ongebroken weken.

Als uw applicatie gebroken weken gebruikt, dan begint het weeknummer op 1 januari en eindigt op de dag voorafgaand aan de systeemvariabele `FirstWeekDay`, ongeacht hoeveel dagen er zijn verstreken.

Als uw toepassing echter ononderbroken weken gebruikt, kan week 1 in het voorgaande jaar of in de eerste paar dagen van januari beginnen. Dit hangt af van hoe u de systeemvariabelen `ReferenceDay` en `FirstWeekDay` gebruikt.

Voorbeeld van de functie Weekname

| Datum | ISO-weeknaam | US-weeknaam |
|---|---|---|
| Za 2020 Dec 26 | 2020/52 | 2020/52 |
| Zo 2020 Dec 27 | 2020/52 | 2020/53 |
| Ma 2020 Dec 28 | 2020/53 | 2020/53 |
| Di 2020 Dec 29 | 2020/53 | 2020/53 |
| Wo 2020 Dec 30 | 2020/53 | 2020/53 |
| Do 2020 Dec 31 | 2020/53 | 2020/53 |
| Vr 2021 Jan 1 | 2020/53 | 2021/01 |
| Za 2021 Jan 2 | 2020/53 | 2021/01 |
| Zo 2021 Jan 3 | 2020/53 | 2021/02 |
| Ma 2021 Jan 4 | 2021/01 | 2021/02 |
| Di 2021 Jan 5 | 2021/01 | 2021/02 |

### Wanneer gebruiken

De functie `weekname()` is handig als u aggregaties per week wilt vergelijken.

Bijvoorbeeld als u de totale verkoop van producten per week wilt zien. Gebruik `weekname()` in plaats van `lunarweekname()` om consistentie te behouden met de omgevingsvariabele `BrokenWeeks` in de toepassing. Als de toepassing ononderbroken weken gebruikt, kan week 1 datums van december van het voorgaande jaar bevatten of datums in januari van het lopende jaar uitsluiten. Als de toepassing gebruikmaakt van gebroken weken, kan week 1 minder dan zeven dagen bevatten.

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **timestamp** | De te evalueren datum of tijdstempel. |

| Argument | Beschrijving |
|---|---|
| **period_no** | **shift** is een geheel getal, waarbij de waarde 0 de week aangeeft die **date** bevat. Negatieve waarden in verschuiving geven voorgaande weken aan en positieve waarden geven volgende weken aan. |
| **first_week_day** | Geeft de dag op waarop de week begint. Bij weglating wordt de waarde van de variabele **FirstWeekDay** gebruikt.<br><br>De mogelijke **first_week_day**-waarden zijn 0 voor maandag, 1 voor dinsdag, 2 voor woensdag, 3 voor donderdag, 4 voor vrijdag, 5 voor zaterdag en 6 voor zondag.<br><br>Zie *FirstWeekDay (page 244)* voor meer informatie over de systeemvariabele. |
| **broken_weeks** | Als u **broken_weeks** (gebroken weken) niet specificeert, dan wordt de waarde van variabele **BrokenWeeks** gebruikt om te definiëren of weken al dan niet gebroken zijn. |
| **reference_day** | Als u **reference_day** (referentiedag) niet specificeert, dan wordt de waarde van variabele **ReferenceDay** gebruikt om te definiëren welke dag in januari wordt ingesteld als referentiedag om week 1 te definiëren. Standaard wordt in Qlik Sense-functies 4 gebruikt als de referentiedag. Dit betekent dat week 1 4 januari moet bevatten of, anders gezegd, dat week 1 altijd ten minste 4 dagen in januari moet hebben. |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

De onderstaande voorbeelden gaan uit van het volgende:

```
Set FirstWeekDay=0;
Set BrokenWeeks=0;
Set ReferenceDay=4;
```

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| weekname('01/12/2013') | Retourneert 2013/02. |
| weekname('01/12/2013', -1) | Retourneert 2013/01. |
| weekname('01/12/2013', 0, 1) | Retourneert 2013/02. |

## Voorbeeld 1 – datum zonder aanvullende argumenten

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties voor de laatste week van 2021 en eerste twee weken van 2022 wordt geladen in een tabel met de naam '`Transactions`'.
- De systeemvariabele `DateFormat` die is ingesteld op de indeling `MM/DD/YYYY`.
- De systeemvariabele `BrokenWeeks` die is ingesteld op `1`.
- De systeemvariabele `FirstWeekDay` die is ingesteld op `6`.
- Een voorgaande lading die het volgende bevat:
    - De functie `weekday()` die is ingesteld als het veld, '`week_number`', dat het jaar- en weeknummer retourneert wanneer de transacties plaatsvonden.
    - De functie `weekname()` die is ingesteld als het veld met de naam '`week_day`', om de weekdagwaarde van elke transactiedatum weer te geven.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `date`
- `week_day`
- `week_number`

Resultatentabel

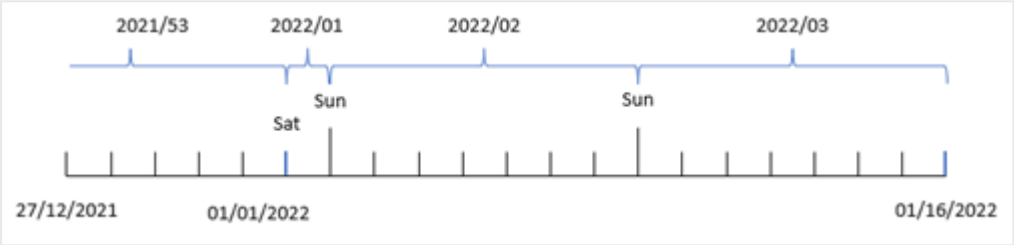
| id | date | week_day | week_nummer |
|---|---|---|---|
| 8183 | 12/27/2021 | Ma | 2021/53 |
| 8184 | 12/28/2021 | Di | 2021/53 |
| 8185 | 12/29/2021 | Wo | 2021/53 |
| 8186 | 12/30/2021 | Do | 2021/53 |
| 8187 | 12/31/2021 | Vr | 2021/53 |
| 8188 | 01/01/2022 | Za | 2022/01 |
| 8189 | 01/02/2022 | Zo | 2022/02 |
| 8190 | 01/03/2022 | Ma | 2022/02 |
| 8191 | 01/04/2022 | Di | 2022/02 |
| 8192 | 01/05/2022 | Wo | 2022/02 |
| 8193 | 01/06/2022 | Do | 2022/02 |
| 8194 | 01/07/2022 | Vr | 2022/02 |
| 8195 | 01/08/2022 | Za | 2022/02 |
| 8196 | 01/09/2022 | Zo | 2022/03 |
| 8197 | 01/10/2022 | Ma | 2022/03 |
| 8198 | 01/11/2022 | Di | 2022/03 |
| 8199 | 01/12/2022 | Wo | 2022/03 |
| 8200 | 01/13/2022 | Do | 2022/03 |
| 8201 | 01/14/2022 | Vr | 2022/03 |

Het veld 'week_number' wordt gemaakt in de voorafgaande load-instructie met behulp van de functie weekname() en geeft het datumveld door als het argument van de functie.

De functie weekname() identificeert aanvankelijk in welke week de datumwaarde valt en retourneert het aantal weeknummers en het jaar waarin de transactie plaatsvindt.

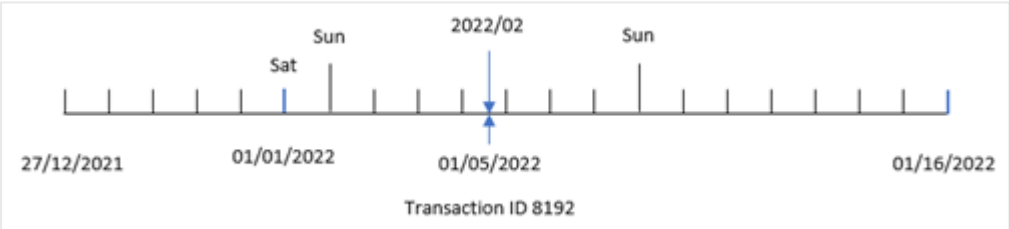
De systeemvariabele FirstWeekDay stelt zondag in als de eerste dag van de week. De systeemvariabele BrokenWeeks stelt de toepassing in op het gebruik van gebroken weken, wat betekent dat week 1 op 1 januari begint.

*Diagram van de functie weekname() met de standaardvariabelen.*




Week 1 begint op 1 januari, wat een zaterdag is, en daarom retourneren transacties die op deze datum plaatsvinden de waarde 2022/01 (het jaar- en weeknummer).

*Diagram van de functie weekname() die het weeknummer van transactie 8192 identificeert.*




Omdat de toepassing gebroken weken gebruikt en de eerste weekdag zondag is, retourneren transacties die plaatsvinden van 2 januari tot 8 januari de waarde 2022/02 (weeknummer 2 in 2022). Een voorbeeld hiervan is transactie 8192 die plaatsvond op 5 januari en de waarde 2022/02 retourneert voor het veld 'week_number'.

## Voorbeeld 2 – period_no

### Load-script en resultaten

#### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is het echter de taak om een veld te maken, 'previous_week_number', dat het jaar en het weeknummer retourneert voordat de transacties plaatsvonden.

Open de Editor voor laden van gegevens en voeg het volgende load-script toe aan een nieuwe sectie.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `date`
- `week_day`
- `week_number`

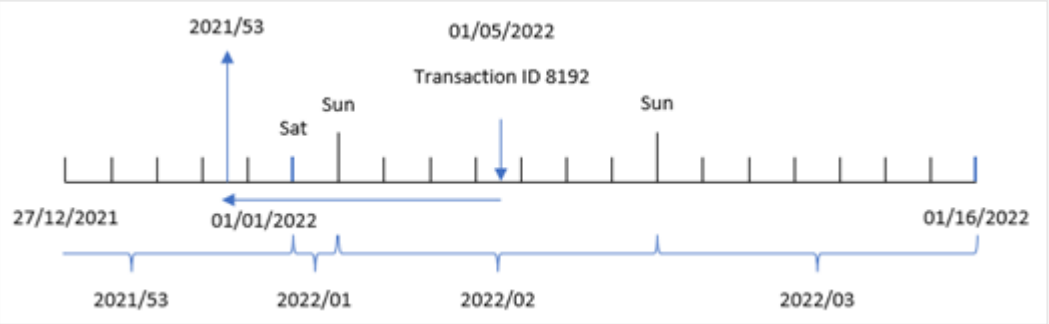
Resultatentabel

| id | date | week_day | week_nummer |
|---|---|---|---|
| 8183 | 12/27/2021 | Ma | 2021/52 |
| 8184 | 12/28/2021 | Di | 2021/52 |

| id | date | week_day | week_nummer |
|---|---|---|---|
| 8185 | 12/29/2021 | Wo | 2021/52 |
| 8186 | 12/30/2021 | Do | 2021/52 |
| 8187 | 12/31/2021 | Vr | 2021/52 |
| 8188 | 01/01/2022 | Za | 2021/52 |
| 8189 | 01/02/2022 | Zo | 2021/53 |
| 8190 | 01/03/2022 | Ma | 2021/53 |
| 8191 | 01/04/2022 | Di | 2021/53 |
| 8192 | 01/05/2022 | Wo | 2021/53 |
| 8193 | 01/06/2022 | Do | 2021/53 |
| 8194 | 01/07/2022 | Vr | 2021/53 |
| 8195 | 01/08/2022 | Za | 2022/01 |
| 8196 | 01/09/2022 | Zo | 2022/02 |
| 8197 | 01/10/2022 | Ma | 2022/02 |
| 8198 | 01/11/2022 | Di | 2022/02 |
| 8199 | 01/12/2022 | Wo | 2022/02 |
| 8200 | 01/13/2022 | Do | 2022/02 |
| 8201 | 01/14/2022 | Vr | 2022/02 |

Omdat een `period_no` van `-1` werd gebruikt als het offset-argument in de functie `weekname()`, identificeert de functie eerst de week waarin de transacties plaatsvinden. Het kijkt dan een week eerder en identificeert de eerste milliseconde van die week.

*Diagram van functie `weekname()` met een offset `period_no` van -1.*




Transactie 8192 vond plaats op 5 januari 2022. De functie `weekname()` kijkt een week eerder, 30 december 2021, en retourneert het weeknummer en het jaar voor die datum - `2021/53`.

## Voorbeeld 3 – first_week_day

Load-script en resultaten

### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is het bedrijfsbeleid echter dat de werkweek op dinsdag begint.

Open de Editor voor laden van gegevens en voeg het volgende load-script toe aan een nieuwe sectie.

### Load-script

```
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
];
```

### Resultaten

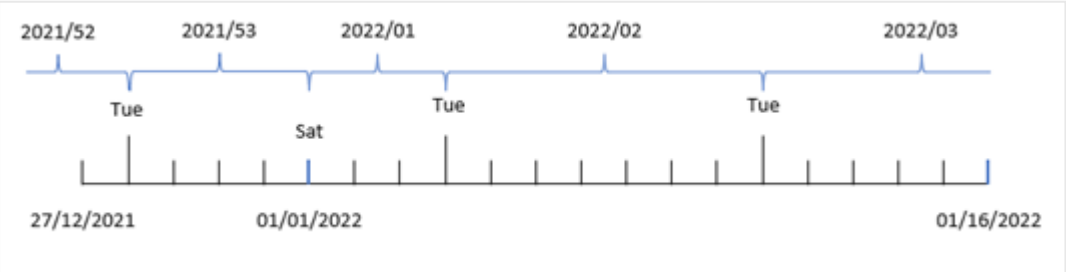
Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date
- week_day
- week_number

Resultatentabel

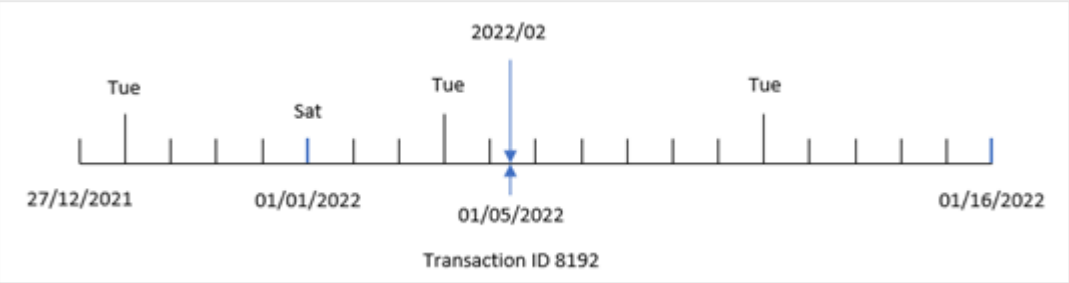
| id | date | week_day | week_nummer |
|---|---|---|---|
| 8183 | 12/27/2021 | Ma | 2021/52 |
| 8184 | 12/28/2021 | Di | 2021/53 |
| 8185 | 12/29/2021 | Wo | 2021/53 |
| 8186 | 12/30/2021 | Do | 2021/53 |
| 8187 | 12/31/2021 | Vr | 2021/53 |
| 8188 | 01/01/2022 | Za | 2022/01 |
| 8189 | 01/02/2022 | Zo | 2022/01 |
| 8190 | 01/03/2022 | Ma | 2022/01 |
| 8191 | 01/04/2022 | Di | 2022/02 |
| 8192 | 01/05/2022 | Wo | 2022/02 |
| 8193 | 01/06/2022 | Do | 2022/02 |
| 8194 | 01/07/2022 | Vr | 2022/02 |
| 8195 | 01/08/2022 | Za | 2022/02 |
| 8196 | 01/09/2022 | Zo | 2022/02 |
| 8197 | 01/10/2022 | Ma | 2022/02 |
| 8198 | 01/11/2022 | Di | 2022/03 |
| 8199 | 01/12/2022 | Wo | 2022/03 |
| 8200 | 01/13/2022 | Do | 2022/03 |
| 8201 | 01/14/2022 | Vr | 2022/03 |

*Diagram van functie `weekname()` met dinsdag als de eerste dag van de week.*



Omdat het argument `first_week_date` van `1` wordt gebruikt in de functie `weekname()`, wordt dinsdag als eerste dag van de week gebruikt. De functie bepaalt dus dat week 53 van 2021 begint op dinsdag 28 december; en vanwege de toepassing met gebroken weken begint week 1 op 1 januari 2022 en eindigt op de laatste milliseconde van maandag 3 januari 2022.

*Diagram met weeknummer van transactie 8192 met dinsdag als eerste dag van de week.*




Transactie 8192 vond plaats op 5 januari 2022. Daarom, met behulp van een parameter `first_week_day` van dinsdag, retourneert de functie `weekname()` de waarde `2022/02` voor het veld '`week_number`'.

## Voorbeeld 4 – diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is de gegevensverzameling echter niet gewijzigd en wordt in de applicatie geladen. De berekening die het jaarnummer van de week retourneert maand waarin de transacties plaatsvonden is gemaakt als meting in een diagramobject van de toepassing.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `date`
- `=week_day (date)`

Maak de volgende meting om de start van de week waarin een transactie plaatsvindt te berekenen:

```
=weekname(date)
```

Resultatentabel

| id | date | =weekday(date) | =weekname(date) |
|---|---|---|---|
| 8183 | 12/27/2021 | Ma | 2021/53 |
| 8184 | 12/28/2021 | Di | 2021/53 |
| 8185 | 12/29/2021 | Wo | 2021/53 |
| 8186 | 12/30/2021 | Do | 2021/53 |
| 8187 | 12/31/2021 | Vr | 2021/53 |
| 8188 | 01/01/2022 | Za | 2022/01 |
| 8189 | 01/02/2022 | Zo | 2022/02 |
| 8190 | 01/03/2022 | Ma | 2022/02 |
| 8191 | 01/04/2022 | Di | 2022/02 |
| 8192 | 01/05/2022 | Wo | 2022/02 |
| 8193 | 01/06/2022 | Do | 2022/02 |
| 8194 | 01/07/2022 | Vr | 2022/02 |
| 8195 | 01/08/2022 | Za | 2022/02 |
| 8196 | 01/09/2022 | Zo | 2022/03 |
| 8197 | 01/10/2022 | Ma | 2022/03 |
| 8198 | 01/11/2022 | Di | 2022/03 |
| 8199 | 01/12/2022 | Wo | 2022/03 |

| id | date | =weekday(date) | =weekname(date) |
|---|---|---|---|
| 8200 | 01/13/2022 | Do | 2022/03 |
| 8201 | 01/14/2022 | Vr | 2022/03 |

Het veld 'week_number' wordt gemaakt als een meting in het diagramobject door de functie weekname() te gebruiken en het datumveld door te geven als het argument van de functie.

De functie weekname() identificeert aanvankelijk in welke week de datumwaarde valt en retourneert het aantal weeknummers en het jaar waarin de transactie plaatsvindt.

De systeemvariabele FirstWeekDay stelt zondag in als de eerste dag van de week. De systeemvariabele BrokenWeeks stelt de toepassing in op het gebruik van gebroken weken, wat betekent dat week 1 op 1 januari begint.

*Diagram met weeknummer met zondag als eerste dag van de week.*

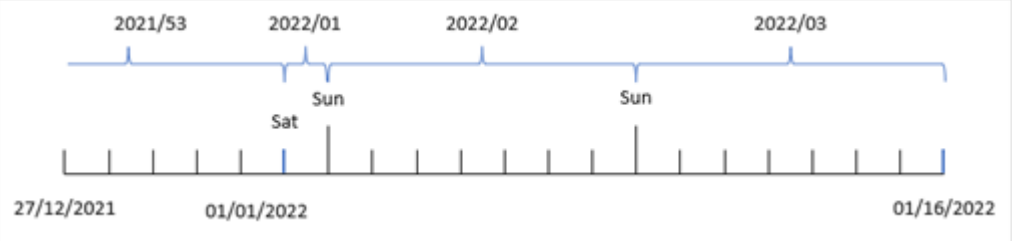
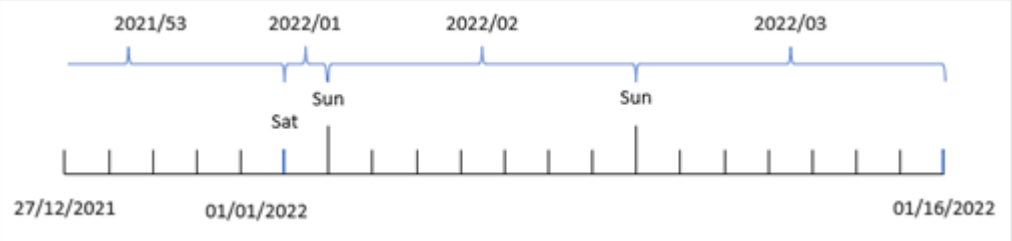



*Diagram waarop te zien is dat transactie 8192 in week nummer twee heeft plaatsgevonden.*




Omdat de toepassing gebroken weken gebruikt en de eerste weekdag zondag is, retourneren transacties die plaatsvinden van 2 januari tot 8 januari de waarde 2022/02, weeknummer 2 in 2022. Transactie 8192 vond plaats op 5 januari en de waarde retourneert 2022/02 voor het veld 'week_number'.

## Voorbeeld 5 – Scenario

Load-script en resultaten

**Overzicht**

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met een set transacties voor de laatste week van 2019 en eerste twee weken van 2020 wordt geladen in een tabel met de naam '`Transactions`'.
- De systeemvariabele `BrokenWeeks` die is ingesteld op `0`.
- De systeemvariabele `ReferenceDay` die is ingesteld op `2`.
- De systeemvariabele `DateFormat` die is ingesteld op de indeling `MM/DD/YYYY`.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel.

Maak een berekende dimensie met behulp van de volgende uitdrukking:

```
=weekname(date)
```

Maak de volgende aggregatiemeting om de totale verkoop te berekenen:

```
=sum(amount)
```

Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

| weekname(date) | =sum(amount) |
|---|---|
| 2019/52 | $125.69 |
| 2020/01 | $346.51 |
| 2020/02 | $347.57 |
| 2020/03 | $122.01 |

Om de resultaten van het gebruik van de functie weekname() in dit scenario te demonstreren, voegt u het volgende veld toe als een dimensie:

`date`

Resultatentabel met datumveld

| 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 |

Omdat de toepassing ononderbroken weken gebruikt en week 1 minimaal twee dagen in januari vereist vanwege de systeemvariabele `ReferenceDay`, bevat week 1 van 2020 transacties vanaf 29 december 2019.

## weekstart

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de eerste milliseconde van de eerste dag van de kalenderweek die **date** bevat. De standaarduitvoernotatie is de **DateFormat** die is ingesteld in het script.

**Syntaxis:**

```
WeekStart(timestamp [, period_no [, first_week_day ]])
```

**Retourgegevenstypen:** dubbele waarde

De `weekstart()`-functie bepaalt welke week de datum in valt. Het retourneert vervolgens een tijdstempel, in datumnotatie, voor de eerste milliseconde van die week. De eerste dag van de week wordt bepaald door de omgevingsvariabele `FirstWeekDay`. Maar dit kan worden vervangen door het `first_week_day`-argument in de `weekstart()`-functie.

Argumenten

| Argument | Beschrijving |
|---|---|
| **timestamp** | De te evalueren datum of tijdstempel. |
| **period_no** | **shift** is een geheel getal, waarbij de waarde 0 de week aangeeft die **date** bevat. Negatieve waarden in verschuiving geven voorgaande weken aan en positieve waarden geven volgende weken aan. |
| **first_week_day** | Geeft de dag op waarop de week begint. Bij weglating wordt de waarde van de variabele **FirstWeekDay** gebruikt.<br><br>De mogelijke **first_week_day**-waarden zijn 0 voor maandag, 1 voor dinsdag, 2 voor woensdag, 3 voor donderdag, 4 voor vrijdag, 5 voor zaterdag en 6 voor zondag.<br><br>Zie *FirstWeekDay (page 244)* voor meer informatie over de systeemvariabele. |

### Wanneer gebruiken

De `weekstart()`-functie wordt doorgaans gebruikt als onderdeel van een uitdrukking als de gebruiker wil dat in de berekening het deel van de week wordt gebruikt dat al is geweest. Bijvoorbeeld: het zou kunnen worden gebruikt om het totale salaris te berekenen dat werknemers op een week tot dusver hebben verdiend.

De volgende voorbeelden gaan uit van het volgende:

```
SET FirstWeekDay=0;
```

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `weekstart('01/12/2013')` | Retourneert 01/07/2013. |
| `weekstart('01/12/2013', -1 )` | Retourneert 11/31/2012. |
| `weekstart('01/12/2013', 0, 1)` | Retourneert 01/08/2013. |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

**Voorbeelden:**

Als u ISO-instellingen wilt voor weken en weeknummers, moet u het volgende opnemen in het script:

```
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
```

Als u US-instellingen wilt, moet u het volgende opnemen in het script:

```
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
```

De bovenstaande voorbeelden resulteren in het volgende op basis van de functie `weekstart()`:

Voorbeeld van de functie Weekstart

| Datum | ISO eerste dag van de week | US eerste dag van de week |
|---|---|---|
| Za 2020 Dec 26 | 2020-12-21 | 12/20/2020 |
| Zo 2020 Dec 27 | 2020-12-21 | 12/27/2020 |
| Ma 2020 Dec 28 | 2020-12-28 | 12/27/2020 |
| Di 2020 Dec 29 | 2020-12-28 | 12/27/2020 |
| Wo 2020 Dec 30 | 2020-12-28 | 12/27/2020 |
| Do 2020 Dec 31 | 2020-12-28 | 12/27/2020 |
| Vr 2021 Jan 1 | 2020-12-28 | 12/27/2020 |
| Za 2021 Jan 2 | 2020-12-28 | 12/27/2020 |
| Zo 2021 Jan 3 | 2020-12-28 | 1/3/2021 |

| Datum | ISO eerste dag van de week | US eerste dag van de week |
|---|---|---|
| Ma 2021 Jan 4 | 2021-01-04 | 1/3/2021 |
| Di 2021 Jan 5 | 2021-01-04 | 1/3/2021 |

*De week begint op maandag in de ISO-kolom en op zondag in de US-kolom.*

## Voorbeeld 1 – geen aanvullende argumenten

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties voor 2022 die wordt geladen in de tabel `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.
- Het maken van een veld, `start_of_week`, dat een tijdstempel voor het begin van de week retourneert waarin de transactie plaatsvond.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `start_of_week`
- `start_of_week_timestamp`

Resultatentabel

| date | start_of_week | start_of_week_timestamp |
|---|---|---|
| 1/7/2022 | 01/02/2022 | 1/2/2022 12:00:00 AM |
| 1/19/2022 | 01/16/2022 | 1/16/2022 12:00:00 AM |
| 2/5/2022 | 01/30/2022 | 1/30/2022 12:00:00 AM |
| 2/28/2022 | 02/27/2022 | 2/27/2022 12:00:00 AM |
| 3/16/2022 | 03/13/2022 | 3/13/2022 12:00:00 AM |
| 4/1/2022 | 03/27/2022 | 3/27/2022 12:00:00 AM |
| 5/7/2022 | 05/01/2022 | 5/1/2022 12:00:00 AM |
| 5/16/2022 | 05/15/2022 | 5/15/2022 12:00:00 AM |
| 6/15/2022 | 06/12/2022 | 6/12/2022 12:00:00 AM |
| 6/26/2022 | 06/26/2022 | 6/26/2022 12:00:00 AM |
| 7/9/2022 | 07/03/2022 | 7/3/2022 12:00:00 AM |
| 7/22/2022 | 07/17/2022 | 7/17/2022 12:00:00 AM |
| 7/23/2022 | 07/17/2022 | 7/17/2022 12:00:00 AM |
| 7/27/2022 | 07/24/2022 | 7/24/2022 12:00:00 AM |
| 8/2/2022 | 07/31/2022 | 7/31/2022 12:00:00 AM |
| 8/8/2022 | 08/07/2022 | 8/7/2022 12:00:00 AM |
| 8/19/2022 | 08/14/2022 | 8/14/2022 12:00:00 AM |

| date | start_of_week | start_of_week_timestamp |
|---|---|---|
| 9/26/2022 | 09/25/2022 | 9/25/2022 12:00:00 AM |
| 10/14/2022 | 10/09/2022 | 10/9/2022 12:00:00 AM |
| 10/29/2022 | 10/23/2022 | 10/23/2022 12:00:00 AM |

Het veld '`start_of_week`' wordt gemaakt in de voorafgaande load-instructie met behulp van de functie `weekstart()` en geeft het datumveld door als het argument van de functie.

De `weekstart()`-functie identificeert aanvankelijk in welke week de datumwaarde valt en retourneert een tijdstempel voor de eerste milliseconde van die week.

*Diagram van `weekstart()`-functie, voorbeeld zonder aanvullende argumenten*

Transactie 8191 vond plaats op 5 februari. De systeemvariabele `FirstWeekDay` stelt de eerste dag van de week in op een zondag. De `weekstart()`-functie identificeert dat de eerste zaterdag vóór 5 februari, en daarom het begin van de week, op 30 februari viel. Daarom retourneert de `start_of_week`-waarde voor die transactie de eerste milliseconde van die dag, die op 30 januari om 12:00:00 AM was.

## Voorbeeld 2 – period_no

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het maken van een veld, `previous_week_start`, dat de tijdstempel voor het begin van het kwartaal retourneert voordat de transactie plaatsvond.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `previous_week_start`
- `previous_week_start_timestamp`

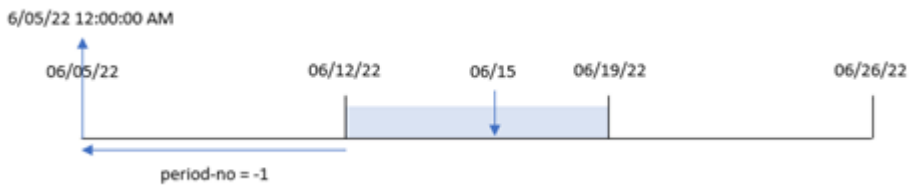
Resultatentabel

| date | previous_week_start | previous_week_start_timestamp |
|---|---|---|
| 1/7/2022 | 12/26/2021 | 12/26/2021 12:00:00 AM |
| 1/19/2022 | 01/09/2022 | 1/9/2022 12:00:00 AM |
| 2/5/2022 | 01/23/2022 | 1/23/2022 12:00:00 AM |
| 2/28/2022 | 02/20/2022 | 2/20/2022 12:00:00 AM |
| 3/16/2022 | 03/06/2022 | 3/6/2022 12:00:00 AM |
| 4/1/2022 | 03/20/2022 | 3/20/2022 12:00:00 AM |
| 5/7/2022 | 04/24/2022 | 4/24/2022 12:00:00 AM |
| 5/16/2022 | 05/08/2022 | 5/8/2022 12:00:00 AM |

| date | previous_week_start | previous_week_start_timestamp |
|---|---|---|
| 6/15/2022 | 06/05/2022 | 6/5/2022 12:00:00 AM |
| 6/26/2022 | 06/19/2022 | 6/19/2022 12:00:00 AM |
| 7/9/2022 | 06/26/2022 | 6/26/2022 12:00:00 AM |
| 7/22/2022 | 07/10/2022 | 7/10/2022 12:00:00 AM |
| 7/23/2022 | 07/10/2022 | 7/10/2022 12:00:00 AM |
| 7/27/2022 | 07/17/2022 | 7/17/2022 12:00:00 AM |
| 8/2/2022 | 07/24/2022 | 7/24/2022 12:00:00 AM |
| 8/8/2022 | 07/31/2022 | 7/31/2022 12:00:00 AM |
| 8/19/2022 | 08/07/2022 | 8/7/2022 12:00:00 AM |
| 9/26/2022 | 09/18/2022 | 9/18/2022 12:00:00 AM |
| 10/14/2022 | 10/02/2022 | 10/2/2022 12:00:00 AM |
| 10/29/2022 | 10/16/2022 | 10/16/2022 12:00:00 AM |

In dit geval, omdat een `period_no` van -1 werd gebruikt als het offset-argument in de `weekstart()`-functie, identificeert de functie eerst de week waarin de transacties plaatsvinden. Het kijkt dan een week eerder en identificeert de eerste milliseconde van die week.

*Diagram van `weekstart()`-functie, voorbeeld period_no*




Transactie 8196 vond plaats op 15 juni. De `weekstart()`-functie identificeert dat de week op 12 juni begint. Daarom begon de vorige week op 5 juni om 12:00:00 AM; dit is de waarde die voor het `previous_week_start`-veld wordt geretourneerd.

## Voorbeeld 3 – first_week_day

### Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld. Maar in dit voorbeeld moeten we dinsdag instellen als de eerste dag van de werkweek.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `start_of_week`
- `start_of_week_timestamp`

Resultatentabel

| date | start_of_week | start_of_week_timestamp |
|---|---|---|
| 1/7/2022 | 01/04/2022 | 1/4/2022 12:00:00 AM |
| 1/19/2022 | 01/18/2022 | 1/18/2022 12:00:00 AM |

| date | start_of_week | start_of_week_timestamp |
|---|---|---|
| 2/5/2022 | 02/01/2022 | 2/1/2022 12:00:00 AM |
| 2/28/2022 | 02/22/2022 | 2/22/2022 12:00:00 AM |
| 3/16/2022 | 03/15/2022 | 3/15/2022 12:00:00 AM |
| 4/1/2022 | 03/29/2022 | 3/29/2022 12:00:00 AM |
| 5/7/2022 | 05/03/2022 | 5/3/2022 12:00:00 AM |
| 5/16/2022 | 05/10/2022 | 5/10/2022 12:00:00 AM |
| 6/15/2022 | 06/14/2022 | 6/14/2022 12:00:00 AM |
| 6/26/2022 | 06/21/2022 | 6/21/2022 12:00:00 AM |
| 7/9/2022 | 07/05/2022 | 7/5/2022 12:00:00 AM |
| 7/22/2022 | 07/19/2022 | 7/19/2022 12:00:00 AM |
| 7/23/2022 | 07/19/2022 | 7/19/2022 12:00:00 AM |
| 7/27/2022 | 07/26/2022 | 7/26/2022 12:00:00 AM |
| 8/2/2022 | 08/02/2022 | 8/2/2022 12:00:00 AM |
| 8/8/2022 | 08/02/2022 | 8/2/2022 12:00:00 AM |
| 8/19/2022 | 08/16/2022 | 8/16/2022 12:00:00 AM |
| 9/26/2022 | 09/20/2022 | 9/20/2022 12:00:00 AM |
| 10/14/2022 | 10/11/2022 | 10/11/2022 12:00:00 AM |
| 10/29/2022 | 10/25/2022 | 10/25/2022 12:00:00 AM |

In dit geval geldt dat omdat het `first_week_date`-argument 1 wordt gebruikt in de `weekstart()`-functie, dinsdag als eerste dag van de week wordt ingesteld.

*Diagram van de `weekstart()`-functie, voorbeeld met first_week_day*

Transactie 8191 vond plaats op 5 februari. De `weekstart()`-functie identificeert dat de eerste dinsdag vóór deze datum, en daarom het begin van de week en geretourneerde waarde, op 1 februari om 12:00:00 AM viel.

## Voorbeeld 4 – diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die een tijdstempel retourneert voor het begin van de week waarin de transacties plaatsvonden, wordt gemaakt als meting in een diagramobject van de toepassing.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `date`.

Voeg de volgende metingen toe om het begin van de week waarin een transactie plaatsvindt te berekenen:

- `=weekstart(date)`
- `=timestamp(weekstart(date))`

Resultatentabel

| date | start_of_week | start_of_week_timestamp |
|---|---|---|
| 1/7/2022 | 01/02/2022 | 1/2/2022 12:00:00 AM |
| 1/19/2022 | 01/16/2022 | 1/16/2022 12:00:00 AM |
| 2/5/2022 | 01/30/2022 | 1/30/2022 12:00:00 AM |
| 2/28/2022 | 02/27/2022 | 2/27/2022 12:00:00 AM |
| 3/16/2022 | 03/13/2022 | 3/13/2022 12:00:00 AM |
| 4/1/2022 | 03/27/2022 | 3/27/2022 12:00:00 AM |
| 5/7/2022 | 05/01/2022 | 5/1/2022 12:00:00 AM |
| 5/16/2022 | 05/15/2022 | 5/15/2022 12:00:00 AM |
| 6/15/2022 | 06/12/2022 | 6/12/2022 12:00:00 AM |
| 6/26/2022 | 06/26/2022 | 6/26/2022 12:00:00 AM |
| 7/9/2022 | 07/03/2022 | 7/3/2022 12:00:00 AM |
| 7/22/2022 | 07/17/2022 | 7/17/2022 12:00:00 AM |
| 7/23/2022 | 07/17/2022 | 7/17/2022 12:00:00 AM |
| 7/27/2022 | 07/24/2022 | 7/24/2022 12:00:00 AM |
| 8/2/2022 | 07/31/2022 | 7/31/2022 12:00:00 AM |
| 8/8/2022 | 08/07/2022 | 8/7/2022 12:00:00 AM |
| 8/19/2022 | 08/14/2022 | 8/14/2022 12:00:00 AM |
| 9/26/2022 | 09/25/2022 | 9/25/2022 12:00:00 AM |
| 10/14/2022 | 10/09/2022 | 10/9/2022 12:00:00 AM |
| 10/29/2022 | 10/23/2022 | 10/23/2022 12:00:00 AM |

De meting `start_of_week` wordt gemaakt in het diagramobject met behulp van de functie `weekstart()` en geeft het veld `date` door als het argument van de functie.

De `weekstart()`-functie identificeert aanvankelijk in welke week de datumwaarde valt en retourneert een tijdstempel voor de eerste milliseconde van die week.

*Diagram van `weekstart()`-functie, diagramobjectvoorbeeld*

Transactie 8191 vond plaats op 5 februari. De systeemvariabele `FirstWeekDay` stelt de eerste dag van de week in op een zondag. De `weekstart()`-functie identificeert dat de eerste zondag vóór 5 februari, en daarom het begin van de week, op 30 januari viel. De `start_of_week`-waarde voor die transactie retourneert daarom de eerste milliseconde van die dag, 30 januari om 12:00:00 AM.

## Voorbeeld 5 – Scenario

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset die wordt geladen in een tabel met de naam `Payroll`.
- Gegevens omvatten werknemers-id's, werknemersnamen en het gemiddelde dagsalaris dat elke werknemer verdient.

Werknemers beginnen hun werk op maandag en werken zes dagen per week. De `FirstWeekDay`-systeemvariabele mag niet worden aangepast.

De eindgebruiker wil een diagramobject dat de salarissen weergeeft op werknemers-id en werknemersnaam, die in de week tot dusver zijn verdiend.

### Load-script

```
Payroll:
Load
*
Inline
[
employee_id,employee_name,day_rate
182,Mark, $150
183,Deryck, $125
184,Dexter, $125
185,Sydney,$270
186,Agatha,$128
];
```

**Resultaten**

**Doe het volgende:**

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:
   - `employee_id`
   - `employee_name`
2. Maak vervolgens een meting om de salarissen te berekenen die in de week tot dusver zijn verdiend: `=if(today(1)-weekstart(today(1),0,0)<7,(today(1)-weekstart(today(1),0,0))*day_rate,day_rate*6)`
3. Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

| 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 | Mark | $600.00 |
| 183 | Deryck | $500.00 |
| 184 | Dexter | $500.00 |
| 185 | Sydney | $1080.00 |
| 186 | Agatha | $512.00 |

De `weekstart()`-functie die door de datum van vandaag als het eerste argument en 0 als het derde argument te gebruiken, maandag instelt als de eerste dag van de week en de begindatum van de huidige week retourneert. Door dat resultaat af te trekken van de huidige datum, retourneert de uitdrukking vervolgens het aantal dagen dat in deze week tot nu toe is verstreken.

De voorwaarde bepaalt vervolgens of er meer dan zes dagen deze week zijn geweest. Als dat zo is dan wordt het `day_rate` van de werknemer vermenigvuldigd met 6 dagen. Anders wordt het `day_rate` vermenigvuldigd met het aantal dagen tot nu toe deze week.

## weekyear

Deze functie retourneert het jaar waarbij het weeknummer hoort, op basis van de omgevingsvariabelen. Het weeknummer varieert van 1 tot ongeveer 52.

**Syntaxis:**

```
weekyear(timestamp [, first_week_day [, broken_weeks [, reference_day]]])
```

**Retourgegevenstypen:** geheel getal

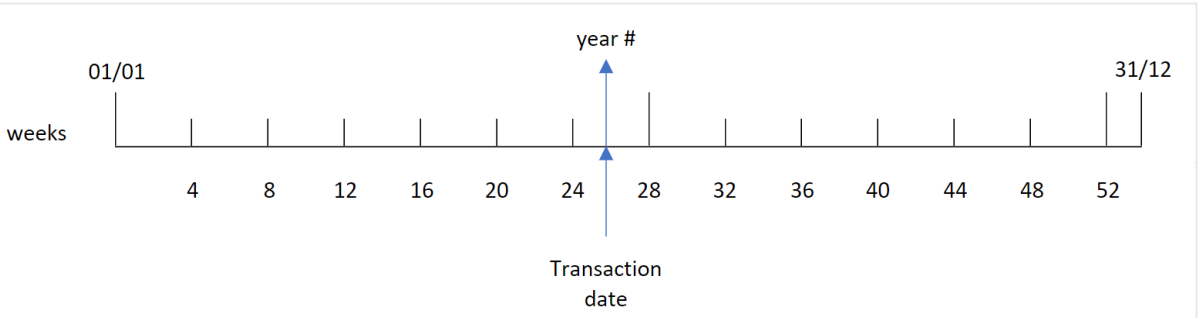
Argumenten

| Argument | Beschrijving |
|---|---|
| **timestamp** | De te evalueren datum of tijdstempel. |
| **first_week_day** | Geeft de dag op waarop de week begint. Bij weglating wordt de waarde van de variabele **FirstWeekDay** gebruikt.<br><br>De mogelijke **first_week_day**-waarden zijn 0 voor maandag, 1 voor dinsdag, 2 voor woensdag, 3 voor donderdag, 4 voor vrijdag, 5 voor zaterdag en 6 voor zondag.<br><br>Zie *FirstWeekDay (page 244)* voor meer informatie over de systeemvariabele. |
| **broken_weeks** | Als u **broken_weeks** (gebroken weken) niet specificeert, dan wordt de waarde van variabele **BrokenWeeks** gebruikt om te definiëren of weken al dan niet gebroken zijn. |
| **reference_day** | Als u **reference_day** (referentiedag) niet specificeert, dan wordt de waarde van variabele **ReferenceDay** gebruikt om te definiëren welke dag in januari wordt ingesteld als referentiedag om week 1 te definiëren. Standaard wordt in Qlik Sense-functies 4 gebruikt als de referentiedag. Dit betekent dat week 1 4 januari moet bevatten of, anders gezegd, dat week 1 altijd ten minste 4 dagen in januari moet hebben. |

De `weekyear()`-functie bepaalt welke week van een jaar de datum in valt. Hij retourneert vervolgens het jaar dat bij dat weeknummer hoort.

Als `BrokenWeeks` is ingesteld op `0` (onwaar), `weekyear()` retourneert hetzelfde als `year()`.

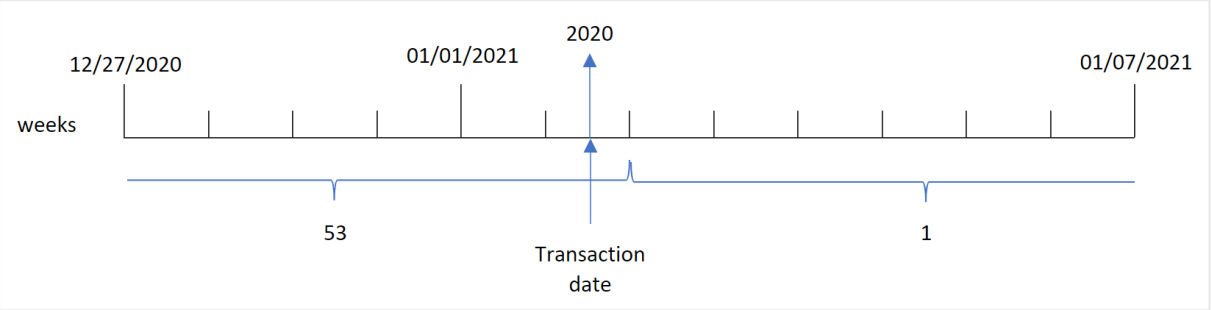
*Diagram van het bereik van de `weekyear()`-functie*




Maar als de `BrokenWeeks`-systeemvariabele is ingesteld op gebruik van niet-opgesplitste weken, mag week 1 alleen een bepaald aantal dagen in januari omvatten op basis van de waarde die is gespecificeerd in de `ReferenceDay`-systeemvariabele.

Als bijvoorbeeld een `ReferenceDay`-waarde van `4` wordt gebruikt, moet week 1 ten minste vier dagen in januari omvatten. Het is mogelijk dat week 1 datums in december van het voorgaande jaar bevat of dat het laatste weeknummer van een jaar datums in januari van het volgende jaar bevat. In situaties zoals deze, retourneert de `weekyear()`-functie een andere waarde dan de `year()`-functie.

*Diagram van het bereik van de `weekyear()`-functie bij gebruik van niet-opgesplitste weken*




### Wanneer gebruiken

De `weekyear()`-functie is handig als u aggregaties per jaar wilt vergelijken. Bijvoorbeeld als u de totale verkoop van producten per jaar wilt zien. De `weekyear()`-functie wordt verkozen boven de `year()`-functie wanneer de gebruiker consistent wil blijven met de `BrokenWeeks`-systeemvariabele in de app.

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `weekyear('12/30/1996',0,0,4)` | Retourneert 1997, omdat week 1 van 1997 begint op 12/30/1996 |
| `weekyear('01/02/1997',0,0,4)` | Retourneert 1997 |
| `weekyear('12/28/1997',0,0,4)` | Retourneert 1997 |
| `weekyear('12/30/1997',0,0,4)` | Retourneert 1998, omdat week 1 van 1998 begint op 12/29/1997 |
| `weekyear('01/02/1999',0,0,4)` | Retourneert 1998, omdat week 53 van 1998 eindigt op 01/03/1999 |

Gerelateerde onderwerpen

| Onderwerp | Interactie |
|---|---|
| *week (page 1181)* | Retourneert een geheel getal dat het weeknummer representeert conform ISO 8601 |
| *year (page 1255)* | Retourneert een geheel getal dat het jaar representeert als de uitdrukking wordt geïnterpreteerd als een datum volgens de standaardinterpretatie van getallen. |

## Voorbeeld 1 - Opgesplitste weken

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties voor de laatste week van 2020 en de eerste week van 2021 wordt geladen in de tabel `Transactions`.
- De variabele `BrokenWeeks` is ingesteld op 1.
- Een voorgaande lading die het volgende bevat:
  - De `weekyear()`-functie die is ingesteld als het veld, `week_year`, dat het jaar retourneert waarin de transacties hebben plaatsgevonden.
  - De `week()`-functie, ingesteld als het veld `week`, dat het weeknummer toont van elke transactiedatum.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `date`
- `week`
- `week_year`

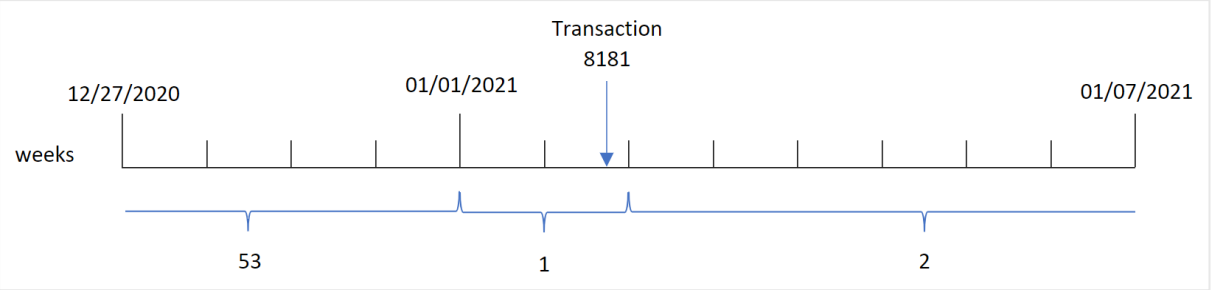
Resultatentabel

| id | date | week | week_year |
|---|---|---|---|
| 8176 | 12/28/2020 | 53 | 2020 |
| 8177 | 12/29/2020 | 53 | 2020 |
| 8178 | 12/30/2020 | 53 | 2020 |
| 8179 | 12/31/2020 | 53 | 2020 |
| 8180 | 01/01/2021 | 1 | 2021 |
| 8181 | 01/02/2021 | 1 | 2021 |
| 8182 | 01/03/2021 | 2 | 2021 |
| 8183 | 01/04/2021 | 2 | 2021 |
| 8184 | 01/05/2021 | 2 | 2021 |
| 8185 | 01/06/2021 | 2 | 2021 |
| 8186 | 01/07/2021 | 2 | 2021 |

Het veld '`week_year`' wordt gemaakt in de voorafgaande load-instructie met behulp van de functie `weekyear()` en geeft het datumveld door als het argument van de functie.

De systeemvariabele `BrokenWeeks` is ingesteld op `1` wat inhoudt dat de app opgesplitste weken gebruikt. Week 1 begint op 1 januari.

*Diagram van het bereik van de `weekyear()`-functie bij gebruik van opgesplitste weken*




Transactie 8181 vindt plaats op 2 januari. Deze datum valt in week 1. Daarom wordt de waarde 2021 geretourneerd voor het veld `week_year`.

## Voorbeeld 2 - Niet-opgesplitste weken

### Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met een set transacties voor de laatste week van 2020 en de eerste week van 2021 wordt geladen in de tabel `Transactions`.
- De `BrokenWeeks`-variabele is ingesteld op `0`.
- Een voorgaande lading die het volgende bevat:
  - De `weekyear()`-functie die is ingesteld als het veld, `week_year`, dat het jaar retourneert waarin de transacties hebben plaatsgevonden.
  - De `week()`-functie, ingesteld als het veld `week`, dat het weeknummer toont van elke transactiedatum.

Maar in dit voorbeeld is het bedrijfsbeleid om niet-opgesplitste weken te gebruiken.

#### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `date`
- `week`
- `week_year`

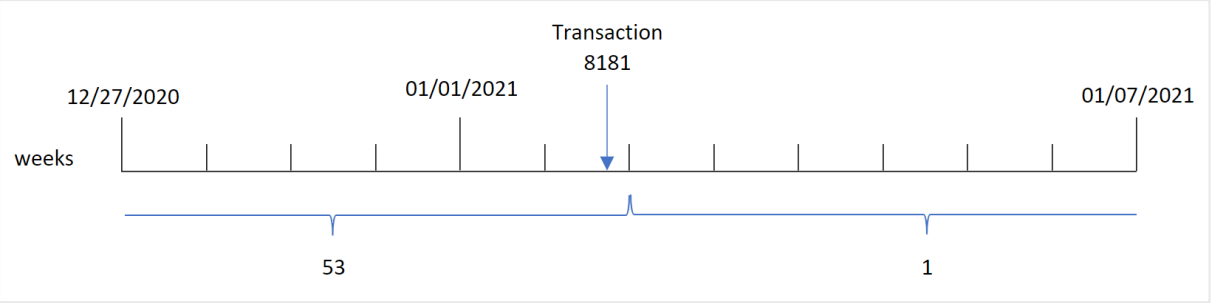
Resultatentabel

| id | date | week | week_year |
|---|---|---|---|
| 8176 | 12/28/2020 | 53 | 2020 |
| 8177 | 12/29/2020 | 53 | 2020 |
| 8178 | 12/30/2020 | 53 | 2020 |
| 8179 | 12/31/2020 | 53 | 2020 |
| 8180 | 01/01/2021 | 53 | 2020 |
| 8181 | 01/02/2021 | 53 | 2020 |
| 8182 | 01/03/2021 | 1 | 2021 |
| 8183 | 01/04/2021 | 1 | 2021 |
| 8184 | 01/05/2021 | 1 | 2021 |
| 8185 | 01/06/2021 | 1 | 2021 |
| 8186 | 01/07/2021 | 1 | 2021 |

De systeemvariabele `BrokenWeeks` is ingesteld op `0` wat inhoudt dat de applicatie niet-opgesplitste weken gebruikt. Daarom hoeft week 1 niet te beginnen op 1 januari.

Week 53 van 2020 gaat verder tot het eind van 2 januari 2021, waarbij week 1 van 2020 op zondag, 3 januari 2021 valt.

*Diagram van het bereik van de `weekyear()`-functie bij gebruik van niet-opgesplitste weken*




Transactie 8181 vindt plaats op 2 januari. Deze datum valt in week 1. Daarom wordt de waarde 2021 geretourneerd voor het veld `week_year`.

## Voorbeeld 3 - Diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is de gegevensverzameling echter niet gewijzigd en wordt in de applicatie geladen. De berekening die het jaarnummer van de week retourneert maand waarin de transacties plaatsvonden is gemaakt als meting in een diagramobject van de toepassing.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date

Maak de volgende meting om de week waarin een transactie plaatsvindt te berekenen:

- =week(date)

Maak de volgende meting om het jaar waarin een transactie plaatsvindt te berekenen op basis van het weeknummer:

- =weekyear(date)

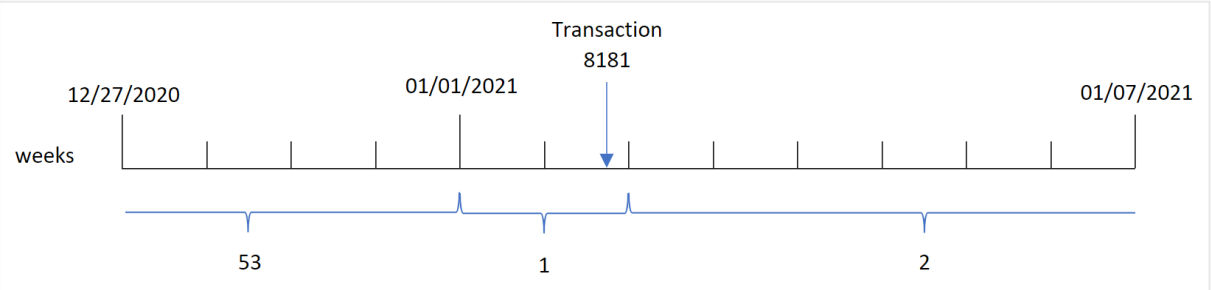
Resultatentabel

| id | date | week | week_year |
|---|---|---|---|
| 8176 | 12/28/2020 | 53 | 2020 |
| 8177 | 12/29/2020 | 53 | 2020 |
| 8178 | 12/30/2020 | 53 | 2020 |
| 8179 | 12/31/2020 | 53 | 2020 |
| 8180 | 01/01/2021 | 1 | 2021 |
| 8181 | 01/02/2021 | 1 | 2021 |
| 8182 | 01/03/2021 | 2 | 2021 |
| 8183 | 01/04/2021 | 2 | 2021 |
| 8184 | 01/05/2021 | 2 | 2021 |
| 8185 | 01/06/2021 | 2 | 2021 |
| 8186 | 01/07/2021 | 2 | 2021 |

Het veld 'week_year' wordt gemaakt in de voorafgaande load-instructie met behulp van de functie weekyear() en geeft het datumveld door als het argument van de functie.

De systeemvariabele BrokenWeeks is ingesteld op 1 wat inhoudt dat de app opgesplitste weken gebruikt. Week 1 begint op 1 januari.

*Diagram van het bereik van de weekyear()-functie bij gebruik van opgesplitste weken*



Transactie 8181 vindt plaats op 2 januari. Deze datum valt in week 1. Daarom wordt de waarde 2021 geretourneerd voor het veld `week_year`.

## Voorbeeld 4 - Scenario

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met een set transacties voor de laatste week van 2020 en de eerste week van 2021 wordt geladen in de tabel `Transactions`.
- De `BrokenWeeks`-variabele is ingesteld op `0`. Dit betekent dat de app niet-opgesplitste weken gebruikt.
- De `ReferenceDay`-variabele is ingesteld op `2`. Dit betekent dat het jaar op 2 januari begint en ten minste twee dagen in januari bevat.
- De `FirstWeekDay`-variabele is ingesteld op `1`. Dit betekent dat dinsdag de eerste dag van de week is.

Het bedrijfsbeleid is om opgesplitste weken te gebruiken. De eindgebruiker wil graag een diagram dat de totale verkoop per jaar weergeeft. De app gebruikt niet-opgesplitste weken waarbij week 1 ten minste twee dagen in januari bevat.

### Load-script

```
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
];
```

**Resultaten**

Laad de gegevens en open een werkblad. Maak een nieuwe tabel.

Maak de volgende meting om het jaar waarin een transactie plaatsvindt te berekenen op basis van het weeknummer:

- `=weekyear(date)`

Maak de volgende meting om de totale verkoop te berekenen:

- `sum(amount)`

Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

| weekyear(datum) | =sum(amount) |
|---|---|
| 2020 | 19.42 |
| 2021 | 373.37 |

## year

Deze functie retourneert een geheel getal dat het jaar representeert als de **expression** wordt geïnterpreteerd als een datum volgens de standaardinterpretatie van getallen.

**Syntaxis:**

```
year(expression)
```

**Retourgegevenstypen:** geheel getal

De `year()`-functie is beschikbaar zowel als een script- als een diagramfunctie. De functie retourneert het jaar voor een bepaalde datum. Dit wordt vaak gebruikt om een jaarveld te maken als een dimensie in een masterkalender.

### Wanneer gebruiken

De `year()`-functie is handig als u aggregaties per jaar wilt vergelijken. De functie kan bijvoorbeeld worden gebruikt om de totale verkoop van producten per jaar te bekijken.

Deze dimensies kunnen in het load-script worden gemaakt door de functie te gebruiken om een veld in een masterkalendertabel te maken. Hij zou ook rechtstreeks in een diagram kunnen worden gebruikt als een berekende dimensie.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `year( '2012-10-12' )` | retourneert 2012 |
| `year( '35648' )` | retourneert 1997, omdat 35648 = 1997-08-06 |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 - gegevensverzameling DateFormat (script)

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met datums die wordt geladen in de tabel `Master Calendar`.
- De standaard `DateFormat`-systeemvariabele mm/dd/jjjj wordt gebruikt.
- Een voorafgaande lading die wordt gebruikt om een extra veld te maken met de naam `year` met behulp van de `year()`-functie.

### Load-script

```
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
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `year`

Resultatentabel

| date | year |
|---|---|
| 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 |

## Voorbeeld 2 – ANSI-datums

#### Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met datums die wordt geladen in de tabel `Master Calendar`.
- De standaard `DateFormat`-systeemvariabele (mm/dd/jjjj) wordt gebruikt. De datums die zijn opgenomen in de gegevensverzameling hebben echter een standaard ANSI-datumnotatie.
- Een voorafgaande lading die wordt gebruikt om een extra veld te maken met de naam `year` met behulp van de `year()`-functie.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `year`

Resultatentabel

| date | jaar |
|---|---|
| 2020-12-28 | 2020 |
| 2020-12-29 | 2020 |
| 2020-12-30 | 2020 |
| 2020-12-31 | 2020 |
| 2021-01-01 | 2021 |

| date | jaar |
|---|---|
| 2021-01-02 | 2021 |
| 2021-01-03 | 2021 |
| 2021-01-04 | 2021 |
| 2021-01-05 | 2021 |
| 2021-01-06 | 2021 |
| 2021-01-07 | 2021 |

## Voorbeeld 3 – Datums zonder notatie

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met datums in numerieke indeling, die wordt geladen in de tabel `Master Calendar`.
- De standaard `DateFormat`-systeemvariabele (mm/dd/jjjj) wordt gebruikt.
- Een voorafgaande lading die wordt gebruikt om een extra veld te maken met de naam `year` met behulp van de `year()`-functie.

De oorspronkelijke, niet-opgemaakte gegevens worden geladen, met de naam `unformatted_date`. Om helderheid te verschaffen, wordt het extra veld `long_date` gebruikt om de numerieke gegevens om te zetten in een opgemaakt datumveld met behulp van de `date()`-functie.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `unformatted_date`
- `long_date`
- `year`

Resultatentabel

| unformatted_date | long_date | jaar |
|---|---|---|
| 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 |

## Voorbeeld 4 – diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

In dit voorbeeld wordt een gegevensset met bestellingen geladen in de tabel Verkoop. De tabel bevat drie velden.

- id
- sales_date
- amount

Garanties bij verkochte producten in de afgelopen twee jaar vanaf de verkoopdatum. Het is de taak om een meting in een diagram te maken om het jaar vast te stellen waarin iedere garantie zal verlopen.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `sales_date`.

Maak de volgende meting:

```
=year(sales_date+365*2)
```

Resultatentabel

| 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 |

| sales_date | =year(sales_date+365*2) |
|---|---|
| 01/03/2021 | 2023 |
| 01/04/2021 | 2023 |
| 01/05/2021 | 2023 |
| 01/06/2021 | 2023 |
| 01/07/2021 | 2023 |

De resultaten van deze meting zijn in de bovenstaande tabel te zien. Om twee jaar bij de datum op te tellen, vermenigvuldigt u 365 met 2 en voegt u het resultaat toe aan de verkoopdatum. De verkoop die in 2020 plaatsvond, heeft een vervaljaar van 2022.

## yearend

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de laatste milliseconde van de laatste dag van het jaar dat **date** bevat. De standaarduitvoernotatie is de **DateFormat** die is ingesteld in het script.

**Syntaxis:**

```
YearEnd( date[, period_no[, first_month_of_year = 1]])
```

Met andere woorden, de functie `yearend()` bepaalt in welk jaar de datum valt. Het retourneert vervolgens een tijdstempel, in datumnotatie, voor de laatste milliseconde van dat jaar. De eerste maand van het jaar is standaard januari. U kunt echter ook wijzigen welke maand als eerste wordt ingesteld met behulp van het argument `first_month_of_year` in de functie `yearend()` te gebruiken.

> *De `yearend()`-functie houdt geen rekening met de systeemvariabele `FirstMonthOfYear`. Het jaar begint op 1 januari, tenzij het argument `first_month_of_year` wordt gebruikt om het te wijzigen.*

*Diagram van `yearend()`-functie.*

### Wanneer gebruiken

De functie `yearend()` wordt gebruikt als onderdeel van een uitdrukking als u wilt dat in de berekening een deel van het jaar wordt gebruikt dat nog niet heeft plaatsgevonden. Bijvoorbeeld als u de totale rente wilt berekenen die nog niet is berekend gedurende het jaar.

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **date** | De te evalueren datum of tijdstempel. |
| **period_no** | **period_no** is een geheel getal, waarbij de waarde 0 het jaar aangeeft dat **date** bevat. Negatieve waarden in **period_no** geven voorgaande jaren aan en positieve waarden geven volgende jaren aan. |
| **first_month_of_year** | Als u wilt werken met (boek)jaren die niet in januari beginnen, geeft u een waarde op tussen 2 en 12 in **first_month_of_year**. |

U kunt de volgende waarden gebruiken om de eerste maand van het jaar in te stellen in het argument `first_month_of_year`:

waarden first_month_of_year

| Maand | Waarde |
|---|---|
| Februari | 2 |
| Maart | 3 |
| April | 4 |
| Mei | 5 |
| Juni | 6 |
| Juli | 7 |
| Augustus | 8 |
| September | 9 |
| Oktober | 10 |
| November | 11 |
| December | 12 |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `yearend('10/19/2001')` | Retourneert 12/31/2001 23:59:59. |
| `yearend('10/19/2001', -1)` | Retourneert 12/31/2000 23:59:59. |
| `yearend('10/19/2001', 0, 4)` | Retourneert 03/31/2002 23:59:59. |

## Voorbeeld 1 – geen aanvullende argumenten

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties tussen 2020 en 2022 wordt geladen in een tabel met de naam '`Transactions`'.
- Het datumveld is opgegeven in de systeemvariabele `DateFormat`, indeling (`(MM/DD/YYYY)`).
- Een voorgaande load-instructie die het volgende bevat:
  - Functie `yearend()` die is ingesteld als het veld `year_end`.
  - Functie `Timestamp()` die is ingesteld als het veld `year_end_timestamp`.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `date`
- `year_end`
- `year_end_timestamp`

Resultatentabel

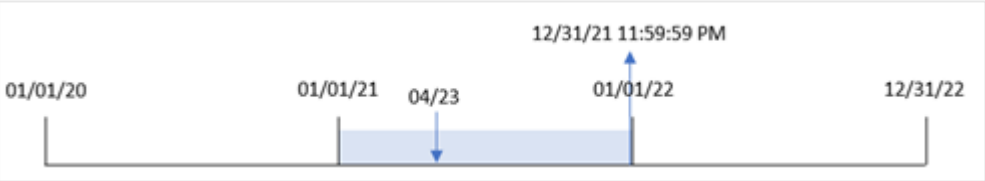
| id | date | year_end | year_end_timestamp |
|---|---|---|---|
| 8188 | 01/13/2020 | 12/31/2020 | 12/31/2020 11:59:59 PM |
| 8189 | 02/26/2020 | 12/31/2020 | 12/31/2020 11:59:59 PM |
| 8190 | 03/27/2020 | 12/31/2020 | 12/31/2020 11:59:59 PM |
| 8191 | 04/16/2020 | 12/31/2020 | 12/31/2020 11:59:59 PM |
| 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 |

Het veld '`year_end`' wordt gemaakt in de voorafgaande load-instructie met behulp van de functie `yearend ()` en geeft het datumveld door als het argument van de functie.

De functie `yearend()` identificeert aanvankelijk in welk jaar de datumwaarde valt en retourneert een tijdstempel voor de laatste milliseconde van dat jaar.

*Diagram van functie `yearend()` met transactie 8199 geselecteerd.*




Transactie 8199 vond plaats op 23 april 2021. De functie `yearend()` retourneert de laatste milliseconde van dat jaar, namelijk 31 december om 11:59:59 P.M.

## Voorbeeld 2 – period_no

Load-script en resultaten

### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is het echter de taak om een veld te maken, '`previous_year_end`', dat de tijdstempel van de einddatum van het jaar voorafgaand aan het jaar waarin de transactie plaatsvond, retourneert.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `date`
- `previous_year_end`
- `previous_ year_end_timestamp`

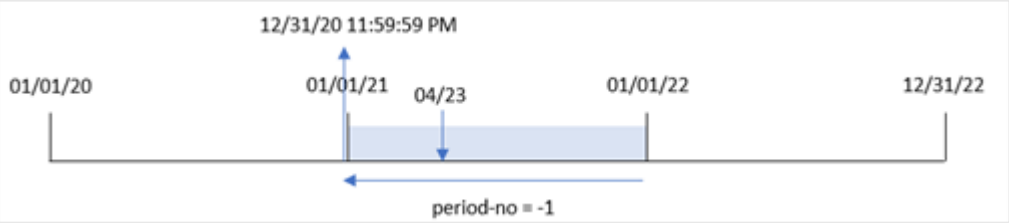
Resultatentabel

| 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 |

Omdat een `period_no` van `-1` werd gebruikt als het offset-argument in de functie `yearend()`, identificeert de functie eerst het jaar waarin de transacties plaatsvinden. Het kijkt dan een jaar eerder en identificeert de laatste milliseconde van dat jaar.

*Diagram van functie `yearend()` met een period_no van -1.*




Transactie 8199 vindt plaats op 23 april 2021. De functie `yearend()` retourneert de laatste milliseconde van het voorgaande jaar, 31 december 2020 om 11:59:59 PM, voor het veld '`previous_year_end`'.

## Voorbeeld 3 – first_month_of_year

### Load-script en resultaten

#### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is het bedrijfsbeleid echter dat het jaar op 1 april begint.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

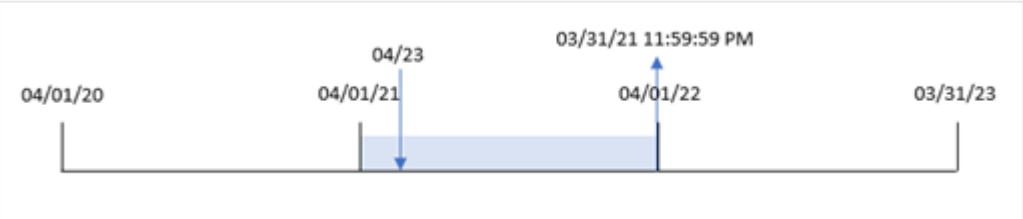
- id
- date
- year_end
- year_end_timestamp

Resultatentabel

| 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 |

Omdat het argument `first_month_of_year` van `4` wordt gebruikt in de functie `yearend()`, stelt het de eerste dag van het jaar in op 1 april en de laatste dag van het jaar op 31 maart.

*Diagram van functie `yearend()` met april als de eerste maand van het jaar.*




Transactie 8199 vindt plaats op 23 april 2021. Omdat de functie `yearend()` het begin van het jaar instelt op 1 april, wordt 31 maart 2022 geretourneerd als de waarde '`year_end`' voor de transactie.

## Voorbeeld 4 – diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is de gegevensverzameling echter niet gewijzigd en wordt in de applicatie geladen. De berekening die een einddatumtijdstempel retourneert van het jaar waarin de transactie plaatsvond is gemaakt als meting in een diagramobject van de toepassing.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `date`

Maak de volgende metingen om te berekenen in welk jaar een transactie plaatsvond:

- `=yearend(date)`
- `=timestamp(yearend(date))`

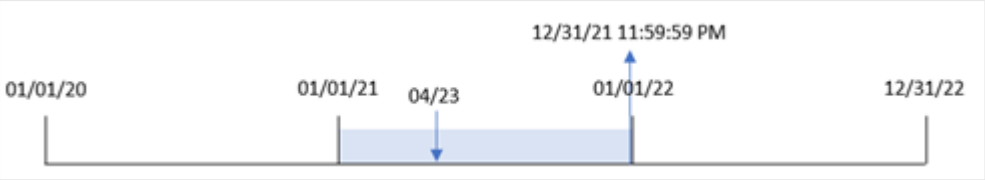
Resultatentabel

| 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 |

De meting '`end_of_year`' wordt gemaakt in het diagramobject met behulp van de functie `yearend()` en geeft het veld door als het argument van de functie.

De functie `yearend()` identificeert aanvankelijk in welk jaar de datumwaarde valt en retourneert een tijdstempel voor de laatste milliseconde van dat jaar.

*Diagram van functie `yearend()` waaruit blijkt dat transactie 8199 in april heeft plaatsgevonden.*




Transactie 8199 vindt plaats op 23 april 2021. De functie `yearend()` retourneert de laatste milliseconde van dat jaar, namelijk 31 december om 11:59:59 P.M.

## Voorbeeld 5 – Scenario

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset wordt geladen in een tabel met de naam '`Employee_Expenses`'. De tabel bevat de volgende velden:
  - werknemers-id's
  - naam werknemer
  - gemiddelde dagelijkse onkostendeclaraties van elke werknemer

De eindgebruiker wil graag een diagramobject dat per werknemer-id en werknemernaam de geschatte nog te maken onkostendeclaraties voor de rest van het jaar weergeeft. Het boekjaar begint in januari.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- employee_id
- employee_name

Maak de volgende meting om de verwachte onkostendeclaraties te berekenen:

```
=(yearend(today(1))-today(1))*avg_daily_claim
```

Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

| employee_id | employee_name | =(yearend(today(1))-today(1))*avg_daily_claim |
|---|---|---|
| 182 | Mark | $3240.00 |
| 183 | Deryck | $2700.00 |
| 184 | Dexter | $2700.00 |
| 185 | Sydney | $5832.00 |
| 186 | Agatha | $3888.00 |

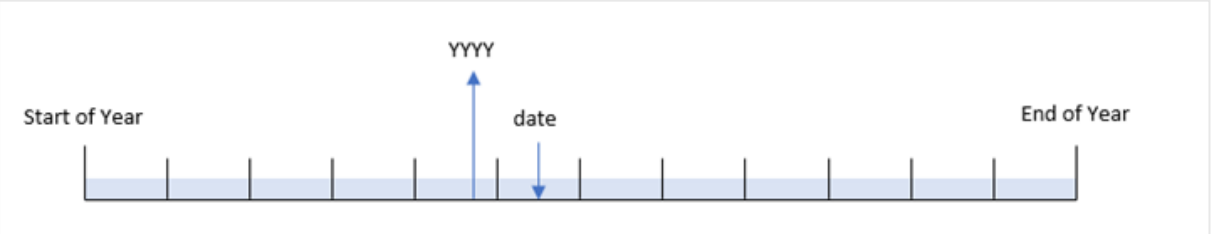
Door de datum van vandaag als enig argument te gebruiken, retourneert de functie `yearend()` de einddatum van het huidige jaar. Door vervolgens de datum van vandaag af te trekken van de einddatum van het jaar, retourneert de uitdrukking het aantal resterende dagen in dit jaar.

Deze waarde wordt vervolgens vermenigvuldigd met de gemiddelde dagelijkse onkostendeclaratie van elke werknemer om de geschatte waarde te berekenen van de declaraties die elke werknemer naar verwachting in het resterende jaar zal indienen.

## yearname

Deze functie retourneert een jaar in vier cijfers als weergavewaarde met een onderliggende numerieke waarde die overeenkomt met een tijdstempel van de eerste milliseconde van de eerste dag van het jaar dat **date** bevat.

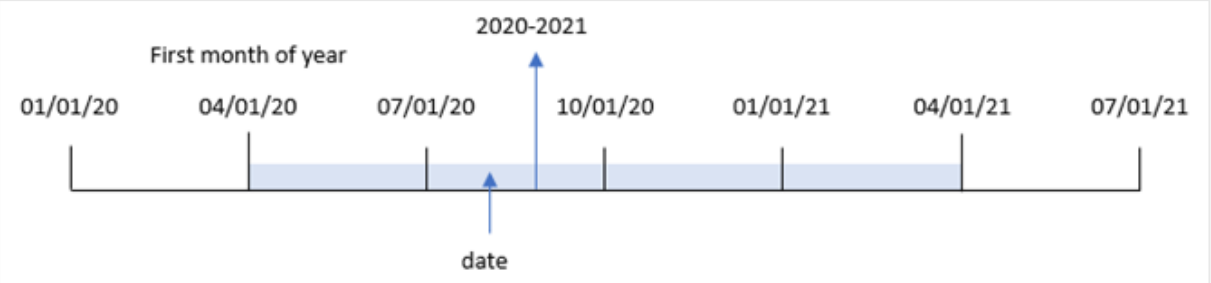
*Schema van het tijdsbereik van de functie `yearname()`.*




De functie `yearname()` is anders dan de functie `year()`, omdat u hiermee de datum die u wilt evalueren kunt verschuiven en de eerste maand van het jaar kunt instellen.

Als de eerste maand van het jaar niet januari is, retourneert de functie de twee viercijferige jaren over de periode van twaalf maanden die de datum bevatten. Als het begin van het jaar bijvoorbeeld april is en de datum die wordt geëvalueerd is 06/30/2020, zou het geretourneerde resultaat `2020-2021`zijn.

*Diagram van functie `yearname()` met april ingesteld als de eerste maand van het jaar.*




**Syntaxis:**

```
YearName(date[, period_no[, first_month_of_year]] )
```

**Retourgegevenstypen:** dubbele waarde

| Argument | Beschrijving |
|---|---|
| **date** | De te evalueren datum of tijdstempel. |
| **period_no** | **period_no** is een geheel getal, waarbij de waarde 0 het jaar aangeeft dat **date** bevat. Negatieve waarden in **period_no** geven voorgaande jaren aan en positieve waarden geven volgende jaren aan. |
| **first_month_of_year** | Als u wilt werken met (boek)jaren die niet in januari beginnen, geeft u een waarde op tussen 2 en 12 in **first_month_of_year**. De weergavewaarde is dan een tekenreeks die twee jaren toont. |

U kunt de volgende waarden gebruiken om de eerste maand van het jaar in te stellen in het argument `first_month_of_year`:

waarden first_month_of_year

| Maand | Waarde |
|---|---|
| Februari | 2 |
| Maart | 3 |
| April | 4 |
| Mei | 5 |
| Juni | 6 |
| Juli | 7 |
| Augustus | 8 |
| September | 9 |
| Oktober | 10 |
| November | 11 |
| December | 12 |

#### Wanneer gebruiken

De functie `yearname()` is handig om aggregaties per jaar te vergelijken. Bijvoorbeeld als u de totale verkoop van producten per jaar wilt zien.

Deze dimensies kunnen in het load-script worden gemaakt door de functie te gebruiken om een veld in een masterkalendertabel te maken. Ze kunnen ook in een diagram worden gemaakt als berekende dimensies

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `yearname('10/19/2001')` | Retourneert '2001.' |
| `yearname('10/19/2001',-1)` | Retourneert '2000.' |
| `yearname('10/19/2001',0,4)` | Retourneert '2001-2002.' |

Gerelateerde onderwerpen

| Onderwerp | Beschrijving |
|---|---|
| *year (page 1255)* | Deze functie retourneert een geheel getal dat het jaar representeert als de uitdrukking wordt geïnterpreteerd als een datum volgens de standaardinterpretatie van getallen. |

## Voorbeeld 1 – geen aanvullende argumenten

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties tussen 2020 en 2022 wordt geladen in een tabel met de naam '`Transactions`'.
- De systeemvariabele `DateFormat` die is ingesteld op '`MM/DD/YYYY`'.
- Een voorafgaande lading die de `yearname()` gebruikt en die is ingesteld als het veld `year_name`.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `year_name`

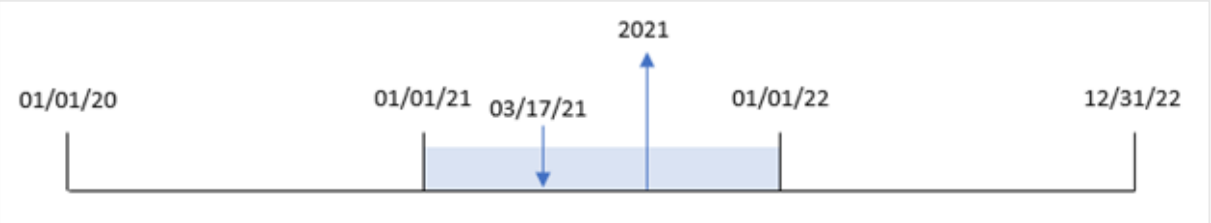
Resultatentabel

| 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 |

Het veld '`year_name`' wordt gemaakt in de voorafgaande load-instructie met behulp van de functie `yearname()` en geeft het datumveld door als het argument van de functie.

De functie `yearname()` identificeert in welk jaar de datumwaarde valt en retourneert dit als een viercijferige jaarwaarde.

*Diagram van de functie `yearname()` die 2021 als jaarwaarde weergeeft.*



## Voorbeeld 2 – period_no

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met een set transacties tussen 2020 en 2022 wordt geladen in een tabel met de naam 'Transacties'.
- De systeemvariabele DateFormat die is ingesteld op 'MM/DD/YYYY'.
- Een voorafgaande lading die de `yearname()` gebruikt en die is ingesteld als het veld `year_name`.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

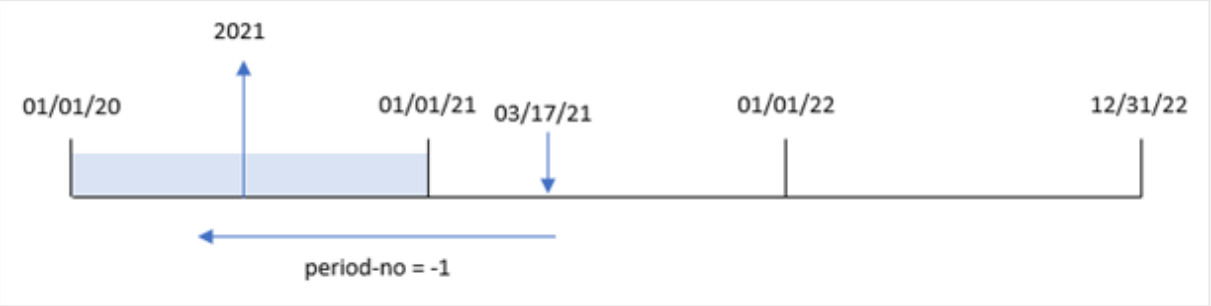
- `date`
- `prior_year_name`

Resultatentabel

| 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 |

Omdat een `period_no` van `-1` wordt gebruikt als het offset-argument in de functie `yearname()`, identificeert de functie eerst het jaar waarin de transacties plaatsvinden. De functie verschuift dan een jaar eerder en retourneert het resulterende jaar.

*Diagram van functie `yearname()` met de `period_no` ingesteld op `-1`.*




## Voorbeeld 3 – first_month_of_year

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset als in het eerste voorbeeld.
- De systeemvariabele `DateFormat` die is ingesteld op '`MM/DD/YYYY`'.
- Een voorafgaande lading die de `yearname()` gebruikt en die is ingesteld als het veld `year_name`.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `date`
- `year_name`

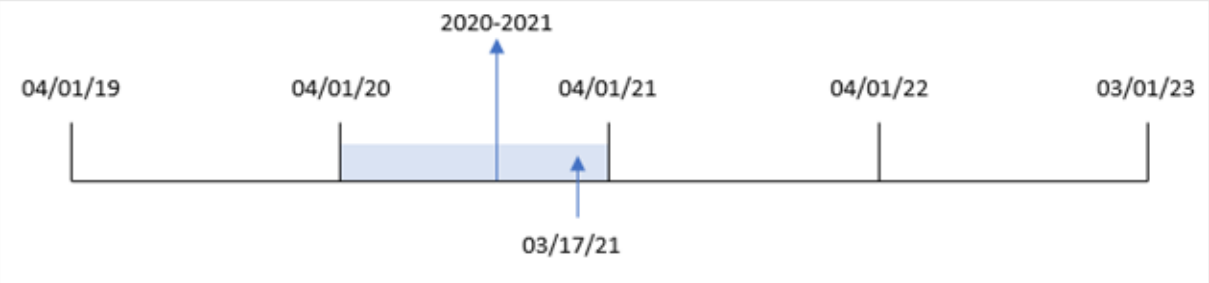
Resultatentabel

| 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 |
| 11/14/2022 | 2022-2023 |
| 12/12/2022 | 2022-2023 |

Omdat het argument [ `first_month_of_year` van `4` wordt gebruikt in de functie `yearname()`, verschuift het begin van het jaar van 1 januari naar 1 april. Daarom kruist elke periode van twaalf maanden twee kalenderjaren en retourneert de functie `yearname()` de twee viercijferige jaar voor geëvalueerde datums.

Transactie 8198 vindt plaats op 17 maart 2021. De functie `yearname()` stelt het begin van het jaar in op 1 april en het einde op 30 maart. Daarom vond transactie 8198 plaats in de jaarperiode van 1 april 2020 en 30 maart 2021. Als resultaat retourneert de functie `yearname()` de waarde 2020-2021.

*Diagram van functie `yearname()` met maart ingesteld als de eerste maand van het jaar.*




## Voorbeeld 4 – diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset als in het eerste voorbeeld.
- De systeemvariabele `DateFormat` die is ingesteld op '`MM/DD/YYYY`'.

Het veld dat het jaar retourneert waarin de transactie heeft plaatsgevonden, wordt echter gemaakt als een meting in een grafiekobject.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

`date`

Om het veld `year_name` te berekenen, maakt u deze meting:

```
=yearname(date)
```

Resultatentabel

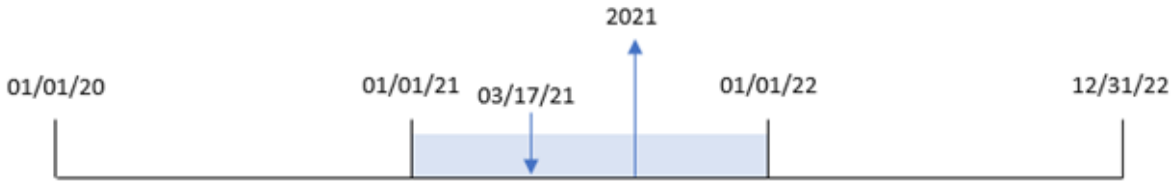
| date | =yearname(date) |
|---|---|
| 01/13/2020 | 2020 |
| 02/26/2020 | 2020 |
| 03/27/2020 | 2020 |
| 04/16/2020 | 2020 |
| 05/21/2020 | 2020 |
| 08/14/2020 | 2020 |
| 10/07/2020 | 2020 |
| 12/05/2020 | 2020 |
| 01/22/2021 | 2021 |
| 02/03/2021 | 2021 |
| 03/17/2021 | 2021 |
| 04/23/2021 | 2021 |
| 05/04/2021 | 2021 |
| 06/30/2021 | 2021 |
| 07/26/2021 | 2021 |
| 12/27/2021 | 2021 |

| date | =yearname(date) |
|---|---|
| 06/06/2022 | 2022 |
| 07/18/2022 | 2022 |
| 11/14/2022 | 2022 |
| 12/12/2022 | 2022 |

De meting '`year_name`' wordt gemaakt in het diagramobject met behulp van de functie `yearname()` en geeft het veld door als het argument van de functie.

De functie `yearname()` identificeert in welk jaar de datumwaarde valt en retourneert dit als een viercijferige jaarwaarde.

*Diagram van de functie `yearname()` met 2021 als jaarwaarde.*




## Voorbeeld 5 – Scenario

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset als in het eerste voorbeeld.
- De systeemvariabele `DateFormat` die is ingesteld op '`MM/DD/YYYY`'.

De eindgebruiker wil graag een diagram die de totale omzet per kwartaal voor de transacties weergeeft. Gebruik de functie `yearname()` als een berekende dimensie om dit diagram te maken wanneer de dimensie `yearname()` niet beschikbaar is in het gegevensmodel.

### Load-script

```
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
];
```

**Resultaten**

Laad de gegevens en open een werkblad. Maak een nieuwe tabel.

Als u aggregaties per jaar wilt vergelijken, maakt u deze berekende dimensie:

```
=yearname(date)
```

Maak deze meting:

```
=sum(amount)
```

Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

| yearname(date) | =sum(amount) |
|---|---|
| 2020 | $463.55 |
| 2021 | $457.69 |
| 2022 | $294.35 |

## yearstart

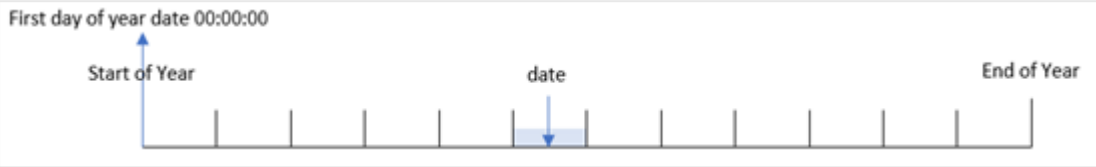
Deze functie retourneert een tijdsstempel die overeenkomt met het begin van de eerste dag van het jaar dat **date** bevat. De standaarduitvoernotatie is de **DateFormat** die is ingesteld in het script.

**Syntaxis:**

```
YearStart(date[, period_no[, first_month_of_year]])
```

Met andere woorden, de functie `yearstart()` bepaalt in welk jaar de datum valt. Het retourneert vervolgens een tijdstempel, in datumnotatie, voor de eerste milliseconde van dat jaar. De eerste maand van het jaar is standaard januari, u kunt echter wijzigen welke maand wordt ingesteld als eerste met behulp van het argument `first_month_of_year` in de functie `yearstart()`.

*Diagram van de functie `yearstart()` dat het tijdsbereik toont dat de functie kan bestrijken.*




### Wanneer gebruiken

De `yearstart()`-functie wordt gebruikt als onderdeel van een uitdrukking als u wilt dat in de berekening een deel van het jaar wordt gebruikt dat tot nu toe is verstreken. Bijvoorbeeld als u de rente wilt berekenen die in een jaar tot nu toe is opgebouwd.

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **date** | De te evalueren datum of tijdstempel. |
| **period_no** | **period_no** is een geheel getal, waarbij de waarde 0 het jaar aangeeft dat **date** bevat. Negatieve waarden in **period_no** geven voorgaande jaren aan en positieve waarden geven volgende jaren aan. |
| **first_month_of_year** | Als u wilt werken met (boek)jaren die niet in januari beginnen, geeft u een waarde op tussen 2 en 12 in **first_month_of_year**. |

De volgende maanden kunnen worden gebruikt in de `first_month_of_year argument`:

waarden first_month_of_year

| Maand | Waarde |
|---|---|
| Februari | 2 |
| Maart | 3 |
| April | 4 |
| Mei | 5 |
| Juni | 6 |
| Juli | 7 |
| Augustus | 8 |

| Maand | Waarde |
|---|---|
| September | 9 |
| Oktober | 10 |
| November | 11 |
| December | 12 |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `yearstart('10/19/2001')` | Retourneert 01/01/2001 00:00:00. |
| `yearstart('10/19/2001',-1)` | Retourneert 01/01/2000 00:00:00. |
| `yearstart('10/19/2001',0,4)` | Retourneert 04/01/2001 00:00:00. |

## Voorbeeld 1 – Basisvoorbeeld

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties tussen 2020 en 2022 wordt geladen in een tabel met de naam '`Transactions`'.
- Het datumveld is opgegeven in de systeemvariabele `DateFormat`, indeling (`MM/DD/YYYY`).
- Een voorgaande load-instructie die het volgende bevat:

- Functie `yearstart()` die is ingesteld als het veld `year_start`.
- Functie `Timestamp()` die is ingesteld als het veld `year_start_timestamp`

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `date`
- `year_start`
- `year_start_timestamp`

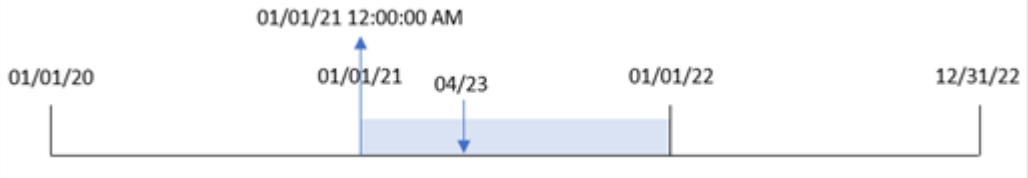
Resultatentabel

| 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 | 1/1/2021 12:00:00 AM |
| 8197 | 02/03/2021 | 01/01/2021 | 1/1/2021 12:00:00 AM |
| 8198 | 03/17/2021 | 01/01/2021 | 1/1/2021 12:00:00 AM |
| 8199 | 04/23/2021 | 01/01/2021 | 1/1/2021 12:00:00 AM |
| 8200 | 05/04/2021 | 01/01/2021 | 1/1/2021 12:00:00 AM |
| 8201 | 06/30/2021 | 01/01/2021 | 1/1/2021 12:00:00 AM |
| 8202 | 07/26/2021 | 01/01/2021 | 1/1/2021 12:00:00 AM |
| 8203 | 12/27/2021 | 01/01/2021 | 1/1/2021 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 |

Het veld '`year_start`' wordt gemaakt in de voorafgaande load-instructie met behulp van de functie `yearstart()` en geeft het datumveld door als het argument van de functie.

De functie `yearstart()` identificeert aanvankelijk in welk jaar de datumwaarde valt en retourneert een tijdstempel voor de eerste milliseconde van dat jaar.

*Diagram van de functie `yearstart()` en transactie 8199.*



Transactie 8199 vond plaats op 23 april 2021. De functie `yearstart()` retourneert de eerste milliseconde van dat jaar, 1 januari om 12:00:00 AM.

## Voorbeeld 2 – period_no

Load-script en resultaten

### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is het echter de taak om een veld te maken, '`previous_year_start`', dat de tijdstempel van de startdatum van het jaar voorafgaand aan het jaar waarin de transactie plaatsvond, retourneert.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

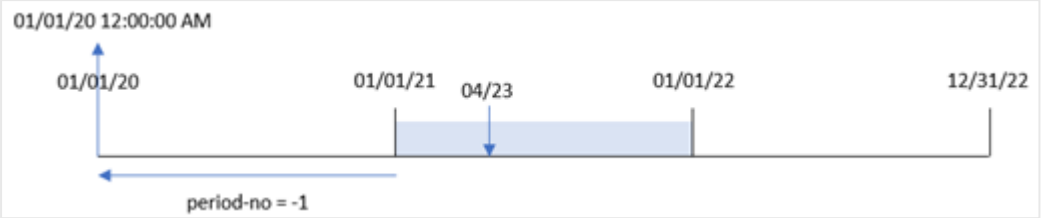
- `id`
- `date`
- `previous_year_start`
- `previous_ year_start_timestamp`

Resultatentabel

| 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 | 1/1/2021 12:00:00 AM |
| 8205 | 07/18/2022 | 01/01/2021 | 1/1/2021 12:00:00 AM |
| 8206 | 11/14/2022 | 01/01/2021 | 1/1/2021 12:00:00 AM |
| 8207 | 12/12/2022 | 01/01/2021 | 1/1/2021 12:00:00 AM |

Omdat in deze instantie een `period_no` van `-1` wordt gebruikt als het offset-argument in de functie `yearstart()`, identificeert de functie eerst het jaar waarin de transacties plaatsvinden. Het kijkt dan een jaar eerder en identificeert de eerste milliseconde van dat jaar.

*Diagram van de functie `yearstart()` met een `period_no` van -1.*




Transactie 8199 vond plaats op 23 april 2021. De functie `yearstart()` retourneert de eerste milliseconde van het voorgaande jaar, 1 januari 2020 om 12:00:00 AM, voor het veld '`previous_year_start`'.

## Voorbeeld 3 – first_month_of_year

### Load-script en resultaten

#### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is het bedrijfsbeleid echter dat het jaar op 1 april begint.

#### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

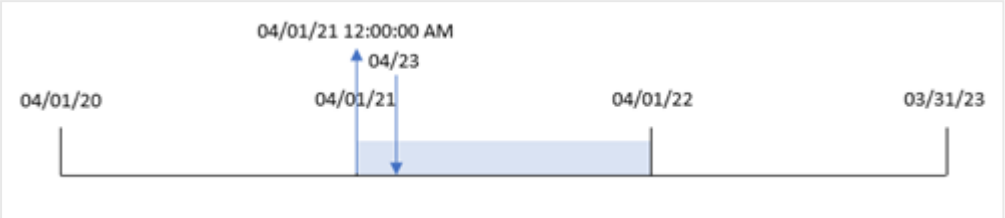
- id
- date
- year_start
- year_start_timestamp

Resultatentabel

| 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 | 4/1/2021 12:00:00 AM |
| 8200 | 05/04/2021 | 04/01/2021 | 4/1/2021 12:00:00 AM |
| 8201 | 06/30/2021 | 04/01/2021 | 4/1/2021 12:00:00 AM |
| 8202 | 07/26/2021 | 04/01/2021 | 4/1/2021 12:00:00 AM |
| 8203 | 12/27/2021 | 04/01/2021 | 4/1/2021 12:00:00 AM |
| 8204 | 06/06/2022 | 04/01/2022 | 4/1/2022 12:00:00 AM |
| 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 |

Omdat in deze instantie het argument `first_month_of_year` van `4` wordt gebruikt in de functie `yearstart()`, stelt het de eerste dag van het jaar in op 1 april en de laatste dag van het jaar op 31 maart.

*Diagram van de functie `yearstart()` met de eerste maand ingesteld als april.*




Transactie 8199 vond plaats op 23 april 2021. Omdat de functie `yearstart()` het begin van het jaar instelt op 1 april en dit retourneert als de waarde '`year_start`' voor de transactie.

## Voorbeeld 4 – diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is de gegevensverzameling echter niet gewijzigd en wordt in de applicatie geladen. De berekening die een startdatumtijdstempel retourneert van het jaar waarin de transactie plaatsvond is gemaakt als meting in een diagramobject van de toepassing.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `date`

Maak de volgende metingen om te berekenen in welk jaar een transactie plaatsvond:

- `=yearstart(date)`
- `=timestamp(yearstart(date))`

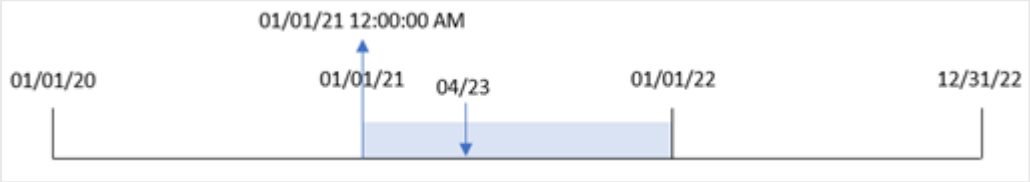
Resultatentabel

| 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 | 1/1/2021 12:00:00 AM |
| 8193 | 02/03/2021 | 01/01/2021 | 1/1/2021 12:00:00 AM |
| 8194 | 03/17/2021 | 01/01/2021 | 1/1/2021 12:00:00 AM |
| 8195 | 04/23/2021 | 01/01/2021 | 1/1/2021 12:00:00 AM |
| 8196 | 05/04/2021 | 01/01/2021 | 1/1/2021 12:00:00 AM |
| 8197 | 06/30/2021 | 01/01/2021 | 1/1/2021 12:00:00 AM |
| 8198 | 07/26/2021 | 01/01/2021 | 1/1/2021 12:00:00 AM |
| 8199 | 12/27/2021 | 01/01/2021 | 1/1/2021 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 |
| 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 |

De meting '`start_of_year`' wordt gemaakt in het diagramobject met behulp van de functie `yearstart()` en geeft het veld door als het argument van de functie.

De functie `yearstart()` identificeert aanvankelijk in welk jaar de datumwaarde valt en retourneert een tijdstempel voor de eerste milliseconde van dat jaar.

*Diagram van de functie `yearstart()` en transactie 8199.*




Transactie 8199 vond plaats op 23 april 2021. De functie `yearstart()` retourneert de eerste milliseconde van dat jaar, 1 januari om 12:00:00 AM.

## Voorbeeld 5 – Scenario

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset wordt geladen in een tabel met de naam '`Loans`'. De tabel bevat de volgende velden:
    - Lening-id's.
    - Het saldo aan het begin van het jaar.
    - De enkelvoudige rente die op elke lening per jaar wordt berekend.

De eindgebruiker wil een diagramobject dat per lening-id de huidige rente weergeeft die is opgebouwd op elke lening in het jaar tot heden.

### Load-script

```
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
];
```

**Resultaten**

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `loan_id`
- `start_balance`

Maak de volgende meting om de opgebouwde rente te berekenen:

```
=start_balance*(rate*(today(1)-yearstart(today(1)))/365)
```

Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

| 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 |

De functie `yearstart()` gebruikt de datum van vandaag als enige argument en retourneert de begindatum van het huidige jaar. Door dat resultaat af te trekken van de huidige datum, retourneert de uitdrukking het aantal dagen dat in dit jaar tot nu toe is verstreken.

Deze waarde wordt vervolgens vermenigvuldigd met het rentepercentage en gedeeld door 365 om het effectieve rentepercentage voor de periode te retourneren. Het effectieve rentepercentage over de periode wordt vervolgens vermenigvuldigd met het beginsaldo van de lening om de rente terug te betalen die dit jaar tot nu toe is opgebouwd.

## yeartodate

Deze functie bepaalt of de invoertijdstempel binnen het jaar valt waarin het script voor het laatst is geladen en retourneert True als dat het geval is en False als dat niet het geval is.

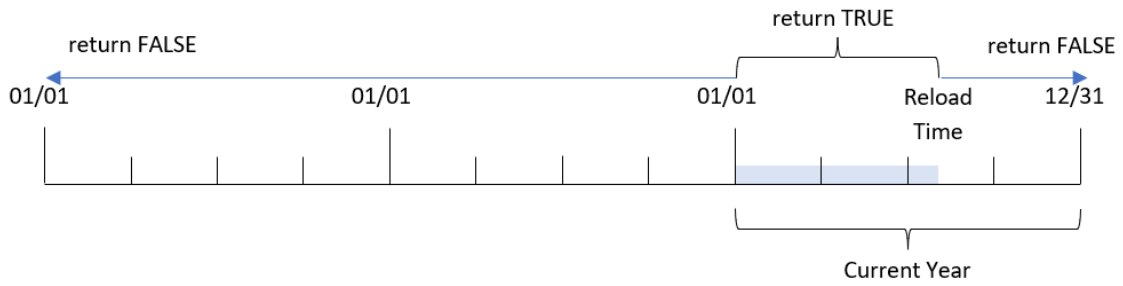
**Syntaxis:**

```
YearToDate(timestamp[ , yearoffset [ , firstmonth [ , todaydate] ] ])
```

**Retourgegevenstypen:** Booleaanse waarde

> *In Qlik Sense wordt de booleaanse waarde Waar vertegenwoordigd door -1 en de waarde Onwaar door 0.*

*Voorbeelddiagram van de `yeartodate()`-functie*




Als geen van de optionele parameters wordt gebruikt, betekent het jaar tot op heden een datum binnen één kalenderjaar vanaf 1 januari tot en met de datum waarop het script voor het laatst is uitgevoerd.

Met andere woorden: wanneer de `yeartodate()`-functie zonder extra parameters wordt getriggerd, wordt deze gebruikt om een tijdstempel te evalueren en een booleaans resultaat te retourneren op basis van of de datum in het kalenderjaar viel tot en met de datum waarop er opnieuw is geladen.

Maar het is tevens mogelijk om de begindatum van het jaar te vervangen met het `firstmonth`-argument, en om vergelijkingen te maken met voorgaande of volgende jaren met behulp van het `yearoffset`-argument.

In gevallen van historische gegevenssets, biedt de `yeartodate()`-functie een parameter om `todaydate` in te stellen, waarmee de tijdstempel met het kalenderjaar wordt vergeleken tot en met de datum die in het `todaydate`-argument is opgegeven.

Argumenten

| Argument | Beschrijving |
|---|---|
| timestamp | De tijdstempel die moet worden geëvalueerd, bijvoorbeeld '10/12/2012'. |
| yearoffset | Als een **yearoffset** wordt opgegeven, retourneert **yeartodate** de waarde True voor dezelfde periode in een ander jaar. Een negatieve **yearoffset** geeft een eerder jaar aan en een positieve verschuiving een toekomstig jaar. Het meest recente jaar tot heden wordt geretourneerd door yearoffset = -1 op te geven. Bij weglating, wordt 0 aangenomen. |
| firstmonth | Door een **firstmonth** tussen 1 en 12 op te geven (1 indien weggelaten), kunt u het begin van het jaar naar voren verplaatsen naar de eerste dag van een andere maand. Als u bijvoorbeeld wilt werken met een fiscaal jaar dat begint op 1 mei, geeft u **firstmonth** = 5 op. Een waarde van 1 zou aangeven dat een boekjaar begint op 1 januari en een waarde van 12 zou aangeven dat een boekjaar begint op 1 december. |
| todaydate | Door een **todaydate** op te geven (tijdstempel van de laatste scriptuitvoering indien weggelaten), kunt u de dag verschuiven die als bovengrens van de periode wordt gebruikt. |

## Wanneer gebruiken

De functie `yeartodate()` retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als voorwaarde in een if-uitdrukking. Dit retourneert een aggregatie of berekening afhankelijk van of een geëvalueerde datum plaatsvond in het jaar tot en met de datum waarop de applicatie voor het laatst opnieuw is geladen.

De YearToDate()-functie kan bijvoorbeeld worden gebruikt om alle materieel te identificeren dat tot dusver in het huidige jaar is geproduceerd.

Voor de volgende voorbeelden is de laatste datum waarop het script werd geladen = 11/18/2011.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `yeartodate( '11/18/2010')` | retourneert False |
| `yeartodate( '02/01/2011')` | retourneert True |
| `yeartodate( '11/18/2011')` | retourneert True |
| `yeartodate( '11/19/2011')` | retourneert False |
| `yeartodate( '11/19/2011', 0, 1, '12/31/2011')` | retourneert True |
| `yeartodate( '11/18/2010', -1)` | retourneert True |
| `yeartodate( '11/18/2011', -1)` | retourneert False |
| `yeartodate( '04/30/2011', 0, 5)` | retourneert False |
| `yeartodate( '05/01/2011', 0, 5)` | retourneert True |

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 – Basisvoorbeeld

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling met een set transacties tussen 2020 en 2022 die wordt geladen in de tabel `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.
- Het aanmaken van een veld, `year_to_date`, dat bepaalt welke transacties hebben plaatsgevonden in hetzelfde kalenderjaar als de laatste keer dat er is geladen.

Op dit moment is de datum 26 april 2022.

### Load-script

```
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
];
```

### Resultaten

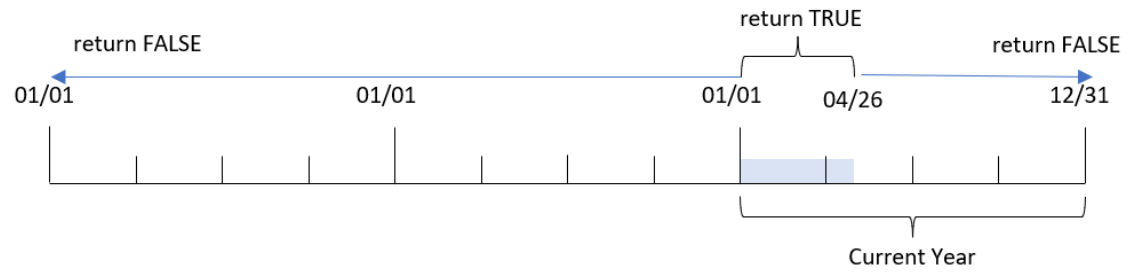
Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `year_to_date`

Resultatentabel

| date | year_to_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 |
| 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 |

*Diagram van `yeartodate()`-functie, basisvoorbeeld*




Het `year_to_date`-veld wordt gemaakt in de voorafgaande load-instructie met behulp van de `yeartodate()`-functie en geeft het `date`-veld door als het argument van de functie.

Omdat er geen andere parameters worden doorgegeven in de functie, identificeert de `yeartodate()`-functie aanvankelijk de herlaaddatum en daarom de grenswaarden voor het huidige kalenderjaar (vanaf 1 januari) die een booleaanse waarde van `TRUE` retourneren.

Daarom zal elke transactie die plaatsvindt tussen 1 januari en 26 april, de datum van opnieuw laden, een booleaans resultaat van `TRUE` retourneren. Een transactie die vóór het begin van 2022 plaatsvindt, retourneert een booleaans resultaat voor `FALSE`.

## Voorbeeld 2 – yearoffset

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het aanmaken van een veld, `two_years_prior`, dat bepaalt welke transacties twee jaar voor het kalenderjaar tot heden hebben plaatsgevonden.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `two_years_prior`

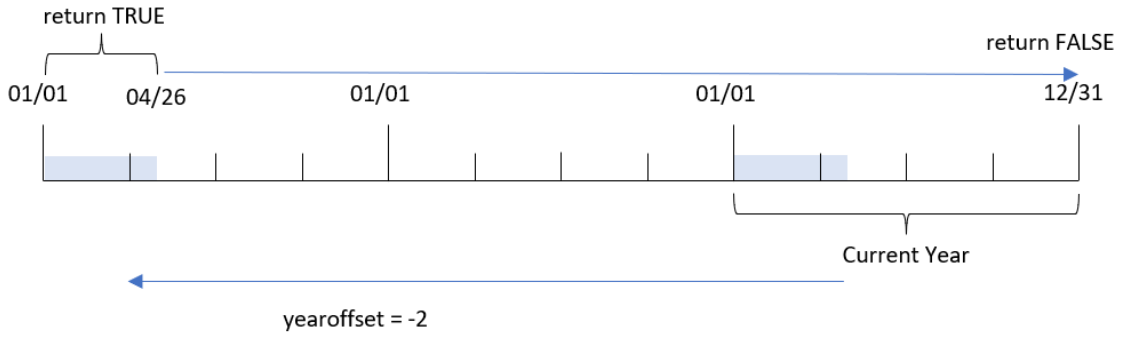
Resultatentabel

| date | two_years_prior |
|---|---|
| 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 |
| 12/05/2020 | 0 |
| 01/22/2021 | 0 |
| 02/03/2021 | 0 |
| 03/17/2021 | 0 |
| 04/23/2021 | 0 |

| date | two_years_prior |
|---|---|
| 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 |

Door -2 als het `yearoffset`-argument te gebruiken in de `yeartodate()`-functie, verschuift de functie de grenswaarden van het kalenderjaarsegment voor vergelijking met twee hele jaren. Aanvankelijk komt het jaarsegment overeen met de periode tussen 1 januari en 26 april 2022. Het `yearoffset`-argument verschuift dit segment vervolgens naar twee jaar eerder. De datumgrenzen vallen tussen 1 januari en 26 april.

*Diagram van `yeartodate()`-functie, voorbeeld yearoffset*




Daarom zal elke transactie die plaatsvindt tussen 1 januari en 26 april 2020 een booleaans resultaat van TRUE retourneren. Iedere transactie die vóór of na dit segment valt, retourneert FALSE.

## Voorbeeld 3 – firstmonth

### Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het aanmaken van een veld, `year_to_date`, dat bepaalt welke transacties hebben plaatsgevonden in hetzelfde kalenderjaar als de laatste keer dat er is geladen.

In dit voorbeeld stellen de het begin van het boekjaar in op 1 juli.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

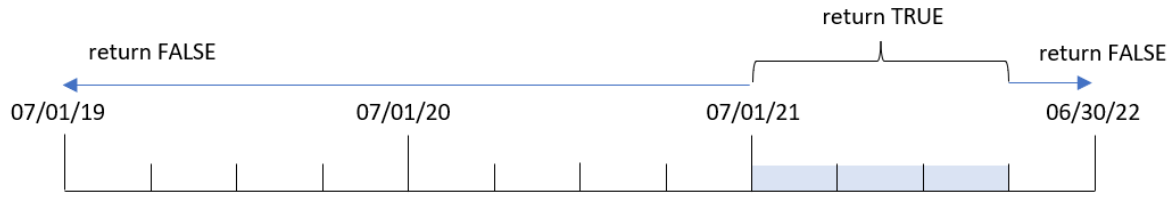
- `date`
- `year_to_date`

Resultatentabel

| date | year_to_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 |
| 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 |

Omdat in deze instantie het `firstmonth`-argument 7 wordt gebruikt in de `yeartodate()`-functie, stelt het de eerste dag van het jaar in op 1 juli en de laatste dag van het jaar op 30 juni.

*Diagram van de `yeartodate()`-functie, voorbeeld firstmonth*



Daarom zal elke transactie die plaatsvindt tussen 1 juli 2021 en 26 april 2022, de datum van opnieuw laden, een booleaans resultaat voor TRUE retourneren. Een transactie die vóór 1 juli 2021 plaatsvindt, retourneert een booleaans resultaat voor FALSE.

## Voorbeeld 4 – todaydate

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het aanmaken van een veld, `year_to_date`, dat bepaalt welke transacties hebben plaatsgevonden in hetzelfde kalenderjaar als de laatste keer dat er is geladen.

Maar in dit voorbeeld moeten we alle transacties identificeren die plaatsvonden in het kalenderjaar tot en met 1 maart 2022.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

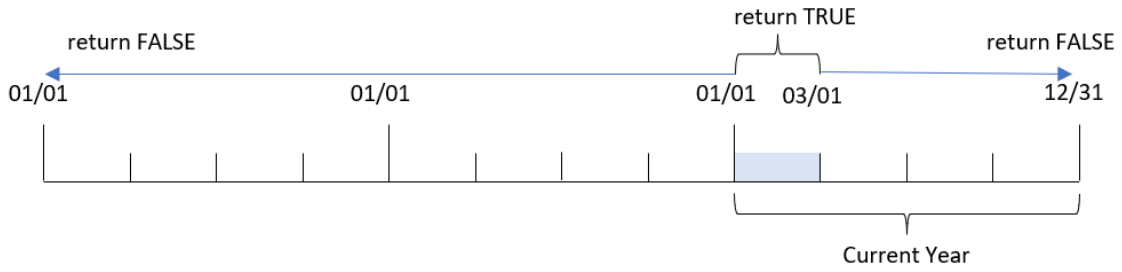
- `date`
- `year_to_date`

Resultatentabel

| date | year_to_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 |
| 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 | 0 |
| 03/11/2022 | 0 |

Omdat het `todaydate`-argument van 03/01/2022 in dit geval wordt gebruikt in de `yeartodate()`-functie, stelt het de eindgrens van het kalenderjaarsegment waarmee wordt vergeleken in op 1 maart 2022. Het is cruciaal om de `firstmonth`-parameter (tussen 1 en 12) in te voeren omdat de functie anders geen resultaten oplevert.

*Diagram van de `yeartodate()`-functie, voorbeeld met het todaydate-argument*




Daarom zal elke transactie die plaatsvindt tussen 1 januari 2022 en 1 maart 2022, de parameter `todaydate`, een booleaans resultaat voor `TRUE`retourneren. Een transactie die vóór 1 januari 2022 of na 1 maart 2022 plaatsvindt, retourneert een booleaans resultaat van `FALSE`.

## Voorbeeld 5 – Diagramobjectvoorbeeld

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die bepaalt welke transacties plaatsvonden in het kalenderjaar tot de datum van de laatste keer dat er is geladen, wordt gemaakt als een meting in een diagramobject van de applicatie.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `date`.

Voeg de volgende meting toe:

```
=yeartodate(date)
```

Resultatentabel

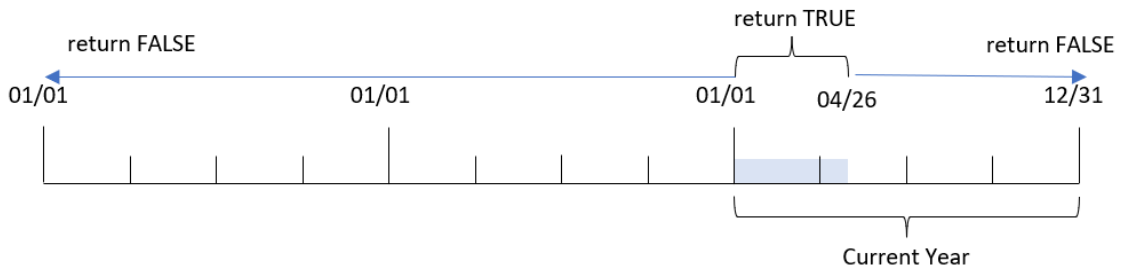
| date | =yeartodate(date) |
|---|---|
| 01/10/2020 | 0 |
| 02/28/2020 | 0 |
| 04/09/2020 | 0 |
| 04/16/2020 | 0 |
| 05/21/2020 | 0 |
| 08/14/2020 | 0 |
| 10/07/2020 | 0 |
| 12/05/2020 | 0 |
| 01/22/2021 | 0 |
| 02/03/2021 | 0 |
| 03/17/2021 | 0 |
| 04/23/2021 | 0 |
| 05/04/2021 | 0 |
| 06/30/2021 | 0 |
| 07/26/2021 | 0 |
| 12/27/2021 | 0 |
| 02/02/2022 | -1 |

| date | =yeartodate(date) |
|---|---|
| 02/26/2022 | -1 |
| 03/07/2022 | -1 |
| 03/11/2022 | -1 |

De meting `year_to_date` wordt gemaakt in het diagramobject met behulp van de `yeartodate()`-functie en geeft het veld `date` door als het argument van de functie.

Omdat er geen andere parameters worden doorgegeven in de functie, identificeert de `yeartodate()`-functie aanvankelijk de herlaaddatum en daarom de grenswaarden voor het huidige kalenderjaar (vanaf 1 januari) die een booleaanse waarde van `TRUE` retourneren.

*Diagram van `yeartodate()`-functie, voorbeeld met gebruik van een diagramobject*




Elke transactie die plaatsvindt tussen 1 januari en 26 april, de datum van opnieuw laden, zal een booleaans resultaat voor `TRUE` retourneren. Een transactie die vóór het begin van 2022 plaatsvindt, retourneert een booleaans resultaat voor `FALSE`.

## Voorbeeld 6 – Scenario

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensset met een set transacties tussen 2020 en 2022 die wordt geladen in een tabel met de naam `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.

De eindgebruiker wil een KPI-object dat de totale verkoop voor de overeenkomstige periode in 2021 weergeeft als het huidige jaar tot heden als de datum van de laatste keer dat er is geladen.

Op dit moment is de datum 16 juni 2022.

### Load-script

```
SET DateFormat='MM/DD/YYYY';

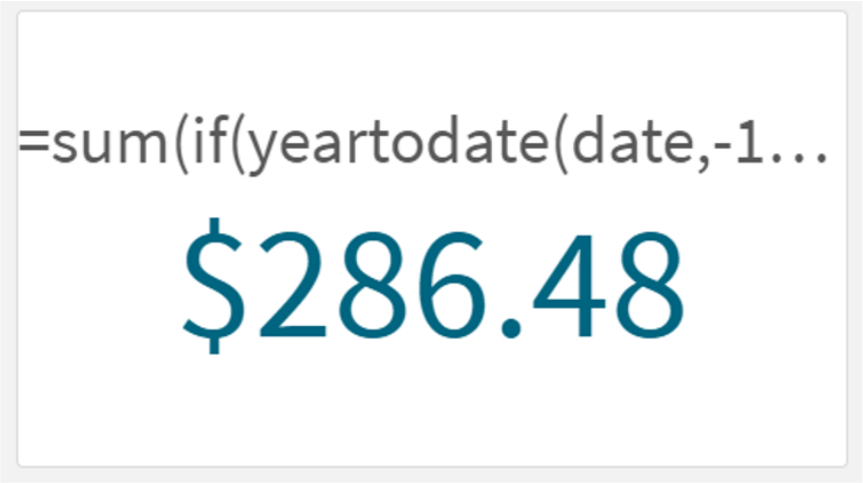
Transactions:
Load
*
Inline
[
id,date,amount
8188,01/10/2020,37.23
8189,02/28/2020,17.17
8190,04/09/2020,88.27
8191,04/16/2020,57.42
8192,05/21/2020,53.80
8193,08/14/2020,82.06
8194,10/07/2020,40.39
8195,12/05/2020,87.21
8196,01/22/2021,95.93
8197,02/03/2021,45.89
8198,03/17/2021,36.23
8199,04/23/2021,25.66
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
];
```

### Resultaten

### Doe het volgende:

1. Maak een KPI-object.
2. Maak de volgende aggregatiemeting om de totale verkoop te berekenen:
   `=sum(if(yeartodate(date,-1),amount,0))`
3. Stel de **Getalnotatie** van de meting in op **Geld**.

*KPI `yeartodate()`-diagram voor 2021*




De `yeartodate()`-functie retourneert een booleaanse waarde bij het evalueren van de productiedatums van elke transactie-id. Omdat er op 16 juni 2022 opnieuw is geladen, segmenteert de `yeartodate`-functie de periode van het jaar tussen 01/01/2022 en 06/16/2022. Maar omdat er een `period_no`-waarde van -1 in de functie is gebruikt, worden deze grenzen naar het vorige jaar verschoven. Daarom retourneert de `yeartodate()`-functie voor iedere transactie die tussen 01/01/2021 en 06/16/2021 plaatsvindt, een booleaanse waarde van `TRUE` en telt hij het bedrag op.

## 8.8 Exponentiële en logaritmische functies

In dit hoofdstuk worden functies beschreven die verband houden met exponentiële en logaritmische berekeningen. Alle functies kunnen zowel worden gebruikt in het load-script voor gegevens als in diagramuitdrukkingen.

In de onderstaande functies zijn de parameters uitdrukkingen waarbij **x** en **y** moeten worden geïnterpreteerd als reële getallen.

### exp

De natuurlijk exponentiële functie, e^x, die de natuurlijke logaritme **e** als grondtal gebruikt. Het resultaat is een positief getal.

```
exp(x )
```

**Voorbeelden en resultaten:**

`exp(3)` retourneert 20,085.

### log

De natuurlijke logaritme van **x**. De functie is alleen gedefinieerd als **x**> 0. Het resultaat is een getal.

```
log(x )
```

**Voorbeelden en resultaten:**

`log(3)` retourneert 1,0986

### log10

De algemene logaritme (grondtal 10) van **x**. De functie is alleen gedefinieerd als **x**> 0. Het resultaat is een getal.

```
log10(x )
```

**Voorbeelden en resultaten:**

`log10(3)` retourneert 0,4771

### pow

Retourneert **x** tot de macht **y**. Het resultaat is een getal.

```
pow(x,y )
```

**Voorbeelden en resultaten:**

`pow(3, 3)` retourneert 27

### sqr

**x** kwadraat (**x**tot de macht 2). Het resultaat is een getal.

```
sqr (x )
```

**Voorbeelden en resultaten:**

`sqr(3)` retourneert 9

### sqrt

Vierkantswortel van **x**. De functie is alleen gedefinieerd als **x** >= 0. Het resultaat is een positief getal.

```
sqrt(x )
```

**Voorbeelden en resultaten:**

`sqrt(3)` retourneert 1,732

## 8.9 Veldfuncties

Deze functies kunnen alleen worden gebruikt in diagramuitdrukkingen.

Veldfuncties moeten gehele getallen of tekenreeksen retourneren waarin verschillende aspecten van veldselecties worden geïdentificeerd.

## Tellerfuncties

GetAlternativeCount

**GetAlternativeCount()** wordt gebruikt om het aantal alternatieve (lichtgrijze) waarden in het vastgestelde veld te zoeken.

```
GetAlternativeCount - diagramfunctie (field_name)
```

GetExcludedCount

**GetExcludedCount()** vindt het aantal uitgesloten distinctieve waarden in het vastgestelde veld. Alleen uitgesloten velden (donkergrijs) worden geteld. Alternatieve (lichtgrijs) en geselecteerde uitgesloten (donkergrijs met vinkje) waarden worden niet geteld.

```
GetExcludedCount - diagramfunctie(field_name)
```

GetPossibleCount

**GetPossibleCount()** wordt gebruikt om het aantal mogelijke waarden in het vastgestelde veld te zoeken. Als het vastgestelde veld selecties bevat, worden de geselecteerde (groene) velden geteld. Anders worden de geassocieerde (witte) waarden geteld.

```
GetPossibleCount - diagramfunctie(field_name)
```

GetSelectedCount

**GetSelectedCount()** retourneert het aantal geselecteerde (groene) waarden in een veld.

```
GetSelectedCount - diagramfunctie (field_name [, include_excluded])
```

GetStateCounts

De **GetStateCounts()** diagramfunctie wordt gebruikt om het totale aantal unieke waarden te berekenen dat overeenkomt met de opgegeven selectiestatussen

```
GetStateCounts - diagramfunctie (field_name, state_name [, state_
type1,...state_typeN])
```

## Veld- en selectiefuncties

GetCurrentSelections

**GetCurrentSelections()** retourneert een lijst van de huidige selecties in de app. Als de selecties in plaats daarvan zijn gemaakt met een zoekreeks in een zoekvak, retourneert **GetCurrentSelections()** de zoekreeks.

```
GetCurrentSelections - diagramfunctie([record_sep [,tag_sep [,value_sep
[,max_values]]]])
```

GetFieldSelections

**GetFieldSelections()** retourneert een **tekenreeks** met de huidige selecties in een veld.

```
GetFieldSelections - diagramfunctie ( field_name [, value_sep [, max_
values]])
```

GetObjectDimension

**GetObjectDimension()** retourneert de naam van de dimensie. **Index** is een optioneel geheel getal dat de dimensie aangeeft die moet worden geretourneerd.

```
GetObjectDimension - diagramfunctie ([index])
```

GetObjectField

**GetObjectField()** retourneert de velduitdrukking van de dimensie. **Index** is een optioneel geheel getal dat de dimensie aangeeft die moet worden geretourneerd.

```
GetObjectField - diagramfunctie ([index])
```

GetObjectMeasure

**GetObjectMeasure()** retourneert de naam van de meting. **Index** is een optioneel geheel getal dat de meting aangeeft die moet worden geretourneerd.

```
GetObjectMeasure - diagramfunctie ([index])
```

## GetAlternativeCount - diagramfunctie

**GetAlternativeCount()** wordt gebruikt om het aantal alternatieve (lichtgrijze) waarden in het vastgestelde veld te zoeken.

**Syntaxis:**

```
GetAlternativeCount (field_name)
```

**Retourgegevenstypen:** geheel getal

> *De kleuren die worden gebruikt in de selectiebalk en voor elke selectiestatus kunnen worden aangepast met een aangepast thema. Als u werkt met een app die een aangepast thema gebruikt, merkt u mogelijk dat uw selecties niet worden weergegeven met dezelfde kleuren die worden beschreven in dit Help-onderwerp.*

Argumenten

| Argument | Beschrijving |
|---|---|
| **field_name** | Het veld met het bereik van gegevens die moeten worden gemeten. |

In de volgende tabel staan andere functies die aan deze functie gerelateerd zijn.

Gerelateerde functies

| Functie | Interactie |
|---|---|
| *GetStateCounts - diagramfunctie (page 1356)* | Met **GetStateCounts()** kunt u de berekening van de volgende aantallen combineren met één enkele functieaanroep:<br>• Aantal geselecteerde opgenomen waarden.<br>• Aantal mogelijke waarden.<br>• Aantal alternatieve waarden.<br>• Aantal uitgesloten waarden, exclusief alternatieve en geselecteerde opgenomen waarden.<br>• Aantal geselecteerde uitgesloten waarden. |
| *GetSelectedCount - diagramfunctie (page 1351)* | Retourneert het aantal geselecteerde opgenomen waarden. |
| *GetPossibleCount - diagramfunctie (page 1347)* | Retourneert het aantal mogelijke waarden. |
| *GetAlternativeCount - diagramfunctie (page 1317)* | Retourneert het aantal uitgesloten waarden, exclusief alternatieve en geselecteerde opgenomen waarden. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `GetAlternativeCount ([First name])` | Retourneert `0` als er geen selectie is gemaakt in het veld `First Name`. Als er selecties zijn gemaakt, dan is de geretourneerde waarde het totale aantal verschillende beschikbare waarden min het aantal geselecteerde waarden. |

## Voorbeeld - Basisprincipes voor GetAlternativeCount

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `FirstName`
  - `LastName`
  - `Initials`
  - `HasCellphone`

### Load-script

```
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 '|');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuw filtervak en voeg dit veld toe als dimensie:

- `FirstName`

Voeg een KPI toe en maak dan de volgende meting:

- `=GetAlternativeCount (FirstName)`, om het aantal alternatieve waarden in het veld `FirstName` te vinden.

De KPI met de functie `GetAlternativeCount` verschijnt met een waarde van `0`. Als u `Jane` selecteert in het filtervak `FirstName`, retourneert de KPI de waarde `4` omdat er vier unieke en uitgezonderde (grijze) waarden zijn. Selecteer nu `Mark`, zodat zowel `Jane` als `Mark` geselecteerd zijn. De KPI-waarde voor de meting `GetAlternativeCount` retourneert `3` omdat er 3 unieke en uitgesloten waarden zijn, `John`, `Peter`, en `Sue`.

*KPI-resultaten met 3 unieke en uitgesloten waarden na het selecteren van Jane en Mark in het filtervak FirstName*

U kunt zien hoe de waarden die door de functie `GetAlternativeCount` worden geretourneerd, veranderen naarmate er meer selecties worden gemaakt.

## Voorbeeld - Scenario met GetAlternativeCount

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `CustomerTypeData`.
- De volgende velden in de gegevenstabel:
    - `CustomerID`
    - `CustomerType`

### Load-script

```
CustomerTypeData:
LOAD * INLINE [
CustomerID, CustomerType
C001, Retail
C002, Wholesale
C003, Distributor
C004, Retail
C005, Wholesale
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuw filtervak en voeg dit veld toe als dimensie:

- `CustomerType`

Voeg een KPI toe en maak dan de volgende meting:

- `=GetAlternativeCount (CustomerType)`, om het aantal alternatieve waarden in het veld `CustomerType` te vinden.
  De KPI met de functie `GetAlternativeCount` verschijnt met een waarde van `0` omdat er geen waarden zijn geselecteerd in het veld `CustomerType`.

Selecteer `Retail` uit het filtervak voor `CustomerType`. De `GetAlternativeCount` retourneert de waarde `2` omdat er twee unieke en uitgesloten (lichtgrijze) waarden in het veld `CustomerType` staan.

*KPI-resultaten met 2 unieke en uitgesloten waarden na het selecteren van Retail in het filtervak CustomerType*

De resultaten laten zien dat hoewel er meerdere waarden zijn, de functie `GetAlternativeCount` alleen de unieke telling van de niet geselecteerde items weergeeft.

## Voorbeeld - Scenario met GetAlternativeCount met beschikbare waarden

Diagramuitdrukking

### Overzicht

Dit voorbeeld bouwt voort op het vorige scenario. De gegevensverzameling bevat een tweede gegevenstabel om te laten zien hoe selecties in elk veld de resultaten van de functie `GetAlternativeCount` veranderen.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die in de volgende 2 inline-gegevenstabellen en bijbehorende velden wordt geladen:
  - `CustomerTypeData`: omvat de velden `CustomerID` en `CustomerType`.
  - `CustomerSegment`: omvat de velden `CustomerType` en `Segment`.

### Load-script

```
CustomerTypeData:
LOAD * INLINE [
CustomerID, CustomerType
C001, Retail
C002, Wholesale
C003, Distributor
C004, Retail
C005, Wholesale
];

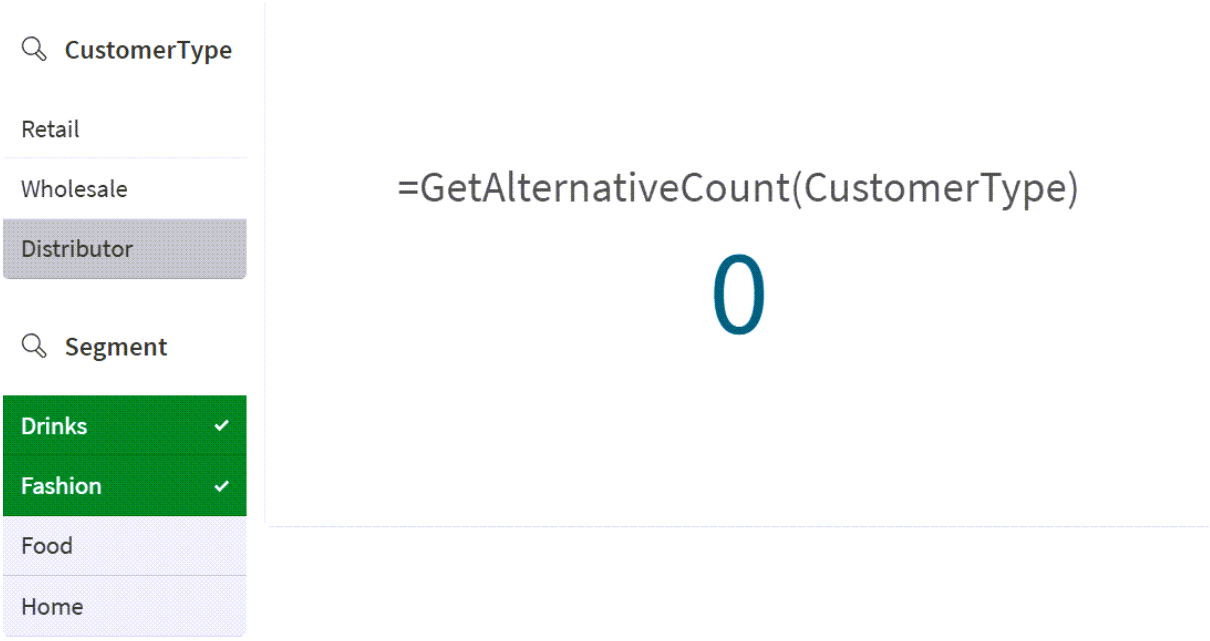
CustomerSegment:
```

```
Load * Inline [
CustomerType, Segment
Retail, Fashion
Retail, Home
Wholesale, Fashion
Wholesale, Drinks
Distributor, Food
];
```

### Resultaten

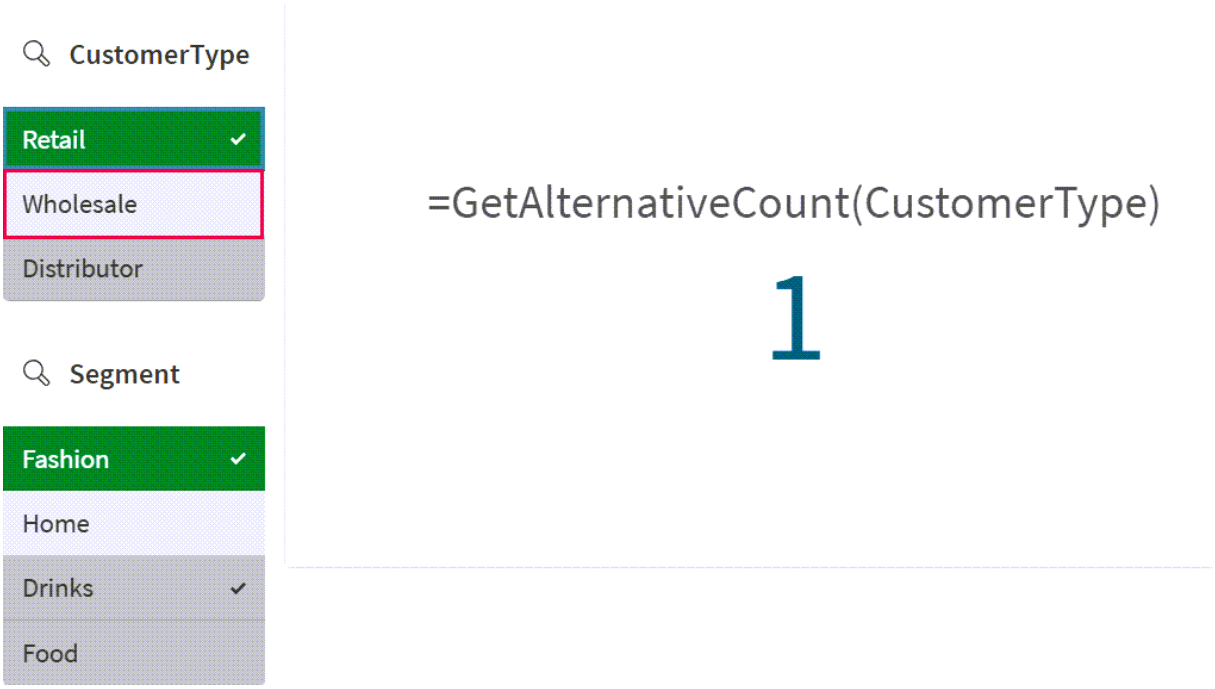
1. Laad de gegevens en open een werkblad. Maak een nieuw filtervak en voeg deze velden toe als dimensies:
   - `CustomerType`
   - `Segment`
2. Voeg een KPI toe en maak dan de volgende meting:
   - `=GetAlternativeCount (CustomerType)`, om het aantal alternatieve waarden in het veld `CustomerType` te vinden.
     De KPI met de functie `GetAlternativeCount` verschijnt met een waarde van `0` omdat er geen waarden zijn geselecteerd in het veld `CustomerType` of `Segment`.
3. Selecteer `Drinks` en `Fashion` in het filter `Segment`. De KPI retourneert nog steeds `0` omdat er geen waarde is geselecteerd in het veld `CustomerType`.

   *KPI-resultaten met 0 unieke en uitgezonderde waarden na het maken van selecties in het filter `Segment`*




4. Selecteer `Retail` in het veld `CustomerType` en behoud de selecties in het filter `Segment`.
   De functie `GetAlternativeCount` retourneert de waarde `1` omdat er slechts één unieke en uitgezonderde (lichtgrijze) waarde, `Wholesale`, in het veld `CustomerType` staat.

*KPI-resultaten met 1 unieke en uitgezonderde waarde na het maken van selecties in de filters `CustomerType` en `Segment`*




Hoewel twee waarden, `Wholesale` en `Distributor`, niet geselecteerd zijn in het veld `CustomerType` (deze verschijnen lichtgrijs en donkergrijs), is er slechts één waarde, `Wholesale`, *beschikbaar* om te selecteren in `CustomerType` vanwege selecties in het filter `Segment`.

## GetCurrentSelections - diagramfunctie

**GetCurrentSelections()** retourneert een lijst van de huidige selecties in de app. Als de selecties in plaats daarvan zijn gemaakt met een zoekreeks in een zoekvak, retourneert **GetCurrentSelections()** de zoekreeks.

Als opties worden gebruikt, moet u record_sep opgeven. U kunt een nieuw regel opgeven door **record_sep** in te stellen op **chr(13)&chr(10)**.

Als alle waarden op één of twee na worden geselecteerd, wordt respectievelijk de indeling 'NOT x,y' of 'NOT y' gebruikt. Als u alle waarden selecteert en het aantal waarden is groter dan max_values, wordt de tekst ALL geretourneerd.

**Syntaxis:**

```
GetCurrentSelections ([record_sep [, tag_sep [, value_sep [, max_values [, state_name]]]]])
```

**Retourgegevenstypen:** tekenreeks

Argumenten

| Argumenten | Beschrijving |
|---|---|
| **record_sep** | Het scheidingsteken dat tussen veldrecords moet worden geplaatst. De standaardinstelling is <CR><LF>, oftewel een nieuwe regel. |
| **tag_sep** | Het scheidingsteken dat tussen de tag van de veldnaam en de veldwaarden moet worden geplaatst. De standaardinstelling is ': '. |
| **value_sep** | Het scheidingsteken dat tussen twee veldwaarden moet worden geplaatst. De standaardwaarde is ', '. |
| **max_values** | Het maximale aantal veldwaarden dat afzonderlijk moet worden vermeld. Als een groter aantal waarden wordt geselecteerd, wordt de notatie 'x van y waarden' gebruikt. De standaardwaarde is 6. |
| **state_name** | De naam van een alternatieve state die voor de specifieke visualisatie is gekozen. Als het argument **state_name** wordt gebruikt, wordt alleen rekening gehouden met de selecties die zijn gekoppeld aan de opgegeven statenaam. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `GetCurrentSelections ()` | Retourneert (-) als er geen selectie is gemaakt in een van de velden. Als er selecties zijn gemaakt, dan is de geretourneerde waarde de veldnaam en de waarde, bijvoorbeeld `First Name : John`. |

## Voorbeeld - Basisprincipes voor GetCurrentSelections

Diagramuitdrukking

### Overzicht

Dit voorbeeld laat zien hoe de waarden die de functie `GetCurrentSelections` retourneert, veranderen volgens de selecties die u maakt.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `FirstName`
  - `LastName`
  - `Initials`
  - `HasCellphone`

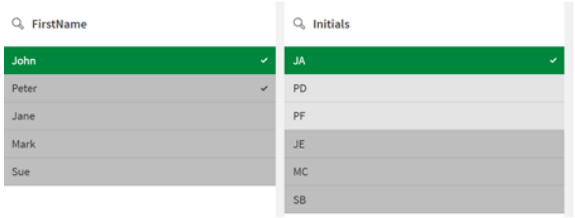
### Load-script

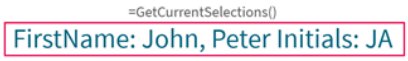
```
Example:
LOAD * inline [
FirstName|LastName|Initials|HasCellphone
John|Anderson|JA|Yes
Sue|Brown|SB|Yes
Mark|Carr|MC|No
Peter|Devonshire|PD|No
Jane|Elliot|JE|Yes
Peter|Franc|PF|Yes ] (delimiter is '|');
```

### Resultaten

1. Laad de gegevens en open een werkblad. Maak een nieuw filtervak en voeg deze velden toe als dimensies:
   - `FirstName`
   - `Initials`
2. Voeg een KPI-object toe en maak dan de volgende meting:
   - `=GetCurrentSelections ()`, om de huidige selecties te vinden.
     De KPI retourneert (-) omdat er geen waarden zijn geselecteerd.

   > *In dit voorbeeld is de opmaak van de KPI aangepast om een responsief lay-outgedrag te gebruiken, zodat de uitvoer gemakkelijker te bekijken is. .*
3. Klik op **Werkblad bewerken** om over te schakelen naar de analysemodus en selecteer vervolgens `John` in het veld `FirstName`. De KPI retourneert het volgende resultaat: `FirstName: John`.
   - Selecteer nu `Peter`, zodat zowel `John` als `Peter` geselecteerd zijn. De KPI retourneert het volgende: `FirstName: John, Peter`.
   - Voeg toe aan de selectie door `JA` te selecteren in het veld `Initials`. De KPI retourneert het volgende: `First name: John, Peter Initials: JA`.

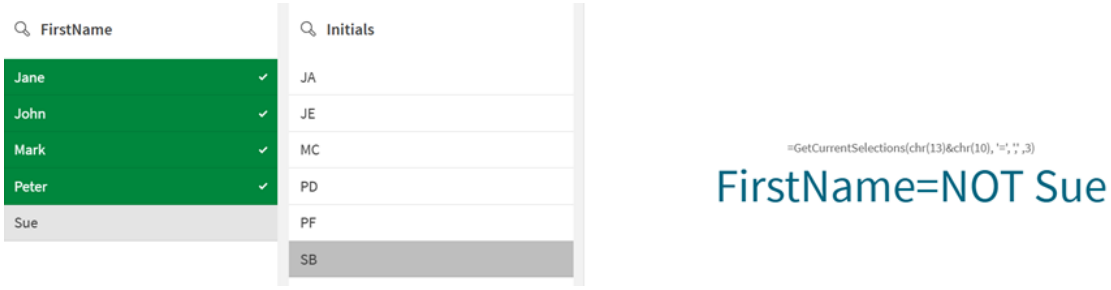


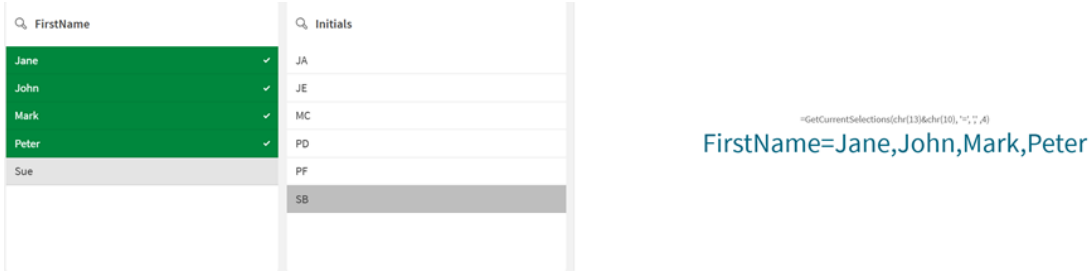

4. Wis de selectie en doe het volgende:
   - Selecteer `John` in het veld `FirstName` en `JA` in het veld `Initials`.
   - Bewerk de KPI-uitdrukking als volgt: `=GetCurrentSelections ( chr(13)&chr(10) , ' = ' )`, om het tagscheidingsteken te wijzigen in een gelijkheidsteken (=).

     > *KPI ondersteunt geen regeleinden en voegt een spatie toe als recordscheidingsteken.*

5. Wis de selectie en doe het volgende:
    - Selecteer alle namen behalve `Sue` in het veld `FirstName`.
    - Bewerk de KPI-uitdrukking als volgt: `=GetCurrentSelections (chr(13)&chr(10), '=', ',' ,3)`, om een tagscheidingsteken (`=`), een waardescheidingsteken (`,`), en het maximale aantal waarden toe te voegen om op te sommen (`3`).




De KPI retourneert de waarde : `FirstName=NOT Sue` omdat er vier namen zijn geselecteerd, maar het argument `max_values` geeft er maar drie. Als u het argument `max_values` wijzigt in `4`, worden alle vier de waarden afzonderlijk weergegeven met dezelfde selectie.




## Voorbeeld - Een diagramtitel aanpassen om selecties weer te geven

Diagramuitdrukking

### Overzicht

Dit voorbeeld laat zien hoe u de functie `GetCurrentSelections` kunt gebruiken om dynamische diagramtitels te maken op basis van de gebruikersselectie.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
    - `SalesID`
    - `Product`
    - `Year`

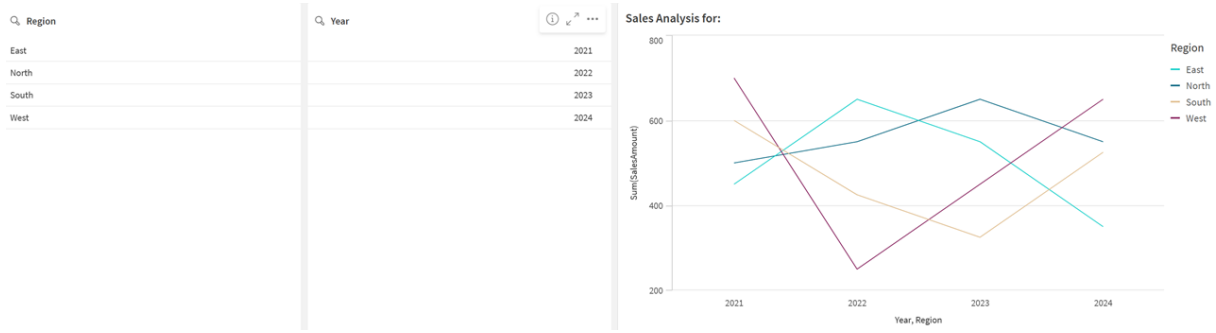
- Region
- SalesAmount

### Load-script

```
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
];
```

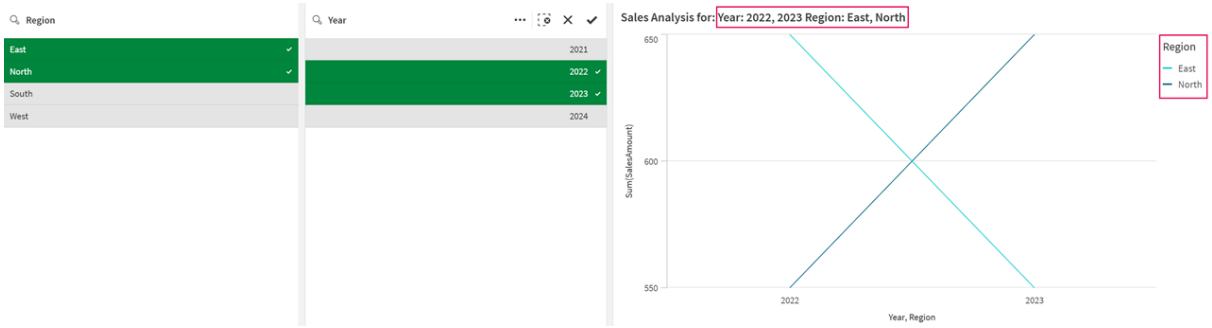
### Resultaten

1. Laad de gegevens en open een werkblad. Maak een nieuw filtervak en voeg deze velden toe als dimensies:
    - Region
    - Year
2. Maak een lijndiagram en voeg de volgende dimensies toe:
    - Year
    - Region
3. Maak de volgende meting:
    - `=Sum(SalesAmount)`, om de totale verkoop weer te geven.
4. In het eigenschappenvenster van de meting, onder **Uiterlijk** > **Algemeen** > **Titel**, voert u de volgende uitdrukking in: `='Sales Analysis for: ' & GetCurrentSelections()`.




De resultaten tonen de som van verkopen voor alle regio's en jaren. De diagramtitel verschijnt als `Sales Analysis for:`.

5. Klik op **Werkblad bewerken** om over te schakelen naar de analysemodus en maak de volgende selecties:
    - Selecteer in het veld **Regio** `East` en `North`.
    - Selecteer in het veld **Jaar** `2022` en `2023`.




De diagramtitel verandert om de filterselecties voor `Region` en `Year` weer te geven.

## GetExcludedCount - diagramfunctie

**GetExcludedCount()** vindt het aantal uitgesloten distinctieve waarden in het vastgestelde veld. Alleen uitgesloten velden (donkergrijs) worden geteld. Alternatieve (lichtgrijs) en geselecteerde uitgesloten (donkergrijs met vinkje) waarden worden niet geteld.

**Syntaxis:**

```
GetExcludedCount (field_name)
```

**Retourgegevenstypen:** geheel getal

> *De kleuren die worden gebruikt in de selectiebalk en voor elke selectiestatus kunnen worden aangepast met een aangepast thema. Als u werkt met een app die een aangepast thema gebruikt, merkt u mogelijk dat uw selecties niet worden weergegeven met dezelfde kleuren die worden beschreven in dit Help-onderwerp.*

Argumenten

| Argumenten | Beschrijving |
|---|---|
| **field_name** | Het veld met het bereik van gegevens die moeten worden gemeten. |

In de volgende tabel staan andere functies die aan deze functie gerelateerd zijn.

Gerelateerde functies

| Functie | Interactie |
|---|---|
| *GetStateCounts - diagramfunctie (page 1356)* | Met **GetStateCounts()** kunt u de berekening van de volgende aantallen combineren met één enkele functieaanroep:<br>• Aantal geselecteerde opgenomen waarden.<br>• Aantal mogelijke waarden.<br>• Aantal alternatieve waarden.<br>• Aantal uitgesloten waarden, exclusief alternatieve en geselecteerde opgenomen waarden.<br>• Aantal geselecteerde uitgesloten waarden. |
| *GetSelectedCount - diagramfunctie (page 1351)* | Retourneert het aantal geselecteerde opgenomen waarden. |
| *GetPossibleCount - diagramfunctie (page 1347)* | Retourneert het aantal mogelijke waarden. |
| *GetAlternativeCount - diagramfunctie (page 1317)* | Retourneert het aantal alternatieve waarden. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `GetExcludedCount( Initials )` | Retourneert `0` als er geen selectie is gemaakt. Als er een selectie is gemaakt, dan is de geretourneerde waarde het aantal uitgezonderde waarden in `Initials` met donkergrijze kleur. |

## Voorbeeld - Basisprincipes voor GetExcludedCount

Diagramuitdrukking

### Overzicht

Dit voorbeeld toont hoe de waarden die door de functie `GetExcludedCount` worden geretourneerd, veranderen naarmate er meer selecties worden gemaakt.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `FirstName`
  - `LastName`
  - `Initials`
  - `HasCellphone`

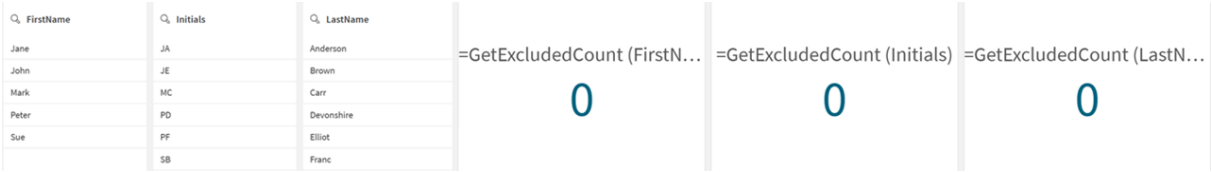
### Load-script

```
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 '|');
```

### Resultaten

1. Laad de gegevens en open een werkblad. Maak een nieuw filtervak en voeg deze velden toe als dimensies:
   - `FirstName`
   - `LastName`
   - `Initials`
2. Maak drie KPI-objecten en voeg de volgende uitdrukkingen toe als metingen, één per KPI-object:
   a. `=GetExcludedCount(FirstName)`, om het aantal uitgezonderde waarden in het veld `FirstName` te tellen.
   b. `=GetExcludedCount(Initials)`, om het aantal uitgezonderde waarden in het veld `Initials` te tellen.
   c. `=GetExcludedCount(LastName)`, om het aantal uitgezonderde waarden in het veld `LastName` te tellen.

   Uw werkblad ziet eruit als de volgende afbeelding omdat er geen waarden zijn geselecteerd. De functie `GetExcludedCount` retourneert `0` voor elke KPI.



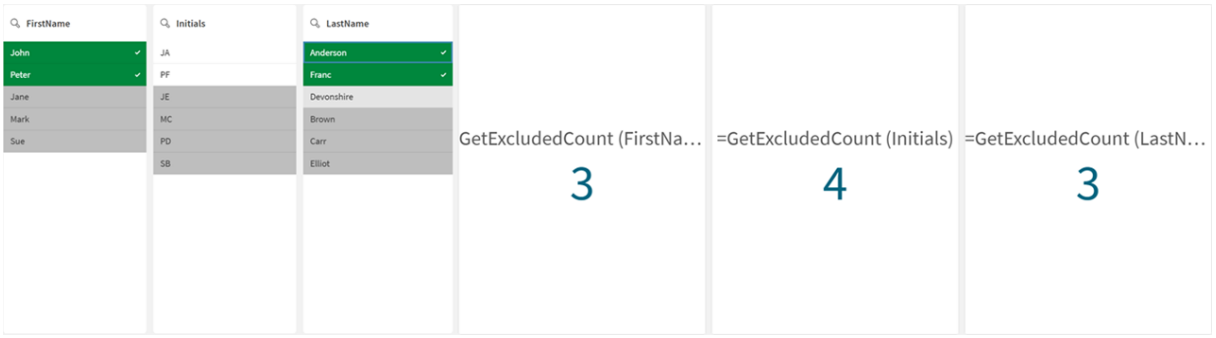

3. Klik op **Werkblad bewerken** om over te schakelen naar de analysemodus en maak de volgende selecties:

Filterselecties en -resultaten in de analysemodus

| Selectie filtervak | KPI-resultaten |
|---|---|
| Selecteer `John` in het filter `FirstName`. | • `GetExcludedCount (Initials)` retourneert een waarde van `5` omdat er `5` uitgesloten waarden met donkergrijze kleur zijn in `Initials`. De `JA` waarde blijft wit, aangezien deze is verbonden met de selectie `John` in `FirstName`.<br>• `GetExcludedCount (LastName)` retourneert een waarde van `5` omdat er `5` uitgesloten waarden met donkergrijze kleur zijn in `LastName`. De `Anderson` waarde blijft wit, aangezien deze is verbonden met de selectie `John` in `FirstName`. |
| Selecteer `John` en `Peter` in het filter `FirstName`. | • `GetExcludedCount (Initials)` retourneert een waarde van `3` omdat er `3` uitgesloten waarden met donkergrijze kleur zijn in `Initials`.<br>• `GetExcludedCount (LastName)` retourneert een waarde van `3` omdat er `3` uitgesloten waarden met donkergrijze kleur zijn in `LastName`. |
| Met `John` en `Peter` geselecteerd, selecteert u `Franc` in het filter `LastName`. | • `GetExcludedCount (FirstName)` retourneert een waarde van `3` omdat er `3` uitgesloten waarden met donkergrijze kleur en zonder selectievakje zijn in `FirstName`. De functie `The GetExcludedCount` telt alleen uitgesloten waarden. Alternatieve en geselecteerde uitgesloten waarden worden niet meegeteld. `John` is een geselecteerde uitgesloten waarde.<br>• `GetExcludedCount (Initials)` retourneert een waarde van `5` omdat er `5` uitgesloten waarden met donkergrijze kleur zijn in `Initials`.<br>• `GetExcludedCount (LastName)` retourneert een waarde van `3` omdat er `3` uitgesloten waarden met donkergrijze kleur zijn in `LastName`. De 2 waarden met lichtgrijze kleur, `Anderson` en `Devonshire`, zijn alternatieve waarden. |
| Behoud de vorige selecties en selecteer vervolgens `Anderson` in het filter `LastName`. | • `GetExcludedCount (FirstName)` retourneert een waarde van `3` omdat er `3` uitgesloten waarden met donkergrijze kleur zijn in `FirstName`.<br>• `GetExcludedCount (Initials)` retourneert een waarde van `4` omdat er `4` uitgesloten waarden met donkergrijze kleur zijn in `Initials`.<br>• `GetExcludedCount (LastName)` retourneert een waarde van `3` omdat er `3` uitgesloten waarden met donkergrijze kleur zijn in `LastName`. |

U kunt de eindresultaten van de selecties in het filtervak in de volgende afbeelding zien.

*KPI-resultaten van de selectie van de voornamen, John en Peter, en achternamen, Franc en Anderson*




## Voorbeeld 5 – Scenario met GetExcludedCount

Diagramuitdrukking

### Overzicht

Een verkoopmanager gebruikt de functie `GetExcludedCount` om een KPI-diagram te maken om de verkochte producten per verkoper te analyseren.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Salesperson`
  - `Product`
  - `SalesAmount`

### Load-script

```
Example:
LOAD * INLINE [
Salesperson, Product, SalesAmount
John, Laptops, 1000
Alice, Smartphones, 800
Bob, Laptops, 1200
Carol, Tablets, 700
Dave, Tablets, 900
];
```

### Resultaten

1. Laad de gegevens en open een werkblad. Maak een nieuw filtervak en voeg deze velden toe als dimensies:
   - `Salesperson`

- `Product`

2. Voeg een nieuw KPI-object toe en maak dan de volgende meting:
   - `=GetExcludedCount(Product)`, om het aantal uitgesloten waarden in het veld `Product` te tellen.
3. Klik op **Werkblad bewerken** om over te schakelen naar de analysemodus en selecteer vervolgens `Alice` in het filter `Salesperson`.
   De functie `GetExcludedCount` retourneert de waarde `2` omdat er `2` unieke en uitgesloten (donkergrijze) waarden in het veld `Product` staan.

De resultaten laten zien hoe u de functie `GetExcludedCount` kunt gebruiken om de gegevens te analyseren. De verkoper `Alice` heeft geen artikelen verkocht in 2 van de 3 productgroepen, `Laptop` en `Tablet` producten.

## GetFieldSelections - diagramfunctie

**GetFieldSelections()** retourneert een **tekenreeks** met de huidige selecties in een veld.

Als alle waarden op één of twee na worden geselecteerd, wordt respectievelijk de indeling 'NOT x,y' of 'NOT y' gebruikt. Als u alle waarden selecteert en het aantal waarden is groter dan max_values, wordt de tekst ALL geretourneerd.

**Syntaxis:**

```
GetFieldSelections ( field_name [, value_sep [, max_values [, state_name]]])
```

**Retourgegevenstypen:** tekenreeks

Retourtekenreeksindelingen

| Opmaak | Beschrijving |
|---|---|
| 'a, b, c' | Als het aantal geselecteerde waarden max_values of minder is, is de tekenreeks die wordt geretourneerd, een lijst met de geselecteerde waarden.<br><br>De waarden worden gescheiden met value_sep als scheidingsteken. |

| Opmaak | Beschrijving |
|---|---|
| 'NOT a, b, c' | Als het aantal niet-geselecteerde waarden max_values of minder is, is de tekenreeks die wordt geretourneerd, een lijst met niet-geselecteerde waarden met NOT als een voorvoegsel.<br><br>De waarden worden gescheiden met value_sep als scheidingsteken. |
| 'x of y' | x = het aantal geselecteerde waarden<br><br>y = het totale aantal waarden<br><br>Dit wordt geretourneerd wanneer max_values < x < ( y - max_values). |
| 'ALL' | Geretourneerd als alle waarden zijn geselecteerd. |
| '-' | Geretourneerd als geen waarde is geselecteerd. |
| <search string> | Als u een selectie hebt gemaakt met behulp van een zoekopdracht, wordt de zoekreeks geretourneerd. |

Argumenten

| Argumenten | Beschrijving |
|---|---|
| **field_name** | Het veld met het bereik van gegevens die moeten worden gemeten. |
| **value_sep** | Het scheidingsteken dat tussen twee veldwaarden moet worden geplaatst. De standaardwaarde is ', '. |
| **max_values** | Het maximale aantal veldwaarden dat afzonderlijk moet worden vermeld. Als een groter aantal waarden wordt geselecteerd, wordt de notatie 'x van y waarden' gebruikt. De standaardwaarde is 6. |
| **state_name** | De naam van een alternatieve state die voor de specifieke visualisatie is gekozen. Als het argument **state_name** wordt gebruikt, wordt alleen rekening gehouden met de selecties die zijn gekoppeld aan de opgegeven statenaam. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `GetFieldSelections( FirstName )` | Retourneert '-' (NULL) als er geen selectie is gemaakt in het veld `FirstName`. Anders wordt de waarde van de selecties geretourneerd, bijvoorbeeld `John`. |

## Voorbeeld - Basisprincipes voor GetFieldSelections

Diagramuitdrukking

**Overzicht**

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `FirstName`
  - `LastName`
  - `Initials`
  - `HasCellphone`

### Load-script

```
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 '|');
```

### Resultaten

1. Laad de gegevens en open een werkblad. Maak een nieuw filtervak en voeg dit veld toe als dimensie:
   - `FirstName`
2. Voeg een KPI-object toe en maak dan de volgende meting:
   - `=GetFieldSelections(FirstName)`, om de waarden van de selectie in het veld `FirstName` te retourneren.

   *In dit voorbeeld is de opmaak van de KPI aangepast om een responsief lay-outgedrag te gebruiken, zodat de uitvoer gemakkelijker te bekijken is.*
3. Klik op **Werkblad bewerken** om over te schakelen naar de analysemodus en voer de volgende acties uit in de volgorde zoals ze verschijnen in de tabel:

Acties en resultaten in de analysemodus

| Actie | KPI-resultaten |
|---|---|
| Selecteer `John` in het filter `FirstName`. | De meting retourneert de waarde: `John`. |
| Selecteer `John` en `Peter` in het filter `FirstName`. | De meting retourneert de waarden: `John, Peter`. |
| Klik op **Werkblad bewerken** om over te schakelen naar de bewerkingsmodus en bewerk vervolgens de KPI- | De meting retourneert de waarden met de nieuwe `value_sep` indeling: `John; Peter`. |

| Actie | KPI-resultaten |
| --- | --- |
| metinguitdrukking als volgt: `=GetFieldSelections(FirstName,'; ')` | |
| Bewerk de KPI-metinguitdrukking als volgt: `=GetFieldSelections (FirstName,';', 2)` | De meting retourneert de waarden met de nieuwe `value_sep` indeling: `John;Peter`. |
| Klik op **Werkblad bewerken** om terug te schakelen naar de analysemodus. Wis de selecties van het filter `FirstName` en selecteer vervolgens `John`, `Sue` en `Mark`. | De meting retourneert `NOT Jane;Peter`, omdat het aantal selecties groter is dan het argument `max_value` van `2`. Als er geen `max_value` is ingesteld, worden alle geselecteerde namen weergegeven: `John; Sue; Mark`. |

De volgende afbeelding toont de resultaten van de laatste actie in de tabel waarbij het aantal selecties groter is dan het argument `max_values`.

*KPI-resultaten van de selectie van de voornamen John, Sue, en Mark*

In dit voorbeeld kunt u zien hoe de waarden die door de functie `GetFieldSelections` worden geretourneerd, veranderen naarmate er selecties worden gemaakt.

## Voorbeeld - Scenario met GetFieldSelections

Diagramuitdrukking

### Overzicht

De functie `GetFieldSelections` wordt gebruikt om namen van klanten te markeren wanneer ze worden geselecteerd.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `OrderID`
  - `CustomerName`

- Product
- Quantity

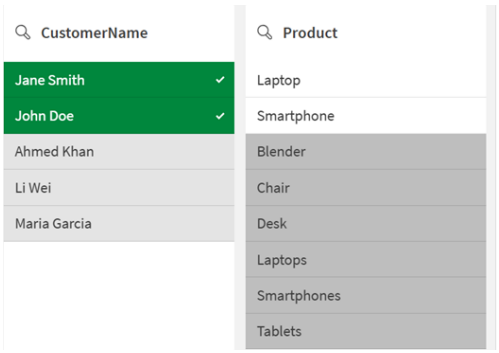
### Load-script

```
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
];
```

### Resultaten

1. Laad de gegevens en open een werkblad. Maak een nieuw filtervak en voeg deze velden toe als dimensies:
   - CustomerName
   - Product
2. Voeg een KPI-object toe en maak dan de volgende meting:
   - `=GetFieldSelections(CustomerName).`

   > *In dit voorbeeld is de opmaak van de KPI aangepast om een responsief lay-outgedrag te gebruiken, zodat de uitvoer gemakkelijker te bekijken is. .*
3. Klik op **Werkblad bewerken** om over te schakelen naar de analysemodus en selecteer vervolgens `Jane Smith` en `John Doe` in het filter `CustomerName`.
   De functie `GetFieldSelections` retourneert de waarden: `Jane Smith, John Doe`.



U kunt de geselecteerde klanten zien in het filtervak in de KPI.

## GetObjectDimension - diagramfunctie

**GetObjectDimension()** retourneert de naam van de dimensie. **Index** is een optioneel geheel getal dat de dimensie aangeeft die moet worden geretourneerd.

*U kunt deze functie niet gebruiken in een diagram in de volgende locaties: titel, subtitel, voettekst, uitdrukking referentielijn en uitdrukking min/max.*

*U kunt niet naar de naam van een dimensie of meting in een ander object verwijzen met behulp van de Object ID.*

**Syntaxis:**

```
GetObjectDimension ([index])
```

**Retourgegevenstypen:** Tekenreeks

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `GetObjectDimension ()`<br>`GetObjectDimension (0)` | Retourneert de naam van de eerste dimensie in het diagram. |
| `GetObjectDimension (1)` | Retourneert de naam van de tweede dimensie in het diagram. |

## Voorbeeld - Basisprincipes voor GetObjectDimension

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `TransactionDate`
  - `CustomerID`
  - `TransactionQuantity`

### Load-script

```
Example:
LOAD * INLINE [
TransactionDate, CustomerID, TransactionQuantity
2018/08/30, 049681, 13
2018/08/30, 203521, 6
2018/08/30, 203521, 21
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `TransactionDate`
- `CustomerID`
- `TransactionQuantity`

Maak de volgende metingen:

- `=GetObjectDimension ()`, om de eerste dimensie in de tabel te vinden.
- `=GetObjectDimension (0)`, om de eerste dimensie in de tabel te vinden.
- `=GetObjectDimension (1)`, om de tweede dimensie in de tabel te vinden.

Resultatentabel

| TransactionDate | 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 |

Als u naar de resultaten kijkt, kunt u zien dat de functie `GetObjectDimension` de naam van de dimensie retourneert die in de functieparameters is aangegeven.

## Voorbeeld - Geavanceerd scenario met GetObjectDimension

#### Diagramuitdrukking

#### Overzicht

Dit voorbeeld gebruikt de functie `GetObjectDimension` in combinatie met filterselecties. De dimensie- en metingberekening die in het diagramobject verschijnt, verandert op basis van het filter dat u selecteert.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die is geladen in twee inline-gegevenstabellen: `Sales` en `Filter`. De eerste tabel, `Sales`, bevat verkoopgegevens. De tweede tabel, `Filter`, bevat een lijst met veldnamen die gebruikt zullen worden om dimensies te wisselen. Met de wisselknop kunt u schakelen tussen dimensies in een diagram en de metingberekening aanpassen die verschijnt afhankelijk van de geselecteerde dimensie.
- De volgende velden in de tabel `Sales`:

    - `Country`
    - `Salesperson`
    - `SalesValue`
- Het volgende veld in de tabel `Filter`: `FilterField`.

### Load-script

```
Sales:
LOAD * INLINE [
Country, SalesPerson, SalesValue
USA, John, 500
USA, Alice, 700
Canada, Bob, 300
Canada, Carol, 400
Mexico, Dave, 200
];

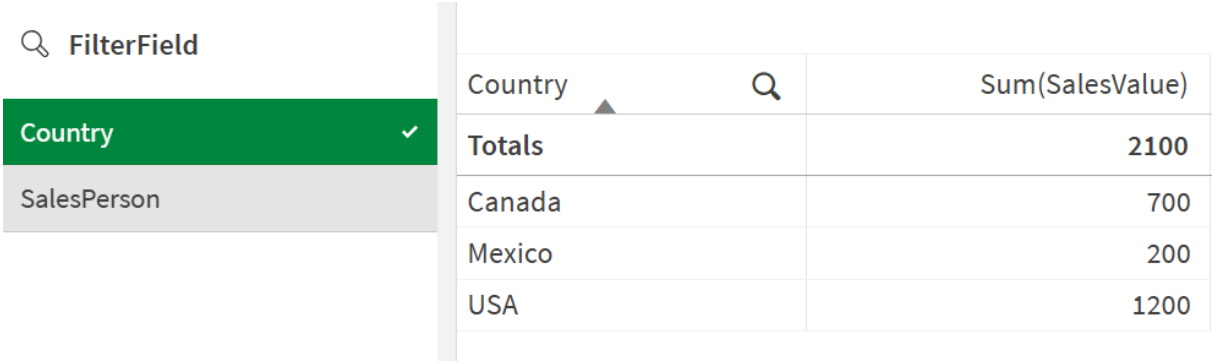
Filter:
LOAD * INLINE [
FilterField
Country
SalesPerson
];
```

### Resultaten

1. Laad de gegevens en open een werkblad. Maak een nieuw filtervak en voeg dit veld toe als dimensie:
    - `FilterField`
2. Klik in het bedrijfsmiddelenvenster onder **Velden** met de rechtermuisknop op **Filterveld** en selecteer **Veldinstellingen**. Selecteer het selectievakje **Altijd één geselecteerde waarde** en klik op **Opslaan**.
3. Maak een tabel en voeg een dimensie toe door de volgende uitdrukking in te voeren:
    - `=$(= FilterField)`
    - Voer onder **Label** de volgende uitdrukking in om de kolomnaam voor de dimensie te definiëren: `='$(= FilterField)'`
4. Maak de volgende meting:
    - `=If(GetObjectDimension() = 'Country', Sum(SalesValue), Avg(SalesValue))`, om de verkoopwaarden bij elkaar op te tellen als de geselecteerde dimensie `Country` is, anders om het gemiddelde van de verkoopwaarde te retourneren als de geselecteerde dimensie `SalesPerson` is.
    - Voer onder `Label` de volgende uitdrukking in om de kolomnaam voor de meting te definiëren op basis van de filterselectie: `=If(GetObjectDimension() = 'Country', 'Sum (SalesValue)', 'Avg(SalesValue)')`

Wanneer u in de analysemodus schakelt tussen `SalesPerson` en `Country` in het filter `FieldFilter`, verandert de tabel om de geselecteerde dimensie en de bijbehorende metingberekening voor die dimensie op te nemen. Als u bijvoorbeeld `Country` selecteert in het filter, dan toont de tabel `Country` als dimensie in de eerste kolom. De meting gebruikt dan de functie `GetObjectDimension`, stelt dit gelijk aan `Country`, en retourneert de `Sum(SalesValue)`.

*Filtervak en tabelresultaten voor Country*




De volgende tabel toont de resultaten wanneer u `Country` selecteert in het filter `FilterField`.

Resultatentabel voor Country

| Country | Sum(SalesValue) |
|---|---|
| **Totals** | **2100** |
| Canada | 700 |
| Mexico | 200 |
| USA | 1200 |

De volgende tabel toont de resultaten wanneer u `SalesPerson` selecteert in het filter `FilterField`.

Resultatentabel voor SalesPerson

| SalesPerson | Avg(SalesValue) |
|---|---|
| **Totals** | **420** |
| Alice | 700 |
| Bob | 300 |
| Carol | 400 |
| Dave | 200 |
| John | 500 |

## GetObjectField - diagramfunctie

**GetObjectField()** retourneert de velduitdrukking van de dimensie. **Index** is een optioneel geheel getal dat de dimensie aangeeft die moet worden geretourneerd.

*U kunt deze functie niet gebruiken in een diagram in de volgende locaties: titel, subtitel, voettekst, uitdrukking referentielijn en uitdrukking min/max.*

*U kunt niet naar de naam van een dimensie of meting in een ander object verwijzen met behulp van de Object ID.*

**Syntaxis:**

```
GetObjectField ([index])
```

**Retourgegevenstypen:** tekenreeks

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `GetObjectField ()`<br>`GetObjectField (0)` | Retourneert de velduitdrukking van de eerste dimensie in het diagram. |
| `GetObjectField (1)` | Retourneert de velduitdrukking van de tweede dimensie in het diagram. |

Als u de naam van een meting wilt retourneren, gebruikt u de functie **GetObjectMeasure**.

## Voorbeeld - Basisprincipes voor GetObjectField

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `TransactionDate`
  - `CustomerID`
  - `TransactionQuantity`

### Load-script

```
Example:
LOAD * INLINE [
TransactionDate, CustomerID, TransactionQuantity
2018/08/30, 049681, 13
2018/08/30, 203521, 6
2018/08/30, 203521, 21
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `TransactionDate`
- `CustomerID`
- `TransactionQuantity`

Maak de volgende metingen:

- `=GetObjectField()`, om de eerste dimensie in de tabel te vinden.
- `=GetObjectField(0)`, om de eerste dimensie in de tabel te vinden.
- `=GetObjectField(1)`, om de tweede dimensie in de tabel te vinden.

Resultatentabel

| 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 |

Als u naar de resultaten kijkt, kunt u zien dat de functie `GetObjectField` de naam van het veld retourneert dat in de functieparameters is aangegeven.

## Voorbeeld - Geavanceerd scenario met GetObjectField

#### Diagramuitdrukking

#### Overzicht

Dit voorbeeld gebruikt de functie `GetObjectField` in combinatie met filterselecties. De dimensie- en metingberekening die in het diagramobject verschijnt, verandert op basis van het filter dat u selecteert.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die is geladen in twee inline-gegevenstabellen: `Sales` en `Filter`. De eerste tabel, `Sales`, bevat verkoopgegevens. De tweede tabel, `Filter`, bevat een lijst met veldnamen die gebruikt zullen worden om dimensies te wisselen. Met de wisselknop kunt u schakelen tussen dimensies in een diagram en de metingberekening aanpassen die verschijnt afhankelijk van de geselecteerde dimensie.
- De volgende velden in de tabel `Sales`:

    - Country
    - Salesperson
    - SalesValue
- Het volgende veld in de tabel Filter: FilterField.

### Load-script

```
Sales:
LOAD * INLINE [
Country, SalesPerson, SalesValue
USA, John, 500
USA, Alice, 700
Canada, Bob, 300
Canada, Carol, 400
Mexico, Dave, 200
];

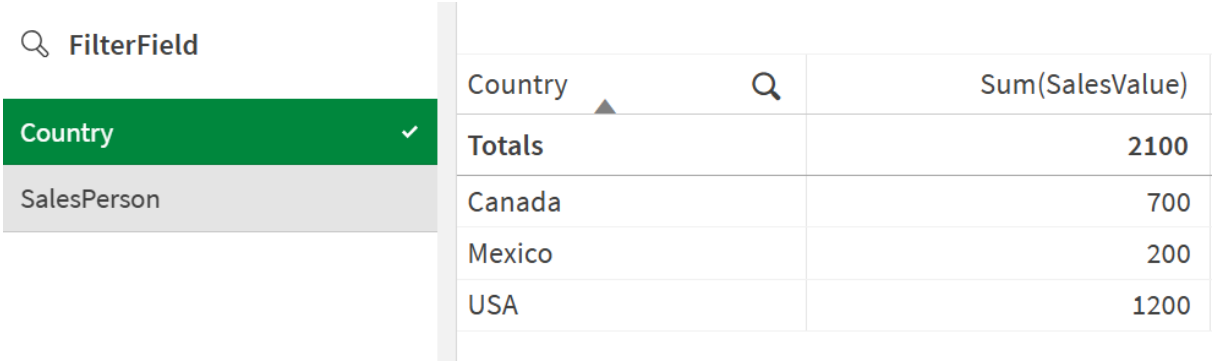
Filter:
LOAD * INLINE [
FilterField
Country
SalesPerson
];
```

### Resultaten

1. Laad de gegevens en open een werkblad. Maak een nieuw filtervak en voeg dit veld toe als dimensie:
    - FilterField
2. Klik in het bedrijfsmiddelenvenster onder **Velden** met de rechtermuisknop op **Filterveld** en selecteer **Veldinstellingen**. Selecteer het selectievakje **Altijd één geselecteerde waarde** en klik op **Opslaan**.
3. Maak een tabel en voeg een dimensie toe door de volgende uitdrukking in te voeren:
    - =$(= FilterField)
    - Voer onder **Label** de volgende uitdrukking in om de kolomnaam voor de dimensie te definiëren: ='$(= FilterField)'
4. Maak de volgende meting:
    - =If(GetObjectField() = 'Country', Sum(SalesValue), Avg(SalesValue)), om de verkoopwaarden bij elkaar op te tellen als de geselecteerde dimensie Country is, anders om het gemiddelde van de verkoopwaarde te retourneren als de geselecteerde dimensie SalesPerson is.
    - Voer onder Label de volgende uitdrukking in om de kolomnaam voor de meting te definiëren die wordt weergegeven voor de meting op basis van de filterselectie: =If(GetObjectField() = 'Country', 'Sum(SalesValue)', 'Avg(SalesValue)') Als bijvoorbeeld Country in het filter is geselecteerd, zal de naam van de meting de waarde Sum(SalesValue) retourneren.

Wanneer u in de analysemodus schakelt tussen `SalesPerson` en `Country` in het filter `FieldFilter`, verandert de tabel om de geselecteerde dimensie en de bijbehorende metingberekening voor die dimensie op te nemen. Als u bijvoorbeeld `Country` selecteert in het filter, dan toont de tabel `Country` als dimensie in de eerste kolom. De meting gebruikt dan de functie `GetObjectField`, stelt dit gelijk aan `Country`, en retourneert de `Sum(SalesValue)`.

*Filtervak en tabelresultaten voor Country*




De volgende tabel toont de resultaten wanneer u `Country` selecteert in het filter `FilterField`.

Resultatentabel voor `Country`

| Country | Sum(SalesValue) |
|---|---|
| Canada | 700 |
| Mexico | 200 |
| USA | 1200 |

De volgende tabel toont de resultaten wanneer u `SalesPerson` selecteert in het filter `FilterField`.

Resultatentabel voor `SalesPerson`

| SalesPerson | Avg(SalesValue) |
|---|---|
| Alice | 700 |
| Bob | 300 |
| Carol | 400 |
| Dave | 200 |
| John | 500 |

## GetObjectMeasure - diagramfunctie

**GetObjectMeasure()** retourneert de naam van de meting. **Index** is een optioneel geheel getal dat de meting aangeeft die moet worden geretourneerd.

*U kunt deze functie niet gebruiken in een diagram in de volgende locaties: titel, subtitel, voettekst, uitdrukking referentielijn en uitdrukking min/max.*

*U kunt niet naar de naam van een dimensie of meting in een ander object verwijzen met behulp van de Object ID.*

**Syntaxis:**

```
GetObjectMeasure ([index])
```

**Retourgegevenstypen:** Tekenreeks

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `GetObjectMeasure ()`<br>`GetObjectMeasure (0)` | Retourneert de naam van de eerste meting in het diagram. |
| `GetObjectMeasure (1)` | Retourneert de naam van de tweede meting in het diagram. |

## Voorbeeld - Basisprincipes voor GetObjectMeasure

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `CustomerID`
  - `TransactionQuantity`

### Load-script

```
Example:
LOAD * INLINE [
CustomerID, TransactionQuantity
049681, 13
203521, 6
203521, 21
];
```

**Resultaten**

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `CustomerID`

Maak de volgende metingen:

- `Sum(TransactionQuantity)` - eerste meting.
- `Avg(TransactionQuantity)` - tweede meting
- `=GetObjectMeasure()`, om de eerste meting in de tabel te vinden.
- `=GetObjectMeasure(0)`, om de eerste meting in de tabel te vinden.
- `=GetObjectMeasure(1)`, om de tweede meting in de tabel te vinden.

Resultatentabel

| CustomerID | Sum (TransactionQuantity) | Avg (TransactionQuantity) | GetObjectMeasure () | GetObjectMeasure (0) | GetObjectMeasure (1) |
|---|---|---|---|---|---|
| **Totals** | **40** | **13.333333** | **Sum (TransactionQuantity)** | **Sum (TransactionQuantity)** | **Avg (TransactionQuantity)** |
| 049681 | 13 | 13 | =Sum (TransactionQuantity) | =Sum (TransactionQuantity) | =Avg (TransactionQuantity) |
| 203521 | 27 | 13.5 | =Sum (TransactionQuantity) | =Sum (TransactionQuantity) | =Avg (TransactionQuantity) |

Als u naar de resultaten kijkt, kunt u zien dat de functie `GetObjectMeasure` de meting retourneert die in de functieparameters is aangegeven.

## GetPossibleCount - diagramfunctie

**GetPossibleCount()** wordt gebruikt om het aantal mogelijke waarden in het vastgestelde veld te zoeken. Als het vastgestelde veld selecties bevat, worden de geselecteerde (groene) velden geteld. Anders worden de geassocieerde (witte) waarden geteld.

Voor velden met selecties retourneert **GetPossibleCount()** het aantal geselecteerde (groene) velden.

**Syntaxis:**

```
GetPossibleCount (field_name)
```

**Retourgegevenstypen:** geheel getal

*De kleuren die worden gebruikt in de selectiebalk en voor elke selectiestatus kunnen worden aangepast met een aangepast thema. Als u werkt met een app die een aangepast thema gebruikt, merkt u mogelijk dat uw selecties niet worden weergegeven met dezelfde kleuren die worden beschreven in dit Help-onderwerp.*

Argumenten

| Argumenten | Beschrijving |
|---|---|
| **field_name** | Het veld met het bereik van gegevens die moeten worden gemeten. |

In de volgende tabel staan andere functies die aan deze functie gerelateerd zijn.

Gerelateerde functies

| Functie | Interactie |
|---|---|
| *GetStateCounts - diagramfunctie (page 1356)* | Met **GetStateCounts()** kunt u de berekening van de volgende aantallen combineren met één enkele functieaanroep:<br>• Aantal geselecteerde opgenomen waarden.<br>• Aantal mogelijke waarden.<br>• Aantal alternatieve waarden.<br>• Aantal uitgesloten waarden, exclusief alternatieve en geselecteerde opgenomen waarden.<br>• Aantal geselecteerde uitgesloten waarden. |
| *GetSelectedCount - diagramfunctie (page 1351)* | Retourneert het aantal geselecteerde opgenomen waarden. |
| *GetAlternativeCount - diagramfunctie (page 1317)* | Retourneert het aantal alternatieve waarden. |
| *GetPossibleCount - diagramfunctie (page 1347)* | Retourneert het aantal uitgesloten waarden, exclusief alternatieve en geselecteerde opgenomen waarden. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `GetPossibleCount( Initials )` | Als er geen selecties worden gemaakt, wordt het totale aantal unieke waarden in het veld `Initials` geretourneerd. Als er één waarde in dit veld geselecteerd is, dan wordt er 1 geretourneerd. Als er een selectie wordt gemaakt in een ander veld, dan is de geretourneerde waarde het aantal unieke waarden in het veld `Initials` dat betrekking heeft op de waarden die in het tweede veld zijn geselecteerd. |

## Voorbeeld - Basisprincipes voor GetPossibleCount

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `FirstName`
  - `LastName`
  - `Initials`
  - `HasCellphone`

### Load-script

```
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 '|');
```

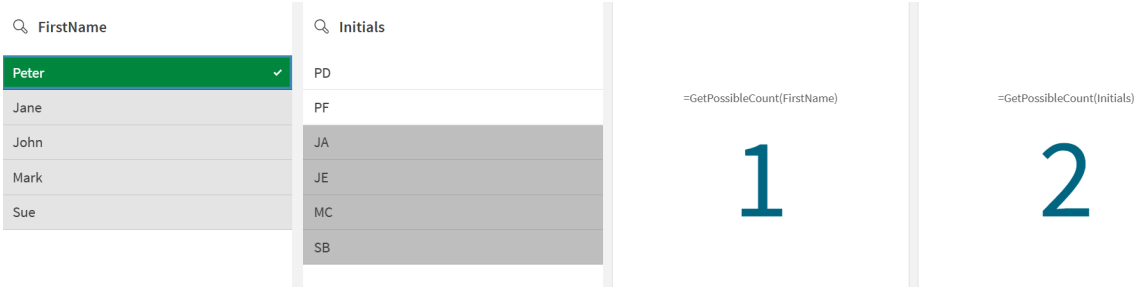
### Resultaten

1. Laad de gegevens en open een werkblad. Maak een nieuw filtervak en voeg deze velden toe als dimensies:
   - `FirstName`
   - `Initials`
2. Maak twee KPI-objecten, één voor elk van de volgende metinguitdrukkingen:
   - KPI 1: `=GetPossibleCount(FirstName)` om het aantal mogelijke unieke waarden in het veld `FirstName` te retourneren.
   - KPI 2: `=GetPossibleCount(Initials)` om het aantal mogelijke unieke waarden in het veld `Initials` te retourneren.

   Elke KPI retourneert het aantal mogelijke unieke waarden. Zonder selecties in het filter retourneert KPI 1 de waarde `5` en KPI 2 de waarde `6`.

   *In dit voorbeeld is de opmaak van de KPI aangepast om een responsief lay-outgedrag te gebruiken, zodat de uitvoer gemakkelijker te bekijken is.*

3. Klik op **Werkblad bewerken** om over te schakelen naar de analysemodus en selecteer `John` in het filter `FirstName`.
   Beide KPI's retourneren de waarde `1` omdat er één waarde is geselecteerd in het veld `FirstName` en `JA` de enige bijbehorende waarde is in het veld `Initials`.
4. Wis de filerselectie `FirstName` en selecteer vervolgens `Peter`.
   KPI 1 retourneert de waarde `1` omdat er one mogelijke unieke waarde beschikbaar zijn in het veld `FirstName`. KPI 2 retourneert `2` omdat er two mogelijke unieke waarden beschikbaar zijn in het veld `Initials`.




## Voorbeeld 5 – Scenario met GetPossibleCount

Diagramuitdrukking

### Overzicht

Een manager wil de gegevens van werknemers filteren om te bepalen welke werknemers ervaring hebben in verschillende vaardigheidsgebieden.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Project`
  - `Employee`
  - `Skill`

### Load-script

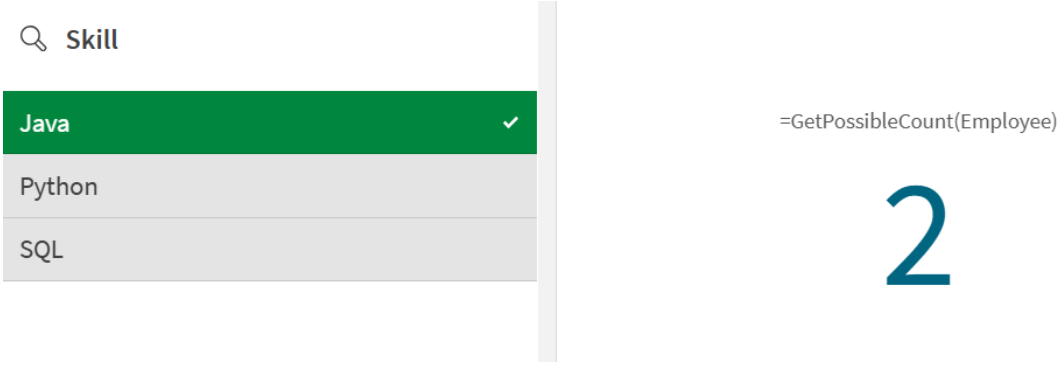
```
Example:
LOAD * INLINE [
Project, Employee, Skill
Project X, John, Java
Project X, Mary, SQL
Project Y, James, Python
Project Y, Linda, Java
Project Z, Robert, SQL
];
```

**Resultaten**

1. Laad de gegevens en open een werkblad. Maak een nieuw filtervak en voeg dit veld toe als dimensie:
   - `Skill`
2. Maak een KPI-object en voeg de volgende meting toe:
   - `=GetPossibleCount(Employee)` om het aantal werknemers te bepalen.

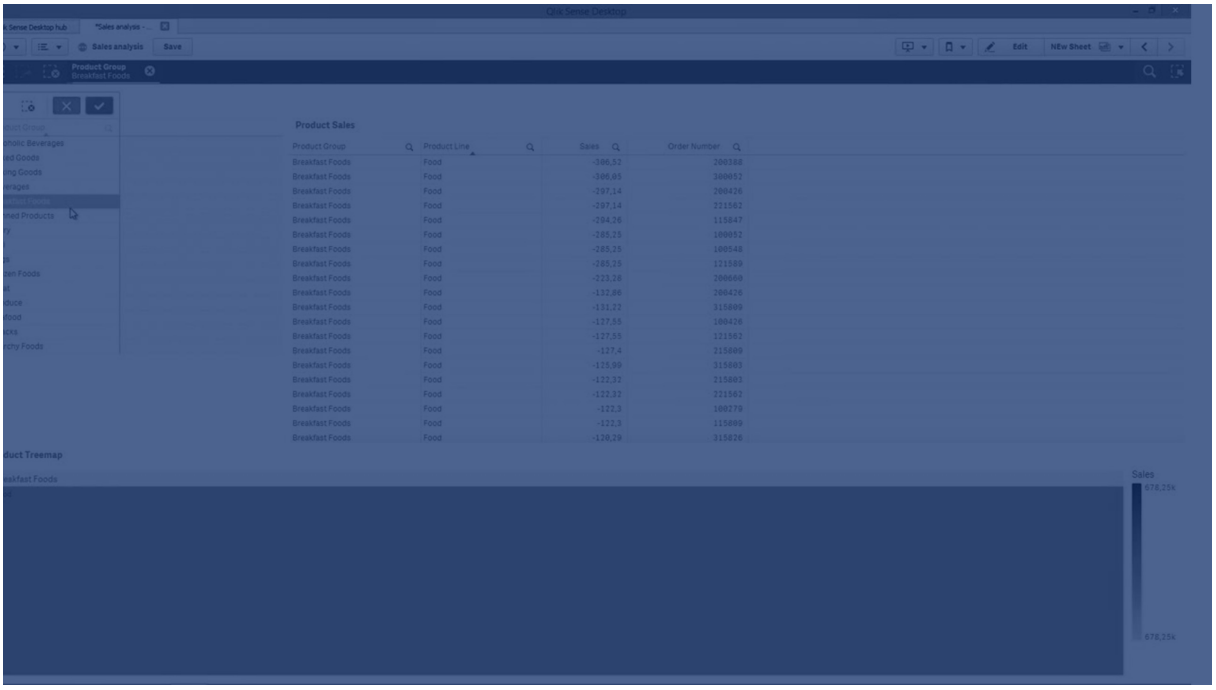
   Zonder selecties in het filtervak retourneert de KPI de waarde `5` omdat er five werknemers zijn.

   > *In dit voorbeeld is de opmaak van de KPI aangepast om een responsief lay-outgedrag te gebruiken, zodat de uitvoer gemakkelijker te bekijken is.*
3. Klik op **Werkblad bewerken** om over te schakelen naar de analysemodus en selecteer `Java` in het filter `Skill`.
   De KPI geeft de waarde `2` omdat er two werknemers zijn die de vaardigheid `Java` hebben.




## GetSelectedCount - diagramfunctie

**GetSelectedCount()** retourneert het aantal geselecteerde (groene) waarden in een veld.




**Syntaxis:**

```
GetSelectedCount (field_name [, include_excluded [, state_name]])
```

**Retourgegevenstypen:** geheel getal

> *De kleuren die worden gebruikt in de selectiebalk en voor elke selectiestatus kunnen worden aangepast met een aangepast thema. Als u werkt met een app die een aangepast thema gebruikt, merkt u mogelijk dat uw selecties niet worden weergegeven met dezelfde kleuren die worden beschreven in dit Help-onderwerp.*

Argumenten

| Argumenten | Beschrijving |
|---|---|
| **field_name** | Het veld met het bereik van gegevens die moeten worden gemeten. |
| **include_ excluded** | Als de parameter is ingesteld op **True()**, worden ook geselecteerde waarden meegeteld die momenteel worden uitgesloten door selecties in andere velden. Als de parameter is ingesteld op False of wordt weggelaten, worden deze waarden niet opgenomen. |
| **state_name** | De naam van een alternatieve state die voor de specifieke visualisatie is gekozen. Als het argument **state_name** wordt gebruikt, wordt alleen rekening gehouden met de selecties die zijn gekoppeld aan de opgegeven statenaam. |

In de volgende tabel staan andere functies die aan deze functie gerelateerd zijn.

Gerelateerde functies

| Functie | Interactie |
|---|---|
| *GetStateCounts - diagramfunctie (page 1356)* | Met **GetStateCounts()** kunt u de berekening van de volgende aantallen combineren met één enkele functieaanroep:<br>• Aantal geselecteerde opgenomen waarden.<br>• Aantal mogelijke waarden.<br>• Aantal alternatieve waarden.<br>• Aantal uitgesloten waarden, exclusief alternatieve en geselecteerde opgenomen waarden.<br>• Aantal geselecteerde uitgesloten waarden. |
| *GetPossibleCount - diagramfunctie (page 1347)* | Retourneert het aantal mogelijke waarden. |
| *GetAlternativeCount - diagramfunctie (page 1317)* | Retourneert het aantal alternatieve waarden. |
| *GetSelectedCount - diagramfunctie (page 1351)* | Retourneert het aantal uitgesloten waarden, exclusief alternatieve en geselecteerde opgenomen waarden. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `GetSelectedCount (FirstName)` | Als er geen selecties worden gemaakt, wordt `0` geretourneerd. Als er selecties zijn gemaakt in het veld `FirstName`, dan wordt het aantal geselecteerde items geretourneerd. |
| `GetSelectedCount (FirstName, True ())` | Als er geen selecties worden gemaakt, wordt `0` geretourneerd. Als er selecties zijn gemaakt in het veld `FirstName`, dan wordt het aantal geselecteerde items geretourneerd, ongeacht eventuele latere selecties in andere velden die de beschikbare waarden in het veld `FirstName` zouden kunnen verkleinen. |

## Voorbeeld - Basisprincipes voor GetSelectedCount

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `FirstName`
  - `LastName`

- `Initials`
- `HasCellphone`

### Load-script

```
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 '|');
```

### Resultaten

1. Laad de gegevens en open een werkblad. Maak een nieuw filtervak en voeg deze velden toe als dimensies:
   - `FirstName`
   - `Initials`
2. Maak twee KPI-objecten, één voor elk van de volgende metinguitdrukkingen:
   - KPI 1: `=GetSelectedCount(FirstName)` om het aantal geselecteerde items in het veld `FirstName` te retourneren.
   - KPI 2: `=GetSelectedCount(Initials)` om het aantal geselecteerde items in het veld `Initials` te retourneren.

   Elke KPI retourneert het aantal geselecteerde items in elk veld. Zonder selecties in de filters retourneren beide KPI's de waarde `0`.

   > *In dit voorbeeld is de opmaak van de KPI aangepast om een responsief lay-outgedrag te gebruiken, zodat de uitvoer gemakkelijker te bekijken is.*
3. Klik op **Werkblad bewerken** om over te schakelen naar de analysemodus en selecteer vervolgens `John` in het filter `FirstName`.
   KPI 1 retourneert de waarde `1` omdat er één item geselecteerd is in het filter `FirstName`. KPI 2 retourneert nog steeds `0` omdat het filter `Initials` geen geselecteerde items heeft.

## Voorbeeld - GetSelectedCount met include_excluded parameter

Diagramuitdrukking

### Overzicht

Dit voorbeeld gebruikt dezelfde gegevensverzameling als het vorige voorbeeld, maar toont het verschil in resultaten wanneer de parameter `include_excluded` is ingesteld op `True()`.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `FirstName`
  - `LastName`
  - `Initials`
  - `HasCellphone`

### Load-script

```
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 '|');
```

### Resultaten

1. Laad de gegevens en open een werkblad. Maak een nieuw filtervak en voeg deze velden toe als dimensies:
   - `Initials`
   - `HasCellPhone`
2. Maak twee KPI-objecten, één voor elk van de volgende metinguitdrukkingen:
   - KPI 1: `=GetselectedCount(Initials)` om het aantal geselecteerde waarden in het veld `Initials` te bepalen.
   - KPI 2: `=GetSelectedCount([Initials], True())` om ook geselecteerde waarden op te nemen die momenteel worden uitgesloten door selecties in andere velden.

   Zonder selecties in het filtervak, retourneert elke KPI de waarde `0`.

> *In dit voorbeeld is de opmaak van de KPI aangepast om een responsief lay-outgedrag te gebruiken, zodat de uitvoer gemakkelijker te bekijken is.*

3. Klik op **Werkblad bewerken** om over te schakelen naar de analysemodus. Selecteer alle waarden in het filter `Initials` en selecteer vervolgens `Yes` in het filter `HasCellphone`.

Wanneer u de resultaten van elke KPI vergelijkt, kunt u zien dat elke KPI een andere waarde oplevert, afhankelijk van of de parameter `include_excluded` is ingesteld op `True()` of niet. KPI 2 retourneert de waarde `6` omdat het alle geselecteerde waarden in het filter `Initials` omvat, ook al heeft de selectie in het filter `HasCellphone` de beschikbare items gereduceerd tot `4`. KPI 1 retourneert `4` omdat de geselecteerde uitgesloten waarden niet worden geteld in `Initials`.

## GetStateCounts - diagramfunctie

De **GetStateCounts()** diagramfunctie wordt gebruikt om het totale aantal unieke waarden te berekenen dat overeenkomt met de opgegeven selectiestatussen.

Met **GetStateCounts()** kunt u de berekeningen van de volgende functies combineren in één enkele functieaanroep: **GetSelectedCount()**, **GetNotSelectedCount()**, **GetAlternativeCount()**, **GetPossibleCount()** en **GetExcludedCount()**. Het aantal geselecteerde uitgesloten waarden kan ook aan de berekening worden toegevoegd. U kunt opgeven of elke functieberekening een waarde optelt of aftrekt van het geretourneerde totaal.

**Syntaxis:**

```
GetStateCounts (field_name, state_name [, state_type1,...state_typeN])
```

**Retourgegevenstypen:** geheel getal

Argumenten

| Argument | Beschrijving |
|---|---|
| **field_name** | Het veld waarvoor u de selectiestatus berekent. Een niet-bestaande veldnaam leidt tot het resultaat null. |
| **state_name** | De naam van de alternatieve states. Als het argument leeg (`''`) of null is, wordt de overgenomen alternatieve state gebruikt. Gebruik `$` om expliciet de standaardstate te gebruiken. Een benoemde (niet-lege) statenaam die niet overeenkomt met een bestaande state resulteert in de uitkomst null. |

| Argument | Beschrijving |
|---|---|
| **state_type** | Een lijst van één of meer statetypen voor de veldwaarde. Deze statetypen worden samengevoegd tot een telling. Een statetype wordt gespecificeerd met behulp van een sleutel. Voer elke sleutel in tussen enkele aanhalingstekens.<br><br>Als dit argument wordt weggelaten, retourneert de functie een tekenreeks met alle beschikbare statetellingen voor het veld, in dezelfde volgorde als de opsomming.<br><br>Zie onderstaande tabel voor een lijst van states die u kunt gebruiken. |

Er wordt naar statetypen verwezen met behulp van specifieke sleutels. U kunt de numerieke of tekstuele versie van de sleutel gebruiken. Combineer meer dan één sleutel in dezelfde uitdrukking om het resultaat verder aan te passen. U kunt het aantal states aftrekken van het totaal, in plaats van het toe te voegen. Gebruik hiervoor de tekstsleutel en zet een minteken (-) voor het statetype.

Het statetype en, indien van toepassing, het minteken moeten tussen enkele aanhalingstekens staan.

Sleutels voor elk veld statetype

| Statetype Veld | Beschrijving | Numerieke sleutel | Tekstuele sleutel |
|---|---|---|---|
| Geselecteerd | Bevat geselecteerde waarden in de berekening. Zie *GetSelectedCount - diagramfunctie (page 1351)* voor de soortgelijke functie. | 1 | S |
| Optioneel | Bevat optionele (niet geselecteerde, maar wel te selecteren) waarden in de berekening. Zie *GetPossibleCount - diagramfunctie (page 1347)* voor de soortgelijke functie. | 2 | O |
| Selectie opgeheven | Bevat niet-geselecteerde waarden in de berekening. Dit statetype is alleen beschikbaar als het veld in de AND-modus staat. | 3 | D |
| Alternatief | Bevat alternatieve waarden in de berekening. Zie *GetAlternativeCount - diagramfunctie (page 1317)* voor de soortgelijke functie. | 4 | A |
| Uitgesloten | Bevat uitgesloten (niet-geselecteerde) waarden in de berekening. Zie *GetExcludedCount - diagramfunctie (page 1328)* voor de soortgelijke functie. | 5 | X |
| Geselecteerde uitgesloten | Bevat geselecteerde uitgesloten waarden in de berekening. | 6 | XS |

## Wanneer gebruiken

Door **GetStateCounts()** kunt u een aangepaste selectiestatus berekenen. Met deze functie kunt u meerdere functieaanroepen consolideren in één functieaanroep, waardoor het proces van het schrijven van uw uitdrukking wordt vereenvoudigd.

U moet bijvoorbeeld het totale aantal uitgesloten, alternatieve en geselecteerde waarden berekenen voor een veld. U kunt **GetStateCounts()** gebruiken om dit totaal te berekenen.

Voorbeelden: diagramuitdrukkingen

| Voorbeelden | Resultaten |
|---|---|
| `=GetStateCounts( ProductName, Null(), 'S' )` | Retourneert het geselecteerde aantal voor `ProductName`, in de overgenomen alternatieve state. |
| `=GetStateCounts( ProductName, '', 'X', 'A', 'XS' )` | Geeft het totale aantal uitgesloten, geselecteerde waarden en alternatieve waarden voor `ProductName`. De overgenomen alternatieve state wordt gebruikt. |
| `=GetStateCounts( ProductName, '', 'S', 'XS' )` | Retourneert het totale aantal gebruikersselecties voor `ProductName`, in de overgenomen alternatieve state. |
| Gegeven dat het veld *ProductName* in and-modus staat.<br>`=GetStateCounts (ProductName, '', 'D', '-O')` | Retourneert het aantal niet-geselecteerde waarden, afgetrokken van het aantal mogelijke waarden, voor `ProductName`. De overgenomen alternatieve state wordt gebruikt. |
| `=GetStateCounts (ProductName, '', 'X', , 'A', 'XS')` | Geeft het totale aantal uitgesloten, geselecteerde waarden en alternatieve waarden voor `ProductName`. De overgenomen alternatieve state wordt gebruikt. |
| `=GetStateCounts (ProductName, '$', 'O')` | Retourneert het mogelijke aantal voor `ProductName`, in de standaard alternatieve state. |
| `=GetStateCounts (ProductName, 'StateA', 'S')` | Retourneert het geselecteerde aantal van `ProductName`, in de alternatieve state met de naam `StateA`. |

## Voorbeeld 1 - Het totale aantal gebruikersselecties tellen (inclusief geselecteerde uitgesloten waarden)

Diagramuitdrukking

### Overzicht

De functie **GetSelectedCount()** retourneert het totale aantal geselecteerde waarden, exclusief het totale aantal geselecteerde uitgesloten waarden. Met **GetExcludedCount()** kunt u het aantal geselecteerde uitgesloten waarden opnemen in de berekening van het diagram.

Open de Editor voor laden van gegevens en voeg het volgende load-script toe aan een nieuwe sectie.

### Load-script

```
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 '|');
```

### Resultaten

### Doe het volgende:

1. Laad de gegevens en open een werkblad.
2. Maak twee filtervakken.
3. Voeg **First name** toe als dimensie in het eerste filtervak. Voeg **Last name** toe als dimensie in het tweede filtervak.
4. Maak vervolgens een KPI met deze meting:
   `=GetStateCounts([First Name],'', 'S','XS')`
5. Selecteer in het filtervak **First name** de waarden **Mark** en **Peter**. Selecteer in het filtervak **Last name** de waarden **Carr**.
6. Merk op dat de KPI een waarde laat zien van 2. Dit geeft aan dat alle waarden die u hebt geselecteerd, inclusief de geselecteerde uitgesloten waarde **Peter**, worden geteld.

## Voorbeeld 2 - Combinatie van uitgesloten, geselecteerde uitgesloten en alternatieve tellingen

Diagramuitdrukkingen

### Overzicht

De functie **GetExcludedCount()** retourneert het totale aantal unieke waarden die niet zijn geselecteerd en uitgesloten. Als u geselecteerde uitgesloten waarden en alternatieve waarden wilt opnemen in uw visualisatie, kunt u de functie **GetStateCounts()** als volgt gebruiken.

Open de Editor voor laden van gegevens en voeg het volgende load-script toe aan een nieuwe sectie.

### Load-script

```
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 '|');
```

### Resultaten

### Doe het volgende:

1. Laad de gegevens en open een werkblad.
2. Maak twee filtervakken.
3. Voeg **First name** toe als dimensie in het eerste filtervak. Voeg **Last name** toe als dimensie in het tweede filtervak.
4. Maak vervolgens een KPI met deze meting:
   `=GetStateCounts([First name], '', 'X', 'A', 'XS')`
5. Selecteer in het filtervak **First name** de waarden **John** en **Peter**. Selecteer in het filtervak **Last name** de waarden **Franc**.
6. U ziet dat de KPI een waarde van 4 laat zien. Dit geeft aan dat de geselecteerde uitgesloten waarde **John** is opgenomen in de telling, naast de telling van de uitgesloten waarden.
7. Bewerk de KPI en vervang de bestaande meting door het volgende:
   `=GetStateCounts([Last name], '', 'X', 'A', 'XS')`
8. Behoud de bestaande selecties en selecteer de waarde **Anderson** in het filtervak **Last name**.
9. U ziet dat de KPI nog steeds een waarde van 4 laat zien. Dit geeft aan dat de alternatieve waarde **Devonshire** is opgenomen in de telling, naast de telling van de uitgesloten waarden.

## 8.10 Bestandsfuncties

De bestandsfuncties (alleen beschikbaar in scriptuitdrukkingen) retourneren informatie over het tabelbestand dat op dat moment wordt gelezen. Deze functies retourneren een voor alle gegevensbronnen, behalve tabelbestanden (uitzondering: ).NULL ConnectString

### Overzicht van bestandsfuncties

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

#### Attribute

Deze scriptfunctie retourneert de waarde van de metatags van verschillende mediabestanden als tekst. De volgende bestandsindelingen worden ondersteund: MP3, WMA, WMV, PNG en JPG. Als het bestand **filename** niet bestaat, niet een ondersteunde bestandsindeling heeft of niet een metatag bevat met de naam **attributename**, wordt NULL geretourneerd.

```
Attribute (filename, attributename)
```

#### ConnectString

De functie **ConnectString()** retourneert de naam van de actieve gegevensverbinding voor ODBC- of OLE DB-verbindingen. De functie retourneert een lege tekenreeks als geen **connect**-opdracht is uitgevoerd of na een **disconnect**-opdracht.

```
ConnectString ()
```

#### FileBaseName

De functie **FileBaseName** retourneert een tekenreeks met de naam van het tabelbestand dat op dat moment wordt gelezen, zonder pad of extensie.

```
FileBaseName ()
```

#### FileDir

De functie **FileDir** retourneert een tekenreeks met het pad naar de map van het tabelbestand dat op dat moment wordt gelezen.

```
FileDir ()
```

#### FileExtension

De functie **FileExtension** retourneert een tekenreeks met de extensie van het tabelbestand dat op dat moment wordt gelezen.

```
FileExtension ()
```

#### FileName

De functie **FileName** retourneert een tekenreeks met de naam van het tabelbestand dat op dat moment wordt gelezen, zonder pad maar met de extensie.

```
FileName ()
```

### FilePath

De functie **FilePath** retourneert een tekenreeks met het volledige pad naar het tabelbestand dat op dat moment wordt gelezen.

```
FilePath ()
```

### FileSize

De functie retourneert een geheel getal met de grootte in bytes van het bestand of, als geen is opgegeven, van het tabelbestand dat op dat moment wordt gelezen.filenamefilename

```
FileSize ()
```

### FileTime

De functie **FileTime** geeft een tijdstempel in UTC-indeling van de laatste wijziging van een opgegeven bestand. Als er geen bestand is opgegeven, geeft de functie een tijdstempel in UTC-indeling van de laatste wijziging van het momenteel gelezen tabelbestand.

```
FileTime ([ filename ])
```

### GetFolderPath

De functie **GetFolderPath** retourneert de waarde van de functie Microsoft Windows *SHGetFolderPath*. Deze functie neemt de naam van een map in Microsoft Windows  als invoer en retourneert het volledige pad van de map.

```
GetFolderPath ()
```

### QvdCreateTime

Deze scriptfunctie retourneert de tijdstempel voor de XML-koptekst uit een QVD-bestand, indien aanwezig. Anders wordt NULL geretourneerd. In de tijdstempel wordt tijd weergegeven in UTC.

```
QvdCreateTime (filename)
```

### QvdFieldName

Deze script-functie retourneert de naam van veldnummer **fieldno** in een QVD-bestand. Als het veld niet bestaat, wordt NULL geretourneerd.

```
QvdFieldName (filename , fieldno)
```

### QvdNoOfFields

Deze scriptfunctie retourneert het aantal velden in een QVD-bestand.

```
QvdNoOfFields (filename)
```

### QvdNoOfRecords

Deze scriptfunctie retourneert het huidige aantal records in een QVD-bestand.

```
QvdNoOfRecords (filename)
```

### QvdTableName

Deze scriptfunctie retourneert de naam van de tabel die is opgeslagen in een QVD-bestand.

```
QvdTableName (filename)
```

## Attribute

Deze scriptfunctie retourneert de waarde van de metatags van verschillende mediabestanden als tekst. De volgende bestandsindelingen worden ondersteund: MP3, WMA, WMV, PNG en JPG. Als het bestand **filename** niet bestaat, niet een ondersteunde bestandsindeling heeft of niet een metatag bevat met de naam **attributename**, wordt NULL geretourneerd.

**Syntaxis:**

```
Attribute(filename, attributename)
```

Een groot aantal metatags kan worden gelezen. De voorbeelden in dit onderwerp laten zien welke tags kunnen worden gelezen voor de respectievelijke ondersteunde bestandstypen.

> *U kunt alleen metatags lezen die volgens de relevante specificatie in het bestand zijn opgeslagen, bijvoorbeeld ID2v3 voor MP3-bestanden of EXIF voor JPG-bestanden. U kunt geen meta-informatie die is opgeslagen in **Windows Bestandsbeheer** lezen.*

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| filename | De naam van een mediabestand, inclusief pad, indien nodig, als mapgegevensverbinding.<br><br>**Voorbeeld:** *'lib://Table Files/'*<br><br>In de bestaande scriptmodus, worden tevens de volgende padindelingen ondersteund:<br>• absoluut<br>**Voorbeeld:** *c:\data*<br>• relatief ten opzichte van de werkmap van de Qlik Sense-app.<br>**Voorbeeld:** *data* |
| attributename | De naam van een metatag. |

In de voorbeelden wordt de functie **GetFolderPath** gebruikt om de paden naar mediabestanden te vinden. Aangezien **GetFolderPath** alleen wordt ondersteund in de bestaande modus, moet u de verwijzingen naar **GetFolderPath** vervangen door een lib:// gegevensverbindingspad als u deze functie gebruikt in de standaardmodus of in Qlik Sense SaaS.

*Beperking van toegang tot bestandssysteem (page 1938)*

### Example 1: MP3-bestanden

Dit script leest alle mogelijke MP3-metatags in de map *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, 'TCON') as TCON, // 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, // Unsychronized 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

Dit script leest alle mogelijke EXIF-metatags uit JPG-bestanden in de map *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: Windows-mediabestanden

Dit script leest alle mogelijke WMA/WMV ASF-metatags in de map *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

Dit script leest alle mogelijke PNG-metatags in de map *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

De functie **ConnectString()** retourneert de naam van de actieve gegevensverbinding voor ODBC- of OLE DB-verbindingen. De functie retourneert een lege tekenreeks als geen **connect**-opdracht is uitgevoerd of na een **disconnect**-opdracht.

#### Syntaxis:

```
ConnectString()
```

Voorbeelden en resultaten:

Scriptvoorbeelden

| Voorbeeld | Resultaat |
|---|---|
| `LIB CONNECT TO 'Tutorial ODBC';`<br><br>`ConnectString:`<br><br>`Load ConnectString() as ConnectString AutoGenerate 1;` | Retourneert 'Tutorial ODBC' in het veld ConnectString.<br><br>In dit voorbeeld wordt ervan uitgegaan dat u over een gegevensverbinding beschikt met de naam Tutorial ODBC. |

## FileBaseName

De functie **FileBaseName** retourneert een tekenreeks met de naam van het tabelbestand dat op dat moment wordt gelezen, zonder pad of extensie.

**Syntaxis:**

```
FileBaseName()
```

Voorbeelden en resultaten:

Scriptvoorbeelden

| Voorbeeld | Resultaat |
|---|---|
| `LOAD *, filebasename( ) as X from`<br><br>`C:\UserFiles\abc.txt` | Retourneert 'abc' in veld X in elke gelezen record. |

## FileDir

De functie **FileDir** retourneert een tekenreeks met het pad naar de map van het tabelbestand dat op dat moment wordt gelezen.

**Syntaxis:**

```
FileDir()
```

*Deze functie ondersteunt alleen mapgegevensverbindingen in de standaardmodus.*

Voorbeelden en resultaten:

Scriptvoorbeelden

| Voorbeeld | Resultaat |
|---|---|
| `Load *, filedir( ) as X from`<br><br>`C:\UserFiles\abc.txt` | Retourneert 'C:\UserFiles' in veld X in elke gelezen record. |

## FileExtension

De functie **FileExtension** retourneert een tekenreeks met de extensie van het tabelbestand dat op dat moment wordt gelezen.

**Syntaxis:**

```
FileExtension()
```

Voorbeelden en resultaten:

Scriptvoorbeelden

| Voorbeeld | Resultaat |
|---|---|
| `LOAD *, FileExtension( ) as X from`<br><br>`C:\UserFiles\abc.txt` | Retourneert 'txt' in veld X in elke gelezen record. |

## FileName

De functie **FileName** retourneert een tekenreeks met de naam van het tabelbestand dat op dat moment wordt gelezen, zonder pad maar met de extensie.

**Syntaxis:**

```
FileName()
```

Voorbeelden en resultaten:

Scriptvoorbeelden

| Voorbeeld | Resultaat |
|---|---|
| `LOAD *, FileName( ) as X from`<br><br>`C:\UserFiles\abc.txt` | Retourneert ' 'abc.txt" in veld X in elke gelezen record. |

## FilePath

De functie **FilePath** retourneert een tekenreeks met het volledige pad naar het tabelbestand dat op dat moment wordt gelezen.

**Syntaxis:**

```
FilePath()
```

*Deze functie ondersteunt alleen mapgegevensverbindingen in de standaardmodus.*

Voorbeelden en resultaten:

Scriptvoorbeelden

| Voorbeeld | Resultaat |
|---|---|
| `Load *, FilePath( ) as X from C:\UserFiles\abc.txt` | Retourneert ' 'C:\UserFiles\abc.txt" in veld X in elke gelezen record. |

## FileSize

De functie retourneert een geheel getal met de grootte in bytes van het bestand of, als geen is opgegeven, van het tabelbestand dat op dat moment wordt gelezen.filenamefilename

**Syntaxis:**

**FileSize(**[filename]**)**

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| filename | De naam van een bestand, indien nodig met het pad, als een gegevensverbinding in een map of webbestand. Als u geen bestandsnaam opgeeft, wordt het tabelbestand gebruikt dat momenteel wordt gelezen.<br><br>**Voorbeeld:** *'lib://Table Files/'*<br><br>In de bestaande scriptmodus, worden tevens de volgende padindelingen ondersteund:<br>• absoluut<br>**Voorbeeld:** *c:\data*<br>• relatief ten opzichte van de werkmap van de Qlik Sense-app.<br>**Voorbeeld:** *data*<br>• URL-adres (HTTP of FTP) dat verwijst naar een locatie op internet of intranet.<br>**Voorbeeld:** *http://www.qlik.com* |

Voorbeelden en resultaten:

Scriptvoorbeelden

| Voorbeeld | Resultaat |
|---|---|
| `LOAD *, FileSize( ) as X from abc.txt;` | Retourneert de grootte van het opgegeven bestand (abc.txt) als een geheel getal in veld X in elke gelezen record. |
| `FileSize( 'lib://DataFiles/xyz.xls' )` | Retourneert de grootte van het bestand xyz.xls. |

## FileTime

De functie **FileTime** geeft een tijdstempel in UTC-indeling van de laatste wijziging van een opgegeven bestand. Als er geen bestand is opgegeven, geeft de functie een tijdstempel in UTC-indeling van de laatste wijziging van het momenteel gelezen tabelbestand.

**Syntaxis:**

```
FileTime([ filename ])
```

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| filename | De naam van een bestand, indien nodig met pad, als map- of webgegevensverbinding.<br><br>**Voorbeeld:** *'lib://Table Files/'*<br><br>In de bestaande scriptmodus, worden tevens de volgende padindelingen ondersteund:<br>• absoluut<br>**Voorbeeld:** *c:\data*<br>• relatief ten opzichte van de werkmap van de Qlik Sense-app.<br>**Voorbeeld:** *data*<br>• URL-adres (HTTP of FTP) dat verwijst naar een locatie op internet of intranet.<br>**Voorbeeld:** *http://www.qlik.com* |

Voorbeelden en resultaten:

Scriptvoorbeelden

| Voorbeeld | Resultaat |
|---|---|
| `LOAD *, FileTime( ) as X from abc.txt;` | Retourneert de tijdstempel van de laatste wijziging van het bestand (abc.txt) in veld X in elke gelezen record. |
| `FileTime( 'xyz.xls' )` | Retourneert de tijdstempel van de laatste wijziging van het bestand xyz.xls. |

## GetFolderPath

De functie **GetFolderPath** retourneert de waarde van de functie Microsoft Windows *SHGetFolderPath*. Deze functie neemt de naam van een map in Microsoft Windows als invoer en retourneert het volledige pad van de map.

*Deze functie wordt niet ondersteund in de standaardmodus. .*

**Syntaxis:**

```
GetFolderPath(foldername)
```

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| **foldername** | Naam van de Microsoft Windows-map.<br>De mapnaam mag geen spatie bevatten. Alle spaties in de mapnaam in Windows Explorer moeten worden verwijderd uit de mapnaam.<br>Voorbeelden:<br>*MyMusic*<br>*MyDocuments* |

**Voorbeelden en resultaten:**

Het doel van dit voorbeeld is het ophalen van de paden van de volgende Microsoft Windows-mappen: *MyMusic*, *MyPictures* en *Windows*. Voeg het voorbeeldscript toe aan uw app en laad het opnieuw.

```
LOAD
 GetFolderPath('MyMusic') as MyMusic,
 GetFolderPath('MyPictures') as MyPictures,
 GetFolderPath('Windows') as Windows
AutoGenerate 1;
```

Nadat de app opnieuw is geladen, worden de velden *MyMusic*, *MyPictures* en *Windows* toegevoegd aan het gegevensmodel. Elk veld bevat het pad naar de map die is gedefinieerd in de invoer. Bijvoorbeeld:

- *C:\Users\smu\Music* for the folder *MyMusic*
- *C:\Users\smu\Pictures* for the folder *MyPictures*
- *C:\Windows* for the folder *Windows*

## QvdCreateTime

Deze scriptfunctie retourneert de tijdstempel voor de XML-koptekst uit een QVD-bestand, indien aanwezig. Anders wordt NULL geretourneerd. In de tijdstempel wordt tijd weergegeven in UTC.

**Syntaxis:**

```
QvdCreateTime(filename)
```

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| filename | De naam van een QVD-bestand, indien nodig met pad, als map- of webgegevensverbinding.<br><br>**Voorbeeld:** *'lib://Table Files/'*<br><br>In de bestaande scriptmodus, worden tevens de volgende padindelingen ondersteund:<br><br>• absoluut<br><br>**Voorbeeld:** *c:\data*<br><br>• relatief ten opzichte van de werkmap van de Qlik Sense-app.<br><br>**Voorbeeld:** *data*<br><br>• URL-adres (HTTP of FTP) dat verwijst naar een locatie op internet of intranet.<br><br>**Voorbeeld:** *http://www.qlik.com* |

**Voorbeeld:**

```
QvdCreateTime('MyFile.qvd')

QvdCreateTime('C:\MyDir\MyFile.qvd')

QvdCreateTime('lib://DataFiles/MyFile.qvd')
```

## QvdFieldName

Deze script-functie retourneert de naam van veldnummer **fieldno** in een QVD-bestand. Als het veld niet bestaat, wordt NULL geretourneerd.

**Syntaxis:**

```
QvdFieldName(filename , fieldno)
```

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| filename | De naam van een QVD-bestand, indien nodig met pad, als map- of webgegevensverbinding.<br>**Voorbeeld:** *'lib://Table Files/'*<br>In de bestaande scriptmodus, worden tevens de volgende padindelingen ondersteund:<br>• absoluut<br>**Voorbeeld:** *c:\data*<br>• relatief ten opzichte van de werkmap van de Qlik Sense-app.<br>**Voorbeeld:** *data*<br>• URL-adres (HTTP of FTP) dat verwijst naar een locatie op internet of intranet.<br>**Voorbeeld:** *http://www.qlik.com* |
| fieldno | Het nummer van het veld in de tabel binnen het QVD-bestand. |

**Voorbeelden:**

```
QvdFieldName ('MyFile.qvd', 5)

QvdFieldName ('C:\MyDir\MyFile.qvd', 5)

QvdFieldName ('lib://DataFiles/MyFile.qvd', 5)
```

Alle drie de exemplaren retourneren de naam van het vijfde veld van de tabel binnen het QVD-bestand.

## QvdNoOfFields

Deze scriptfunctie retourneert het aantal velden in een QVD-bestand.

**Syntaxis:**

```
QvdNoOfFields(filename)
```

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| filename | De naam van een QVD-bestand, indien nodig met pad, als map- of webgegevensverbinding.<br><br>**Voorbeeld:** *'lib://Table Files/'*<br><br>In de bestaande scriptmodus, worden tevens de volgende padindelingen ondersteund:<br><br>• absoluut<br><br>**Voorbeeld:** *c:\data*<br><br>• relatief ten opzichte van de werkmap van de Qlik Sense-app.<br><br>**Voorbeeld:** *data*<br><br>• URL-adres (HTTP of FTP) dat verwijst naar een locatie op internet of intranet.<br><br>**Voorbeeld:** *http://www.qlik.com* |

**Voorbeelden:**

```
QvdNoOfFields ('MyFile.qvd')

QvdNoOfFields ('C:\MyDir\MyFile.qvd')

QvdNoOfFields ('lib://DataFiles/MyFile.qvd')
```

## QvdNoOfRecords

**Voorbeeld: Deze scriptfunctie retourneert het huidige aantal records in een QVD-bestand.**

**Syntaxis:**

```
QvdNoOfRecords(filename)
```

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| filename | De naam van een QVD-bestand, indien nodig met pad, als map- of webgegevensverbinding.<br><br>**Voorbeeld:** *'lib://Table Files/'*<br><br>In de bestaande scriptmodus, worden tevens de volgende padindelingen ondersteund:<br><br>• absoluut<br><br>**Voorbeeld:** *c:\data*<br><br>• relatief ten opzichte van de werkmap van de Qlik Sense-app.<br><br>**Voorbeeld:** *data*<br><br>• URL-adres (HTTP of FTP) dat verwijst naar een locatie op internet of intranet.<br><br>**Voorbeeld:** *http://www.qlik.com* |

**Voorbeelden:**

```
QvdNoOfRecords ('MyFile.qvd')

QvdNoOfRecords ('C:\MyDir\MyFile.qvd')

QvdNoOfRecords ('lib://DataFiles/MyFile.qvd')
```

## QvdTableName

Deze scriptfunctie retourneert de naam van de tabel die is opgeslagen in een QVD-bestand.

**Syntaxis:**

```
QvdTableName(filename)
```

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| filename | De naam van een QVD-bestand, indien nodig met pad, als map- of webgegevensverbinding.<br><br>**Voorbeeld:** *'lib://Table Files/'*<br><br>In de bestaande scriptmodus, worden tevens de volgende padindelingen ondersteund:<br><br>• absoluut<br><br>**Voorbeeld:** *c:\data*<br><br>• relatief ten opzichte van de werkmap van de Qlik Sense-app.<br><br>**Voorbeeld:** *data*<br><br>• URL-adres (HTTP of FTP) dat verwijst naar een locatie op internet of intranet.<br><br>**Voorbeeld:** *http://www.qlik.com* |

**Voorbeelden:**

```
QvdTableName ('MyFile.qvd')

QvdTableName ('C:\MyDir\MyFile.qvd')

QvdTableName ('lib://data\MyFile.qvd')
```

## 8.11 Financiële functies

Financiële functies kunnen zowel worden gebruikt in het load-script voor gegevens als in diagramuitdrukkingen voor het berekenen van betalingen en rentepercentages.
Voor alle argumenten worden contante betalingen als negatieve getallen weergegeven. Contant geld dat u ontvangt, wordt als positief getal weergegeven.
Hierna volgt een lijst met de argumenten die u kunt gebruiken in de financiële functies (uitgezonderd de argumenten die beginnen met **range**-).

*Voor alle financiële functies is het belangrijk dat u consequent bent bij het opgeven van eenheden voor **rate** en **nper**. Als er maandelijkse betalingen worden gedaan voor een lening van vijf jaar tegen een rente van 6% per jaar, gebruikt u 0,005 (6%/12) voor **rate** en 60 (5*12) voor **nper**. Als er jaarlijkse betalingen worden gedaan voor dezelfde lening, gebruikt u 6% voor **rate** en 5 voor **nper**.*

## Overzicht van financiële functies

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

### FV

Deze functie retourneert de toekomstige waarde van een investering op basis van periodieke, constante betalingen en een enkelvoudige jaarlijkse rente.

```
FV (rate, nper, pmt [ ,pv [ , type ] ])
```

### nPer

Deze functie retourneert het aantal perioden voor een investering op basis van periodieke, constante betalingen en een constant rentepercentage.

```
nPer (rate, pmt, pv [ ,fv [ , type ] ])
```

### Pmt

Deze functie retourneert de betaling voor een lening op basis van periodieke, constante betalingen en een constant rentepercentage. Dit bedrag moet gedurende de hele duur van de annuïteit gelijk blijven. Een betaling wordt weergegeven als een negatief getal, bijvoorbeeld, -20.

```
Pmt (rate, nper, pv [ ,fv [ , type ] ] )
```

### PV

Deze functie retourneert de huidige waarde van een investering.

```
PV (rate, nper, pmt [ ,fv [ , type ] ])
```

### Rate

Deze functie retourneert het rentepercentage per periode op annuïteit. Het resultaat heeft de standaardgetalnotatie van **Fix** twee decimalen en %.

```
Rate (nper, pmt , pv [ ,fv [ , type ] ])
```

## BlackAndSchole

Het Black and Scholes-model is een wiskundig model voor afgeleide financiële instrumenten. De formule berekent de theoretische waarde van een optie. In Qlik Sense retourneert de functie **BlackAndSchole** de waarde volgens de ongewijzigde Black and Scholes-formule (opties Europese stijl).

```
BlackAndSchole(strike , time_left , underlying_price , vol , risk_free_rate , type)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| strike | De toekomstige aankoopprijs van het aandeel. |
| time_left | Het resterende aantal tijdsperioden. |
| underlying_price | De huidige waarde van het aandeel. |
| vol | Volatiliteit (van de aandelenprijs) uitgedrukt als percentage in decimale vorm, per tijdsperiode. |
| risk_free_rate | De risicovrije rente uitgedrukt als percentage in decimale vorm, per tijdsperiode. |
| call_or_put | Het type optie:<br>'c', 'call' of enige numerieke waarde die niet gelijk is aan nul voor callopties<br>'p', 'put' of 0 voor putopties. |

**Beperkingen:**

De waarde van strike, time_left en underlying_price moet >0 zijn.

De waarde van vol en risk_free_rate moet <) of >0 zijn.

Voorbeelden en resultaten:

Scriptvoorbeelden

| Voorbeeld | Resultaat |
|---|---|
| `BlackAndSchole(130, 4, 68.5, 0.4, 0.04, 'call')`<br>Hiermee wordt de theoretische prijs berekend van een optie voor het aankopen van een aandeel dat vandaag 68,5 waard is, bij een waarde van 130 over 4 jaar. In de formule wordt gebruikgemaakt van een volatiliteit van 0,4 (40%) per jaar en een risicovrij rentetarief van 0,04 (4%). | retourneert 11,245 |

## FV

Deze functie retourneert de toekomstige waarde van een investering op basis van periodieke, constante betalingen en een enkelvoudige jaarlijkse rente.

**Syntaxis:**

```
FV(rate, nper, pmt [ ,pv [ , type ] ])
```

**Retourgegevenstypen:** numeriek. Het resultaat wordt standaard geformatteerd als een valuta..

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| rate | Het rentepercentage per periode. |
| nper | Het totale aantal betalingsperioden in een annuïteit. |
| pmt | De betaling die elke periode wordt gedaan. Dit bedrag moet gedurende de hele duur van de annuïteit gelijk blijven. Een betaling wordt weergegeven als een negatief getal, bijvoorbeeld, -20. |
| pv | De huidige waarde die of het totale bedrag dat een reeks toekomstige betalingen op dit moment waard is. Als **pv** wordt weggelaten, wordt dit verondersteld 0 (nul) te zijn. |
| type | Moet 0 zijn als de betalingen aan het eind van de periode verschuldigd zijn en 1 als de betalingen aan het begin van de periode verschuldigd zijn. Als **type** niet wordt aangegeven, wordt dit verondersteld 0 te zijn. |

Voorbeelden en resultaten:

Scriptvoorbeeld

| Voorbeeld | Resultaat |
|---|---|
| U betaalt een nieuw huishoudelijk apparaat af in 36 maandelijkse termijnen van $20. De rente is 6% op jaarbasis. De rekening komt aan het einde van elke maand. Wat is het geïnvesteerde totaal wanneer de laatste rekening is betaald?<br>`FV(0.005,36,-20)` | Retourneert $786.72 |

## nPer

Deze functie retourneert het aantal perioden voor een investering op basis van periodieke, constante betalingen en een constant rentepercentage.

**Syntaxis:**

```
nPer(rate, pmt, pv [ ,fv [ , type ] ])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| rate | Het rentepercentage per periode. |

| Argument | Beschrijving |
|---|---|
| nper | Het totale aantal betalingsperioden in een annuïteit. |
| pmt | De betaling die elke periode wordt gedaan. Dit bedrag moet gedurende de hele duur van de annuïteit gelijk blijven. Een betaling wordt weergegeven als een negatief getal, bijvoorbeeld, -20. |
| pv | De huidige waarde die of het totale bedrag dat een reeks toekomstige betalingen op dit moment waard is. Als **pv** wordt weggelaten, wordt dit verondersteld 0 (nul) te zijn. |
| fv | De toekomstige waarde die of het kassaldo dat u wilt bereiken nadat de laatste betaling is gedaan. Als **fv** niet wordt aangegeven, wordt dit verondersteld 0 te zijn. |
| type | Moet 0 zijn als de betalingen aan het eind van de periode verschuldigd zijn en 1 als de betalingen aan het begin van de periode verschuldigd zijn. Als **type** niet wordt aangegeven, wordt dit verondersteld 0 te zijn. |

Voorbeelden en resultaten:

Scriptvoorbeeld

| Voorbeeld | Resultaat |
|---|---|
| U wilt een huishoudelijk apparaat verkopen en dat er in maandelijks termijnen van $ 20 wordt betaald. De rente is 6% op jaarbasis. De rekening komt aan het einde van elke maand. Hoeveel perioden zijn vereist als de waarde van het geld dat u hebt ontvangen na de laatste rekening gelijk moet zijn aan $ 800?<br>`nPer(0.005,-20,0,800)` | retourneert 36,56 |

## Pmt

Deze functie retourneert de betaling voor een lening op basis van periodieke, constante betalingen en een constant rentepercentage. Dit bedrag moet gedurende de hele duur van de annuïteit gelijk blijven. Een betaling wordt weergegeven als een negatief getal, bijvoorbeeld, -20.

```
Pmt(rate, nper, pv [ ,fv [ , type ] ] )
```

**Retourgegevenstypen:** numeriek. Het resultaat wordt standaard geformatteerd als een valuta..

Het totaalbedrag dat is betaald tijdens de looptijd van de lening is gelijk aan de waarde van **pmt** vermenigvuldigd met **nper**.

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| rate | Het rentepercentage per periode. |

| Argument | Beschrijving |
|---|---|
| nper | Het totale aantal betalingsperioden in een annuïteit. |
| pv | De huidige waarde die of het totale bedrag dat een reeks toekomstige betalingen op dit moment waard is. Als **pv** wordt weggelaten, wordt dit verondersteld 0 (nul) te zijn. |
| fv | De toekomstige waarde die of het kassaldo dat u wilt bereiken nadat de laatste betaling is gedaan. Als **fv** niet wordt aangegeven, wordt dit verondersteld 0 te zijn. |
| type | Moet 0 zijn als de betalingen aan het eind van de periode verschuldigd zijn en 1 als de betalingen aan het begin van de periode verschuldigd zijn. Als **type** niet wordt aangegeven, wordt dit verondersteld 0 te zijn. |

Voorbeelden en resultaten:

Scriptvoorbeelden

| Voorbeeld | Resultaat |
|---|---|
| De volgende formule retourneert de maandelijkse betaling op een lening van € 20.000 tegen een jaarrente van 10 procent, die in 8 maanden moet worden afbetaald:<br>`Pmt(0.1/12,8,20000)` | Retourneert -$2,594.66 |
| Als voor dezelfde lening betalingen aan het begin van de maand moeten plaatsvinden, is de betaling als volgt:<br>`Pmt(0.1/12,8,20000,0,1)` | Retourneert -$2,573.21 |

# PV

Deze functie retourneert de huidige waarde van een investering.

```
PV(rate, nper, pmt [ ,fv [ , type ] ])
```

**Retourgegevenstypen:** numeriek. Het resultaat wordt standaard geformatteerd als een valuta..

De huidige waarde is het totale bedrag dat een reeks toekomstige betalingen op dit moment waard is. Als u bijvoorbeeld geld leent, is het geleende bedrag de huidige waarde voor de verstrekker van de lening.

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| rate | Het rentepercentage per periode. |
| nper | Het totale aantal betalingsperioden in een annuïteit. |
| pmt | De betaling die elke periode wordt gedaan. Dit bedrag moet gedurende de hele duur van de annuïteit gelijk blijven. Een betaling wordt weergegeven als een negatief getal, bijvoorbeeld, -20. |

| Argument | Beschrijving |
|---|---|
| fv | De toekomstige waarde die of het kassaldo dat u wilt bereiken nadat de laatste betaling is gedaan. Als **fv** niet wordt aangegeven, wordt dit verondersteld 0 te zijn. |
| type | Moet 0 zijn als de betalingen aan het eind van de periode verschuldigd zijn en 1 als de betalingen aan het begin van de periode verschuldigd zijn. Als **type** niet wordt aangegeven, wordt dit verondersteld 0 te zijn. |

Voorbeelden en resultaten:

Scriptvoorbeeld

| Voorbeeld | Resultaat |
|---|---|
| Wat is de huidige waarde van een schuld, wanneer u $100 moet betalen aan het eind van elke maand gedurende een periode van vijf jaar, met een rentepercentage van 7%?<br>`PV(0.07/12,12*5,-100,0,0)` | Retourneert $5,050.20 |

## Rate

Deze functie retourneert het rentepercentage per periode op annuïteit. Het resultaat heeft de standaardgetalnotatie van **Fix** twee decimalen en %.

**Syntaxis:**

```
Rate(nper, pmt , pv [ ,fv [ , type ] ])
```

**Retourgegevenstypen:** numeriek.

De **rate** wordt per iteratie berekend en kan nul of meer oplossingen hebben. Als de opeenvolgende resultaten van **rate** niet samenvallen, wordt een NULL-waarde geretourneerd.

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| nper | Het totale aantal betalingsperioden in een annuïteit. |
| pmt | De betaling die elke periode wordt gedaan. Dit bedrag moet gedurende de hele duur van de annuïteit gelijk blijven. Een betaling wordt weergegeven als een negatief getal, bijvoorbeeld, -20. |
| pv | De huidige waarde die of het totale bedrag dat een reeks toekomstige betalingen op dit moment waard is. Als **pv** wordt weggelaten, wordt dit verondersteld 0 (nul) te zijn. |
| fv | De toekomstige waarde die of het kassaldo dat u wilt bereiken nadat de laatste betaling is gedaan. Als **fv** niet wordt aangegeven, wordt dit verondersteld 0 te zijn. |

| Argument | Beschrijving |
|---|---|
| type | Moet 0 zijn als de betalingen aan het eind van de periode verschuldigd zijn en 1 als de betalingen aan het begin van de periode verschuldigd zijn. Als **type** niet wordt aangegeven, wordt dit verondersteld 0 te zijn. |

Voorbeelden en resultaten:

Scriptvoorbeeld

| Voorbeeld | Resultaat |
|---|---|
| Wat is het rentepercentage van een annuïteitenlening van € 10.000 met een looptijd van vijf jaar en maandelijkse betalingen van € 300?<br>`Rate(60,-300,10000)` | Retourneert 2.00% |

# 8.12 Opmaakfuncties

De opmaakfuncties leggen de weergave-indeling op aan de ingevoerde numerieke velden of uitdrukkingen. Afhankelijk van het gegevenstype kunt u de tekens voor het scheidingsteken voor decimalen, duizendtallen enzovoort opgeven.

Alle functies retourneren een dubbele waarde met zowel de tekenreeks als de getalwaarde, maar kunnen worden beschouwd als een conversie van getal naar tekenreeks. **Dual()** is een speciaal geval, maar de andere opmaakfuncties nemen de numerieke waarde van de invoeruitdrukking en genereren een tekenreeks die het getal vertegenwoordigt.

De interpretatiefuncties, daarentegen, doen het omgekeerde: zij nemen tekenreeksuitdrukkingen en evalueren deze als getallen, waarbij de notatie van het resulterende getal wordt opgegeven.

De functies kunnen zowel worden gebruikt in load-scripts voor gegevens als in diagramuitdrukkingen.

*Alle getallen moeten worden weergegeven met een decimale punt als scheidingsteken voor decimalen.*

## Overzicht van opmaakfuncties

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

### ApplyCodepage

**ApplyCodepage()** hiermee past u een andere codepagina-tekenset toe op het veld of de tekst die wordt aangegeven in de uitdrukking. Gebruik een getalnotatie voor het **codepage**-argument.

```
ApplyCodepage (text, codepage)
```

### Date

**Date()** maakt een uitdrukking op als een datum met de opmaak die in de systeemvariabelen in het load-script voor gegevens, in het besturingssysteem of in een opmaakreeks is ingesteld, indien opgegeven.

```
Date (number[, format])
```

### Dual

**Dual()** combineert een getal en een tekenreeks tot één record, zodat de nummerweergave van de record kan worden gebruikt voor sorterings- en berekeningsdoeleinden, terwijl de tekenreekswaarde kan worden gebruikt voor weergavedoeleinden.

```
Dual (text, number)
```

### Interval

**Interval()** maakt een getal op als een tijdsinterval met de opmaak die in de systeemvariabelen in het load-script voor gegevens, in het besturingssysteem of in een opmaakreeks is ingesteld, indien opgegeven.

```
Interval (number[, format])
```

### Money

**Money()** maakt een uitdrukking op numerieke wijze op als een geldwaarde in de opmaak die is ingesteld in de systeemvariabelen die zijn ingesteld in het load-script voor gegevens of in het besturingssysteem, tenzij een opmaakreeks is opgegeven, alsmede optionele scheidingstekens voor decimalen en duizendtallen.

```
Money (number[, format[, dec_sep [, thou_sep]]])
```

### Num

**Num()** geeft de opmaak van een getal, wat wil zeggen dat deze functie de numerieke waarde van de invoer converteert naar tekst met gebruikmaking van de opmaak die is opgegeven in de tweede parameter. Als de tweede parameter wordt weggelaten, worden de scheidingstekens voor decimalen en duizendtallen gebruikt die zijn ingesteld in het load-script voor gegevens. Aangepaste scheidingstekens voor decimalen en scheidingstekens voor duizendtallen zijn optionele parameters.

```
Num (number[, format[, dec_sep [, thou_sep]]])
```

### Time

**Time()** maakt een uitdrukking op als een tijdwaarde in de tijdnotatie die in de systeemvariabelen in het load-script voor gegevens of in het besturingssysteem is ingesteld, tenzij een opmaakreeks is opgegeven.

```
Time (number[, format])
```

### Timestamp

**TimeStamp()** maakt een uitdrukking op als een datum- en tijdwaarde in de notatie voor tijdstempels die in de systeemvariabelen in het load-script voor gegevens of in het besturingssysteem is ingesteld, tenzij een opmaakreeks is opgegeven.

```
Timestamp (number[, format])
```

**Zie ook:**

p *Interpretatiefuncties (page 1447)*

## ApplyCodepage

**ApplyCodepage()** hiermee past u een andere codepagina-tekenset toe op het veld of de tekst die wordt aangegeven in de uitdrukking. Gebruik een getalnotatie voor het **codepage**-argument.

*Hoewel ApplyCodepage in diagramuitdrukkingen kan worden gebruikt, dient het vaker als scriptfunctie in de editor voor laden van gegevens. Terwijl u bijvoorbeeld bestanden laadt die mogelijk zijn opgeslagen in andere tekensets waarover u geen controle hebt, kunt u de codepagina toepassen die de tekenset vertegenwoordigt die u nodig hebt.*

**Syntaxis:**

```
ApplyCodepage(text, codepage)
```

**Retourgegevenstypen:** tekenreeks

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | Veld of tekst waarop u een andere codepagina wilt toepassen, opgegeven via het argument **codepage**. |
| **codepage** | Nummer dat de codepagina aangeeft die moet worden toegepast op het veld of de uitdrukking die is opgegeven via **text**. |

Voorbeeld: load-script

**Voorbeeld**

```
LOAD
ApplyCodepage(ROWX,1253) as GreekProduct,
ApplyCodepage (ROWY, 1255) as HebrewProduct,
ApplyCodepage (ROWZ, 65001) as EnglishProduct;
SQL SELECT ROWX, ROWY, ROWZ From Products;
```

**Resultaat**

Bij het laden vanuit SQL bevat de bron mogelijk een aantal verschillende tekensets: Cyrillisch, Hebreeuws, enzovoort, die afwijken van de indeling UTF-8.

| Voorbeeld | Resultaat |
| --- | --- |
| | Deze zouden rij voor rij moeten worden geladen waarbij voor elke rij een andere codepagina zou moeten worden toegepast. De **codepage**-waarde 1253 staat voor de Griekse tekenset van Windows, de waarde 1255 voor Hebreeuws en de waarde 65001 voor standaard Latijnse UTF-8-tekens. |

**Zie ook:** *Tekenset (page 176)*

## Date

**Date()** maakt een uitdrukking op als een datum met de opmaak die in de systeemvariabelen in het load-script voor gegevens, in het besturingssysteem of in een opmaakreeks is ingesteld, indien opgegeven.

**Syntaxis:**

```
Date(number[, format])
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
| --- | --- |
| **number** | Het getal dat moet worden opgemaakt. |

| Argument | Beschrijving |
|---|---|
| format | Tekenreeks waarmee de notatie van de resulterende tekenreeks wordt beschreven. Als er geen opmaakreeks wordt opgegeven, wordt de datumopmaak gebruikt die is ingesteld in de systeemvariabelen in het load-script voor gegevens of in het besturingssysteem.<br>De functie `Date` ondersteunt vele opmaakopties voor verschillende datumstructuren, bijvoorbeeld:<br>`DD/MM/YYYY`: dag/maand/jaar (bijvoorbeeld 15/02/2022)<br>`MM-DD-YYYY`: maand-dag-jaar met koppeltekens (bijvoorbeeld 02-15-2022)<br>`YYYY-MM-DD`: jaar-maand-dag, ISO-formaat (bijvoorbeeld 2022-02-15)<br>`D-MMM-YYYY`: dag, korte maandnaam, jaar, met koppeltekens (bijvoorbeeld 15-Feb-2022)<br>`MMMM YYYY`: lange maandnaam en jaar (bijvoorbeeld February 2022)<br>`MM/YYYY`: maand en jaar (bijvoorbeeld 02/2022)<br>`DD MMMM YYYY`: dag, lange maandnaam en jaar (bijvoorbeeld 15 February 2022) |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `Date( 35648 )` | Retourneert `97-08-06` als de `DateFormat` instelling `YY-MM-DD` is. |
| `Date( 35648, 'YY.MM.DD' )` | Retourneert `97.08.06` als de `DateFormat` instelling `YY-MM-DD` is. |
| `Date( '8/6/97', 'YY.MM.DD' )` | Retourneert `NULL` als de `DateFormat` instelling `YY-MM-DD` is. |

## Voorbeeld - Basisprincipes voor Date

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `RowNo`
  - `InputDate`
- De variabele `DateFormat`, die de notatie `YY-MM-DD` gebruikt.

### Load-script

```
SET DateFormat='YY-MM-DD';
Example:
Load * inline [
RowNo, InputDate
1, 35648
2, 35648.375
3, 8/6/97
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `RowNo`
- `InputDate`

Maak de volgende metingen:

- `=Date(InputDate)`, om de waarde weer te geven als een datum die is opgemaakt met behulp van de instellingen van de variabele `DateFormat`.
- `=Date(InputDate, 'DD.MM.YYYY')`, om de waarde weer te geven als een datum die is opgemaakt volgens de notatie die in de functie is opgegeven.
- `=Date(InputDate, 'YY.MM.DD')`, om de waarde weer te geven als een datum die is opgemaakt volgens de notatie die in de functie is opgegeven.
- `=Num(InputDate)` om de numerieke waarde van de datum te retourneren.

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 | - | - | - | - |

De resultaten laten zien hoe de functie `Date` gebruikt kan worden om datums weer te geven en op te maken. In dit voorbeeld wordt de `DateFormat` ingesteld als `YY-MM-DD` in het load-script. Het systeem verwacht dat de uitdrukking voor de datum deze notatie gebruikt. U kunt zien dat de waarde voor `InputDate` in rij 3 (`8/6/97`) deze notatie niet volgt, daarom herkent het systeem het niet als een datum en retourneert het `NULL`.

Verander de instelling `DateFormat` in het load-script in `SET DateFormat='M/D/YY';` en laad de gegevens opnieuw. De waarde in rij 3 komt nu overeen met de verwachte notatie. Het systeem zal dit herkennen als een geldige datum en de datum dienovereenkomstig opmaken. De volgende tabel toont de resultaten met de nieuwe instelling `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 |

## Voorbeeld – Scenario met Date

Diagramuitdrukking

### Overzicht

Een gegevensverzameling bevat verkoopgegevens die de datumnotatie `MM-DD-YYYY` gebruiken. U wilt de datumnotatie `DD/MM/YYYY` gebruiken.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Customer`
  - `SalesDate`
  - `Amount`
- De variabele `DateFormat`, die de notatie `MM/DD/YYYY` gebruikt.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Customer`

Maak de volgende metingen:

- `=Date(SalesDate ,'DD/MM/YYYY')` om de waarde weer te geven als een datum die is opgemaakt volgens de notatie die in de functie is opgegeven.
- `=Sum(Amount)` om de totale waarde van de verkoopkosten te berekenen.

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 |

Als u naar de resultaten kijkt, ziet u dat hoewel de datums in de gegevensverzameling één notatie gebruikten, u met de functie `Date` deze datums in een andere notatie kunt presenteren.

## Dual

**Dual()** combineert een getal en een tekenreeks tot één record, zodat de nummerweergave van de record kan worden gebruikt voor sorterings- en berekeningsdoeleinden, terwijl de tekenreekswaarde kan worden gebruikt voor weergavedoeleinden.

**Syntaxis:**

```
Dual(text, number)
```

**Retourgegevenstypen:** dubbele waarde

> *Alle dubbele retourwaarden worden rechts uitgelijnd.*

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De tekenreekswaarde die moet worden gebruikt in combinatie met het getalargument. |
| **number** | Het getal dat moet worden gebruikt in combinatie met het tekenreeksargument. |

In Qlik Sense zijn alle veldwaarden mogelijk dubbele waarden. Dit betekent dat de veldwaarden zowel een numerieke waarde als een tekstuele waarde kunnen hebben. Een voorbeeld is een datum die een numerieke waarde 40908 en de tekstuele representatie '2011-12-31' zou kunnen hebben.

> *Als diverse gegevens die in één veld worden gelezen, verschillende tekenreeksweergaven hebben maar wel dezelfde geldige getalswaarde, delen ze allemaal de eerst aangetroffen tekenreeksweergave.*

*De functie **dual** wordt meestal vroeg in het script gebruikt, nog voordat andere gegevens in het desbetreffende veld worden gelezen, zodat die eerste tekenreeksweergave wordt gemaakt en dan in filtervakken wordt getoond.*

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Beschrijving |
|---|---|
| `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)` | Het veld `DayOfWeek` kan als dimensie in een visualisatie worden gebruikt. In een tabel met de weekdagen worden de dagen automatisch gesorteerd op juiste getalvolgorde in plaats van op alfabetische volgorde. |
| `Dual(Pick( Priority , 'Low','Medium','High'), Priority)` | Het veld `Priority` kan als dimensie in een visualisatie worden gebruikt. In een tabel wordt het veld `Priority` automatisch gesorteerd op juiste getalvolgorde in plaats van op alfabetische volgorde. |
| `Dual('Q' & Ceil(Month (Date)/3), Ceil(Month (Date)/3))` | Het veld `Date` kan als dimensie in een visualisatie worden gebruikt. De tekstwaarden `Q1` tot `Q4` worden aangemaakt en krijgen de numerieke waarden `1` tot `4` toegewezen. |
| `Dual(WeekYear(Date) & '-W' & Week(Date), WeekStart (Date))` | Het veld `Date` kan als dimensie in een visualisatie worden gebruikt. In dit voorbeeld worden tekstwaarden van het formulier `'2012-W22'` gemaakt en wordt tegelijkertijd een numerieke waarde toegewezen die overeenkomt met het datumnummer van de eerste dag van de week, bijvoorbeeld: `41057`. |

Voorbeeld: load-scripts

| Voorbeeld | Beschrijving |
| --- | --- |
| Voeg de volgende voorbeelden toe aan uw script en voer het uit.<br>`Load dual ( NameDay,NumDay ) as DayOfWeek inline`<br>`[ NameDay,NumDay`<br>`Monday,0`<br>`Tuesday,1`<br>`Wednesday,2`<br>`Thursday,3`<br>`Friday,4`<br>`Saturday,5`<br>`Sunday,6 ];` | Het veld `DayOfWeek` kan bijvoorbeeld als dimensie in een visualisatie worden gebruikt. In een tabel met de weekdagen worden de dagen automatisch gesorteerd op juiste getalvolgorde in plaats van op alfabetische volgorde. |
| `Load Dual('Q' & Ceil(Month(Now ())/3), Ceil(Month (Now())/3)) as Quarter AutoGenerate 1;` | Dit voorbeeld retourneert het huidige kwartaal. Het wordt weergegeven als `Q1` als de functie **Now** wordt uitgevoerd in de eerste drie maanden van het jaar, `Q2` voor de tweede reeks van drie maanden enzovoort. Als het veld `Quarter` echter wordt gebruikt bij sorteren, gedraagt het zich als een numerieke waarde: 1 t/m 4. |
| `Dual('Q' & Ceil (Month(Date)/3), Ceil(Month (Date)/3)) as Quarter` | Net als in het voorafgaande voorbeeld, wordt het veld `Quarter` gemaakt met de tekstwaarden `Q1` t/m `Q4` en worden hieraan de numerieke waarden `1` t/m `4` toegewezen. Als u dit in het script wilt gebruiken, moeten de waarden voor `Date` worden geladen. |
| `Dual(WeekYear(Date) & '-w' & Week (Date), WeekStart (Date)) as YearWeek` | In dit voorbeeld wordt een veld `YearWeek` gemaakt met tekstwaarden van het formulier `2012-W22` en wordt tegelijkertijd een numerieke waarde toegewezen die overeenkomt met het datumnummer van de eerste dag van de week, bijvoorbeeld: `41057`. Als u dit in het script wilt gebruiken, moeten de waarden voor `Date` worden geladen. |

## Voorbeeld - Basisprincipes voor Dual

Diagramuitdrukking

### Overzicht

Een gegevensverzameling bevat gegevens over de klantenservice. U wilt het aantal oproepen per dag van de week bekijken.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `DayOfWeek`
  - `NoOfCalls`

### Load-script

```
Example:
Load * inline [
DayOfWeek, NoOfCalls
0, 25000
1, 30000
2, 27500
3, 32000
4, 23250
5, 28950
6, 32350
];
```

### Resultaten

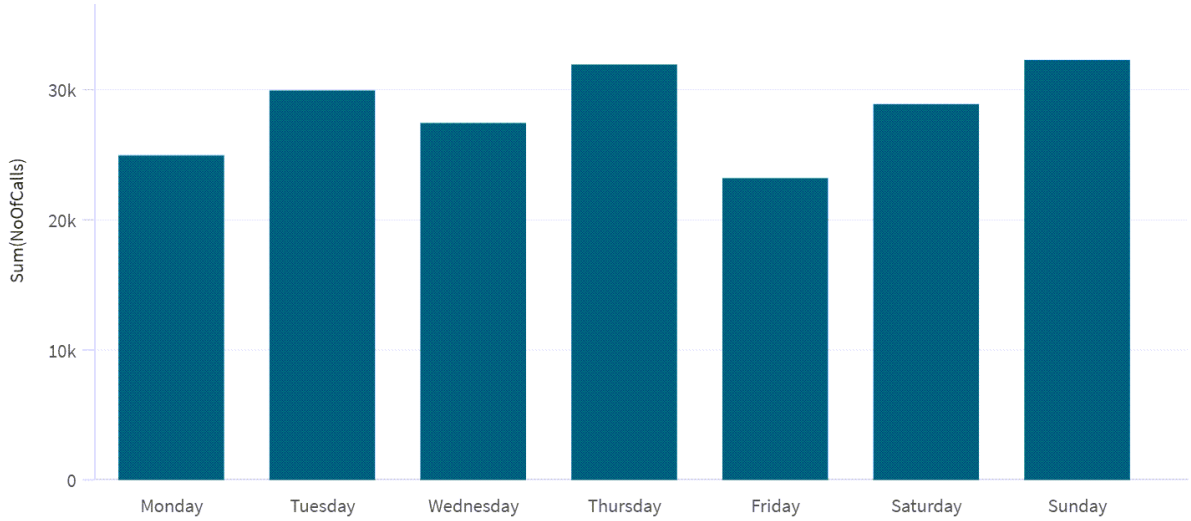
Laad de gegevens en open een werkblad. Maak een nieuw staafdiagram en voeg vervolgens een dimensie toe met een van de volgende methoden:

a. Voeg in de dimensie-uitdrukking het volgende toe:
```
=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. Deze methode is efficiënter. Voeg in de dimensie-uitdrukking het volgende toe:
```
=Dual(pick( DayOfWeek +1 , 'Monday','Tuesday','Wednesday', 'Thursday','Friday',
'Saturday','Sunday' ), DayOfWeek)
```

Maak de volgende meting:

- `=Sum(NoOfCalls)`, om het aantal oproepen te berekenen.

*Resultaten: Staafdiagram met het aantal oproepen per weekdag*




Het staafdiagram toont het aantal oproepen per dag van de week. Het sorteert de dagen van de week automatisch in de juiste volgorde.

## Voorbeeld – Scenario met Dual

### Diagramuitdrukking

#### Overzicht

Een gegevensverzameling bevat feedback van klantenondersteuning. U wilt de feedback sorteren en weergeven met de negatieve reacties eerst gevolgd door de positieve reacties.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `ResponseType`
  - `NoOfResponses`

#### Load-script

```
Example:
Load * inline [
ResponseType, NoOfResponses
1, 350
2, 560
3, 4350
4, 2350
```

```
5, 5350
];
```

**Resultaten**

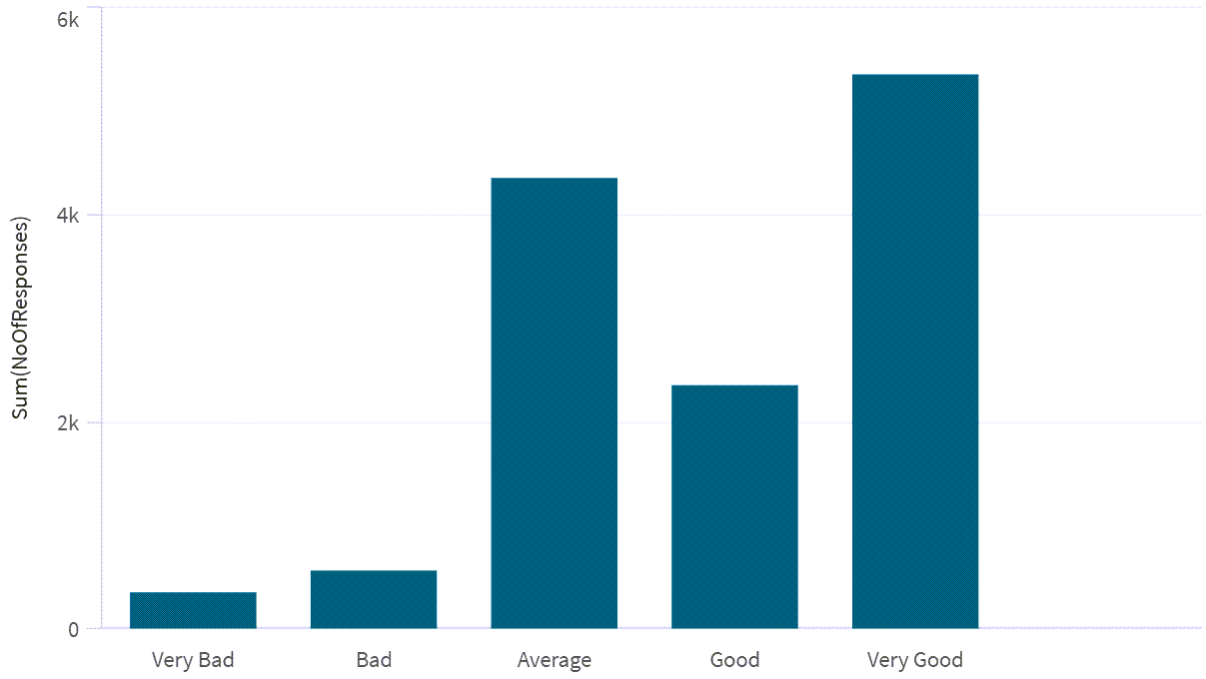
Laad de gegevens en open een werkblad. Maak een nieuw staafdiagram en voeg de volgende uitdrukking toe aan een dimensie.

- `=Dual(pick( ResponseType , 'Very Bad','Bad','Average', 'Good','Very Good'), ResponseType)`

Maak de volgende meting:

- `=Sum(NoOfResponses)`, om het aantal reacties te berekenen.

*Resultaten: Staafdiagram met het aantal reacties per type reactie*




De staafdiagram toont het aantal reacties per type reactie. De staven worden automatisch gesorteerd van negatief naar positief. Het reactietype, dat een numerieke waarde is, heeft een tekstbeschrijving gekregen.

## Interval

**Interval()** maakt een getal op als een tijdsinterval met de opmaak die in de systeemvariabelen in het load-script voor gegevens, in het besturingssysteem of in een opmaakreeks is ingesteld, indien opgegeven.

Intervallen kunnen worden opgemaakt als een tijd, als dagen of als een combinatie van dagen, uren, minuten, seconden en fracties van seconden.

**Syntaxis:**

```
Interval(number[, format])
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **number** | Het getal dat moet worden opgemaakt. |
| **format** | Tekenreeks die beschrijft hoe de resulterende intervalreeks moet worden opgemaakt. Bij weglating wordt gebruikgemaakt van de tijdnotatie en het scheidingsteken voor decimalen die zijn ingesteld in het besturingssysteem. Bij weglating wordt gebruikgemaakt van de tijdnotatie en het scheidingsteken voor decimalen die zijn ingesteld in het besturingssysteem.<br>De functie `Interval` ondersteunt vele opmaakopties voor verschillende tijdstructuren, bijvoorbeeld:<br>`hh:mm`–eenvoudige uren en minuten<br>`hh:mm:ss`–uren, minuten en seconden<br>`d hh:mm`–dagen, uren en minuten |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `Interval( 0.375 )` | Retourneert `9:00:00` als de `TimeFormat` `h:mm:ss` is. |
| `Interval(.375, 'hh:mm' )` | Retourneert `9:00` als de in de functie geleverde notatie `hh.mm` is. |
| `Interval( 0.375, 'd hh' )` | Retourneert `0 09` als de in de functie geleverde notatie `d hh` is. |

## Voorbeeld - Basisprincipes voor Interval

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Het volgende veld in een gegevenstabel:
  - `IntervalValue`
- De variabele `TimeFormat`, die de notatie `h:mm:ss` gebruikt.

### Load-script

```
SET TimeFormat='h:mm:ss';
Example:
Load * inline [
```

```
IntervalValue
0.375
1.375
35.648
36.875
368.75
468.95
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `IntervalValue`

Maak de volgende metingen:

- `=Interval(IntervalValue)`, om de waarde als intervalnotatie weer te geven met behulp van de systeeminstelling.
- `=Interval(IntervalValue, 'd hh')`, om de waarde weer te geven als een intervalnotatie die is opgemaakt volgens de notatie die in de functie is opgegeven.
- `=Interval(IntervalValue, 'hh:mm')`, om de waarde weer te geven als een intervalnotatie die is opgemaakt volgens de notatie die in de functie is opgegeven.
- `=Interval(IntervalValue, 'd hh:mm')`, om de waarde weer te geven als een intervalnotatie die is opgemaakt volgens de notatie die in de functie is opgegeven.

Resultatentabel

| IntervalValue | Interval (IntervalValue) | Interval (IntervalValue, 'd hh') | Interval (IntervalValue, 'hh:mm' | Interval (IntervalValue, 'd 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 |

De resultaten laten zien hoe de functie `Interval` waarden weergeeft volgens de notatieparameter in de functie. In de laatste rij bijvoorbeeld, retourneert de waarde `468.95`, indien deze wordt opgemaakt met de functie `Interval` end e notatieparameter, `d hh:mm`, de waarde `468` dagen, `22` uur en `48` minuten.

## Voorbeeld – Scenario met Interval

Diagramuitdrukking

### Overzicht

Gegevens van supporttickets berekenen de oplostijd per ticket in seconden. U wilt deze waarde converteren naar een andere notatie.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `TicketID`
  - `ResolutionTime`

### Load-script

```
Example:
Load * inline [
TicketID, ResolutionTime
Ticket A, 4350
Ticket B, 560
Ticket C, 86400
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `TicketID`

Maak de volgende meting:

- `=Interval(ResolutionTime / 86400, 'hh:mm:ss')`, om de waarden om te zetten naar dagen, door `ResolutionTime` te delen door `86,400`, en vervolgens op te maken volgens de notatie die in de functie is opgegeven.

Resultatentabel

| TicketID | Interval(ResolutionTime / 86400, 'hh:mm:ss') |
|---|---|
| Ticket A | 01:12:30 |
| Ticket B | 00:09:20 |
| Ticket C | 24:00:00 |

De waarden voor oplostijd die oorspronkelijk in seconden werden geleverd, worden nu geconverteerd naar een fractie van een dag en vervolgens opgemaakt in uren, minuten en seconden. De tabel laat bijvoorbeeld zien dat `Ticket B` er `9` uur en `20` minuten over deed om te worden opgelost.

## Money

**Money()** maakt een uitdrukking op numerieke wijze op als een geldwaarde in de opmaak die is ingesteld in de systeemvariabelen die zijn ingesteld in het load-script voor gegevens of in het besturingssysteem, tenzij een opmaakreeks is opgegeven, alsmede optionele scheidingstekens voor decimalen en duizendtallen.

**Syntaxis:**

```
Money(number[, format[, dec_sep[, thou_sep]]])
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **number** | Het getal dat moet worden opgemaakt. |
| **format** | Tekenreeks die beschrijft hoe de resulterende valutareeks moet worden opgemaakt. Bij weglating wordt gebruikgemaakt van de waarde voor `MoneyFormat` die is ingesteld in het load-script voor gegevens.<br>De functie `Money` ondersteunt vele opmaakopties voor verschillende datumstructuren, bijvoorbeeld:<br>`$#,##0.00`: voegt een dollarteken en twee decimalen toe.<br>`€ #,##0`: voegt een euroteken en geen decimalen toe, met een spatie na het euroteken.<br>`¥#,##0`: voegt een Japans yen-teken zonder decimalen toe.<br>`$#,##0;($#,##0)`: plaatst negatieve waarden tussen haakjes.<br>`₣ #.##0,00`: voegt een Zwitserse frank-teken toe met punten als scheidingsteken voor duizendtallen en een komma voor decimalen (bijvoorbeeld `₣ 1.000,00`).<br>`R #,##0.00`: Zuid-Afrikaanse rand-teken met twee decimalen en spaties na het rand-teken.<br>`$ #,##0.000`: voegt drie decimalen toe aan de dollarnotatie. |

| Argument | Beschrijving |
|---|---|
| **dec_sep** | Tekenreeks die het scheidingsteken voor decimalen opgeeft. Bij weglating wordt gebruikgemaakt van de waarde voor `MoneyDecimalSep` die is ingesteld in het load-script voor gegevens.<br>De functie `Money` ondersteunt vele notaties voor scheidingstekens voor decimalen, bijvoorbeeld:<br><br>Komma ' , ': Bijvoorbeeld `$1,000,00` voor duizend dollar.<br><br>Punt ' . ': bijvoorbeeld `€ 1,000.00`.<br><br>Koppelteken ' - ': als dit wordt gebruikt in plaats van de traditionele punt of komma als scheidingsteken voor decimalen, geeft het waarden weer als `¥1,000-00`. |
| **thou_sep** | Tekenreeks die het scheidingsteken voor duizendtallen opgeeft. Bij weglating wordt gebruikgemaakt van de waarde voor `MoneyThousandSep` die is ingesteld in het load-script voor gegevens.<br>De functie `Money` ondersteunt vele opties voor het scheidingsteken voor duizendtallen:<br><br>Punt ' . ': gebruikt een punt als scheidingsteken voor duizendtallen (gebruikelijk in sommige Europese notaties), wordt weergegeven als `$1.000,00` indien gecombineerd met een komma als scheidingsteken voor decimalen.<br><br>Spatie ' ': gebruikt een spatie voor het scheidingsteken voor duizendtallen, wat gebruikelijk is in sommige Europese landen. Wordt weergegeven als `€ 1 000.00`.<br><br>Onderstrepingsteken '_': gebruikt een onderstrepingsteken voor het scheidingsteken voor duizendtallen, dat soms wordt gebruikt in technische of specifieke regionale notaties. Geeft waarden weer zoals `$1_000.00`. |

Als de argumenten **format**, **dec_sep** en **thou_sep** worden weggelaten, wordt de valutanotatie die in het besturingssysteem is ingesteld, gebruikt bij het formatteren van elk specifiek deel van de uitvoer. De instellingen van het besturingssysteem worden automatisch opgenomen als systeemvariabelen bij het maken van het script, zodat deze consistent zijn voor de hele applicatie.

Deze systeemvariabelen zijn **MoneyFormat**, **MoneyDecimalSep**, en **MoneyThousandSep** en wijzen respectievelijk naar de **format**, **dec_sep**, en **thou_sep** functieargumenten.

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaten |
|---|---|
| `Money( 35648 )` | Retourneert `kr 35 648,00` als de `MoneyFormat` instelling `kr ##0,00` is en de `MoneyThousandSep` een punt `' '` is. |
| `Money( 35648, '#,##0 ¥', '.' , ',' )` | Retourneert `3,564,800 ¥` door de instelling `MoneyFormat` en de instelling `MoneyThousandSep` te negeren, omdat de optionele parameters `Format`, `dec_sep` en `thou_sep` in de functie worden geleverd. |
| `Money(35648)` | Retourneert `$ 35,648.00` als de `MoneyFormat` instelling `$ #,##0.00` is en de `MoneyThousandSep` een komma `','` is. |

## Voorbeeld - Basisprincipes voor Money

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `InputValue`.
- De volgende variabele notaties:
  - De variabele `MoneyFormat`, die de notatie `kr ##0,00` gebruikt.
  - De variabele `MoneyThousandSep` is ingesteld om een spatie `' '` te gebruiken

### Load-script

```
SET MoneyFormat='kr ##0,00';
SET MoneyThousandSep=' ';
Example:
Load * inline [
InputValue
35648
3564800
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `InputValue`

Maak de volgende metingen:

- `=Money(InputValue)`, om de waarde weer te geven als een geldbedrag dat is opgemaakt met behulp van de systeeminstelling.
- `=Money(InputValue, '#,##0 ¥', '.' , ',')`, om de waarde weer te geven als een geldbedrag dat is opgemaakt volgens de notatie die in de functie is opgegeven.
- `=Num(InputValue)` om de numerieke waarde te retourneren.

Results table

| InputValue | Money(InputValue) | Money(InputValue, '#,##0 ¥', '.' , ',') | Num(InputValue) |
|---|---|---|---|
| 35648 | kr35 648,00 | 35,648 ¥ | 35648 |
| 3564800 | kr3 564 800,00 | 3,564,800 ¥ | 3564800 |

In dit voorbeeld kunt u zien hoe de functie `Money` waarden weergeeft en hoe de waarden in de opmaakparameter de uitvoer kunnen veranderen.

## Voorbeeld - Basisprincipes voor Money met behulp van een andere opmaakinstelling

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `InputValue`.
- De volgende variabele notaties:
  - De variabele `MoneyFormat`, die de notatie `$ #,##0.00` gebruikt.
  - De variabele `MoneyThousandSep` is ingesteld om een komma `','` te gebruiken

### Load-script

```
SET MoneyFormat='$ #,##0.00';
SET MoneyThousandSep=',';
Example:
Load * inline [
InputValue
35648
3564800
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `InputValue`

Maak de volgende metingen:

- `=Money(InputValue)`, om de waarde weer te geven als een geldbedrag dat is opgemaakt met behulp van de systeeminstelling.
- `=Money(InputValue, '#,##0 ¥', '.' , ', )`, om de waarde weer te geven als een geldbedrag dat is opgemaakt volgens de notatie die in de functie is opgegeven.
- `=Num(InputValue)` om de numerieke waarde te retourneren.

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 dit voorbeeld kunt u zien hoe de functie `Money` waarden weergeeft en hoe de waarden in de opmaakparameter de uitvoer kunnen veranderen.

## Voorbeeld – Scenario met Money

Diagramuitdrukking

### Overzicht

Verkoopgegevens van klanten worden in verschillende valuta ontvangen. U wilt de gegevens weergeven in een tabel met de relevante valuta-opmaak.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Customer`
  - `CurrencyCode`
  - `Sales`
- De volgende variabele notaties:
  - De variabele `MoneyFormat`, die de notatie `$ ###0.00;-$ ###0.00` gebruikt.
  - De variabele `MoneyThousandSep` is ingesteld om een komma `','` te gebruiken
  - De variabele `MoneyDecimalSep` is ingesteld om een punt '.' te gebruiken

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Customer`

Maak de volgende meting:

- `=Money(Sales, If(CurrencyCode = 'USD', '$#,##0.00', If(CurrencyCode = 'EUR', '€ #,##0.00', '¥#,##0')))`, om de valuta in de juiste valuta-opmaak weer te geven.

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 |

De verkoopgegevens gebruiken nu de valuta-opmaak die is gedefinieerd in de metinguitdrukking. De parameters decimaal en duizend gebruiken de instellingen in het load-script.

## Num

**Num()** geeft de opmaak van een getal, wat wil zeggen dat deze functie de numerieke waarde van de invoer converteert naar tekst met gebruikmaking van de opmaak die is opgegeven in de tweede parameter. Als de tweede parameter wordt weggelaten, worden de scheidingstekens voor decimalen en duizendtallen gebruikt die zijn ingesteld in het load-script voor gegevens. Aangepaste scheidingstekens voor decimalen en scheidingstekens voor duizendtallen zijn optionele parameters.

**Syntaxis:**

```
Num(number[, format[, dec_sep [, thou_sep]]])
```

**Retourgegevenstypen:** dubbele waarde

De functie Num retourneert een dubbele waarde met zowel de tekenreeks als de numerieke waarde. De functie neemt de numerieke waarde van de invoeruitdrukking en genereert een tekenreeks die het getal vertegenwoordigt.

Argumenten

| Argument | Beschrijving |
|---|---|
| **number** | De teksttekenreeks die moet worden geëvalueerd. |

| Argument | Beschrijving |
|---|---|
| **format** | Tekenreeks die specificeert hoe de resulterende tekenreeks moet worden opgemaakt. Bij weglating worden de scheidingstekens voor decimalen en duizendtallen gebruikt die zijn ingesteld in het load-script voor gegevens.<br>De functie `Num` ondersteunt vele opmaakopties voor verschillende numerieke structuren, bijvoorbeeld:<br><br>`#,##0`:geeft een getal zonder decimalen weer.<br><br>`#,##0.00`:geeft een getal met twee decimalen weer.<br><br>`##0.00%`: geeft een getal als een percentage met twee decimalen weer.<br><br>`#,##0;(#,##0)`: plaatst getallen tussen haakjes.<br><br>`$#,##0.00`: maakt getallen op als valuta met een $-symbool.<br><br>`0.00E+0`:gGeeft getallen in wetenschappelijke notatie weer. |
| **dec_sep** | Tekenreeks die het scheidingsteken voor decimalen opgeeft. Bij weglating wordt gebruikgemaakt van de waarde voor de variabele DecimalSep die is ingesteld in het load-script voor gegevens.<br>De functie `Num` ondersteunt vele notaties voor scheidingstekens voor decimalen, bijvoorbeeld:<br><br>Komma ' , ': Bijvoorbeeld `$1,000,00` voor duizend dollar.<br><br>Punt ' . ': bijvoorbeeld `€ 1,000.00`.<br><br>Koppelteken ' - ': als dit wordt gebruikt in plaats van de traditionele punt of komma als scheidingsteken voor decimalen, geeft het waarden weer als `¥1,000-00`.<br><br>Ga voor meer informatie naar *DecimalSep (page 241)*. |
| **thou_sep** | Tekenreeks die het scheidingsteken voor duizendtallen opgeeft. Bij weglating wordt gebruikgemaakt van de waarde voor de variabele ThousandSep die is ingesteld in het load-script voor gegevens.<br>De functie `Num` ondersteunt vele opties voor het scheidingsteken voor duizendtallen:<br><br>Punt ' . ': gebruikt een punt als scheidingsteken voor duizendtallen (gebruikelijk in sommige Europese notaties), wordt weergegeven als `$1.000,00` indien gecombineerd met een komma als scheidingsteken voor decimalen.<br><br>Spatie ' ': gebruikt een spatie voor het scheidingsteken voor duizendtallen, wat gebruikelijk is in sommige Europese landen. Wordt weergegeven als `€ 1 000.00`.<br><br>Onderstrepingsteken '_': gebruikt een onderstrepingsteken voor het scheidingsteken voor duizendtallen, dat soms wordt gebruikt in technische of specifieke regionale notaties. Geeft waarden weer zoals `$1_000.00`.<br><br>Ga voor meer informatie naar *ThousandSep (page 279)*. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaten (geretourneerde tekenreekswaarde) |
|---|---|
| `Num(35648.312)` | Retourneer `35648.312` (afhankelijk van omgevingsvariabelen in het script) |
| `Num(35648.312, '0.0', '.')` | Retourneert `35648.3` |
| `Num(35648.312, '0,00', ',')` | Retourneert `3,56,48` |
| `Num(35648.312, '#,##0.0', '.', ',')` | Retourneert `35,648.3` |
| `Num(35648.312, '# ##0', ',', ' ')` | Retourneert `35 648` |

## Voorbeeld - Basisprincipes voor Num

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Het volgende veld in een gegevenstabel:
    - `InputValue`

### Load-script

```
Example:
Load * inline [
InputValue
35648
3564800
356.48
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `InputValue`

Maak de volgende metingen:

- `=Num(InputDate)`, om de waarde weer te geven als een getal dat is opgemaakt volgens de systeeminstellingen.

- `=Num(InputValue, '0.0', '.')`, om de waarde weer te geven als een getal dat is opgemaakt volgens de notatie die in de functie is opgegeven.
- `=Num(InputValue, '0,00', ',')`, om de waarde weer te geven als een getal dat is opgemaakt volgens de notatie die in de functie is opgegeven.
- `=Num(InputValue, '#,##0.0', '.', ',')`, om de waarde weer te geven als een getal dat is opgemaakt volgens de notatie die in de functie is opgegeven.
- `=Num(InputValue, '# ##0', ',', ' ')`, om de waarde weer te geven als een getal dat is opgemaakt volgens de notatie die in de functie is opgegeven.

Results table

| InputValue | Num (InputDate) | Num (InputValue, '0.0', '.') | Num (InputValue, '0,00', ',') | Num (InputValue, '#,##0.0', '.', ',') | Num (InputValue, '# ##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 |

De resultaten laten zien hoe de functie `Num` de waarden voor het veld `InputValue` kan weergeven op basis van de notatie die in de functie-uitdrukking is opgegeven.

### Load-script

Zelfs als de scheidingstekens voor duizendtallen en decimalen al in het script zijn ingesteld, kan de functie `Num` worden gebruikt voor de notatie van een getal in een load-script. Het onderstaande load-script bevat specifieke scheidingstekens voor duizendtallen en decimalen, maar vervolgens is `Num` gebruikt om de gegevens op andere manieren te noteren.

#### Load-script

Maak een nieuwe sectie aan in de Editor voor laden van gegevens, voeg vervolgens het voorbeeldscript toe en voer het uit. Voeg vervolgens ten minste de velden die in de resultatentabel staan aan een werkblad in uw app toe om het resultaat te bekijken.

```
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, 0.333333333333333, 2038593, M, Blue
3757, 20180923, 3177.4, 21, .14, 203521, XL, Black
];
```

#### Resultaten

Qlik Sense-tabel met de resultaten van verschillende manieren waarop de functie *Num* in het load-script wordt gebruikt. Ter illustratie bevat de vierde kolom van de tabel het gebruik van incorrecte notatie.

| No formatting | 0 | #,##0 | # ###,00 | # ###,00 , ',' , '' | #,###.00 , '.' , ',' | $#,###.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 |

## Voorbeeld – Scenario met Num

Diagramuitdrukking

#### Overzicht

Een gegevensverzameling bevat verkoopgegevens die u wilt presenteren in een eenvoudiger numerieke notatie.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Customer`
  - `Sales`

#### Load-script

```
Example:
Load * inline [
```

```
Customer, Sales
Customer A, 2550000
Customer B, 12500
Customer C, 15000
Customer D, 250
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Customer`

Maak de volgende meting:

- `=If(Sales > 1000000, Num(Sales/1000000, '#,##0.00M'), If(Sales > 1000, Num(Sales/1000, '#,##0.00K'), Num(Sales, '#,##0.00')))`, om de numerieke weergave van de waarden in te stellen.

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 |

De metinguitdrukking in kolom 2 gebruikt de functie `Num` om de gegevens in een vereenvoudigde numerieke notatie weer te geven.

#### Load-script

De functie `Num` kan worden gebruikt in een load-script om getallen als percentage op te maken.

### Load-script

Maak een nieuwe sectie aan in de Editor voor laden van gegevens, voeg vervolgens het voorbeeldscript toe en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan aan een werkblad in uw app toe om het resultaat te bekijken.

```
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, 0.333333333333333, 2038593, M, Blue
3757, 20180923, 3177.4, 21, .14, 203521, XL, Black
];
```

Qlik Sense-tabel met de resultaten van de functie `Num` die in het load-script wordt gebruikt voor het noteren van percentages.

| Discount | Discount #,##0% |
|---|---|
| 0 | 0% |
| 0.1 | 10% |
| 0.22 | 22% |
| 0 | 0% |
| 75 | 7,500% |
| 0.333333333333333 | 33% |
| .14 | 14% |

## Time

**Time()** maakt een uitdrukking op als een tijdwaarde in de tijdnotatie die in de systeemvariabelen in het load-script voor gegevens of in het besturingssysteem is ingesteld, tenzij een opmaakreeks is opgegeven.

**Syntaxis:**

```
Time(number[, format])
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **number** | Het getal dat moet worden opgemaakt. |

| Argument | Beschrijving |
|---|---|
| format | Tekenreeks die beschrijft hoe de resulterende tijdreeks moet worden opgemaakt. Bij weglating wordt gebruikgemaakt van de tijdnotatie en het scheidingsteken voor decimalen die zijn ingesteld in het besturingssysteem.<br><br>De functie `Time` ondersteunt vele opmaakopties voor verschillende tijdstructuren, bijvoorbeeld:<br><br>`hh:mm:ss`–Maakt een tijdwaarde op in een standaard tijdnotatie en geeft uren, minuten en seconden met voorloopnullen weer.<br><br>`hh:mm TT`–Gebruikt een 12-uursnotatie met AM/PM achtervoegsel.<br><br>`hh:mm`-Toont alleen uren en minuten, zonder seconden.<br><br>`hh.mm.ss`–Tijdnotatie met verschillende scheidingstekens, zoals een punt (`.`) in plaats van een dubbele punt.<br><br>`hh TT`–Alleen uur en AM/PM.<br><br>`hh "Hrs" mm "Mins"`–Toont de tijd met aangepaste tekst, zoals Hrs. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `Time( 0.375 )` | Retourneert `09:00:00` als de `TimeFormat` instelling `'hh:mm:ss'` is. |
| `Time( 35648.375 )` | Retourneert `09:00:00` als de geleverde `Timeformat` `'hh:mm:ss'` is. |
| `Time( 0.99999, 'hh-mm' )` | Retourneert `23-59` door de instelling `TimeFormat` te negeren, aangezien de notatie in de meting wordt geleverd. |

## Voorbeeld - Basisprincipes voor Time

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Het volgende veld in een gegevenstabel:
  - `TimeValue`
- De variabele `TimeFormat`, die de notatie `hh:mm:ss` gebruikt.

### Load-script

```
SET TimeFormat='hh:mm:ss';
Example:
Load * inline [
TimeValue
0.375
35648.375
0.99999
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `TimeValue`

Maak de volgende metingen:

- `=Time(TimeValue )`, om de waarde weer te geven als een tijd die is opgemaakt met behulp van de systeeminstelling.
- `=Time(TimeValue, 'hh-mm' )`, om de waarde weer te geven als een tijd volgens de notatie die in de functie is opgegeven.
- `=Time(TimeValue, 'hh:mm:ss' )`, om de waarde weer te geven als een tijd volgens de notatie die in de functie is opgegeven.
- `=Time(TimeValue, 'hh:mm TT' )`, om de waarde weer te geven als een tijd volgens de notatie die in de functie is opgegeven.
- `=Time(TimeValue, 'hh:mm' )`, om de waarde weer te geven als een tijd volgens de notatie die in de functie is opgegeven.
- `=Time(TimeValue, 'hh.mm.ss' )`, om de waarde weer te geven als een tijd volgens de notatie die in de functie is opgegeven.
- `=Time(TimeValue, 'hh TT' )`, om de waarde weer te geven als een tijd volgens de notatie die in de functie is opgegeven.
- `=Time(TimeValue, 'hh "Hrs" mm "Mins"' )`, om de waarde weer te geven als een tijd volgens de notatie die in de functie is opgegeven.

Resultatentabel

| TimeValue | Time (TimeValue) | Time (TimeValue, 'hh-mm') | Time (TimeValue, 'hh:mm:ss') | Time (TimeValue, 'hh:mm TT') | Time (TimeValue, 'hh:mm') | Time (TimeValue, 'hh.mm.ss') | Time (TimeValue, 'hh TT') | Time (TimeValue, 'hh "Hrs" mm "Mins"') |
|---|---|---|---|---|---|---|---|---|
| 0.375 | 09:00:00 | 09-00 | 09:00:00 | 09:00 AM | 09:00 | 09.00.00 | 09 AM | 09 Hrs 00 Mins |

| TimeValue | Time (TimeValue) | Time (TimeValue, 'hh-mm') | Time (TimeValue, 'hh:mm:ss') | Time (TimeValue, 'hh:mm TT') | Time (TimeValue, 'hh:mm') | Time (TimeValue, 'hh.mm.ss') | Time (TimeValue, 'hh TT') | Time (TimeValue, 'hh "Hrs" mm "Mins"') |
|---|---|---|---|---|---|---|---|---|
| 0.99999 | 23:59:59 | 23-59 | 23:59:59 | 11:59 PM | 23:59 | 23.59.59 | 11 PM | 23 Hrs 59 Mins |
| 35648.375 | 09:00:00 | 09-00 | 09:00:00 | 09:00 AM | 09:00 | 09.00.00 | 09 AM | 09 Hrs 00 Mins |

De resultaten laten zien hoe de functie `Time` de waarden kan weergeven op basis van de notatie die in de functie-uitdrukking is opgegeven.

## Voorbeeld - Basisprincipes voor Time met behulp van een andere TimeFormat instelling

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Het volgende veld in een gegevenstabel:
  - `TimeValue`
- De variabele `TimeFormat`, die de notatie `hh.mm.ss` gebruikt.

### Load-script

```
SET TimeFormat='hh.mm.ss';
Example:
Load * inline [
TimeValue
0.375
35648.375
0.99999
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `TimeValue`

Maak de volgende metingen:

- `=Time(TimeValue )`, om de waarde weer te geven als een tijd die is opgemaakt met behulp van de systeeminstelling.
- `=Time(TimeValue, 'hh-mm' )`, om de waarde weer te geven als een tijd volgens de notatie die in de functie is opgegeven.
- `=Time(TimeValue, 'hh:mm:ss' )`, om de waarde weer te geven als een tijd volgens de notatie die in de functie is opgegeven.
- `=Time(TimeValue, 'hh:mm TT' )`, om de waarde weer te geven als een tijd volgens de notatie die in de functie is opgegeven.
- `=Time(TimeValue, 'hh:mm' )`, om de waarde weer te geven als een tijd volgens de notatie die in de functie is opgegeven.
- `=Time(TimeValue, 'hh.mm.ss' )`, om de waarde weer te geven als een tijd volgens de notatie die in de functie is opgegeven.
- `=Time(TimeValue, 'hh TT' )`, om de waarde weer te geven als een tijd volgens de notatie die in de functie is opgegeven.
- `=Time(TimeValue, 'hh "Hrs" mm "Mins"' )`, om de waarde weer te geven als een tijd volgens de notatie die in de functie is opgegeven.

Resultatentabel

| TimeValue | Time (TimeValue) | Time (TimeValue, 'hh-mm') | Time (TimeValue, 'hh:mm:ss') | Time (TimeValue, 'hh:mm TT') | Time (TimeValue, 'hh:mm') | Time (TimeValue, 'hh.mm.ss') | Time (TimeValue, 'hh TT') | Time (TimeValue, 'hh "Hrs" mm "Mins"') |
|---|---|---|---|---|---|---|---|---|
| 0.375 | 09.00.00 | 09-00 | 09:00:00 | 09:00 AM | 09:00 | 09.00.00 | 09 AM | 09 Hrs 00 Mins |
| 0.99999 | 23.59.59 | 23-59 | 23:59:59 | 11:59 PM | 23:59 | 23.59.59 | 11 PM | 23 Hrs 59 Mins |
| 35648.375 | 09.00.00 | 09-00 | 09:00:00 | 09:00 AM | 09:00 | 09.00.00 | 09 AM | 09 Hrs 00 Mins |

De resultaten laten zien hoe de functie `Time` de waarden kan weergeven op basis van de notatie die in de functie-uitdrukking is opgegeven.

## Timestamp

**TimeStamp()** maakt een uitdrukking op als een datum- en tijdwaarde in de notatie voor tijdstempels die in de systeemvariabelen in het load-script voor gegevens of in het besturingssysteem is ingesteld, tenzij een opmaakreeks is opgegeven.

**Syntaxis:**

```
Timestamp(number[, format])
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **number** | Het getal dat moet worden opgemaakt. |
| **format** | Tekenreeks die beschrijft hoe de resulterende tekenreeks voor de tijdstempel moet worden opgemaakt. Bij weglating wordt gebruikgemaakt van de korte datumnotatie, tijdnotatie en decimaal scheidingsteken die zijn ingesteld in het besturingssysteem.<br><br>De functie `Timestamp` ondersteunt vele opmaakopties voor verschillende tijdstructuren, bijvoorbeeld:<br><br>`DD/MM/YYYY`: richt zich alleen op de datum (bijvoorbeeld: `15/11/2024`).<br><br>`YYYY-MM-DD hh:mm:ss`:geeft het volledige tijdstempel weer (bijvoorbeeld: `2024-11-15 13:30:45`).<br><br>`MMM DD, YYYY`: geeft een korte versie van de maand en het volledige jaar weer (bijvoorbeeld: `Nov 15, 2024`).<br><br>`hh:mm:ss TT`: toont alleen de tijd in 12-uursnotatie (bijvoorbeeld: `01:30:45 PM`)<br><br>`HH:mm:ss`: geeft de tijd in 24-uursnotatie weer (bijvoorbeeld: `13:30:45`)<br><br>`wwww, MMM DD YYYY`:toont de volledige weekdag en datum (bijvoorbeeld: `Friday, Nov 15 2024`)<br><br>`YYYY-MM-DD hh:mm:ss.fff`: inclusief milliseconden (bijvoorbeeld: `2024-11-15 13:30:45.123`). |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `Timestamp( 35648 )` | Retourneert `97-08-06 00:00:00` als de `TimestampFormat` `YY-MM-DD hh:mm:ss` is. |
| `Timestamp(35648,'YYYY-MM-DD hh.mm')` | Retourneert `1997-08-06 00.00` als de geleverde notatie `YYYY-MM-DD hh.mm` is. |
| `Timestamp( 35648,'M/D/YYYY h:mm:ss[.fff] TT')` | Retourneert `8/6/1997 12:00:00 AM` als de geleverde notatie `M/D/YYYY h:mm:ss[.fff] TT` is. |
| `Timestamp( 35648,'www, DD MMM YYYY hh:mm TT')` | Retourneert `Wed, 06 Aug 1997 12:00 AM` als de geleverde notatie `www, DD MMM YYYY hh:mm TT` is. |

## Voorbeeld - Basisprincipes voor Timestamp

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Het volgende veld in een gegevenstabel:
  - `TimeStampValue`
- De variabele `TimestampFormat`, die de notatie `YY-MM-DD hh:mm:ss` gebruikt.

### Load-script

```
SET TimestampFormat='YY-MM-DD hh:mm:ss';
Example:
Load * inline [
TimeStampValue
35648.375
35648
36648.875
36648.8756789
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `TimeStampValue`

Maak de volgende metingen:

- `=Timestamp(TimeStampValue)`, om de waarde weer te geven als een tijdstempelnotatie met behulp van de systeeminstelling.
- `=Timestamp(TimeStampValue,'YYYY-MM-DD hh.mm')`, om de waarde weer te geven als een tijdstempelnotatie volgens de notatie die in de functie is opgegeven.
- `=Timestamp(TimeStampValue,'M/D/YYYY h:mm:ss[.fff] TT')`, om de waarde weer te geven als een tijdstempelnotatie volgens de notatie die in de functie is opgegeven.
- `=Timestamp(TimeStampValue,'YYYY-MM-DD hh:mm:ss.fff')`, om de waarde weer te geven als een tijdstempelnotatie volgens de notatie die in de functie is opgegeven.
- `=Timestamp(TimeStampValue,'www, DD MMM YYYY hh:mm TT')`, om de waarde weer te geven als een tijdstempelnotatie volgens de notatie die in de functie is opgegeven.
- `=Timestamp(TimeStampValue,'wwww, MMM DD YYYY')`, om de waarde weer te geven als een tijdstempelnotatie volgens de notatie die in de functie is opgegeven.

Resultatentabel

| TimeStampValue | Timestamp (TimeStampValue) | Timestamp (TimeStampValue,'YYYY-MM-DD hh.mm' | Timestamp (TimeStampValue,'M/D/YYYY h:mm:ss[.fff] TT') | Timestamp (TimeStampValue,'YYYY-MM-DD hh:mm:ss.fff') | Timestamp (TimeStampValue,'www, DD MMM YYYY hh:mm TT') | Timestamp (TimeStampValue,'www, MMM DD YYYY') |
|---|---|---|---|---|---|---|
| 35648 | 97-08-06 00:00:00 | 1997-08-06 00.00 | 8/6/1997 12:00:00 AM | 1997-08-06 00:00:00.000 | Wed, 06 Aug 1997 12:00 AM | Wednesday, Aug 06 1997 |
| 35648.375 | 97-08-06 09:00:00 | 1997-08-06 09.00 | 8/6/1997 9:00:00 AM | 1997-08-06 09:00:00.000 | Wed, 06 Aug 1997 09:00 AM | Wednesday, Aug 06 1997 |
| 36648.875 | 00-05-02 21:00:00 | 2000-05-02 21.00 | 5/2/2000 9:00:00 PM | 2000-05-02 21:00:00.000 | Tue, 02 May 2000 09:00 PM | Tuesday, May 02 2000 |
| 36648.8756789 | 00-05-02 21:00:58 | 2000-05-02 21.00 | 5/2/2000 9:00:58 PM | 2000-05-02 21:00:58.657 | Tue, 02 May 2000 09:00 PM | Tuesday, May 02 2000 |

De resultaten laten zien hoe de functie `TimestampFormat` waarden kan weergeven met verschillende notatieparameters.

## Voorbeeld - Basisprincipes voor Timestamp met behulp van een andere TimestampFormat instelling

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Het volgende veld in een gegevenstabel:
  - `TimeStampValue`
- De variabele `TimestampFormat`, die de notatie `YY-MM-DD hh:mm:ss` gebruikt.

### Load-script

```
SET TimestampFormat='M/D/YY hh:mm:ss';
Example:
Load * inline [
TimeStampValue
```

```
35648.375
35648
36648.875
36648.8756789
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `TimeStampValue`

Maak de volgende metingen:

- `=Timestamp(TimeStampValue)`, om de waarde weer te geven als een tijdstempelnotatie met behulp van de systeeminstelling.
- `=Timestamp(TimeStampValue,'YYYY-MM-DD hh.mm')`, om de waarde weer te geven als een tijdstempelnotatie volgens de notatie die in de functie is opgegeven.
- `=Timestamp(TimeStampValue,'M/D/YYYY h:mm:ss[.fff] TT')`, om de waarde weer te geven als een tijdstempelnotatie volgens de notatie die in de functie is opgegeven.
- `=Timestamp(TimeStampValue,'YYYY-MM-DD hh:mm:ss.fff')`, om de waarde weer te geven als een tijdstempelnotatie volgens de notatie die in de functie is opgegeven.
- `=Timestamp(TimeStampValue,'www, DD MMM YYYY hh:mm TT')`, om de waarde weer te geven als een tijdstempelnotatie volgens de notatie die in de functie is opgegeven.
- `=Timestamp(TimeStampValue,'wwww, MMM DD YYYY')`, om de waarde weer te geven als een tijdstempelnotatie volgens de notatie die in de functie is opgegeven.

Resultatentabel

| TimeStampValue | Timestamp (TimeStampValue) | Timestamp (TimeStampValue,'YYYY-MM-DD hh.mm' | Timestamp (TimeStampValue,'M/D/YYYY h:mm:ss[.fff] TT') | Timestamp (TimeStampValue,'YYYY-MM-DD hh:mm:ss.fff') | Timestamp (TimeStampValue,'www, DD MMM YYYY hh:mm TT') | Timestamp (TimeStampValue,'wwww, MMM DD YYYY') |
|---|---|---|---|---|---|---|
| 35648 | 8/6/97 00:00:00 | 1997-08-06 00.00 | 8/6/1997 12:00:00 AM | 1997-08-06 00:00:00.000 | Wed, 06 Aug 1997 12:00 AM | Wednesday, Aug 06 1997 |
| 35648.375 | 8/6/97 09:00:00 | 1997-08-06 09.00 | 8/6/1997 9:00:00 AM | 1997-08-06 09:00:00.000 | Wed, 06 Aug 1997 09:00 AM | Wednesday, Aug 06 1997 |
| 36648.875 | 5/2/00 21:00:00 | 2000-05-02 21.00 | 5/2/2000 9:00:00 PM | 2000-05-02 21:00:00.000 | Tue, 02 May 2000 09:00 PM | Tuesday, May 02 2000 |
| 36648.8756789 | 5/2/00 21:00:58 | 2000-05-02 21.00 | 5/2/2000 9:00:58 PM | 2000-05-02 21:00:58.657 | Tue, 02 May 2000 09:00 PM | Tuesday, May 02 2000 |

De resultaten laten zien hoe de functie `TimestampFormat` waarden kan weergeven met verschillende notatieparameters.

## Voorbeeld - De tijdstempelnotatie van klantenorders wijzigen

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Customer`
  - `CustomerName`
  - `OrderTimestamp`

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `OrderID`
- `CustomerName`

Maak de volgende meting:

- `=Timestamp(OrderTimestamp, 'WWWW, DD MMM YYYY hh:mm TT')`, om het tijdstempel van de bestelling weer te geven in een notatie die de dag van de week, de datum en de tijd weergeeft.

Resultatentabel

| OrderID | CustomerName | Timestamp(OrderTimestamp, 'WWWW, DD MMM YYYY hh:mm TT') |
|---|---|---|
| 1 | Alice | Tuesday, 20 Feb 2024 11:00 AM |
| 2 | Bob | Friday, 01 Mar 2024 03:30 PM |

| OrderID | CustomerName | Timestamp(OrderTimestamp, 'WWWW, DD MMM YYYY hh:mm TT') |
|---|---|---|
| 3 | Charlie | Friday, 12 Apr 2024 09:45 AM |
| 4 | David | Tuesday, 21 May 2024 09:10 PM |
| 5 | Eve | Sunday, 30 Jun 2024 02:25 PM |

Met de functie `Timestamp` worden de tijdstempelgegevens nu opgemaakt met de dag van de week en de nieuwe notatie voor de datum en tijd.

## 8.13 Algemene numerieke functies

In deze algemene numerieke functies zijn de argumenten uitdrukkingen waarbij **x** moet worden geïnterpreteerd als een reëel getal. Alle functies kunnen zowel worden gebruikt in scripts voor het laden van gegevens als in diagramuitdrukkingen.

### Overzicht van algemene numerieke functies

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

bitcount

**BitCount()** retourneert hoeveel bits in het binaire equivalent van een getal zijn ingesteld op 1. De functie retourneert dus het aantal setbits in **integer_number**, waarbij **integer_number** wordt geïnterpreteerd als een ondertekend 32-bits geheel getal.

```
BitCount(integer_number)
```

div

**Div()** retourneert het gehele deel van de rekenkundige deling van het eerste argument door het tweede argument. Beide parameters worden geïnterpreteerd als reële getallen. Ze hoeven dus geen gehele getallen te zijn.

```
Div (integer_number1, integer_number2)
```

fabs

**Fabs()** retourneert de absolute waarde van **x**. Het resultaat is een positief getal.

```
Fabs (x)
```

fact

**Fact()** retourneert de faculteit van een positief geheel getal **x**.

```
Fact (x)
```

frac

**Frac()** retourneert het breukgedeelte van **x**.

```
Frac (x)
```

sign

**Sign()** retourneert 1, 0 of -1, afhankelijk van het feit of **x** een positief getal, 0 of een negatief getal is.

```
Sign (x)
```

## Combinatie- en permutatiefuncties

combin

**Combin()** retourneert het aantal combinaties van **q** elementen dat kan worden gekozen uit een groep van **p** elementen. Zoals weergegeven door de formule: `Combin(p,q) = p! / q!(p-q)!` De volgorde waarin de items worden geselecteerd is niet van belang.

```
Combin (p, q)
```

permut

**Permut()** retourneert het aantal permutaties van **q** elementen dat kan worden geselecteerd uit een groep van **p** elementen. Zoals weergegeven door de formule: `Permut(p,q) = (p)! / (p - q)!` De volgorde waarin de items worden geselecteerd is van belang.

```
Permut (p, q)
```

## Modulofuncties

fmod

**fmod()** is een gegeneraliseerde modulo-functie die het gehele deel van de rekenkundige deling van het eerste argument (het deeltal) op het tweede argument (de deler) retourneert. Het resultaat is een reëel getal. Beide argumenten worden geïnterpreteerd als reële getallen. Ze hoeven dus geen gehele getallen te zijn.

```
Fmod (a, b)
```

mod

**Mod()** is een wiskundige modulo-functie die de niet-negatieve restwaarde van een deling van gehele getallen retourneert. Het eerste argument is het deeltal en het tweede argument de deler. Beide argumenten moeten gehele getallen zijn.

```
Mod (integer_number1, integer_number2)
```

## Pariteitsfuncties

even

**Even()** retourneert True (-1), als **integer_number** een even geheel getal of nul is. Retourneert False (0), als **integer_number** een oneven geheel getal is en NULL als **integer_number** niet een geheel getal is.

```
Even (integer_number)
```

odd

**Odd()** retourneert True (-1), als **integer_number** een oneven geheel getal of nul is. Retourneert False (0), als **integer_number** een even geheel getal is en NULL als **integer_number** niet een geheel getal is.

```
Odd (integer_number)
```

## Afrondingsfuncties

ceil

**Ceil()** rondt een getal naar boven af op het dichtstbijzijnde veelvoud van de **step** die is verschoven met het **offset** -getal.

```
Ceil (x[, step[, offset]])
```

floor

**Floor()** rondt een getal naar beneden af op het dichtstbijzijnde veelvoud van de **step** die is verschoven met het **offset** -getal.

```
Floor (x[, step[, offset]])
```

round

**Round()** retourneert het resultaat van de afronding omhoog of omlaag naar het dichtstbijzijnde veelvoud van de **step** die is verschoven met het **offset** -getal.

```
Round ( x [ , stap [ , offset ]])
```

## BitCount

**BitCount()** retourneert hoeveel bits in het binaire equivalent van een getal zijn ingesteld op 1. De functie retourneert dus het aantal setbits in **integer_number**, waarbij **integer_number** wordt geïnterpreteerd als een ondertekend 32-bits geheel getal.

**Syntaxis:**

```
BitCount(integer_number)
```

**Retourgegevenstypen:** geheel getal

**Voorbeelden en resultaten:**

Voorbeelden en resultaten

| Voorbeelden | Resultaten |
|---|---|
| `BitCount ( 3 )` | 3 is binair 101, dus wordt 2 geretourneerd |
| `BitCount ( -1 )` | -1 is 64 enen in binaire code, dus wordt hierbij 64 geretourneerd |

## Ceil

**Ceil()** rondt een getal naar boven af op het dichtstbijzijnde veelvoud van de **step** die is verschoven met het **offset** -getal.

Vergelijk deze functie met de functie **floor**, waarbij ingevoerde cijfers naar beneden worden afgerond.

**Syntaxis:**

```
Ceil(x[, step[, offset]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| **x** | Invoergetal. |
| **step** | Increment van interval. De standaardwaarde is 1. |
| **offset** | Definieert de basis van de stapinterval. De standaardwaarde is 0. |

**Voorbeelden en resultaten:**

Voorbeelden en resultaten

| Voorbeelden | Resultaten |
|---|---|
| `Ceil(2.4 )` | retourneert 3<br>In dit voorbeeld is de grootte van de stap 1 en is de basis van de stapinterval 0.<br>De intervallen zijn ...0 < x <=1, 1 < x <= 2, **2< x <=3**, 3< x <=4... |
| `Ceil(4.2 )` | retourneert 5 |
| `Ceil(3.88 ,0.1)` | retourneert 3,9<br>In dit voorbeeld is de grootte van de interval 0,1 en is de basis van de interval 0.<br>De intervallen zijn ... 3.7 < x <= 3.8, **3.8 < x <= 3.9**, 3.9 < x <= 4.0... |
| `Ceil(3.88 ,5)` | retourneert 5 |
| `Ceil(1.1 ,1)` | retourneert 2 |
| `Ceil(1.1 ,1,0.5)` | retourneert 1,5<br>In dit voorbeeld is de grootte van de stap 1 en is de beginwaarde 0,5. Dit betekent dat de basis van de stapinterval 0,5 is en niet 0.<br>De intervallen zijn ...**0.5 < x <=1.5**, 1.5 < x <= 2.5, 2.5< x <=3.5, 3.5< x <=4.5... |
| `Ceil(1.1 ,1,-0.01)` | retourneert 1,99<br>De intervallen zijn ...-0.01< x <= 0.99, **0.99< x <= 1.99**, 1.99 < x <=2.99... |

## Combin

**Combin()** retourneert het aantal combinaties van **q** elementen dat kan worden gekozen uit een groep van **p** elementen. Zoals weergegeven door de formule: `Combin(p,q) = p! / q!(p-q)!` De volgorde waarin de items worden geselecteerd is niet van belang.

**Syntaxis:**

```
Combin(p, q)
```

**Retourgegevenstypen:** geheel getal

**Beperkingen:**

Elementen die niet een geheel getal zijn, worden afgekapt.

**Voorbeelden en resultaten:**

Voorbeelden en resultaten

| Voorbeelden | Resultaten |
|---|---|
| Hoeveel combinaties van 7 getallen kunnen worden gekozen uit een totaal van 35 lottogetallen?<br>`Combin( 35,7 )` | retourneert 6.724.520 |

## Div

**Div()** retourneert het gehele deel van de rekenkundige deling van het eerste argument door het tweede argument. Beide parameters worden geïnterpreteerd als reële getallen. Ze hoeven dus geen gehele getallen te zijn.

**Syntaxis:**

```
Div(integer_number1, integer_number2)
```

**Retourgegevenstypen:** geheel getal

**Voorbeelden en resultaten:**

Voorbeelden en resultaten

| Voorbeelden | Resultaten |
|---|---|
| `Div( 7,2 )` | retourneert 3 |
| `Div( 7.1,2.3 )` | retourneert 3 |
| `Div( 9,3 )` | retourneert 3 |
| `Div( -4,3 )` | retourneert -1 |
| `Div( 4,-3 )` | retourneert -1 |
| `Div( -4,-3 )` | retourneert 1 |

## Even

**Even()** retourneert True (-1), als **integer_number** een even geheel getal of nul is. Retourneert False (0), als **integer_number** een oneven geheel getal is en NULL als **integer_number** niet een geheel getal is.

**Syntaxis:**

```
Even(integer_number)
```

**Retourgegevenstypen:** Booleaanse waarde

**Voorbeelden en resultaten:**

Voorbeelden en resultaten

| Voorbeelden | Resultaten |
|---|---|
| `Even( 3 )` | Retourneert 0, False |
| `Even( 2 * 10 )` | Retourneert -1, True |
| `Even( 3.14 )` | Retourneert NULL |

## Fabs

**Fabs()** retourneert de absolute waarde van **x**. Het resultaat is een positief getal.

**Syntaxis:**

```
fabs(x)
```

**Retourgegevenstypen:** numeriek

**Voorbeelden en resultaten:**

Voorbeelden en resultaten

| Voorbeelden | Resultaten |
|---|---|
| `fabs( 2.4 )` | retourneert 2,4 |
| `fabs( -3.8 )` | retourneert 3,8 |

## Fact

**Fact()** retourneert de faculteit van een positief geheel getal **x**.

**Syntaxis:**

```
Fact(x)
```

**Retourgegevenstypen:** geheel getal

**Beperkingen:**

Als het getal **x** geen geheel getal is, wordt het afgekapt. Niet-positieve getallen retourneren NULL.

**Voorbeelden en resultaten:**

Voorbeelden en resultaten

| Voorbeelden | Resultaten |
|---|---|
| `Fact( 1 )` | retourneert 1 |
| `Fact( 5 )` | Retourneert 120 (1 * 2 * 3 * 4 * 5 = 120) |
| `Fact( -5 )` | Retourneert NULL |

## Floor

**Floor()** rondt een getal naar beneden af op het dichtstbijzijnde veelvoud van de **step** die is verschoven met het **offset** -getal.

Vergelijk deze functie met de functie **ceil**, waarbij ingevoerde getallen naar boven worden afgerond.

**Syntaxis:**

```
Floor(x[, step[, offset]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| **x** | Invoergetal. |
| **step** | Increment van interval. De standaardwaarde is 1. |
| **offset** | Definieert de basis van de stapinterval. De standaardwaarde is 0. |

**Voorbeelden en resultaten:**

Voorbeelden en resultaten

| Voorbeelden | Resultaten |
|---|---|
| `Floor(2.4)` | retourneert 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 <= x <1, 1 <= x < 2, **2<= x <3**, 3<= x <4.... |
| `Floor(4.2)` | retourneert 4 |

| Voorbeelden | Resultaten |
|---|---|
| `Floor(3.88 ,0.1)` | retourneert 3,8<br><br>In dit voorbeeld is de grootte van de interval 0,1 en is de basis van de interval 0.<br><br>De intervallen zijn ... 3.7 <= x < 3.8, **3.8 <= x < 3.9**, 3.9 <= x < 4.0... |
| `Floor(3.88 ,5)` | Retourneert 0 |
| `Floor(1.1 ,1)` | retourneert 1 |
| `Floor(1.1 ,1,0.5)` | retourneert 0,5<br><br>In dit voorbeeld is de grootte van de stap 1 en is de beginwaarde 0,5. Dit betekent dat de basis van de stapinterval 0,5 is en niet 0.<br><br>De intervallen zijn ...**0.5 <= x <1.5**, 1.5 <= x < 2.5, 2.5<= x <3.5,... |

## Fmod

**fmod()** is een gegeneraliseerde modulo-functie die het gehele deel van de rekenkundige deling van het eerste argument (het deeltal) op het tweede argument (de deler) retourneert. Het resultaat is een reëel getal. Beide argumenten worden geïnterpreteerd als reële getallen. Ze hoeven dus geen gehele getallen te zijn.

**Syntaxis:**

```
fmod(a, b)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| **a** | Deeltal |
| **b** | Deler |

**Voorbeelden en resultaten:**

Voorbeelden en resultaten

| Voorbeelden | Resultaten |
|---|---|
| `fmod( 7,2 )` | retourneert 1 |
| `fmod( 7.5,2 )` | retourneert 1,5 |
| `fmod( 9,3 )` | retourneert 0 |

| Voorbeelden | Resultaten |
|---|---|
| `fmod( -4,3 )` | retourneert -1 |
| `fmod( 4,-3 )` | retourneert 1 |
| `fmod( -4,-3 )` | retourneert -1 |

## Frac

**Frac()** retourneert het breukgedeelte van **x**.

De fractie is zodanig geformuleerd dat `Frac(x ) + Floor(x ) = x`. In eenvoudige termen betekent dit dat het fractionele deel van een positief getal het verschil is tussen het getal (x) en het gehele getal dat aan het fractionele deel voorafgaat.

Bijvoorbeeld: Het fractionele deel van 11,43 = 11,43 - 11 = 0,43

Voor een negatief getal, bijvoorbeeld -1,4, Floor(-1.4) = -2, hetgeen het volgende resultaat oplevert:

Het fractionele deel van -1,4 = 1,4 - (-2) = -1,4 + 2 = 0,6

**Syntaxis:**

```
Frac(x)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| **x** | Getal waarvoor fractie wordt geretourneerd. |

**Voorbeelden en resultaten:**

Voorbeelden en resultaten

| Voorbeelden | Resultaten |
|---|---|
| `Frac( 11.43 )` | retourneert 0,43 |
| `Frac( -1.4 )` | retourneert 0,6 |
| Extraheer het tijdelement uit de numerieke representatie van een tijdstempel waardoor de datum wordt weggelaten.<br><br>`Time(Frac(44518.663888889))` | retourneert 3:56:00 PM |

## Mod

**Mod()** is een wiskundige modulo-functie die de niet-negatieve restwaarde van een deling van gehele getallen retourneert. Het eerste argument is het deeltal en het tweede argument de deler. Beide argumenten moeten gehele getallen zijn.

**Syntaxis:**

```
Mod(integer_number1, integer_number2)
```

**Retourgegevenstypen:** geheel getal

**Beperkingen:**

**integer_number2** moet groter zijn dan 0.

**Voorbeelden en resultaten:**

Voorbeelden en resultaten

| Voorbeelden | Resultaten |
|---|---|
| `Mod( 7,2 )` | retourneert 1 |
| `Mod( 7.5,2 )` | Retourneert NULL |
| `Mod( 9,3 )` | retourneert 0 |
| `Mod( -4,3 )` | retourneert 2 |
| `Mod( 4,-3 )` | Retourneert NULL |
| `Mod( -4,-3 )` | Retourneert NULL |

## Odd

**Odd()** retourneert True (-1), als **integer_number** een oneven geheel getal of nul is. Retourneert False (0), als **integer_number** een even geheel getal is en NULL als **integer_number** niet een geheel getal is.

**Syntaxis:**

```
Odd(integer_number)
```

**Retourgegevenstypen:** Booleaanse waarde

**Voorbeelden en resultaten:**

Voorbeelden en resultaten

| Voorbeelden | Resultaten |
|---|---|
| `Odd( 3 )` | Retourneert -1, True |
| `Odd( 2 * 10 )` | Retourneert 0, False |
| `Odd( 3.14 )` | Retourneert NULL |

## Permut

**Permut()** retourneert het aantal permutaties van **q** elementen dat kan worden geselecteerd uit een groep van **p** elementen. Zoals weergegeven door de formule: `Permut(p,q) = (p)! / (p - q)!` De volgorde waarin de items worden geselecteerd is van belang.

**Syntaxis:**

```
Permut(p, q)
```

**Retourgegevenstypen:** geheel getal

**Beperkingen:**

Argumenten die niet een geheel getal zijn, worden afgekapt.

**Voorbeelden en resultaten:**

Voorbeelden en resultaten

| Voorbeelden | Resultaten |
|---|---|
| Op hoeveel manieren kunnen de gouden, zilveren en bronzen medailles worden verdeeld na een finale op de 100 m met 8 deelnemers?<br>`Permut( 8,3 )` | retourneert 336 |

## Round

**Round()** retourneert het resultaat van de afronding omhoog of omlaag naar het dichtstbijzijnde veelvoud van de **step** die is verschoven met het **offset** -getal.

Als het getal dat moet worden afgerond zich precies in het midden van een interval bevindt, wordt er naar boven afgerond.

**Syntaxis:**

```
Round(x[, step[, offset]])
```

**Retourgegevenstypen:** numeriek

*Als u een getal met zwevend decimaalteken afrondt, krijgt u mogelijk onjuiste resultaten. Deze afrondingsfouten worden veroorzaakt doordat getallen met zwevend decimaalteken worden weergegeven door een eindig aantal binaire cijfers. Dit heeft tot gevolg dat er een berekening wordt uitgevoerd met een getal dat al is afgerond. Als deze afrondingsfouten van invloed zijn op uw werk, vermenigvuldigt u de getallen om ze om te zetten in gehele getallen voordat u afrondt.*

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| **x** | Invoergetal. |
| **step** | Increment van interval. De standaardwaarde is 1. |
| **offset** | Definieert de basis van de stapinterval. De standaardwaarde is 0. |

**Voorbeelden en resultaten:**

Voorbeelden en resultaten

| Voorbeelden | Resultaten |
|---|---|
| `Round(3.8 )` | retourneert 4<br>In dit voorbeeld is de grootte van de stap 1 en is de basis van de stapinterval 0.<br>De intervallen zijn ...0 <= x <1, 1 <= x < 2, 2<= x <3, **3<= x <4**... |
| `Round(3.8,4 )` | retourneert 4 |
| `Round(2.5 )` | Retourneert 3.<br>In dit voorbeeld is de grootte van de stap 1 en is de basis van de stapinterval 0.<br>De intervallen zijn ...0 <= x <1, 1 <= x <2, **2<= x <3**... |
| `Round(2,4 )` | Retourneert 4. Wordt naar boven afgerond omdat 2 precies de helft van de stapinterval van 4 is.<br>In dit voorbeeld is de grootte van de stap 4 en is de basis van de stapinterval 0.<br>De intervallen zijn ...**0 <= x <4**, 4 <= x <8, 8<= x <12... |
| `Round(2,6 )` | Retourneert 0. Wordt naar beneden afgerond omdat 2 minder dan de helft van de stapinterval van 6 is.<br>In dit voorbeeld is de grootte van de stap 6 en is de basis van de stapinterval 0.<br>De intervallen zijn ...**0 <= x <6**, 6 <= x <12, 12<= x <18... |
| `Round(3.88 ,0.1)` | retourneert 3,9<br>In dit voorbeeld is de grootte van de stap 0,1 en is de basis van de stapinterval 0.<br>De intervallen zijn ... 3.7 <= x <3.8, **3.8 <= x <3.9**, 3.9 <= x < 4.0... |

| Voorbeelden | Resultaten |
|---|---|
| `Round (3.88875,1/1000)` | Retourneert 3,889<br><br>In dit voorbeeld is de stapgrootte 0,001, waardoor het getal naar boven wordt afgerond op drie decimalen achter de komma. |
| `Round(3.88 ,5)` | retourneert 5 |
| `Round(1.1 ,1,0.5)` | retourneert 1,5<br><br>In dit voorbeeld is de grootte van de stap 1 en is de basis van de stapinterval 0,5.<br><br>De intervallen zijn ...**0.5 <= x <1.5**, 1.5 <= x <2.5, 2.5<= x <3.5... |

## Sign

**Sign()** retourneert 1, 0 of -1, afhankelijk van het feit of **x** een positief getal, 0 of een negatief getal is.

**Syntaxis:**

```
Sign(x)
```

**Retourgegevenstypen:** numeriek

**Beperkingen:**

Als er geen numerieke waarde wordt gevonden, wordt NULL geretourneerd.

**Voorbeelden en resultaten:**

Voorbeelden en resultaten

| Voorbeelden | Resultaten |
|---|---|
| `Sign( 66 )` | retourneert 1 |
| `Sign( 0 )` | retourneert 0 |
| `Sign( - 234 )` | retourneert -1 |

# 8.14 Geospatiale functies

Deze functies worden gebruikt voor het verwerken van geospatiale gegevens in kaartvisualisaties. Qlik Sense volgt de GeoJSON-specificaties voor geospatiale gegevens en ondersteunt het volgende:

- Punt
- Linestring
- Polygon
- Multipolygon

Voor meer informatie over GeoJSON-specificaties raadpleegt u:

≤ GeoJSON.org

## Overzicht van geospatiale functies

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

Er zijn twee categorieën geospatiale functies: aggregatie- en niet-aggregatiefuncties.

Aggregatiefuncties nemen een geometrieset (punten of gebieden) als invoer en geven een enkele geometrie als resultaat. Zo kunnen bijvoorbeeld meerdere gebieden worden samengevoegd, waarbij een enkele grens voor de aggregatie op de kaart kan worden getekend.

Niet-aggregatiefuncties nemen een enkele geometrie en retourneren één geometrie. Als bijvoorbeeld voor de functie \ GeoGetPolygonCenter(), de grensgeometrie van het ene gebied is ingesteld als invoer, wordt de puntgeometrie (lengtegraad en breedtegraad) voor het midden van dat gebied geretourneerd.

De volgende functies zijn aggregatiefuncties:

### GeoAggrGeometry

**GeoAggrGeometry()** wordt gebruikt voor het aggregeren van een aantal gebieden tot een groter gebied, bijvoorbeeld het aggregeren van een aantal subregio's tot een regio.

```
GeoAggrGeometry (field_name)
```

### GeoBoundingBox

**GeoBoundingBox()** wordt gebruikt in scripts voor aggregatie van een geometrie in een gebied en voor berekening van het kleinste kader dat alle coördinaten bevat.

```
GeoBoundingBox (field_name)
```

### GeoCountVertex

**GeoCountVertex()** wordt gebruikt voor het zoeken van het aantal hoekpunten dat een polygoongeometrie bevat.

```
GeoCountVertex(field_name)
```

### GeoInvProjectGeometry

**GeoInvProjectGeometry()** wordt voor aggregatie van een geometrie in een gebied en voor toepassing van de inverse van een projectie.

```
GeoInvProjectGeometry(type, field_name)
```

### GeoProjectGeometry

**GeoProjectGeometry()** wordt voor aggregatie van een geometrie in een gebied en voor toepassing van een projectie.

```
GeoProjectGeometry(type, field_name)
```

### GeoReduceGeometry

**GeoReduceGeometry()** wordt gebruikt om het aantal hoekpunten van een geometrie te verminderen, en om meerdere gebieden te aggregeren tot één gebied waarbij wel de grenslijnen van de afzonderlijke gebieden worden getoond.

```
GeoReduceGeometry (geometry)
```

De volgende functies zijn niet-aggregatiefuncties:

### GeoGetBoundingBox

**GeoGetBoundingBox()** wordt gebruikt in scripts en diagramuitdrukkingen om het kleinste geospatiale kader te berekenen dat alle coördinaten van een geometrie bevat.

```
GeoGetBoundingBox (geometry)
```

### GeoGetPolygonCenter

**GeoGetPolygonCenter()** wordt gebruikt in scripts en diagramuitdrukkingen voor het berekenen en retourneren van het middelpunt van een geometrie.

```
GeoGetPolygonCenter (geometry)
```

### GeoMakePoint

**GeoMakePoint()** wordt gebruikt in scripts en diagramuitdrukkingen om een punt met lengtegraad en breedtegraad te maken en aan te duiden.

```
GeoMakePoint (lat_field_name, lon_field_name)
```

### GeoProject

**GeoProject()** wordt gebruikt in scripts en diagramuitdrukkingen voor toepassing van een projectie op een geometrie.

```
GeoProject (type, field_name)
```

## GeoAggrGeometry

**GeoAggrGeometry()** wordt gebruikt voor het aggregeren van een aantal gebieden tot een groter gebied, bijvoorbeeld het aggregeren van een aantal subregio's tot een regio.

**Syntaxis:**

```
GeoAggrGeometry(field_name)
```

**Retourgegevenstypen:** tekenreeks

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| field_name | Een veld dat of uitdrukking die verwijst naar een veld dat de weer te geven geometrie bevat. Dit kan een punt (of reeks van punten) zijn met lengtegraad en breedtegraad, of een gebied. |

Gewoonlijk kan **GeoAggrGeometry()** worden gebruikt voor het combineren van geospatiale grensgegevens. U beschikt bijvoorbeeld over postcodegebieden voor randgemeenten in een stad en de verkoopomzet voor elk gebied. Als het rayon van een verkoper verschillende postcodegebieden omvat, kan het handig zijn de totale omzet per verkooprayon te presenteren in plaats van afzonderlijke gebieden en de resultaten weer te geven op een kaart met kleurarceringen.

Met **GeoAggrGeometry()** kan de aggregatie van de geometrieën van de afzonderlijke randgemeenten worden berekend en de geometrie voor het samengevoegde rayon worden gegenereerd in het gegevensmodel. Als vervolgens de grenzen van het verkooprayon worden aangepast, worden de nieuwe samengevoegde grenzen en omzetgegevens weergegeven op de kaart als de gegevens opnieuw worden geladen.

**GeoAggrGeometry()** is een aggregatiefunctie. Wanneer u het in een script gebruikt, is een **LOAD**-instructie met de clausule **Sorteren op** vereist.

*De grenslijnen van kaarten die zijn gemaakt met **GeoAggrGeometry()** zijn de grenslijnen van de samengevoegde gebieden. Als u de afzonderlijke grenslijnen van de vooraf geaggregeerde gebieden wilt weergeven, gebruikt u **GeoReduceGeometry()**.*

Voorbeelden:
In dit voorbeeld wordt een KML-bestand met regiogegevens geladen. Daarna wordt een tabel geladen met de samengevoegde gegevens.

```
[MapSource]: LOAD [world.Name], [world.Point], [world.Area] FROM [lib://Downloads/world.kml]
(kml, Tabel is [World.shp/Features]); Map: LOAD world.Name, GeoAggrGeometry(world.Area) as
[AggrArea] resident MapSource Group By world.Name;

 Drop Table MapSource;
```

## GeoBoundingBox

**GeoBoundingBox()** wordt gebruikt in scripts voor aggregatie van een geometrie in een gebied en voor berekening van het kleinste kader dat alle coördinaten bevat.

Een GeoBoundingBox wordt weergegeven als een lijst met vier waarden: links, rechts, boven, beneden.

**Syntaxis:**

```
GeoBoundingBox(field_name)
```

**Retourgegevenstypen:** tekenreeks

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| field_name | Een veld dat of uitdrukking die verwijst naar een veld dat de weer te geven geometrie bevat. Dit kan een punt (of reeks van punten) zijn met lengtegraad en breedtegraad, of een gebied. |

GeoBoundingBox() aggregeert een reeks van geometrieën en rertourneert vier coördinaten voor de kleinste rechthoek die alle coördinaten van die geaggregeerde geometrie bevat.

Als u het resultaat wilt visualiseren op een kaart, brengt u de resulterende reeks van vier coördinaten over naar een polygoonvorm, tagt u het overgedragen veld met een polygoonvorm en sleept u dat veld in het kaartobject. De rechthoekige vakken worden vervolgens weergegeven in de kaartvisualisatie.

## GeoCountVertex

**GeoCountVertex()** wordt gebruikt voor het zoeken van het aantal hoekpunten dat een polygoongeometrie bevat.

**Syntaxis:**

```
GeoCountVertex(field_name)
```

**Retourgegevenstypen:** geheel getal

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| field_name | Een veld dat of uitdrukking die verwijst naar een veld dat de weer te geven geometrie bevat. Dit kan een punt (of reeks van punten) zijn met lengtegraad en breedtegraad, of een gebied. |

## GeoGetBoundingBox

**GeoGetBoundingBox()** wordt gebruikt in scripts en diagramuitdrukkingen om het kleinste geospatiale kader te berekenen dat alle coördinaten van een geometrie bevat.

Een geospatiaal kader, dat is gemaakt met de functie GeoBoundingBox(), wordt weergegeven als een lijst met vier waarden: links, rechts, boven, beneden.

**Syntaxis:**

```
GeoGetBoundingBox(field_name)
```

**Retourgegevenstypen:** tekenreeks

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| field_name | Een veld dat of uitdrukking die verwijst naar een veld dat de weer te geven geometrie bevat. Dit kan een punt (of reeks van punten) zijn met lengtegraad en breedtegraad, of een gebied. |

*Maak geen gebruik van de **Group by**-clausule in de editor voor het laden van gegevens bij deze en andere niet-aggregerende geospatiale functies, omdat dit tot een fout bij het laden leidt.*

## GeoGetPolygonCenter

**GeoGetPolygonCenter()** wordt gebruikt in scripts en diagramuitdrukkingen voor het berekenen en retourneren van het middelpunt van een geometrie.

In sommige gevallen moet een punt worden getekend in plaats van een kleurarcering op een kaart. Als de bestaande geospatiale gegevens niet beschikbaar zijn in de vorm van gebiedsgeometrie (bijvoorbeeld een grens), gebruikt u **GeoGetPolygonCenter()** om een set lengte- en breedtegraden voor op te halen voor het midden van het gebied.

**Syntaxis:**

```
GeoGetPolygonCenter(field_name)
```

**Retourgegevenstypen:** tekenreeks

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| field_name | Een veld dat of uitdrukking die verwijst naar een veld dat de weer te geven geometrie bevat. Dit kan een punt (of reeks van punten) zijn met lengtegraad en breedtegraad, of een gebied. |

*Maak geen gebruik van de **Group by**-clausule in de editor voor het laden van gegevens bij deze en andere niet-aggregerende geospatiale functies, omdat dit tot een fout bij het laden leidt.*

## GeoInvProjectGeometry

**GeoInvProjectGeometry()** wordt voor aggregatie van een geometrie in een gebied en voor toepassing van de inverse van een projectie.

**Syntaxis:**

```
GeoInvProjectGeometry(type, field_name)
```

**Retourgegevenstypen:** tekenreeks

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| type | Projectietype dat wordt gebruikt bij het transformeren van de geometrie van de kaart. Hierbij zijn twee waarden mogelijk: 'eenheid' (standaard), hetgeen in een 1:1-projectie, of 'mercator', waarbij de standaard Mercator-projectie wordt gebruikt. |
| field_name | Een veld dat of uitdrukking die verwijst naar een veld dat de weer te geven geometrie bevat. Dit kan een punt (of reeks van punten) zijn met lengtegraad en breedtegraad, of een gebied. |

Voorbeeld:

Scriptvoorbeeld

| Voorbeeld | Resultaat |
|---|---|
| In een `Load`-opdracht: `GeoInvProjectGeometry ('mercator',AreaPolygon) as InvProjectGeometry` | De geometrie die is geladen als **AreaPolygon** wordt getransformeerd met de inverse transformatie van de Mercator-projectie en opgeslagen als **InvProjectGeometry** voor gebruik in visualisaties. |

## GeoMakePoint

**GeoMakePoint()** wordt gebruikt in scripts en diagramuitdrukkingen om een punt met lengtegraad en breedtegraad te maken en aan te duiden. Door GeoMakePoint worden punten teruggegeven in volgorde van lengte- en breedtegraad.

**Syntaxis:**

```
GeoMakePoint(lat_field_name, lon_field_name)
```

**Retourgegevenstypen:** tekenreeks, opgemaakt [lengtegraad, breedtegraad]

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| lat_field_name | Een veld dat of uitdrukking die verwijst naar een veld die de breedtegraad van het punt aangeeft. |
| lon_field_name | Een veld dat of uitdrukking die verwijst naar een veld die de lengtegraad van het punt aangeeft. |

*Maak geen gebruik van de **Group by**-clausule in de editor voor het laden van gegevens bij deze en andere niet-aggregerende geospatiale functies, omdat dit tot een fout bij het laden leidt.*

## GeoProject

**GeoProject()** wordt gebruikt in scripts en diagramuitdrukkingen voor toepassing van een projectie op een geometrie.

**Syntaxis:**

```
GeoProject(type, field_name)
```

**Retourgegevenstypen:** tekenreeks

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| type | Projectietype dat wordt gebruikt bij het transformeren van de geometrie van de kaart. Hierbij zijn twee waarden mogelijk: 'eenheid' (standaard), hetgeen in een 1:1-projectie, of 'mercator', waarbij de Mercator-projectie voor het web wordt gebruikt. |
| field_name | Een veld dat of uitdrukking die verwijst naar een veld dat de weer te geven geometrie bevat. Dit kan een punt (of reeks van punten) zijn met lengtegraad en breedtegraad, of een gebied. |

*Maak geen gebruik van de **Group by**-clausule in de editor voor het laden van gegevens bij deze en andere niet-aggregerende geospatiale functies, omdat dit tot een fout bij het laden leidt.*

Voorbeeld:

Scriptvoorbeelden

| Voorbeeld | Resultaat |
|---|---|
| In een `Load`-opdracht: `GeoProject ('mercator',Area) as GetProject` | De Mercator-projectie wordt toegepast op de geometrie die is geladen als **Area**, en het resultaat wordt opgeslagen als **GetProject**. |

## GeoProjectGeometry

**GeoProjectGeometry()** wordt voor aggregatie van een geometrie in een gebied en voor toepassing van een projectie.

**Syntaxis:**

```
GeoProjectGeometry(type, field_name)
```

**Retourgegevenstypen:** tekenreeks

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| type | Projectietype dat wordt gebruikt bij het transformeren van de geometrie van de kaart. Hierbij zijn twee waarden mogelijk: 'eenheid' (standaard), hetgeen in een 1:1-projectie, of 'mercator', waarbij de Mercator-projectie voor het web wordt gebruikt. |
| field_name | Een veld dat of uitdrukking die verwijst naar een veld dat de weer te geven geometrie bevat. Dit kan een punt (of reeks van punten) zijn met lengtegraad en breedtegraad, of een gebied. |

Voorbeeld:

| Voorbeeld | Resultaat |
|---|---|
| In een `Load`-opdracht: `GeoProjectGeometry ('mercator',AreaPolygon) as ProjectGeometry` | De geometrie die is geladen als **AreaPolygon** wordt getransformeerd met de Mercator-projectie en opgeslagen als **ProjectGeometry** voor gebruik in visualisaties. |

## GeoReduceGeometry

**GeoReduceGeometry()** wordt gebruikt om het aantal hoekpunten van een geometrie te verminderen, en om meerdere gebieden te aggregeren tot één gebied waarbij wel de grenslijnen van de afzonderlijke gebieden worden getoond.

**Syntaxis:**

```
GeoReduceGeometry(field_name[, value])
```

**Retourgegevenstypen:** tekenreeks

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| field_name | Een veld dat of uitdrukking die verwijst naar een veld dat de weer te geven geometrie bevat. Dit kan een punt (of reeks van punten) zijn met lengtegraad en breedtegraad, of een gebied. |
| value | De mate van reductie die moet worden toegepast op de geometrie. Het bereik loopt van 0 tot 1, waarbij 0 geen reductie is en 1 maximale reductie van de hoekpunten.<br> *Als u een value kiest van 0,9 of hoger voor een complexe gegevensverzameling, kan het aantal hoekpunten dusdanig worden gereduceerd dat de visuele representatie onnauwkeurig is.* |

**GeoReduceGeometry()** voert ook een functie uit die vergelijkbaar is met **GeoAggrGeometry()**. Een aantal gebieden wordt namelijk geaggregeerd tot één gebied. Het verschil is dat afzonderlijke grenslijnen uit de vooraf geaggregeerde gegevens op de kaart worden weergegeven als u **GeoReduceGeometry()** gebruikt.

**GeoReduceGeometry()** is een aggregatiefunctie. Wanneer u het in een script gebruikt, is een **LOAD**-instructie met de clausule **Sorteren op** vereist.

Voorbeelden:
In dit voorbeeld wordt een KML-bestand met regiogegevens geladen. Daarna wordt een tabel geladen met de verminderde samengevoegde gegevens.

```
[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 Interpretatiefuncties

De interpretatiefuncties evalueren de inhoud van invoertekstvelden of uitdrukkingen en leggen een specifieke gegevensopmaak op aan de resulterende numerieke waarde. Met deze functies kunt u de opmaak opgeven van het getal, in overeenstemming met het gegevenstype, met inbegrip van eigenschappen als: scheidingsteken voor decimalen, scheidingsteken voor duizendtallen en datumnotatie.

Alle interpretatiefuncties retourneren een dubbele waarde met zowel de tekenreeks als de getalwaarde, maar kunnen worden beschouwd als een conversie van tekenreeks naar getal. De functies nemen de tekstwaarde van de invoeruitdrukking en genereren een getal dat de tekenreeks vertegenwoordigt.

De opmaakfuncties, daarentegen, doen het omgekeerde: zij nemen numerieke uitdrukkingen en evalueren deze als tekenreeksen, waarbij de weergave-indeling van de resulterende tekst wordt opgegeven.

Als geen interpretatiefuncties worden gebruikt, worden de gegevens in Qlik Sense geïnterpreteerd als een combinatie van getallen, datums, tijden, tijdstempels en tekenreeksen. Daarbij wordt uitgegaan van de standaardinstellingen voor de notatie van getallen, datum en tijd zoals die in de scriptvariabelen en het besturingssysteem zijn gedefinieerd.

Alle interpretatiefuncties kunnen zowel worden gebruikt in scripts voor het laden van gegevens als in diagramuitdrukkingen.

> *Alle getallen moeten worden weergegeven met een decimale punt als scheidingsteken voor decimalen.*

### Overzicht van interpretatiefuncties

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

**Date#**
**Date#** evalueert een uitdrukking als een datum in de opmaak die is gespecificeerd in het tweede argument, indien opgegeven. Als de opmaakcode is weggelaten, wordt de standaard datumnotatie die is ingesteld in het besturingssysteem gebruikt.

```
Date#(text[, format])
```

**Interval#**
**Interval#()** evalueert een tekstuitdrukking als een tijdsinterval in de opmaak die standaard is ingesteld in het besturingssysteem of in de opmaak die is gespecificeerd in het tweede argument, indien opgegeven.

```
Interval#(text[, format])
```

### Money#

**Money#()** converteert een teksttekenreeks naar een geldwaarde, in de indeling die is ingesteld in het load-script of in het besturingssysteem, tenzij een opmaakreeks is opgegeven. Aangepaste scheidingstekens voor decimalen en scheidingstekens voor duizendtallen zijn optionele parameters.

```
Money#(text[, format[, dec_sep[, thou_sep ] ] ])
```

### Num#

**Num#()** interpreteert een tekenreeks als een numerieke waarde, wat wil zeggen dat deze functie de invoerreeks converteert naar een getal met gebruikmaking van de opmaak die is opgegeven in de tweede parameter. Als de tweede parameter wordt weggelaten, worden de scheidingstekens voor decimalen en duizendtallen gebruikt die zijn ingesteld in het load-script voor gegevens. Aangepaste scheidingstekens voor decimalen en scheidingstekens voor duizendtallen zijn optionele parameters.

```
Num#(text[ , format[, dec_sep[ , thou_sep]]])
```

### Text

**Text()** dwingt af dat de uitdrukking wordt behandeld als tekst, ook als een numerieke interpretatie mogelijk is.

```
Text(expr)
```

### Time#

**Time#()** evalueert een uitdrukking als een tijdwaarde in de tijdnotatie die in het load-script voor gegevens of in het besturingssysteem is ingesteld, tenzij een opmaakreeks is opgegeven.zijn.

```
Time#(text[, format])
```

### Timestamp#

**Timestamp#()** evalueert een uitdrukking als een datum- en tijdwaarde in de notatie voor tijdstempels die in het load-script voor gegevens of in het besturingssysteem is ingesteld, tenzij een opmaakreeks is opgegeven.

```
Timestamp# (text[, format])
```

---

**Zie ook:**

p *Opmaakfuncties (page 1388)*

## Date#

**Date#** evalueert een uitdrukking als een datum in de opmaak die is gespecificeerd in het tweede argument, indien opgegeven.

**Syntaxis:**

```
Date#(text[, format])
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De teksttekenreeks die moet worden geëvalueerd. |
| **format** | Tekenreeks die het formaat beschrijft dat gebruikt moet worden om de teksttekenreeks te interpreteren. Als dit argument wordt weggelaten, wordt de datumopmaak gebruikt die is ingesteld in de systeemvariabelen in het load-script voor gegevens of in het besturingssysteem.<br><br>De functie `Date#` ondersteunt vele opmaakopties voor verschillende datumstructuren, bijvoorbeeld:<br><br>`DD/MM/YYYY`: dag/maand/jaar (bijvoorbeeld 15/02/2022)<br><br>`MM-DD-YYYY`: maand-dag-jaar met koppeltekens (bijvoorbeeld 02-15-2022)<br><br>`YYYY-MM-DD`: jaar-maand-dag, ISO-formaat (bijvoorbeeld 2022-02-15)<br><br>`D-MMM-YYYY`: dag, korte maandnaam, jaar, met koppeltekens (bijvoorbeeld 15-Feb-2022)<br><br>`MMMM YYYY`: lange maandnaam en jaar (bijvoorbeeld February 2022)<br><br>`MM/YYYY`: maand en jaar (bijvoorbeeld 02/2022)<br><br>`DD MMMM YYYY`: dag, lange maandnaam en jaar (bijvoorbeeld 15 February 2022) |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `Date#('8/7/97')` | Retourneert een geldige datum `8/7/97` als de `DateFormat` instelling `M/D/YY`. is. |
| `Date#('21.10.2023', 'DD.MM.YYYY' )` | Retourneert een geldige datum `21.10.2023` als de datuminstelling `DD.MM.YYYY` is. |
| `Date#('21.10.2023')` | Retourneert `NULL` als de `DateFormat` instelling `M/D/YY` is. |

## Voorbeeld - Basisprincipes voor Date#

Load-script

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Het volgende veld in een gegevenstabel:
    - `StringDate`
- De variabele `DateFormat`, die de notatie `M/D/YY` gebruikt.
- Een voorafgaande lading die een nieuw veld genaamd `Date` aanmaakt. De functies `Date#` en `Num` worden gebruikt om de tekenreeks om te zetten naar een datum, en de waarde vervolgens als getal te retourneren.

### Load-script

```
SET DateFormat='M/D/YY';
Example:
Load *,
Num(Date#(StringDate, 'D/M/YYYY')) as Date
;
LOAD * INLINE [
StringDate
13/12/2024
31/1/23
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `StringDate`
- `Date`

Maak de volgende metingen:

- `=Date(Date)`, om de waarde weer te geven als een datum die is opgemaakt volgens de variabele van de datumnotatie in het load-script.
- `=IsNum(StringDate)`, om te controleren of de waarde een tekenreeks is en niet als datum wordt geëvalueerd omdat de datumnotatie van de veldwaarden niet overeenkomt met de systeeminstelling.

Results table

| StringDate | Date | Date(Date) | IsNum(StringDate) |
|---|---|---|---|
| 13/12/2024 | 45639 | 12/13/24 | 0 |
| 31/1/23 | 44957 | 1/31/23 | 0 |

De resultaten laten zien dat het veld `Date` in kolom 2, dat door de vorige lading is gemaakt, de `StringDate` waarden heeft geconverteerd naar een datum die als een getal wordt uitgedrukt. De `Date` notatiefunctie in kolom 3 herkent de `Date` waarden als datums en maakt ze op volgens de `DateFormat` variabele in het load-script.

## Voorbeeld – Scenario met Date#

Load-script

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Customer`
  - `BirthDate`
- Een voorafgaande lading die een nieuw veld genaamd `ParsedBirthDate` aanmaakt.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `Customer`
- `BirthDate`
- `ParsedBirthDate`

Maak de volgende meting:

- `=Date(ParsedBirthDate,'DD/MM/YYYY')`, om de waarde als een opgemaakte datum weer te geven.

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 dit voorbeeld kunnen we zien dat in kolom 2 het veld `BirthDate` door het systeem als tekst wordt behandeld. De functie `Date#` die in de voorgaande lading wordt gebruikt, converteert `BirthDate` naar een herkenbare datum in kolom 3, `ParsedBirthDate`. De `Date` notatiefunctie in kolom 4 kan dan de `ParsedBirthDate` waarden in de notatie `DD/MM/YYYY` weergeven.

## Interval#

**Interval#()** evalueert een tekstuitdrukking als een tijdsinterval in de opmaak die standaard is ingesteld in het besturingssysteem of in de opmaak die is gespecificeerd in het tweede argument, indien opgegeven.

**Syntaxis:**

```
Interval#(text[, format])
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De teksttekenreeks die moet worden geëvalueerd. |
| **format** | Tekenreeks die de verwachte invoerindeling beschrijft bij het converteren van de tekenreeks naar een numeriek interval.<br>Bij weglating wordt gebruikgemaakt van de tijdnotatie en het scheidingsteken voor decimalen die zijn ingesteld in het besturingssysteem.<br><br>De functie `Interval#` ondersteunt vele opmaakopties voor verschillende tijdstructuren, bijvoorbeeld:<br><br>`hh:mm`–eenvoudige uren en minuten<br><br>`hh:mm:ss`–uren, minuten en seconden<br><br>`d hh:mm`–dagen, uren en minuten |

Met de functie **Interval#** wordt een tijdinterval in tekstvorm geconverteerd naar een numeriek equivalent.

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `Interval#( '12:30', 'hh:mm' )` | Retourneert de numerieke waarde `0.5208`, wat 12.5 uur is als het wordt omgezet naar een dagdeel. |
| `Interval#( '1 05', 'd hh' )` | Retourneert een numerieke weergave die gelijk is aan `1.2083` dagen. |

## Voorbeeld - Basisprincipes voor Interval#

Load-script

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Het volgende veld in een gegevenstabel:
  - `IntervalValue`
- Een voorgaande lading die de functie `Interval#` gebruikt om de tekst in het veld `IntervalValue` om te zetten naar een getal, dat als tijd herkend kan worden.

### Load-script

```
Example:
Load *, Interval#(IntervalValue, 'D hh') as FormattedIntervalValue;
Load * inline [
IntervalValue
1 05
2 15
1 09
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `IntervalValue`

Maak de volgende metingen:

- `=Num(FormattedIntervalValue)`, om de waarde als getal te bekijken nadat deze in het load-script is verwerkt met de functie `Interval#`.
- `=Interval(FormattedIntervalValue, 'dd:hh:mm')`, om de waarde weer te geven als een intervalnotatie die is opgemaakt volgens de notatie die in de functie is opgegeven.

Resultatentabel

| IntervalValue | Num(FormattedIntervalValue) | Interval(FormattedIntervalValue, 'dd:hh:mm' |
|---|---|---|
| 1 05 | 1.2083333333333 | 01:05:00 |
| 1 09 | 1.375 | 01:09:00 |
| 2 15 | 2.625 | 02:15:00 |

De resultaten laten zien hoe de functie `Interval#` waarden converteert naar een getal dat vervolgens kan worden weergegeven met behulp van een gespecificeerde notatieparameter. Bijvoorbeeld, in de kolom `IntervalValue` wordt de waarde `2 15` geconverteerd naar een getal in de voorgaande lading (veld`FormattedIntervalValue` ). De tweede kolom gebruikt de functie `Num` om deze geconverteerde waarde als getal weer te geven, `2.625`. De derde kolom gebruikt de functie `Interval` om dit getal op te maken in dagen, uren en minuten en weer te geven als `02:15:00`, of 2 dagen 15 uur.

## Voorbeeld – Scenario met Interval#

Load-script

### Overzicht

Een gegevensverzameling met supportticketgegevens bevat een veld dat tekst- en tijdgegevens mengt, bijvoorbeeld `2 days 04 hours`. We willen deze waarden omzetten in normale tijdswaarden.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `TicketID`
  - `ResolutionTime`
- Een voorgaande lading die de functie `Interval#` gebruikt om de tekst in het veld `ResolutionTime` om te zetten naar een getal, dat herkend kan worden in dagen en uren.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `TicketID`
- `ResolutionTime`

Maak de volgende metingen:

- `=Interval(FormattedResolutionTime, 'dd:hh')`, om de waarden om te zetten naar de gewenste notatie met een dubbele waarde.

- `=Num(FormattedResolutionTime)`, om de numerieke waarden van het veld weer te geven.

Resultatentabel

| TicketID | ResolutionTime | Interval (FormattedResolutionTime, 'dd hh') | Num (FormattedResolutionTime) |
|---|---|---|---|
| Ticket A | 2 dagen 04 uur | 02 04 | 2.1666666666667 |
| Ticket B | 2 dagen 06 uur | 02 06 | 2.25 |
| Ticket C | 3 dagen 10 uren | 03 10 | 3.4166666666667 |

De resultaten laten zien hoe u de functie `Interval#` kunt gebruiken om waarden om te zetten naar een getal en ze vervolgens weer te geven in een meer typische tijdsnotatie. Bijvoorbeeld, `Ticket B` heeft een resolutietijd van `2 days 06 hours`. Deze waarde wordt geconverteerd naar een getal in de voorgaande lading (veld `FormattedResolutionTime`). De nieuwe waarde wordt weergegeven als `02 06` (2 dagen en 6 uur) wanneer deze is opgemaakt met de functie `Interval` in kolom 3. In kolom 4 geeft de functie `Num` de numerieke waarde weer als `2.25`.

## Money#

**Money#()** converteert een teksttekenreeks naar een geldwaarde, in de indeling die is ingesteld in het load-script of in het besturingssysteem, tenzij een opmaakreeks is opgegeven. Aangepaste scheidingstekens voor decimalen en scheidingstekens voor duizendtallen zijn optionele parameters.

**Syntaxis:**

```
Money#(text[, format[, dec_sep [, thou_sep ] ] ])
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De teksttekenreeks die moet worden geëvalueerd. |

| Argument | Beschrijving |
|---|---|
| **format** | Tekenreeks die de verwachte invoerindeling beschrijft bij het converteren van de tekenreeks naar een numeriek interval.<br>Bij weglating wordt gebruikgemaakt van de waarde voor `MoneyFormat` die is ingesteld in het load-script voor gegevens.<br><br>De functie `Money#` ondersteunt vele opmaakopties voor verschillende geldbedragstructuren, bijvoorbeeld:<br><br>`$#,##0.00`: voegt een dollarteken en twee decimalen toe.<br><br>`€ #,##0`: voegt een euroteken en geen decimalen toe, met een spatie na het euroteken.<br><br>`¥#,##0`: voegt een Japans yen-teken zonder decimalen toe.<br><br>`$#,##0;($#,##0)`: plaatst negatieve waarden tussen haakjes.<br><br>`₣ #.##0,00`: voegt een Zwitserse frank-teken toe met punten als scheidingsteken voor duizendtallen en een komma voor decimalen (bijvoorbeeld `₣ 1.000,00`).<br><br>`R #,##0.00`: Zuid-Afrikaanse rand-teken met twee decimalen en spaties na het rand-symbool.<br><br>`$ #,##0.000`: voegt drie decimalen toe aan de dollarnotatie. |
| **dec_sep** | Tekenreeks die het scheidingsteken voor decimalen opgeeft. Bij weglating wordt gebruikgemaakt van de waarde voor `MoneyDecimalSep` die is ingesteld in het load-script voor gegevens.<br>De functie `Money#` ondersteunt vele notaties voor scheidingstekens voor decimalen, bijvoorbeeld:<br><br>Komma ' , ': Bijvoorbeeld `$1,000,00` voor duizend dollar.<br><br>Punt ' . ': bijvoorbeeld `€ 1,000.00`.<br><br>Koppelteken ' - ': als dit wordt gebruikt in plaats van de traditionele punt of komma als scheidingsteken voor decimalen, geeft het waarden weer als `¥1,000-00`. |

| Argument | Beschrijving |
|---|---|
| thou_sep | Tekenreeks die het scheidingsteken voor duizendtallen opgeeft. Bij weglating wordt gebruikgemaakt van de waarde voor `MoneyThousandSep` die is ingesteld in het load-script voor gegevens.<br><br>De functie `Money#` ondersteunt vele opties voor het scheidingsteken voor duizendtallen:<br><br>Punt `'.'`: gebruikt een punt als scheidingsteken voor duizendtallen (gebruikelijk in sommige Europese notaties), wordt weergegeven als `$1.000,00` indien gecombineerd met een komma als scheidingsteken voor decimalen.<br><br>Spatie `' '`: gebruikt een spatie voor het scheidingsteken voor duizendtallen, wat gebruikelijk is in sommige Europese landen. Wordt weergegeven als `€ 1 000.00`.<br><br>Onderstrepingsteken `'_'`: gebruikt een onderstrepingsteken voor het scheidingsteken voor duizendtallen, dat soms wordt gebruikt in technische of specifieke regionale notaties. Geeft waarden weer zoals `$1_000.00`. |

De functie `Money#` gedraagt zich zoals de functie `Num#`, maar haalt de standaardwaarden voor de scheidingstekens voor decimalen en duizendtallen uit de scriptvariabelen voor geldnotatie of de systeeminstellingen voor valuta.

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaten |
|---|---|
| `Money#('35 648,37 kr', '# ##0,00 kr',',',' ' )` | Retourneert `35 648,37 kr` als de `MoneyFormat` instelling `# ##0,00 kr` is. Het volgende moet in uw load-script worden ingesteld om dit correct te evalueren als getal:<br>• `SET MoneyDecimalSep=',';`<br>• `SET MoneyThousandSep=' ';` |
| `Money#( '$35,648.37', ' $#', '.', ',' )` | Retourneert `$35,648.37` als het volgende is ingesteld:<br>• `MoneyFormat`is `$#`<br>• `SET MoneyDecimalSep=',';`<br>• `SET MoneyThousandSep=' ';` |

## Voorbeeld - Basisprincipes voor Money#

Load-script

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `MoneyValue`.
- Een voorafgaande lading die de functie `Money#` gebruikt om de tekstwaarden in de gegevens om te zetten in een waarde die Qlik Sense herkent als een geldige waarde.

### Load-script

```
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 ';');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `MoneyValue`
- `FormattedMoneyValue`

Maak de volgende metingen:

- `=Money( FormattedMoneyValue, 'kr #,##0', '.' , ',' )`, om de waarde als geld weer te geven, geformatteerd met de argumenten die in de functie `Money` zijn ingesteld.
- `=Num(FormattedMoneyValue)` om de numerieke waarde te retourneren nadat deze verwerkt is door de functie `Money#`.

Results table

| MoneyValue | FormattedMoneyValue | Money( FormattedMoneyValue, 'kr #,##0', '.' , ',' ) | Num (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 |

De tabel met resultaten laat zien hoe waarden kunnen worden weergegeven met behulp van de volgende functies en notatie: `Money#`, `Money` en `Num`. De tweede kolom toont de dimensie `FormattedMoneyValue`, die werd berekend met de functie `Money#` om de waarde te wijzigen in geld met de indeling `'# ##0,00 kr',',',' '`. De derde kolom gebruikt de functie `Money` om de valutanotatie van `FormattedMoneyValue` te wijzigen in `'kr #,##0', '.' , ','`. De vierde kolom gebruikt de functie `Num` om de valutanotatie om de numerieke waarde van de dimensie `FormattedMoneyValue` weer te geven met behulp van de notatie in de systeemvariabele.

## Voorbeeld – Scenario met Money#

Load-script

### Overzicht

Een gegevensverzameling met verkooporders bevat velden die verschillende valuta's gebruiken. Dit voorbeeld laat zien hoe u de verkoopwaarden kunt scheiden en optellen op basis van valuta.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `TransactionID`
  - `Currency`
  - `Amount`
- De systeemvariabelen `MoneyDecimalSep`, `MoneyFormat` en `MoneyThousandSep` zijn voor demonstratiedoeleinden in een andere notatie ingesteld dan in de gegevens.
- Een voorafgaande lading die de velden `TransactionID`, `Currency` en `Amount` laadt en het volgende veld aanmaakt:
  - `AmountValue`, dit veld gebruikt de functie `Money#` om de ontvangen tekst om te zetten naar een waarde die het systeem begrijpt.

### Load-script

```
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"
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- TransactionID
- Currency
- Amount
- AmountValue

Maak de volgende meting:

- Num(AmountValue)

Resultatentabel

| 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 |

De resultaten laten zien hoe de functie Money# tekstwaarden converteert naar monetaire numerieke waarden die vervolgens kunnen worden geëvalueerd. De waarden in het veld Amount zijn bijvoorbeeld tekst en worden links uitgelijnd weergegeven. Zodra deze echter door de functie Money# zijn verwerkt om het nieuwe veld, AmountValue, te maken, herkent het systeem deze als monetaire numerieke waarden. De functie Num, die in de meest rechtse kolom wordt gebruikt, kan nu de monetaire waarden in AmountValue evalueren en de notatie veranderen van geld naar een getal.

## Num#

**Num#()** interpreteert een tekenreeks als een numerieke waarde, wat wil zeggen dat deze functie de invoerreeks converteert naar een getal met gebruikmaking van de opmaak die is opgegeven in de tweede parameter. Als de tweede parameter wordt weggelaten, worden de scheidingstekens voor decimalen en duizendtallen gebruikt die zijn ingesteld in het load-script voor gegevens. Aangepaste scheidingstekens voor decimalen en scheidingstekens voor duizendtallen zijn optionele parameters.

**Syntaxis:**

```
Num#(text[, format[, dec_sep [, thou_sep ] ] ])
```

**Retourgegevenstypen:** dubbele waarde

De functie **Num#()** retourneert een dubbele waarde met zowel de tekenreeks als de numerieke waarde. De functie neemt de tekstuele representatie van de invoeruitdrukking en genereert een getal. De notatie van het getal wordt niet gewijzigd: de uitvoer wordt in dezelfde notatie weergegeven als de invoer.

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De teksttekenreeks die moet worden geëvalueerd. |
| **format** | Tekenreeks die specificeert welke getalopmaak wordt gebruikt in de eerste parameter. Bij weglating worden de scheidingstekens voor decimalen en duizendtallen gebruikt die zijn ingesteld in het load-script voor gegevens.<br>De functie `Num#` ondersteunt vele opmaakopties voor verschillende numerieke structuren, bijvoorbeeld:<br>`#,##0`:geeft een getal zonder decimalen weer.<br>`#,##0.00`:geeft een getal met twee decimalen weer.<br>`##0.00%`: geeft een getal als een percentage met twee decimalen weer.<br>`#,##0;(#,##0)`: plaatst getallen tussen haakjes.<br>`$#,##0.00`: maakt getallen op als valuta met een $-symbool.<br>`0.00E+0`:gGeeft getallen in wetenschappelijke notatie weer. |
| **dec_sep** | Tekenreeks die het scheidingsteken voor decimalen opgeeft. Bij weglating wordt gebruikgemaakt van de waarde voor de variabele DecimalSep die is ingesteld in het load-script voor gegevens.<br>De functie `Num#` ondersteunt vele notaties voor scheidingstekens voor decimalen, bijvoorbeeld:<br>Komma ' , ': Bijvoorbeeld `$1,000,00` voor duizend dollar.<br>Punt ' . ': bijvoorbeeld `€ 1,000.00`.<br>Koppelteken ' - ': als dit wordt gebruikt in plaats van de traditionele punt of komma als scheidingsteken voor decimalen, geeft het waarden weer als `¥1,000-00`.<br>Ga voor meer informatie naar *DecimalSep (page 241)*. |

| Argument | Beschrijving |
|---|---|
| thou_sep | Tekenreeks die het scheidingsteken voor duizendtallen opgeeft. Bij weglating wordt gebruikgemaakt van de waarde voor de variabele ThousandSep die is ingesteld in het load-script voor gegevens.<br>De functie Num# ondersteunt vele opties voor het scheidingsteken voor duizendtallen:<br><br>Punt '.': gebruikt een punt als scheidingsteken voor duizendtallen (gebruikelijk in sommige Europese notaties), wordt weergegeven als $1.000,00 indien gecombineerd met een komma als scheidingsteken voor decimalen.<br><br>Spatie ' ': gebruikt een spatie voor het scheidingsteken voor duizendtallen, wat gebruikelijk is in sommige Europese landen. Wordt weergegeven als € 1 000.00.<br><br>Onderstrepingsteken '_': gebruikt een onderstrepingsteken voor het scheidingsteken voor duizendtallen, dat soms wordt gebruikt in technische of specifieke regionale notaties. Geeft waarden weer zoals $1_000.00.<br><br>Ga voor meer informatie naar *ThousandSep (page 279)*. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaten (geretourneerde tekenreekswaarde) |
|---|---|
| Num#( '35,648.31', '#', '.' , ',' ) | Retourneert een tekenreeks van 35,648.31 en numerieke waarde van 35648.31. |
| Num#( '35 648.312', '#', '.' , ',' ) | Retourneert een tekenreeks van 35 648.312 en numerieke waarde van 35648.312. |
| Num#( '35.648,3123', '#', '.' , ',' ) | Retourneert een tekenreeks van 35.648,3123 en numerieke waarde van NULL (-). |
| Num#( '35 648,31234', '#', '.' , ',' ) | Retourneert een tekenreeks van 35 648,31234 en numerieke waarde van NULL (-). |
| Num#('1,234.56', '#,##0.00') | Retourneert een tekenreeks van 1,234.56 en numerieke waarde van 1230.45. |
| Num#('$1,230.45', '$#,##0.00') | Retourneert een tekenreeks van $1,230.45 en numerieke waarde van 1234.56. |
| Num#('20220315', 'YYYYMMDD') | Retourneert een tekenreeks van 20220315 en numerieke waarde van 20220315. |

## Voorbeeld - Basisprincipes voor Num#

Load-script

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `ProductID`
  - `ProductName`
  - `SalesAmount`
- De standaardinstellingen voor `ThousandSep` en `DecimalSep` om dit voorbeeld te illustreren.
- Een voorgaande lading die de functie `Num#` gebruikt. Wanneer het wordt uitgevoerd, maakt het een extra veld genaamd `FormattedSalesAmount`.

### Load-script

```
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'
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `ProductID`
- `ProductName`
- `SalesAmount`
- `FormattedSalesAmount`

Maak de volgende meting:

- `=Num(FormattedSalesAmount)`, om de numerieke waarde van het veld weer te geven volgens de systeeminstellingen voor de getalnotatie.

Results table

| ProductID | ProductName | SalesAmount | FormattedSalesAmount | Num (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 dit voorbeeld kunt u zien hoe de functie `Num#` gegevens van `SalesAmount` van de oorspronkelijke opmaak van tekst omzet in een numerieke waarde.

## Voorbeeld – Scenario met Num#

Load-script

### Overzicht

Een gegevensverzameling bevat verkoopgegevens van verschillende landen in verschillende tekstopmaak. We willen de gegevens standaardiseren en in een tabel weergeven.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Country`
  - `Product`
  - `Amount`
- Een voorgaande lading die de functie `Num#` gebruikt om de waarden in het veld `Amount` om te zetten.

### Load-script

```
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'
];
```

**Resultaten**

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `Product`
- `Country`
- `Amount`

Maak de volgende meting:

- `Num(Amount)`, om de numerieke waarde van het veld `Amount` weer te geven volgens de systeeminstellingen voor de getalnotatie.

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 |

Wanneer u de kolom `Amount` vergelijkt met de kolom `Num(Amount)`, ziet u dat de gegevens die worden gepresenteerd met de functie `Num#` de onderliggende numerieke waarden wijzigen, maar de tekenreeksopmaak ongewijzigd laten. Bijvoorbeeld, alle waarden voor `Germany` in de kolom `Amount` gebruiken een komma als scheidingsteken voor decimalen, maar de onderliggende opmaak in `Num(Amount)` gebruikt een punt voor alle waarden.

## Text

**Text()** dwingt af dat de uitdrukking wordt behandeld als tekst, ook als een numerieke interpretatie mogelijk is.

**Syntaxis:**

```
Text (expr)
```

**Retourgegevenstypen:** tekenreeks

Argumenten

| Argument | Beschrijving |
|---|---|
| **expr** | Een uitdrukking die als tekst moet worden behandeld. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaten |
|---|---|
| `Text( 1234 )` | Retourneert `1234` als tekenreeks. Er is geen numerieke waarde. |
| `Text( pi( ) )` | Retourneert `3.1415926535898` als tekenreeks. Er is geen numerieke waarde. |

## Voorbeeld - Basisprincipes voor Text

Load-script

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `ID`
  - `PostalCode`
- Een voorafgaande lading die de functie `Text` gebruikt om een nieuw veld met de naam `PostalCode_Text` te maken om ervoor te zorgen dat de postcodes als tekst blijven en niet naar getallen worden geconverteerd.

### Load-script

```
Example:
LOAD
ID,
PostalCode,
Text(PostalCode) AS PostalCode_Text;
LOAD * INLINE [
ID, PostalCode
1, 12345
2, 54321
3, 67890
4, 11223
5, 33445
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `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 |

De uitvoer van de voorgaande lading gebruikt de functie `Text` om ervoor te zorgen dat numerieke waarden als tekst worden behandeld. U kunt zien dat de waarden in de kolom `PostalCode_Text` als tekst worden verwerkt en standaard links uitgelijnd worden weergegeven, terwijl de waarden in de kolom `PostalCode` getallen zijn en standaard rechts uitgelijnd worden weergegeven.

## Voorbeeld- Scenario met Text

Load-script

### Overzicht

Een gegevensverzameling bevat factuurgegevens van klanten die als numerieke waarden zijn ingevoerd, maar u wilt deze waarden als tekst behouden.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `InvoiceNumber`
  - `CustomerName`
- Een voorafgaande lading die de functie `Text` gebruikt om een nieuw veld met de naam `InvoiceNumber_Text` te maken om ervoor te zorgen dat de factuurnummers als tekst worden behandeld.

### Load-script

```
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
];
```

**Resultaten**

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `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 |

De resultaten laten zien dat de numerieke waarden in de kolom `InvoiceNumber` nu zijn omgezet naar tekstwaarden in de kolom `InvoiceNumber_Text` en links uitgelijnd worden weergegeven.

## Time#

**Time#()** evalueert een uitdrukking als een tijdwaarde in de tijdnotatie die in het load-script voor gegevens of in het besturingssysteem is ingesteld, tenzij een opmaakreeks is opgegeven.

**Syntaxis:**

```
time#(text[, format])
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De teksttekenreeks die moet worden geëvalueerd. |

| Argument | Beschrijving |
|---|---|
| **format** | Tekenreeks die het formaat beschrijft dat gebruikt moet worden om de teksttekenreeks te interpreteren. Bij weglating wordt gebruikgemaakt van de tijdnotatie en het scheidingsteken voor decimalen die zijn ingesteld in het besturingssysteem.<br><br>De functie `Time#` ondersteunt vele opmaakopties voor verschillende tijdstructuren, bijvoorbeeld:<br><br>`hh:mm:ss`–Maakt een tijdwaarde op in een standaard tijdnotatie en geeft uren, minuten en seconden met voorloopnullen weer.<br><br>`hh:mm TT`–Gebruikt een 12-uursnotatie met AM/PM achtervoegsel.<br><br>`hh:mm`-Toont alleen uren en minuten, zonder seconden.<br><br>`hh.mm.ss`–Tijdnotatie met verschillende scheidingstekens, zoals een punt (`.`) in plaats van een dubbele punt.<br><br>`hh TT`–Alleen uur en AM/PM.<br><br>`hh "Hrs" mm "Mins"`–Toont de tijd met aangepaste tekst, zoals Hrs. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `Time#('09:00:00')` | Retourneert `09:00:00` als de `TimeFormat` instelling `'hh:mm:ss'` is. |
| `Time#( '09:00:00','hh:mm:ss' )` | Retourneert `09:00:00` als de geleverde notatie `'hh:mm:ss'` is. |
| `Time#('03:25 PM', 'hh:mm TT')` | Retourneert `03:25 PM` als de geleverde notatie `'hh:mm TT'` is. |

## Voorbeeld - Basisprincipes voor Time#

Load-script

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Het volgende veld in een gegevenstabel:
  - `InputTime`
- De variabele `TimeFormat`, die de notatie `h:mm:ss TT` gebruikt.
- Een voorafgaande lading die de functie `Time#` gebruikt om de tekstwaarde van `InputTime` om te zetten in een tijd/numeriek veld genaamd `ParsedTime` dat het systeem kan gebruiken.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `InputTime`
- `ParsedTime`

Maak de volgende meting:

- `=Time(ParsedTime)`, om de tijd weer te geven in de notatie die standaard in het load-script staat.

Resultatentabel

| 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 |

De waarden in het veld `InputTime` worden als tekst weergegeven en zijn links uitgelijnd. Zodra de waarden in `InputTime` door de functie `Time#` zijn geparseerd, wordt het resulterende veld `ParsedTime` aangemaakt. De waarden in het nieuwe veld zijn rechts uitgelijnd, omdat ze nu worden behandeld als een geldige tijdswaarde. De derde kolom gebruikt de functie `Time` om de `ParsedTime` waarden te evalueren en maakt ze op volgens de `TimeFormat` in het load-script.

## Voorbeeld 5 – Scenario met Time#

Load-script

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Het volgende veld in een gegevenstabel:
  - `NoDelimiterTime`
- De variabele `TimeFormat`, die de notatie `hh.mm.ss` gebruikt.
- Een voorafgaande lading die de functie `Time#` gebruikt om de tekstwaarde van `NoDelimiterTime` om te zetten in een tijd/numeriek veld genaamd `ParsedNoDelimiterTime`.

### Load-script

```
SET TimeFormat='hh.mm.ss';
Example:
Load
NoDelimiterTime,
Time#(NoDelimiterTime, 'hhmmss') as ParsedNoDelimiterTime
Inline [
NoDelimiterTime
143015
091045
235959
060530
184010
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `NoDelimiterTime`

Maak de volgende meting:

- `=Time(ParsedNoDelimiterTime,'hh:mm:ss')`, om de tijd weer te geven in de notatie die standaard in het load-script staat.

Resultatentabel

| NoDelimiterTime | Time(ParsedNoDelimiterTime,'hh:mm:ss') |
|---|---|
| 060530 | 06:05:30 |

| NoDelimiterTime | Time(ParsedNoDelimiterTime,'hh:mm:ss') |
|---|---|
| 091045 | 09:10:45 |
| 143015 | 14:30:15 |
| 184010 | 18:40:10 |
| 235959 | 23:59:59 |

De resultaten laten zien hoe u de functie `Time#` kunt gebruiken om waarden van tekst te parseren en te converteren naar een tijd/numerieke waarde.

## Timestamp#

**Timestamp#()** evalueert een uitdrukking als een datum- en tijdwaarde in de notatie voor tijdstempels die in het load-script voor gegevens of in het besturingssysteem is ingesteld, tenzij een opmaakreeks is opgegeven.

**Syntaxis:**

```
timestamp#(text[, format])
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De teksttekenreeks die moet worden geëvalueerd. |

| Argument | Beschrijving |
|---|---|
| **format** | Tekenreeks die het formaat beschrijft dat gebruikt moet worden om de teksttekenreeks te interpreteren. Bij weglating wordt gebruikgemaakt van de korte datumnotatie, tijdnotatie en decimaal scheidingsteken die zijn ingesteld in het besturingssysteem. ISO 8601 wordt ondersteund voor tijdstempels.<br><br>De functie `Timestamp#` ondersteunt vele opmaakopties voor verschillende tijdstructuren, bijvoorbeeld:<br><br>`DD/MM/YYYY`: richt zich alleen op de datum (bijvoorbeeld: `15/11/2024`).<br><br>`YYYY-MM-DD hh:mm:ss`:geeft het volledige tijdstempel weer (bijvoorbeeld: `2024-11-15 13:30:45`).<br><br>`MMM DD, YYYY`: geeft een korte versie van de maand en het volledige jaar weer (bijvoorbeeld: `Nov 15, 2024`).<br><br>`hh:mm:ss TT`: toont alleen de tijd in 12-uursnotatie (bijvoorbeeld: `01:30:45 PM`)<br><br>`HH:mm:ss`: geeft de tijd in 24-uursnotatie weer (bijvoorbeeld: `13:30:45`)<br><br>`wwww, MMM DD YYYY`:toont de volledige weekdag en datum (bijvoorbeeld: `Friday, Nov 15 2024`)<br><br>`YYYY-MM-DD hh:mm:ss.fff`: inclusief milliseconden (bijvoorbeeld: `2024-11-15 13:30:45.123`). |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `Timestamp#('12-03-2023 10:25:30', 'DD-MM-YYYY hh:mm:ss')` | Retourneert `12-03-2023 10:25:30` |
| `Timestamp#('2023/03/15 13:45', 'YYYY/MM/DD hh:mm')` | Retourneert `2023/03/15 13:45` |

## Voorbeeld - Basisprincipes voor Timestamp#

Load-script

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Het volgende veld in een gegevenstabel:
  - `InputString`

- Een voorafgaande lading die de functie `Timestamp#` gebruikt om de uitdrukking als een datum- en tijdwaarde te behandelen. Wanneer het wordt uitgevoerd, maakt het een extra veld genaamd `TimeStamp_InputString`.

#### Load-script

```
SET DateFormat='M/D/YYYY'
Example:
Load *,
Timestamp(Timestamp#(InputString)) as TimeStamp_InputString;
LOAD * INLINE [
InputString
2015-09-15T12:13:14
1952-10-16T13:14:00+0200
1109-03-01T14:15
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `InputString`
- `TimeStamp_InputString`

Resultatentabel

| InputString | TimeStamp_InputString |
|---|---|
| 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 dit voorbeeld kunt u zien hoe tekenreekswaarden kunnen worden geëvalueerd als een tijdstempel met behulp van de functie `Timestamp#`.

## Voorbeeld 5 – Scenario met Timestamp#

Load-script

#### Overzicht

Gegevens uit een gebeurtenislogboek bevatten een tijdstempelnotatie die u wilt gebruiken.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:

    - Event
    - EventDateTime
- Een voorgaande lading die de functie Timestamp# gebruikt om de waarden in het veld EventDateTime om te zetten in een datum- en tijdnotatie die het systeem kan begrijpen. Wanneer het wordt uitgevoerd, maakt het een extra veld genaamd EventTimestamp.

### Load-script

```
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"
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- Event
- EventDateTime
- EventTimestamp

Maak de volgende meting:

- =Timestamp(EventTimestamp), om de opgemaakte tijdstempel weer te geven in de notatie die standaard in het load-script staat.

Resultatentabel

| Gebeurtenis | EventDateTime | EventTimestamp | Timestamp(EventTimestamp) |
|---|---|---|---|
| Aanmelden | 12-03-2023 10:25:30 | 12-03-2023 10:25:30 | 3/12/2023 10:25:30 AM |
| Aanmelden | 13-03-2023 09:10:15 | 13-03-2023 09:10:15 | 3/13/2023 9:10:15 AM |
| Aanmelden | 14-03-2023 08:00:00 | 14-03-2023 08:00:00 | 3/14/2023 8:00:00 AM |
| Afmelden | 12-03-2023 12:45:10 | 12-03-2023 12:45:10 | 3/12/2023 12:45:10 PM |
| Afmelden | 13-03-2023 17:30:20 | 13-03-2023 17:30:20 | 3/13/2023 5:30:20 PM |

In dit voorbeeld kunt u zien hoe waarden kunnen worden geconverteerd met behulp van de functie Timestamp#.

## 8.16 Interrecord-functies

Interrecord-functies worden gebruikt:

- In het load-script voor gegevens, wanneer een waarde uit eerder geladen records van gegevens nodig is voor de evaluatie van de huidige record.
- In een diagramuitdrukking, wanneer een andere waarde uit de gegevensverzameling van een visualisatie nodig is.

*Het is niet toegestaan te sorteren op y-waarden in grafieken of op uitdrukkingskolommen in tabellen als interrecord-diagramfuncties worden gebruikt in een van de uitdrukkingen van de grafiek. Deze sorteeropties worden daarom automatisch uitgeschakeld. Als u een interrecord-diagramfunctie gebruikt in een visualisatie of tabel, wordt de sortering van de visualisatie teruggezet naar de gesorteerde invoer van de interrecordfunctie. Deze beperking is niet van toepassing op de bijbehorende scriptfunctie, indien deze aanwezig is.*

*Definities van naar zichzelf verwijzende uitdrukkingen kunnen alleen op betrouwbare wijze worden gemaakt in tabellen met minder dan 100 rijen, maar dit kan variëren afhankelijk van de hardware waarop de Qlik-engine wordt uitgevoerd.*

### Rijfuncties

Deze functies kunnen alleen worden gebruikt in diagramuitdrukkingen.

Above

**Above()** evalueert een uitdrukking in een rij boven de huidige rij binnen een kolomsegment in een tabel. De rij waarvoor deze waarde wordt berekend is afhankelijk van de waarde van **offset**, indien aanwezig. De direct bovenliggende rij is de standaardwaarde. Voor andere diagrammen dan tabellen, evalueert **Above()** voor de rij boven de huidige rij in het equivalent van de standaard tabel in het diagram.

```
Above - diagramfunctie([TOTAL [<fld{,fld}>]] expr [ , offset [,count]])
```

Below

**Below()** evalueert een uitdrukking in een rij onder de huidige rij binnen een kolomsegment in een tabel. De rij waarvoor deze waarde wordt berekend is afhankelijk van de waarde van **offset**, indien aanwezig. De direct onderliggende rij is de standaardwaarde. Voor andere diagrammen dan tabellen, evalueert **Below()** voor de rij onder de huidige kolom in het equivalent van de standaard tabel in het diagram.

```
Below - diagramfunctie([TOTAL[<fld{,fld}>]] expression [ , offset [,count ]])
```

Bottom

**Bottom()** evalueert een uitdrukking in de laatste rij (onderste rij) van een kolomsegment in een tabel. De rij waarvoor deze waarde wordt berekend is afhankelijk van de waarde van **offset**, indien aanwezig. De onderste rij is de standaardwaarde. Voor andere diagrammen dan tabellen, wordt de laatste rij van de huidige kolom in het equivalent van de strakke tabel in het diagram geëvalueerd.

```
Bottom - diagramfunctie([TOTAL[<fld{,fld}>]] expr [ , offset [,count ]])
```

Top

**Top()** evalueert een uitdrukking in de eerste rij (bovenste rij) van een kolomsegment in een tabel. De rij waarvoor deze waarde wordt berekend is afhankelijk van de waarde van **offset**, indien aanwezig. De bovenste rij is de standaardwaarde. Voor andere diagrammen dan tabellen, wordt bij **Top()** de eerste rij van de huidige kolom in het equivalent van de standaard tabel in het diagram geëvalueerd.

```
Top - diagramfunctie([TOTAL [<fld{,fld}>]] expr [ , offset [,count ]])
```

NoOfRows

**NoOfRows()** retourneert het aantal rijen in het huidige kolomsegment in een tabel. Voor bitmapgrafieken wordt met **NoOfRows()** het aantal rijen in het equivalent van de standaard tabel voor de grafiek geretourneerd.

```
NoOfRows ([TOTAL])
```

## Kolomfuncties

Deze functies kunnen alleen worden gebruikt in diagramuitdrukkingen.

Column

**Column()** retourneert de waarde in de kolom die overeenkomt met **ColumnNo** in een strakke tabel, waarbij dimensies buiten beschouwing worden gelaten. Zo retourneert **Column(2)** bijvoorbeeld de waarde van de tweede kolom met metingen.

```
Column - diagramfunctie(ColumnNo)
```

Dimensionality

**Dimensionality()** retourneert het aantal dimensies voor de huidige rij. In het geval van draaitabellen retourneert de functie het totale aantal kolommen met dimensies met een niet-geaggregeerde inhoud, dat wil zeggen kolommen die geen subtotalen of samengevouwen aggregaten bevatten.

```
Dimensionality - diagramfunctie ( )
```

Secondarydimensionality

**SecondaryDimensionality()** retourneert het aantal rijen in een draaitabel met een niet-geaggregeerde inhoud, dat wil zeggen kolommen die geen subtotalen of samengevouwen aggregaten bevatten. Deze functie is het equivalent van de functie **dimensionality()** voor horizontale dimensies in draaitabellen.

```
SecondaryDimensionality - diagramfunctie ( )
```

## Veldfuncties

FieldIndex

**FieldIndex()** retourneert de positie van de veldwaarde **value** in het veld **field_name** (op volgorde van laden).

```
FieldIndex (field_name , value)
```

FieldValue

**FieldValue()** retourneert de waarde die is gevonden in positie **elem_no** van het veld **field_name** (op volgorde van laden).

```
FieldValue (field_name , elem_no)
```

FieldValueCount

**FieldValueCount()** is een functie voor **geheel getal** waarmee het aantal distinctieve waarden in een veld wordt gevonden.

```
FieldValueCount (field_name)
```

## Draaitabelfuncties

Deze functies kunnen alleen worden gebruikt in diagramuitdrukkingen.

After

retourneert de waarde van een uitdrukking berekend aan de hand van de dimensiewaarden van een draaitabel die staan in de kolom voor de huidige kolom in een rijsegment in de draaitabel.

```
After - diagramfunctie([TOTAL] expression [ , offset [,n]])
```

Before

**Before()** retourneert de waarde van een uitdrukking berekend aan de hand van de dimensiewaarden van een draaitabel die staan in de kolom voor de huidige kolom in een rijsegment in de draaitabel.

```
Before - diagramfunctie([TOTAL] expression [ , offset [,n]])
```

First

**First()** retourneert de waarde van een uitdrukking berekend aan de hand van de dimensiewaarden van een draaitabel die staan in de eerste kolom van het huidige rijsegment in de draaitabel. Deze functie retourneert NULL in alle andere diagramtypen dan een draaitabel.

```
First - diagramfunctie([TOTAL] expression [ , offset [,n]])
```

Last

**Last()** retourneert de waarde van een uitdrukking berekend aan de hand van de dimensiewaarden van een draaitabel die staan in de laatste kolom van het huidige rijsegment in de draaitabel. Deze functie retourneert NULL in alle andere diagramtypen dan een draaitabel.

```
Last - diagramfunctie([TOTAL] expression [ , offset [,n]])
```

ColumnNo

**ColumnNo()** retourneert het nummer van de huidige kolom in het huidige rijsegment in een draaitabel. De eerste kolom is nummer 1.

```
ColumnNo - diagramfunctie([TOTAL])
```

NoOfColumns

**NoOfColumns()** retourneert het aantal kolommen in het huidige rijsegment in een draaitabel.

```
NoOfColumns - diagramfunctie([TOTAL])
```

## Interrecord-functies in het load-script voor gegevens

### Exists

**Exists()** bepaalt of een specifieke veldwaarde al is geladen in het veld in het script voor het laden van gegevens. De functie retourneert TRUE of FALSE, zodat deze kan worden gebruikt in de **where**-clausule van een **LOAD**-opdracht of een **IF**-opdracht.

```
Exists (field_name [, expr])
```

### LookUp

**LookUp()** zoekt een tabel die al is geladen en retourneert de waarde van **field_name** die overeenkomt met het eerste exemplaar van de waarde **match_field_value** in het veld **match_field_name**. De tabel kan de huidige tabel zijn of een andere tabel die eerder is geladen.

```
LookUp (field_name, match_field_name, match_field_value [, table_name])
```

### Peek

**Peek()** retourneert de waarde van een veld in een tabel voor een rij die al is geladen. Het rijnummer kan worden opgegeven, net als de tabel. Als er geen rijnummer is opgegeven, wordt het laatst geladen record gebruikt.

```
Peek (field_name[, row_no[, table_name ] ])
```

### Previous

**Previous()** retourneert de waarde van de uitdrukking **expr** met gebruik van de gegevens uit de vorige invoerrecord die niet is genegeerd wegens een **where**-clausule. In de eerste record van een interne tabel retourneert de functie NULL.

```
Previous (expr)
```

---

**Zie ook:**

p *Bereikfuncties (page 1600)*

## Above - diagramfunctie

**Above()** evalueert een uitdrukking in een rij boven de huidige rij binnen een kolomsegment in een tabel. De rij waarvoor deze waarde wordt berekend is afhankelijk van de waarde van **offset**, indien aanwezig. De direct bovenliggende rij is de standaardwaarde. Voor andere diagrammen dan tabellen, evalueert **Above()** voor de rij boven de huidige rij in het equivalent van de standaard tabel in het diagram.

**Syntaxis:**

```
Above([TOTAL] expr [ , offset [,count]])
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **expr** | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| **offset** | Als een offset n ,groter dan 0, wordt opgegeven, wordt de evaluatie van de uitdrukking n rijen verder omhoog verplaatst, geteld vanaf de huidige rij.<br>Als de beginwaarde 0 wordt opgegeven, wordt de uitdrukking geëvalueerd op de huidige rij.<br><br>Als een negatief getal voor de beginwaarde wordt opgegeven, werkt de functie Above op dezelfde wijze als de functie Below met het overeenkomstige positieve getal als beginwaarde. |
| **count** | Als een derde argument **count** groter dan 1 wordt opgegeven, retourneert de functie niet één maar een bereik van **count**-waarden. Eén waarde voor elke **count** tabelrijen, omhoog geteld vanaf de oorspronkelijke cel.<br>In deze vorm kan de functie worden gebruikt als argument voor een van de speciale bereikfuncties. *Bereikfuncties (page 1600)* |
| **TOTAL** | Als de tabel eendimensionaal is of als de kwalificatie **TOTAL** wordt gebruikt als argument, is het huidige kolomsegment altijd gelijk aan de hele kolom. |

Op de eerste rij van een kolomsegment wordt een NULL-waarde geretourneerd, omdat er geen rij boven de eerste rij is.

*Een kolomsegment wordt gedefinieerd als een opeenvolgende subset van cellen die dezelfde waarden hebben voor de dimensies in de huidige sorteervolgorde. Interrecord-diagramfuncties worden berekend in het kolomsegment met uitzondering van de dimensie uiterst rechts in de equivalente strakke tabel. Als er slechts één dimensie is in het diagram, of als de kwalificatie TOTAL is opgegeven, wordt de uitdrukking geëvalueerd voor de volledige tabel.*

*Als de tabel of het equivalent van de tabel meerdere verticale dimensies heeft, omvat het huidige kolomsegment alleen rijen met dezelfde waarden als de huidige rij in alle dimensiekolommen, met uitzondering van de kolom waarin de laatste dimensie wordt weergegeven in de onderlinge sorteervolgorde van de velden.*

**Beperkingen:**

- Recursieve oproepen retourneren een NULL-waarde.
- Het is niet toegestaan te sorteren op y-waarden in grafieken of op uitdrukkingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van de grafiek. Deze sorteeropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie gebruikt in een

visualisatie of tabel, wordt de sortering van de visualisatie teruggezet naar de gesorteerde invoer van deze functie.

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `Above( Sum(Sales) )` | Retourneert de waarde `Sales` voor de rij boven de huidige rij. |

## Voorbeeld - Basisprincipes voor Above

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Year`
  - `Sales`

### Load-script

```
Example:
LOAD * INLINE [
Year, Sales
2019, 1000
2020, 1500
2021, 1200
2022, 1800
2023, 1700
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Year`

Maak de volgende metingen:

- `=Sum(Sales)`, om de jaaromzet te vinden.
- `=Above(Sum(Sales))`, om de verkoopwaarde voor het vorige jaar te vinden.
- `=Sum(Sales) - Above(Sum(Sales))`, om de variantie tussen de `Sales` waarde van het huidige jaar en de waarde van het vorige jaar te berekenen.
- `=RangeSum(Above(Sales, 1, 2))`, om een voortschrijdende tweejarige aggregatie van de verkoopcijfers van de afgelopen twee jaar aan te tonen.
- `=(Sales - Above(Sales)) / Above(Sales)`, om de verandering in `Sales` als een verhouding / percentage weer te geven. Om deze waarde als percentage weer te geven, selecteert u onder **GetalnotatieGetal > Eenvoudige opmaak > 12,34%**.

Resultatentabel

| 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% |

Als u de resultaten bekijkt, kunt u de `Sales` waarden in de huidige periode vergelijken met de vorige periode.

- De meting `Sum(Sales)- Above(Sum(Sales))` retourneert de wijziging in de `Sales` waarden, waarbij de huidige periode wordt vergeleken met de vorige periode.
- De meting `RangeSum(Above(Sales, 1, 2))` retourneert de som van de vorige twee `Sales` waarden.
- De meting `(Sales - Above(Sales)) / Above(Sales)` retourneert de wijziging in `Sales` ten opzichte van de voorgaande periode als een percentage.

*Deze functie kan behalve in tabellen ook worden gebruikt in diagrammen, bijvoorbeeld in staafdiagrammen.*

**Doe het volgende:**

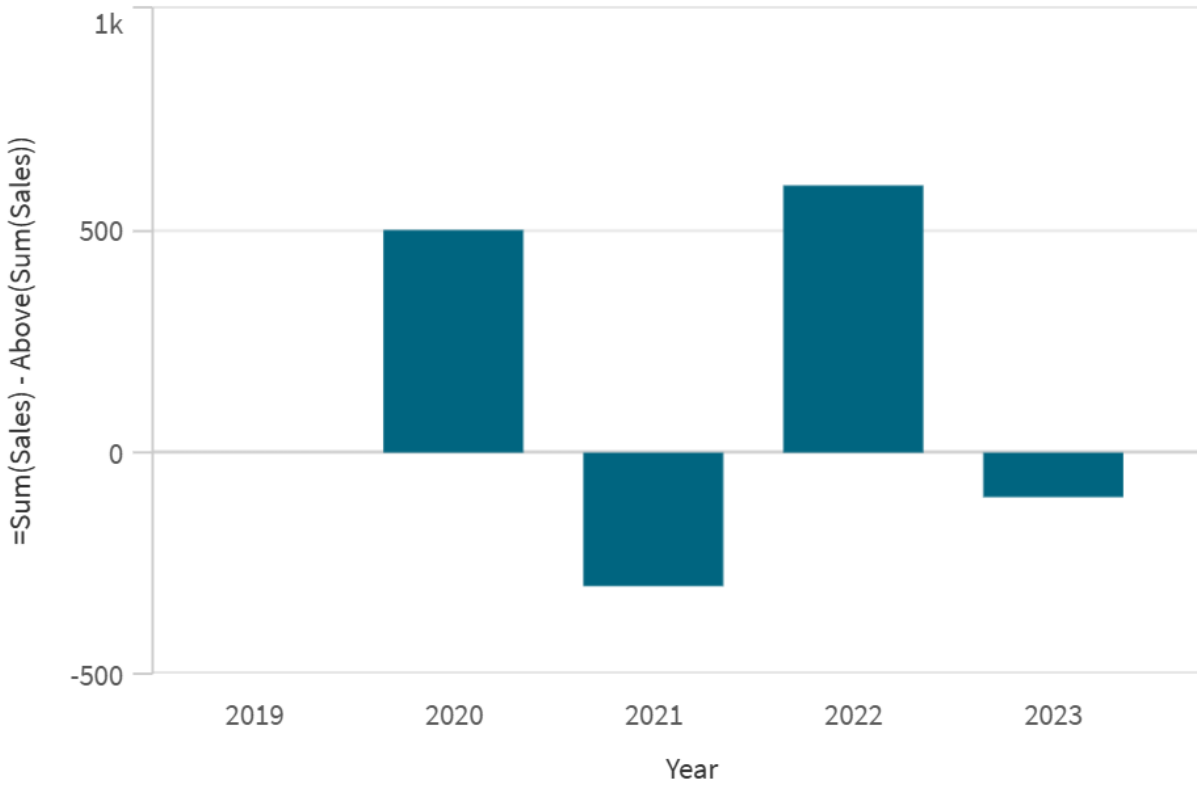
Maak een staafdiagram en voeg dit veld toe als dimensie:

- `Year`

Maak de volgende meting:

- `=Sum(Sales) - Above(Sum(Sales))`, om de jaar op jaar variantie weer te geven.

*Staafdiagram met resultaten voor de functie Above*




Door de functie `Above` in een staafdiagram te gebruiken, krijgt u een visuele vergelijking van de variantie tussen de verkopen van de huidige periode en die van de vorige periode.

## Voorbeeld 5 – Scenario met Above

### Diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `SalesYear`
  - `SalesAmount`

#### Load-script

```
Example:
LOAD * INLINE [
SalesYear, SalesAmount
```

```
2019, 500
2020, 800
2021, 400
2022, 700
2023, 600
];
```

**Resultaten**

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `SalesYear`

Maak de volgende metingen:

- `=Sum(SalesAmount)`, om de verkoop te aggregeren.
- `=RangeSum(Above(SalesAmount, 0, RowNo()))`, om een voorlopig totaal van de `SalesAmount` waarden te berekenen.

Resultatentabel

| 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 |

De resultaten laten zien hoe u een combinatie van de functie `RangeSum` en de functie `Above` kunt gebruiken om een cumulatief voorlopig totaal te maken.

*Deze functie kan behalve in tabellen ook worden gebruikt in diagrammen, bijvoorbeeld in staafdiagrammen.*

**Doe het volgende:**

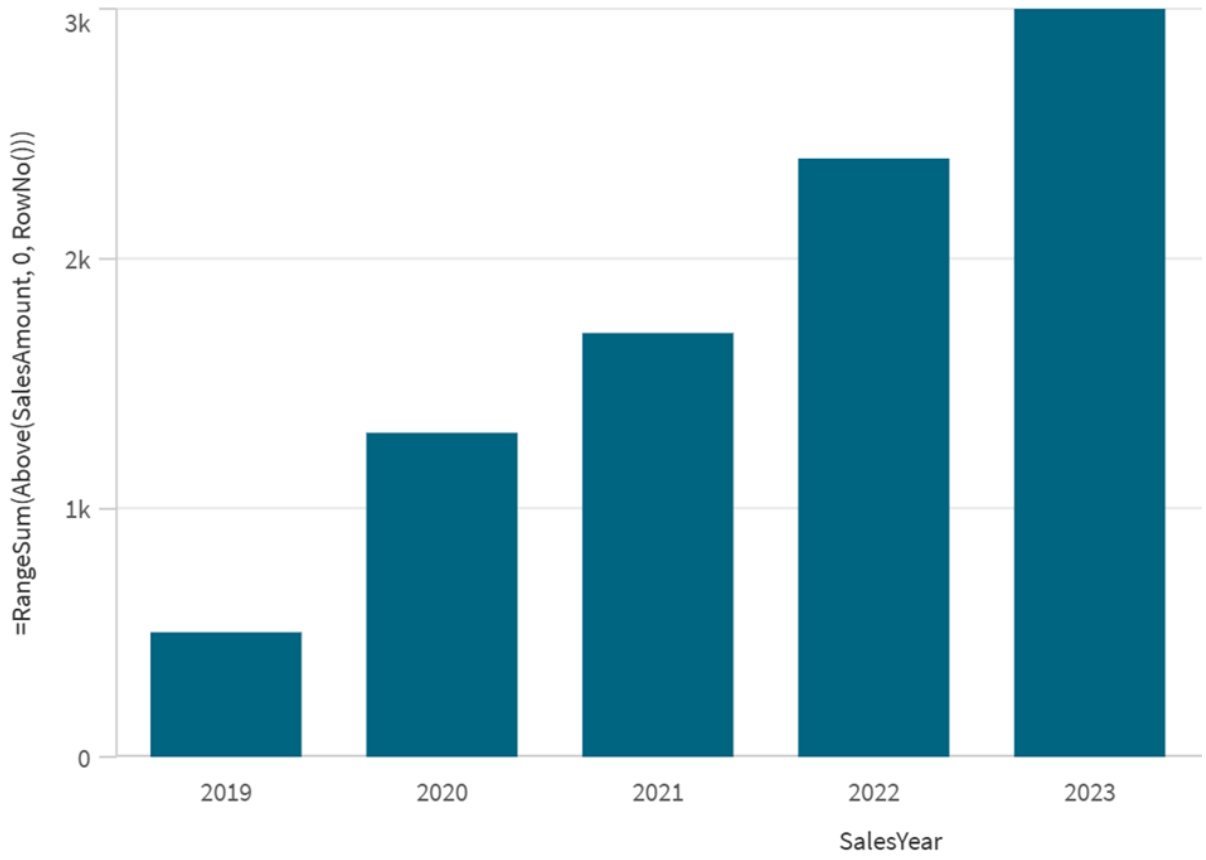
Maak een staafdiagram en voeg dit veld toe als dimensie:

- `SalesYear`

Maak de volgende meting:

- `=RangeSum(Above(SalesAmount, 0, RowNo()))`, om de cumulatieve verkoopwaarde te retourneren.

*Staafdiagram met de cumulatieve verkoopwaarde per jaar*




## Voorbeeld - Diagramuitdrukking met gebruik van Above

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Twee gegevenstabellen genaamd `Monthnames` en `Sales2013`.

### Load-script

```
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 '|');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Customer`

Maak de volgende metingen:

- `=Sum(Sales)`, om de som van verkoop te berekenen.
- `=Above(Sum(Sales))`, om de som van de verkopen in de vorige rij te tonen.
- `=Sum(Sales)+Above(Sum(Sales))`, om de som van de verkoopwaarde op te tellen bij de som van de verkoop voor de klantrij boven de huidige rij.
- `=Sum(Sales)+Above(Sum(Sales), 3)`, om de `Sum(Sales)` waarde voor de huidige `Customer` op te tellen bij de waarde voor de drie bovenliggende rijen van `Customer`.
- `=IF(Sum(Sales)>Above(Sum(Sales)), 'Higher')`, om het label "`Higher`" toe te voegen aan klanten waarvan de resultaten van `Sum(Sales)` groter zijn dan de `Sum(Sales)` van de klant in de rij erboven.

Resultatentabel

| Customer | Sum (Sales) | Above (Sum (Sales)) | Sum (Sales)+Above (Sum(Sales)) | Sum (Sales)+Above (Sum(Sales), 3) | IF(Sum (Sales)>Above (Sum(Sales)), 'Higher') |
|---|---|---|---|---|---|
| **Totals** | **2566** | **-** | **-** | **-** | **-** |
| Astrida | 587 | - | - | - | - |
| Betacab | 539 | 587 | 1126 | - | - |
| Canutility | 683 | 539 | 1222 | - | Higher |
| Divadip | 757 | 683 | 1440 | 1344 | Higher |

De meting `Above(Sum(Sales))` retourneert NULL voor de klant `Astrida`, omdat er geen rij boven is. Het resultaat voor de klant `Betacab` toont de waarde van `Sum(Sales)` voor `Astrida`. Het resultaat voor `Canutility` toont de waarde voor `Sum(Sales)` voor `Betacab`, enzovoort.

Voor de meting `Sum(Sales)+Above(Sum(Sales))` retourneert de rij voor `Betacab` het resultaat van het optellen van de `Sum(Sales)` waarden voor de rijen `Betacab` + `Astrida` (`539+587`). Het resultaat voor de rij `Canutility` bevat het resultaat van de optelsom van de waarden van `Sum(Sales)` voor `Canutility` + `Betacab` (`683+539`).

De meting `Sum(Sales)+Above(Sum(Sales), 3)` heeft het argument `offset`, ingesteld op `3`. Deze neemt de waarde in de rij die 3 rijen boven de huidige rij staat. De waarde van `Sum(Sales)` voor de huidige `Customer` wordt toegevoegd aan de waarde voor de drie bovenliggende rijen van `Customer`. Bijvoorbeeld, de `Sum (Sales)` voor `Divadip`, wordt toegevoegd aan de `Sum(Sales)` voor `Astrida`, de klant die drie rijen boven `Divadip` staat. De waarden die worden geretourneerd voor de eerste drie rijen van `Customer` zijn null, omdat er geen gegevens zijn.

De meting `IF(Sum(Sales)>Above(Sum(Sales)), 'Higher')`, retourneert de waarde `Higher` voor de klanten `Canutility` en `Divadip` omdat hun verkoopresultaten beide hoger zijn dan de som van de verkopen van de klant in de rij erboven, respectievelijk `Betacab` en `Canutility`.

*Deze functie kan behalve in tabellen ook worden gebruikt in diagrammen, bijvoorbeeld in staafdiagrammen.*

*Voor andere diagramtypen converteert u het diagram naar het equivalent van de strakke tabel zodat u gemakkelijk kunt interpreteren op welke rij de functie betrekking heeft.*

## Voorbeeld - Above scenario gebaseerd op sorteervolgorde

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Drie gegevenstabellen genaamd `Monthnames`, `Sales2013` en `Product`.

### Load-script

```
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 '|');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `Customer`
- `Product`
- `Month`

Maak de volgende metingen:

- `=Sum(Sales)`, om de som van verkoop te berekenen.
- `=Above(Sum(Sales))`, om de som van de verkopen in de vorige rij te tonen.

Resultatentabel

| Customer | Product | Month | Sum(Sales) | Above(Sum(Sales)) |
|---|---|---|---|---|
| **Totals** | **-** | **-** | **2566** | **-** |
| Astrida | AA | Jan | 46 | - |
| Astrida | AA | Feb | 60 | 46 |

| Customer | Product | Month | Sum(Sales) | Above(Sum(Sales)) |
|---|---|---|---|---|
| **Totals** | - | - | **2566** | - |
| Astrida | AA | Mar | 70 | 60 |
| 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 |

Voor diagrammen met meer dan één dimensie zijn de resultaten van uitdrukkingen met de functies `Above`, `Below`, `Top` en `Bottom` afhankelijk van de volgorde waarin de kolom met dimensies op Qlik Sense zijn gesorteerd. Qlik Sense evalueert de functies op basis van de kolomsegmenten die het resultaat zijn van de dimensie die als laatste is gesorteerd. De sorteervolgorde van de kolom wordt beheerd in het eigenschappenvenster onder **Sorteren** en is niet per se gelijk aan de volgorde waarin de kolommen in een tabel worden weergegeven.

In de resultatentabel is `Month` de laatst gesorteerde dimensie, dus de functie `Above` wordt geëvalueerd op basis van maanden. Er is een reeks van resultaten voor elke waarde van `Product` voor elke maand (`Jan` t/m `Dec`) - een kolomsegment. Dit wordt gevolgd door een reeks voor het volgende kolomsegment: voor elke `Month` voor het volgende `Product`. Er is een kolomsegment beschikbaar voor elke waarde van `Customer` voor elk `Product`.

## Voorbeeld - Above scenario met een andere sorteervolgorde

### Diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Drie gegevenstabellen genaamd `Monthnames`, `Sales2013` en `Product`.

### Load-script

```
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 '|');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- Customer
- Product
- Month

Maak de volgende metingen:

- `=Sum(Sales)`, om de som van verkoop te berekenen.
- `=Above(Sum(Sales))`, om de som van de verkopen in de vorige rij te tonen.

Wijzig de sorteervolgorde in het eigenschappenvenster. Selecteer **Sorteren** en sleep de dimensie **Product** naar de derde positie in de sorteervolgorde.

Resultatentabel

| Customer | Product | Month | Sum(Sales) | Above(Sum(Sales)) |
|---|---|---|---|---|
| **Totals** | - | - | **2566** | - |
| 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 |

In de resultatentabel is de laatst gesorteerde dimensie `Product`. De functie `Above` wordt geëvalueerd voor elk `Product`, en omdat er maar twee producten zijn, `AA` en `BB`, is er slechts één resultaat dat niet null is in elke reeks. In rij `BB` voor de maand `Jan` bedraagt de waarde voor `Above(Sum(Sales))` `46`. Voor rij `AA` is de waarde null. De waarde in elke rij `AA` voor elke maand is altijd null, aangezien er geen waarde van `Product` boven `AA` is. De tweede reeks wordt geëvalueerd op `AA` en `BB` voor de maand `Feb`, voor de waarde van `Customer`, `Astrida`. Als alle maanden zijn geëvalueerd voor `Astrida`, wordt de reeks herhaald voor de tweede klant `Betacab`, enzovoort.

*De resultaten in de tabel tonen slechts een gedeeltelijke lijst van de gegevens. Om de volledige resultaten te zien, kunt u het voorbeeld in een app uitproberen.*

## Voorbeeld - De functie Above gebruiken met RangeAvg

Diagramuitdrukking

### Overzicht

De functie `Above` kan worden gebruikt als invoer voor de bereikfuncties. Dit voorbeeld gebruikt de functie in een `RangeAvg`.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Drie gegevenstabellen genaamd `Monthnames`, `Sales2013` en `Product`.

### Load-script

```
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 '|');
```

**Resultaten**

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Customer`

Maak de volgende meting:

- `=RangeAvg (Above(Sum(Sales),1,3))`

Resultatentabel*

| Customer | RangeAvg (Above(Sum(Sales),1,3)) |
|---|---|
| **Totals** | **-** |
| Astrida | - |
| Betacab | 587 |
| Canutility | 563 |
| Divadip | 603 |

In de argumenten voor de functie Above wordt `offset` ingesteld op `1` en wordt `count` ingesteld op `3`. De functie retourneert de resultaten van de uitdrukking `Sum(Sales)` voor de drie rijen direct boven de huidige rij in het kolomsegment (waar er een rij is). Deze drie waarden worden gebruikt als invoer voor de functie `RangeAvg`, die het gemiddelde van de waarden in het opgegeven bereik van cijfers als resultaat geeft.

**Zie ook:**

p *Below - diagramfunctie (page 1493)*
p *Bottom - diagramfunctie (page 1503)*
p *Top - diagramfunctie (page 1542)*
p *RangeAvg (page 1603)*

## Below - diagramfunctie

**Below()** evalueert een uitdrukking in een rij onder de huidige rij binnen een kolomsegment in een tabel. De rij waarvoor deze waarde wordt berekend is afhankelijk van de waarde van **offset**, indien aanwezig. De direct onderliggende rij is de standaardwaarde. Voor andere diagrammen dan tabellen, evalueert **Below()** voor de rij onder de huidige kolom in het equivalent van de standaard tabel in het diagram.

**Syntaxis:**

```
Below([TOTAL] expr [ , offset [,count ]])
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **expr** | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| **offset** | Als een **offset** n, groter dan 1 wordt opgegeven, wordt de evaluatie van de uitdrukking n rijen verder omlaag verplaatst, geteld vanaf de huidige rij.<br><br>Als de beginwaarde 0 wordt opgegeven, wordt de uitdrukking geëvalueerd op de huidige rij.<br><br>Als een negatief getal voor de beginwaarde wordt opgegeven, werkt de functie **Below** op dezelfde wijze als de functie **Above** met het overeenkomstige positieve getal als beginwaarde. |
| **count** | Als een derde parameter **count** groter dan 1 wordt opgegeven, retourneert de functie niet één maar een bereik van **count**-waarden. Eén waarde voor elke **count** tabelrijen, omlaag geteld vanaf de oorspronkelijke cel. In deze vorm kan de functie worden gebruikt als argument voor een van de speciale bereikfuncties. *Bereikfuncties (page 1600)* |
| **TOTAL** | Als de tabel eendimensionaal is of als de kwalificatie **TOTAL** wordt gebruikt als argument, is het huidige kolomsegment altijd gelijk aan de hele kolom. |

Op de laatste rij van een kolomsegment wordt een NULL-waarde geretourneerd, omdat er geen rij onder de laatste rij is.

*Een kolomsegment wordt gedefinieerd als een opeenvolgende subset van cellen die dezelfde waarden hebben voor de dimensies in de huidige sorteervolgorde. Interrecord-diagramfuncties worden berekend in het kolomsegment met uitzondering van de dimensie uiterst rechts in de equivalente strakke tabel. Als er slechts één dimensie is in het diagram, of als de kwalificatie TOTAL is opgegeven, wordt de uitdrukking geëvalueerd voor de volledige tabel.*

*Als de tabel of het equivalent van de tabel meerdere verticale dimensies heeft, omvat het huidige kolomsegment alleen rijen met dezelfde waarden als de huidige rij in alle dimensiekolommen, met uitzondering van de kolom waarin de laatste dimensie wordt weergegeven in de onderlinge sorteervolgorde van de velden.*

**Beperkingen:**

- Recursieve oproepen retourneren een NULL-waarde.
- Het is niet toegestaan te sorteren op y-waarden in grafieken of op uitdrukkingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van de grafiek. Deze sorteeropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie gebruikt in een

visualisatie of tabel, wordt de sortering van de visualisatie teruggezet naar de gesorteerde invoer van deze functie.

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `Below( Sum(Sales) )` | Retourneert de waarde van `Sales` voor de rij onder de huidige rij. |

## Voorbeeld - Basisprincipes voor Below

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Year`
  - `Sales`

### Load-script

```
Example:
LOAD * INLINE [
Year, Sales
2019, 1000
2020, 1500
2021, 1200
2022, 1800
2023, 1700
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Year`

Maak de volgende metingen:

- `=Sum(Sales)`, om de jaaromzet te vinden.
- `=Below(Sum(Sales))`, om de verkoopwaarde voor het volgende jaar te vinden.
- `=Sum(Sales) - Below(Sum(Sales))`, om de variantie tussen de `Sales` waarde van het huidige jaar en de waarde van het volgende jaar te berekenen.
- `=RangeSum(Below(Sales, 1, 2))`, om een voortschrijdende tweejarige aggregatie van de verkoopcijfers van de komende twee jaar aan te tonen.
- `=(Sales - Below(Sales)) / Below(Sales)`, om de verandering in `Sales` als een verhouding / percentage weer te geven. Om deze waarde als percentage weer te geven, selecteert u onder **GetalnotatieGetal > Eenvoudige opmaak > 12,34%**.

Resultatentabel

| 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 | - |

Als u de resultaten bekijkt, kunt u de `Sales` waarden in de huidige periode vergelijken met de volgende periode.

- De meting `Sum(Sales) - Below(Sum(Sales))` retourneert de wijziging in de `Sales` waarden, waarbij de huidige periode wordt vergeleken met de volgende periode.
- De meting `RangeSum(Below(Sales, 1, 2))` retourneert de som van de volgende twee `Sales` waarden.
- De meting `(Sales - Below(Sales)) / Below(Sales)` retourneert de wijziging in `Sales` ten opzichte van de volgende periode als een percentage.

*Deze functie kan behalve in tabellen ook worden gebruikt in diagrammen, bijvoorbeeld in staafdiagrammen.*

**Doe het volgende:**

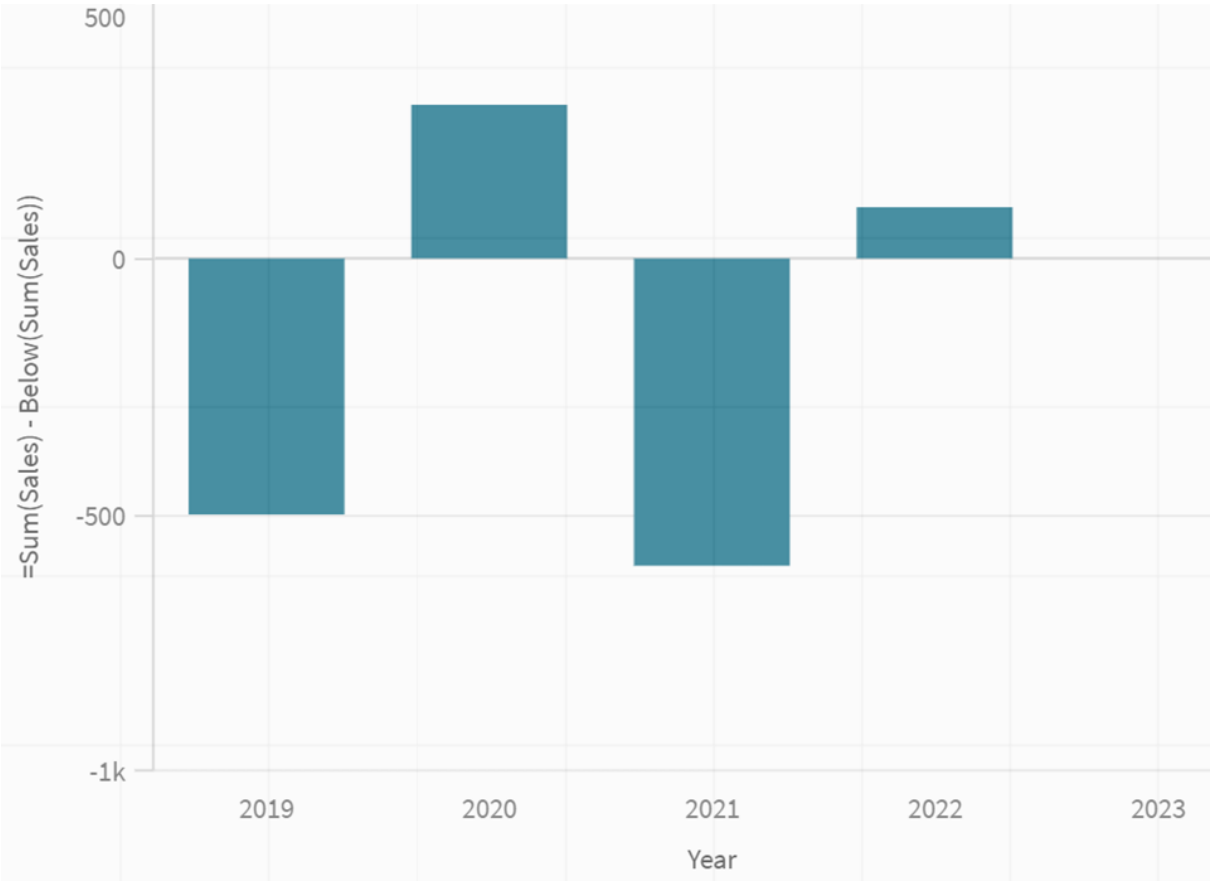
Maak een staafdiagram en voeg dit veld toe als dimensie:

- `Year`

Maak de volgende meting:

- `=Sum(Sales) - Below(Sum(Sales))`, om de jaar op jaar variantie weer te geven.

*Staafdiagram met resultaten voor de functie Below*




Door de functie `Below` in een staafdiagram te gebruiken, krijgt u een visuele vergelijking van de variantie tussen de verkopen van de huidige periode en die van de volgende periode.

## Voorbeeld 5 – Scenario met Below

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `SalesYear`
  - `SalesAmount`

### Load-script

```
Example:
LOAD * INLINE [
SalesYear, SalesAmount
2019, 500
2020, 800
2021, 400
2022, 700
2023, 600
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `SalesYear`

Maak de volgende metingen:

- `=Sum(SalesAmount)`, om de verkoop te aggregeren.
- `=RangeSum(Below(SalesAmount, 0, NoOfRows() - RowNo()))`, om een aflopend totaal van de `SalesAmount` waarden te berekenen.

Resultatentabel

| 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 |

De resultaten laten zien hoe u een combinatie van de functie `RangeSum` en de functie `Below` kunt gebruiken om een cumulatief aflopend totaal te maken.

*Deze functie kan behalve in tabellen ook worden gebruikt in diagrammen, bijvoorbeeld in staafdiagrammen.*

### Doe het volgende:

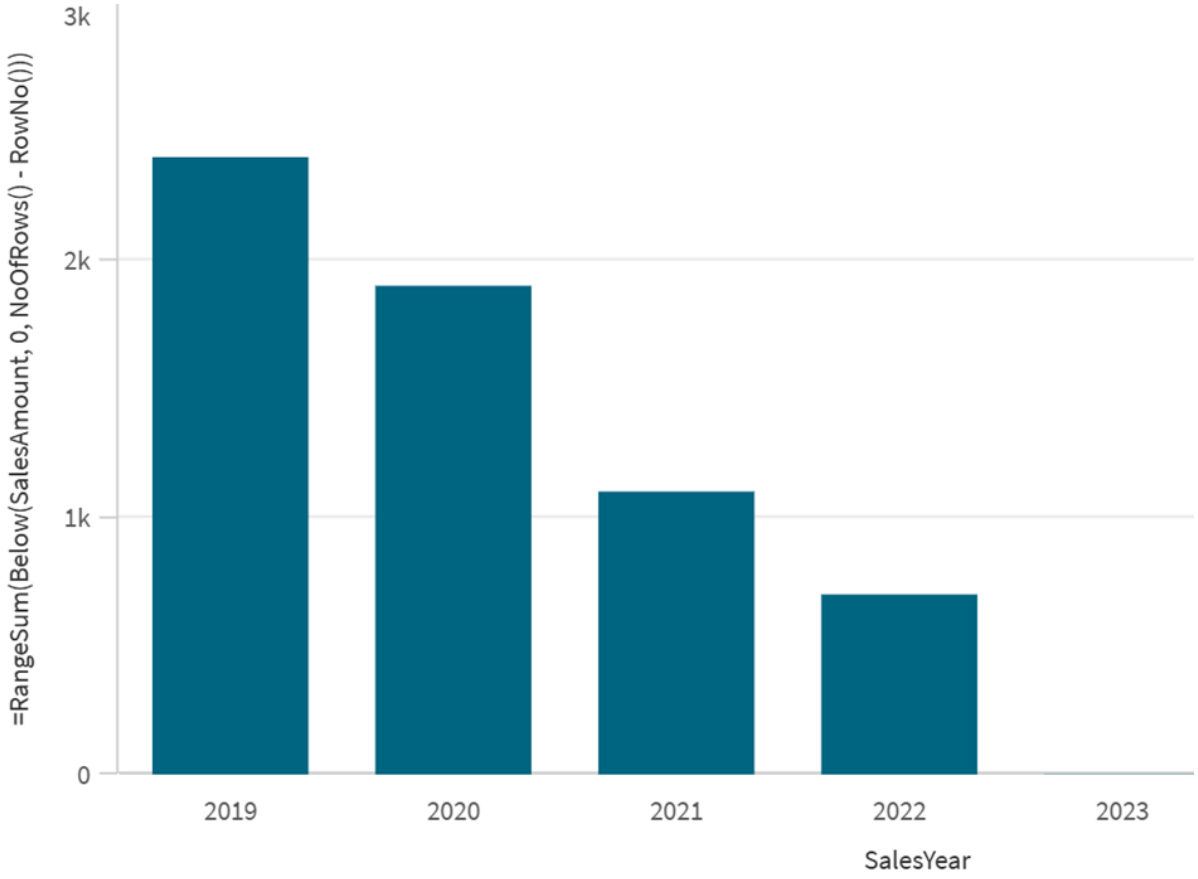
Maak een staafdiagram en voeg dit veld toe als dimensie:

- `SalesYear`

Maak de volgende meting:

- `=RangeSum(Below(SalesAmount, 0, NoOfRows() - RowNo()))`, om de cumulatieve aflopende verkoopwaarde te retourneren.

*Staafdiagram met de cumulatieve aflopende verkoopwaarde per jaar*




Door de functie `Below` in combinatie met `RangeSum` in een staafdiagram te gebruiken, krijgt u een visuele weergave van de cumulatieve aflopende verkoopwaarden.

## Voorbeeld - Diagramuitdrukking met gebruik van Below

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Twee gegevenstabellen genaamd `Monthnames` en `Sales2013`.

### Load-script

```
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 '|');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Customer`

Maak de volgende metingen:

- `=Sum(Sales)`, om de som van verkoop te berekenen.
- `=Below(Sum(Sales))`, om de som van de verkopen in de vorige rij te tonen.
- `=Sum(Sales)+Below(Sum(Sales))`, om de som van de verkoopwaarde op te tellen bij de som van de verkoop voor de klantrij onder de huidige rij.
- `=Sum(Sales)+Below(Sum(Sales), 3)`, om de `Sum(Sales)` waarde voor de huidige `Customer` op te tellen bij de waarde voor de drie onderliggende rijen van `Customer`.
- `=If(Sum(Sales)>Below(Sum(Sales)), 'Higher')`, om het label "`Higher`" toe te voegen aan klanten waarvan de resultaten van `Sum(Sales)` groter zijn dan de `Sum(Sales)` van de klant in de rij eronder.

Resultatentabel

| Customer | Sum (Sales) | Below (Sum (Sales)) | Sum (Sales)+Below (Sum(Sales)) | Sum (Sales)+Below (Sum(Sales), 3) | If(Sum (Sales)>Below(Sum (Sales)), 'Higher') |
|---|---|---|---|---|---|
| **Totals** | **2566** | **-** | **-** | **-** | **-** |
| Astrida | 587 | 539 | 1126 | 1334 | Higher |
| Betacab | 539 | 683 | 1222 | - | - |

| Customer | Sum (Sales) | Below (Sum (Sales)) | Sum (Sales)+Below (Sum(Sales)) | Sum (Sales)+Below (Sum(Sales), 3) | If(Sum (Sales)>Below(Sum (Sales)), 'Higher') |
|---|---|---|---|---|---|
| **Totals** | **2566** | **-** | **-** | **-** | **-** |
| Canutility | 683 | 757 | 1440 | - | - |
| Divadip | 757 | - | - | - | - |

Voor diagrammen met meer dan één dimensie zijn de resultaten van uitdrukkingen met de functies `Above`, `Below`, `Top` en `Bottom` afhankelijk van de volgorde waarin de kolom met dimensies op Qlik Sense zijn gesorteerd. Qlik Sense evalueert de functies op basis van de kolomsegmenten die het resultaat zijn van de dimensie die als laatste is gesorteerd. De sorteervolgorde van de kolom wordt beheerd in het eigenschappenvenster onder **Sorteren** en is niet per se gelijk aan de volgorde waarin de kolommen in een tabel worden weergegeven. Zie *Voorbeeld - Above scenario gebaseerd op sorteervolgorde (page 1487)* voor een voorbeeld waarin het gebruik van sorteervolgorde wordt gedemonstreerd.

De meting `Below(Sum(Sales))` retourneert NULL voor de klant `Divadip`, omdat er geen rij onder is. Het resultaat voor de rij `Canutility` toont de waarde van `Sum(Sales)` voor `Divadip`. Het resultaat voor `Betacab` toont de waarde voor `Sum(Sales)` voor `Canutility`, enzovoort.

Voor de meting `Sum(Sales)+Below(Sum(Sales))` retourneert de rij voor `Astrida` het resultaat van het optellen van de `Sum(Sales)` waarden voor de rijen `Betacab` + `Astrida` (`539+587`). Het resultaat voor de rij `Betacab` bevat het resultaat van de optelsom van de waarden van `Sum(Sales)` voor `Canutility` + `Betacab` (`539+683`).

De meting `Sum(Sales)+Below(Sum(Sales), 3)` bevat het argument `offset`, dat is ingesteld op `3`, en neemt de waarde in de drie rijen onder de huidige rij aan. De waarde van `Sum(Sales)` voor de huidige `Customer` wordt toegevoegd aan de waarde voor de drie onderliggende rijen van `Customer`. De waarden voor de laagste drie rijen van `Customer` zijn null.

De meting `IF(Sum(Sales)>Below(Sum(Sales)), 'Higher')` retourneert de waarde `Higher` voor de klant `Astrida` omdat zijn verkoopresultaten hoger zijn dan de som van de verkopen van de klant in de rij eronder, `Betacab`.

*Deze functie kan behalve in tabellen ook worden gebruikt in diagrammen, bijvoorbeeld in staafdiagrammen.*

*Voor andere diagramtypen converteert u het diagram naar het equivalent van de strakke tabel zodat u gemakkelijk kunt interpreteren op welke rij de functie betrekking heeft.*

## Voorbeeld - De functie Below gebruiken met RangeAvg

Diagramuitdrukking

### Overzicht

De functie `Below` kan worden gebruikt als invoer voor de bereikfuncties. Dit voorbeeld gebruikt de functie in een `RangeAvg`.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Twee gegevenstabellen genaamd `Monthnames`, `Sales2013`.

### Load-script

```
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 '|');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Customer`

Maak de volgende meting:

- `=RangeAvg (Below(Sum(Sales),1,3))`

Resultatentabel

| Customer | RangeAvg (Below(Sum(Sales),1,3)) |
|---|---|
| **Totals** | **-** |
| Astrida | 660 |
| Betacab | 720 |
| Canutility | 757 |
| Divadip | - |

In de argumenten voor de functie `Below` wordt `offset` ingesteld op `1` en wordt `count` ingesteld op `3`. De functie retourneert de resultaten van de uitdrukking `Sum(Sales)` voor de drie rijen direct onder de huidige rij in het kolomsegment (waar er een rij is). Deze drie waarden worden gebruikt als invoer voor de functie `RangeAvg`, die het gemiddelde van de waarden in het opgegeven bereik van cijfers als resultaat geeft.

**Zie ook:**

p *Above - diagramfunctie (page 1479)*
p *Bottom - diagramfunctie (page 1503)*
p *Top - diagramfunctie (page 1542)*
p *RangeAvg (page 1603)*

## Bottom - diagramfunctie

**Bottom()** evalueert een uitdrukking in de laatste rij (onderste rij) van een kolomsegment in een tabel. De rij waarvoor deze waarde wordt berekend is afhankelijk van de waarde van **offset**, indien aanwezig. De onderste rij is de standaardwaarde. Voor andere diagrammen dan tabellen, wordt de laatste rij van de huidige kolom in het equivalent van de strakke tabel in het diagram geëvalueerd.

**Syntaxis:**

```
Bottom([TOTAL] expr [ , offset [,count ]])
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **expr** | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| **offset** | Als een **offset** n groter dan 1 wordt opgegeven, wordt de evaluatie van de uitdrukking n rijen boven de onderste rij geplaatst.<br>Als een negatief getal voor de beginwaarde wordt opgegeven, werkt de functie **Bottom** op dezelfde wijze als de functie **Top** met het overeenkomstige positieve getal als beginwaarde. |

| Argument | Beschrijving |
|---|---|
| **count** | Als een derde parameter **count** groter dan 1 is aangegeven, retourneert de functie niet één maar een bereik van **count** waarden. Een waarde voor elk van de laatste **count** rijen van het huidige kolomsegment. In deze vorm kan de functie worden gebruikt als argument voor een van de speciale bereikfuncties. *Bereikfuncties (page 1600)* |
| **TOTAL** | Als de tabel eendimensionaal is of als de kwalificatie **TOTAL** wordt gebruikt als argument, is het huidige kolomsegment altijd gelijk aan de hele kolom. |

*Een kolomsegment wordt gedefinieerd als een opeenvolgende subset van cellen die dezelfde waarden hebben voor de dimensies in de huidige sorteervolgorde. Interrecord-diagramfuncties worden berekend in het kolomsegment met uitzondering van de dimensie uiterst rechts in de equivalente strakke tabel. Als er slechts één dimensie is in het diagram, of als de kwalificatie TOTAL is opgegeven, wordt de uitdrukking geëvalueerd voor de volledige tabel.*

*Als de tabel of het equivalent van de tabel meerdere verticale dimensies heeft, omvat het huidige kolomsegment alleen rijen met dezelfde waarden als de huidige rij in alle dimensiekolommen, met uitzondering van de kolom waarin de laatste dimensie wordt weergegeven in de onderlinge sorteervolgorde van de velden.*

**Beperkingen:**

- Recursieve oproepen retourneren een NULL-waarde.
- Het is niet toegestaan te sorteren op y-waarden in grafieken of op uitdrukkingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van de grafiek. Deze sorteeropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie gebruikt in een visualisatie of tabel, wordt de sortering van de visualisatie teruggezet naar de gesorteerde invoer van deze functie.

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `Bottom( Sum(Sales) )` | Retourneert de waarde van `Sales` voor de laatste (onderste) rij van een kolomsegment in een tabel. |

## Voorbeeld - Basisprincipes voor Bottom

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Year`
  - `Sales`

### Load-script

```
Example:
LOAD * INLINE [
Year, Sales
2019, 1000
2020, 1500
2021, 1200
2022, 1800
2023, 1700
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Year`

Maak de volgende metingen:

- `=Sum(Sales)`, om de jaaromzet te vinden.
- `=Bottom(Sum(Sales))`, om de verkoopwaarde voor het afgelopen jaar te vinden.
- `=Sum(Sales) - Bottom(Sum(Sales))`, om de variantie tussen de `Sales` waarde van het huidige jaar en de waarde van het afgelopen jaar te berekenen.
- `=Sum(Sales)+Bottom(Sum(Sales), 3)`, om de som van de verkoopwaarde van het huidige jaar en de waarde van het op twee na laatste jaar te berekenen.
- `=(Sales - Bottom(Sales)) / Bottom(Sales)`, om de verandering in `Sales` in vergelijking tot het afgelopen jaar als een verhouding / percentage weer te geven. Om deze waarde als percentage weer te geven, selecteert u onder **GetalnotatieGetal > Eenvoudige opmaak > 12,34%**.

Sorteer de tabel in de dimensie `Year` in oplopende volgorde.

Resultatentabel

| Year | Sum (Sales) | Bottom (Sum (Sales)) | Sum(Sales) - Bottom(Sum (Sales)) | Sum (Sales)+Bottom (Sum(Sales), 3) | (Sales - Bottom (Sales)) / Bottom (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% |

Als u de resultaten bekijkt, kunt u de `Sales` waarden in de huidige periode vergelijken met de afgelopen, of recentste, periode.

- De meting `Sum(Sales) - Bottom(Sum(Sales))` retourneert de wijziging in de `Sales` waarden, waarbij de huidige periode wordt vergeleken met de recentste periode.
- De meting `Sum(Sales)+Bottom(Sum(Sales), 3)` retourneert de som van `Sales` voor de huidige rij en de derde recentste periode.
- De meting `(Sales - Bottom(Sales)) / Bottom(Sales)` retourneert de wijziging in `Sales` ten opzichte van de recentste periode als een percentage.

*Deze functie kan behalve in tabellen ook worden gebruikt in diagrammen, bijvoorbeeld in staafdiagrammen.*

**Doe het volgende:**

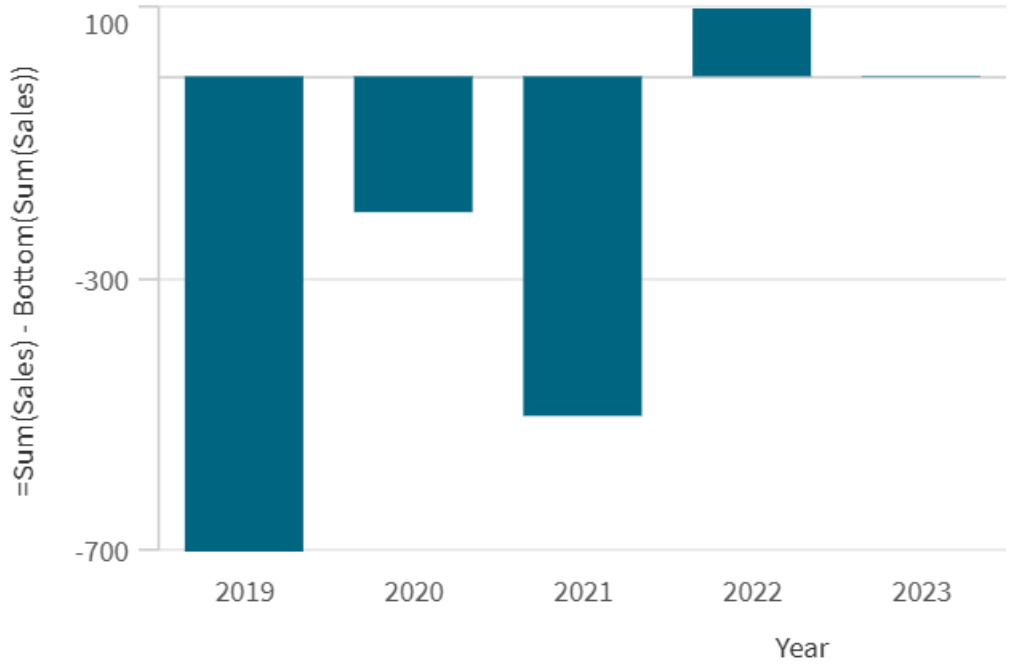
Maak een staafdiagram en voeg dit veld toe als dimensie:

- `Year`

Maak de volgende meting:

- `=Sum(Sales) - Bottom(Sum(Sales))`, om de variantie in verkoop tussen de huidige periode en de recentste periode weer te geven.

*Staafdiagram met resultaten voor de functie Bottom*




Door de functie `Bottom` in een staafdiagram te gebruiken, krijgt u een visuele vergelijking van de variantie tussen de verkopen van de huidige periode en die van de recentste periode.

## Voorbeeld 5 – Scenario met Bottom

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `OrderID`
  - `Product`
  - `SalesValue`
  - `OrderDate`

### Load-script

```
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'
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Product`

Maak de volgende metingen:

- `=RangeAvg (Bottom(Sum(SalesValue),1,3))`, om het gemiddelde van de 3 recentste rijen te retourneren.
- `=Sum(SalesValue) - RangeAvg (Bottom(Sum(SalesValue),1,3))`, om de variantie tussen de huidige rij en het gemiddelde van de 3 recentste rijen te berekenen.
- `=Sum(SalesValue)` om de totale `SalesValue` voor de huidige rij te berekenen.

Resultatentabel

| 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 |

De resultaten laten zien hoe u een combinatie van de functie `RangeSum` en de functie `Bottom` kunt gebruiken om verkoopwaarden te analyseren en vergelijken.

## Voorbeeld - Diagramuitdrukking met gebruik van Bottom

Diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Twee gegevenstabellen genaamd `Monthnames` en `Sales2013`.

### Load-script

```
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 '|');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Customer`

Maak de volgende metingen:

- `=Sum(Sales)`, om de som van verkoop te berekenen.
- `=Bottom(Sum(Sales))`, om de som van de verkopen van de `Customer` in de onderste rij te tonen.
- `=Sum(Sales)+Bottom(Sum(Sales))`, om de `Sum(Sales)` waarde van de huidige `Customer` toe te voegen aan de `Customer` in de onderste rij.
- `=Sum(Sales)+Bottom(Sum(Sales), 3)`, om de `Sum(Sales)` waarde toe te voegen van de huidige `Customer` met de waarde voor de `Customer` die derde is van de onderste rij.

Resultatentabel

| Customer | Sum (Sales) | Bottom(Sum (Sales)) | Sum(Sales)+Bottom (Sum(Sales)) | Sum(Sales)+Bottom(Sum (Sales), 3) |
|---|---|---|---|---|
| **Totals** | **2566** | **757** | **3323** | **3105** |
| Astrida | 587 | 757 | 1344 | 1126 |

| Customer | Sum (Sales) | Bottom(Sum (Sales)) | Sum(Sales)+Bottom (Sum(Sales)) | Sum(Sales)+Bottom(Sum (Sales), 3) |
|---|---|---|---|---|
| **Totals** | **2566** | **757** | **3323** | **3105** |
| Betacab | 539 | 757 | 1296 | 1078 |
| Canutilty | 683 | 757 | 1440 | 1222 |
| Divadip | 757 | 757 | 1514 | 1296 |

De meting `Bottom(Sum(Sales))` retourneert `757` voor alle rijen omdat dit de waarde van de onderste rij is: `Divadip`.

Voor de meting `Sum(Sales)+Bottom(Sum(Sales))` retourneert de rij `Astrida` `1344`, dit het resultaat van het optellen van de `Sum(Sales)` waarde voor `Astrida` en `Divadip`, de `Customer` in de onderste rij.

Voor de meting `Sum(Sales)+Bottom(Sum(Sales), 3)`, is het argument `offset` ingesteld op `3`. De uitdrukking voegt de `Sum(Sales)` waarde voor de huidige rij en de derde rij van onderen toe, `Betacab`. De waarde `1126` wordt bijvoorbeeld geretourneerd voor de klant `Astrida`. Dit is het resultaat van het optellen van de `Sum (Sales)` waarde voor `Astrida` en de `SumSales` voor `Betacab` (`587+539`).

**Zie ook:**

p *Top - diagramfunctie (page 1542)*

## Column - diagramfunctie

**Column()** retourneert de waarde in de kolom die overeenkomt met **ColumnNo** in een strakke tabel, waarbij dimensies buiten beschouwing worden gelaten. Zo retourneert **Column(2)** bijvoorbeeld de waarde van de tweede kolom met metingen.

**Syntaxis:**

```
Column(ColumnNo)
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **ColumnNo** | Kolomnummer van een kolom in de tabel die een meting bevat.<br>*De functie Column() negeert dimensiekolommen.* |

**Beperkingen:**

- Recursieve oproepen retourneren een NULL-waarde.
- Als **ColumnNo** verwijst naar een kolom waarvoor geen meting beschikbaar is, wordt een NULL-waarde geretourneerd.
- Het is niet toegestaan te sorteren op y-waarden in grafieken of op uitdrukkingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van de grafiek. Deze sorteeropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie gebruikt in een visualisatie of tabel, wordt de sortering van de visualisatie teruggezet naar de gesorteerde invoer van deze functie.

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `Column(1)/Column(2)` | Retourneert de waarde van de eerste meting gedeeld door de waarde van de tweede meting in het diagram. |

## Voorbeeld - Basisprincipes voor Column

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Customer`
  - `Product`
  - `UnitSales`
  - `UnitPrice`

### Load-script

```
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 '|');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `Customer`
- `Product`
- `UnitSales`
- `UnitPrice`

Maak de volgende metingen:

- `=Sum(UnitPrice*UnitSales)`, om de jaaromzet te vinden.
- `=Sum(TOTAL UnitPrice*UnitSales)`, om de totale verkoopwaarde te vinden.
- `=100*Column(1)/Column(2)`, om de totale verkooppercentage (%) van de totale verkoop te berekenen.

Resultatentabel

| 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 |

De resultaten laten zien dat u gegevens kunt analyseren door naar een kolompositie te verwijzen en die logica opnieuw te gebruiken in een metinguitdrukking. Bijvoorbeeld, de meting `100*Column(1)/Column(2)` neemt de waarde van de eerste meting, `Sum(UnitPrice*UnitSales)`, deelt deze door de waarde van de tweede meting, `Sum(TOTAL UnitPrice*UnitSales)`, en vermenigvuldigt vervolgens het resultaat met `100`.

## Voorbeeld 5 – Scenario met Column

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
    - `Month`
    - `Sales`
    - `Expenses`
    - `Profit`

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Product`

Maak de volgende metingen:

- `=Sum(Sales)`, om de verkoop te aggregeren.
- `=Sum(Expenses)`, om de verkoop te aggregeren.

- `=Column(1) - Column(2)`, om het verschil tussen de verkoop- en de uitgavenkolom te berekenen.

Resultatentabel

| Month | Sum(Sales) | Sum(Expenses) | Column(1) - Column(2) |
|---|---|---|---|
| **Totals** | **132500** | **46100** | **86400** |
| 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 |

De resultaten van de tabel laten zien hoe u de functie `Column` kunt gebruiken om gegevens te analyseren. De meting `Column(1) - Column(2)` retourneert het overgebleven verkoopbedrag na aftrek van de kosten (kolom van de de tweede meting).

## Voorbeeld - Column scenario voor het analyseren van verkoopgegevens

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `ProductName`
  - `Q1_Sales`
  - `Q2_Sales`
  - `Q3_Sales`
  - `Q4_Sales`

### Load-script

```
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
];
```

**Resultaten**

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `ProductName`

Maak de volgende metingen:

- `=Sum(Q1_Sales)`, om de verkoop van het eerste kwartaal te aggregeren.
- `=Sum(Q2_Sales)`, om de verkoop van het tweede kwartaal te aggregeren.
- `=Sum(Q3_Sales)`, om de verkoop van het derde kwartaal te aggregeren.
- `=Sum(Q4_Sales)`, om de verkoop van het vierde kwartaal te aggregeren.
- `=Column(1) - Column(2)`, om de verhouding van `Q1_Sales` tot `Q2_Sales` te berekenen.
- `=(Column(2) - Column(1)) / Column(1)` om de groeipercentages tussen twee perioden te berekenen.

Resultatentabel

| 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 |

De tabelresultaten laten zien hoe u de functie `Column` kunt gebruiken om gegevens te analyseren.

## Dimensionality - diagramfunctie

**Dimensionality()** retourneert het aantal dimensies voor de huidige rij. In het geval van draaitabellen retourneert de functie het totale aantal kolommen met dimensies met een niet-geaggregeerde inhoud, dat wil zeggen kolommen die geen subtotalen of samengevouwen aggregaten bevatten.

**Syntaxis:**

```
Dimensionality ( )
```

**Retourgegevenstypen:** geheel getal

**Beperkingen:**

Deze functie is alleen beschikbaar in diagrammen. Voor alle diagramtypen, met uitzondering van de draaitabel, retourneert deze functie het aantal dimensies in alle rijen met uitzondering van het totaal dat gelijk is aan 0.

Het is niet toegestaan te sorteren op y-waarden in grafieken of op uitdrukkingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van de grafiek. Deze sorteeropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie gebruikt in een visualisatie of tabel, wordt de sortering van de visualisatie teruggezet naar de gesorteerde invoer van deze functie.

## Voorbeeld: diagramuitdrukking met Dimensionality

Diagramuitdrukking

**Overzicht**

Gebruik de functie `Dimensionality` met een draaitabel als een diagramuitdrukking wanneer u verschillende celopmaakopties wilt toepassen afhankelijk van het aantal dimensies in een rij die niet-geaggregeerde gegevens heeft. In dit voorbeeld wordt een achtergrondkleur toegepast op tabelcellen die overeenkomen met een bepaalde voorwaarde.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Country`
  - `Product`
  - `Sales`
  - `Budget`

**Load-script**

```
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 ',');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe draaotabel en voeg deze velden toe als dimensies:

- `Country`
- `Product`

Maak de volgende metingen:

- `=Sum(Sales)`, om de som van de `Sales` waarden te berekenen.
- `=Sum(Budget)`, om de som van de `Budget` waarden te berekenen.
- `=Dimensionality()`, om het aantal dimensies voor de huidige rij te retourneren.

Voer in het deelvenster **Eigenschappen** de volgende uitdrukking in als de **Uitdrukking achtergrondkleur** voor de `Sum(Sales)`-meting:

```
=If(Dimensionality()=1 and Sum(Sales)<Sum(Budget),RGB(255,156,156),
  If(Dimensionality()=2 and Sum(Sales)<Sum(Budget),RGB(178,29,29)
  )
 )
```

De uitvoer van de achtergrondkleuruitdrukking past een lichtrode of donkerrode celachtergrond toe op de tabelcellen wanneer aan de toepasselijke voorwaarden is voldaan.

Draaitabel met resultaten voor de rij dimensies: Country, Product

| Country<br>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 |

Wanneer de dimensie van de eerste rij (`Country`) wordt weergegeven, retourneert de functie `Dimensionality` `1`. Wanneer u een dimensie van een rij, zoals `Germany`, uitvouwt om de tweede dimensie van de rij (`Product`) weer te geven, retourneert de functie `2`.

De achtergrondkleuruitdrukkingen voor de meting `Sum(Sales)` voeren berekeningen uit op basis van dimensionaliteit en verkoop om verschillende resultaten te markeren. Wanneer de dimensionaliteit `1` is, verschijnt `Country` met `Sum(Sales)` minder dan `Sum(Budget)` lichtrood, in dit voorbeeld `Germany` en `Ireland`. Wanneer de dimensionaliteit `2` is, verschijnt `Product` met `Sum(Sales)` minder dan `Sum(Budget)` donkerrood, in dit voorbeeld `AA` en `BB` voor `Germany`.

## Voorbeeld 5 – Scenario met Dimensionality

Diagramuitdrukking

### Overzicht

Dit voorbeeld gebruikt de functie `Dimensionality` om details op een rapport toe te passen en tabelcellen voor sommige dimensies te markeren.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
    - `Region`
    - `SalesProduct`
    - `Year`
    - `SalesValue`
    - `Quantity`
    - `Product`

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe draaotabel en voeg deze velden toe als dimensies:

- `Year`
- `SalesProduct`
- `Region`

Maak de volgende metingen:

- `=If(Dimensionality()=1, 'Total', 'Detail')`, om aan te geven of het een `Total` of `Detail` rij is.
- `=If(Dimensionality()=3, 'Region-Level', If(Dimensionality()=2, 'Product-Level', 'Summary-Level'))`, om het weergegeven dimensieniveau aan te geven.
- `=Sum(SalesValue)`, om de som van verkoop te berekenen.

Voer in het deelvenster **Eigenschappen** de volgende uitdrukking in als de **Uitdrukking achtergrondkleur** voor de `Sum(SalesValue)`-meting:

```
=If(Dimensionality()=1, LightGray())
```

De uitvoer van de achtergrondkleuruitdrukking past een lichtgrijze celachtergrond toe op de tabelcellen wanneer de dimensionaliteit `1` is.

Draaitabel met resultaten voor de rij dimensies: Country, Product

| **Jaar** | | | |
|---|---|---|---|
| **SalesProduct** | | | |
| **Regio** | | | |
| | **If(Dimensionality()=1, 'Total', 'Detail')** | **If(Dimensionality()=3, 'Region-Level', If(Dimensionality()=2, 'Product-Level', 'Summary-Level'))** | **Sum (SalesValue)** |
| (+) 2022 | Totaal | Summary-Level | 134000 |
| (-) 2023 | Totaal | Summary-Level | 200000 |
| (-)Laptop | Detail | Product-Level | 86000 |
| East | Detail | Region-Level | 21000 |
| North | Detail | Region-Level | 20000 |
| South | Detail | Region-Level | 22000 |
| West | Detail | Region-Level | 23000 |

| Jaar | | | |
|---|---|---|---|
| SalesProduct | | | |
| Regio | | | |
| | If(Dimensionality ()=1, 'Total', 'Detail') | If(Dimensionality()=3, 'Region-Level', If (Dimensionality()=2, 'Product-Level', 'Summary-Level')) | Sum (SalesValue) |
| (-) Mobile | Detail | Product-Level | 63000 |
| (-) Tablet | Detail | Product-Level | 51000 |

U kunt zien hoe de functie `Dimensionality` gebruikt kan worden om het rapport gedetailleerder te maken en alle rijen op het hoogste niveau te markeren.

## Exists

**Exists()** bepaalt of een specifieke veldwaarde al is geladen in het veld in het script voor het laden van gegevens. De functie retourneert TRUE of FALSE, zodat deze kan worden gebruikt in de **where**-clausule van een **LOAD**-opdracht of een **IF**-opdracht.

> *U kunt **Not Exists()** ook gebruiken om te bepalen of een veldwaarde niet is geladen, maar voorzichtigheid is geboden als u **Not Exists()** gebruikt in een where-clausule. De functie **Exists ()** test zowel eerder geladen tabellen als eerder geladen waarden in de huidige tabel. Dit betekent dat alleen het eerste exemplaar wordt geladen. Wanneer een tweede exemplaar wordt gevonden, is de waarde al geladen. Bekijk de voorbeelden voor meer informatie.*

**Syntaxis:**

```
Exists(field_name [, expr])
```

**Retourgegevenstypen:** Booleaanse waarde

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| field_name | De naam van het veld waarin u een waarde wilt zoeken. U kunt een expliciete veldnaam gebruiken zonder aanhalingstekens.<br><br>Het veld moet al door het script zijn geladen. Dit betekent dat u niet naar een veld kunt verwijzen dat in een clausule verderop in het script wordt geladen. |

| Argument | Beschrijving |
|---|---|
| expr | De waarde waarvan u wilt controleren of deze bestaat. U kunt een expliciete waarde gebruiken of een uitdrukking die verwijst naar een of meer velden in de huidige load-opdracht.<br><br>*U kunt niet verwijzen naar velden die niet in de huidige load-opdracht zijn opgenomen.*<br><br>Dit argument is optioneel. Als u dit weglaat, controleert de functie of de waarde van **field_name** al bestaat in het huidige record. |

Voorbeelden en resultaten:

### Voorbeeld 1

```
Exists (Employee)
```

Retourneert -1 (True) als de waarde van het veld **Employee** in de huidige record al bestaat in een eerder gelezen record met dat veld.

De opdrachten `Exists (Employee, Employee)` en `Exists (Employee)` zijn equivalent.

### Voorbeeld 2

```
Exists(Employee, 'Bill')
```

Retourneert -1 (True) als de veldwaarde **'Bill'** wordt gevonden in de huidige inhoud van het veld **Employee**.

### Voorbeeld 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;
```

Dit resulteert in een tabel die u kunt gebruiken in een tabelvisualisatie met de dimensies Employee en Address.

De where-clausule, `where Exists (Employee)`, betekent dat alleen de namen uit de tabel Citizens die tevens in Employees voorkomen in de nieuwe tabel worden geladen. Met de Drop-opdracht wordt de tabel Employees verwijderd om verwarring te voorkomen.

Resultaten

| | |
|---|---|
| Employee | Address |
| Bill | New York |
| John | Miami |
| Steve | Chicago |

### Voorbeeld 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;
```

De where-clausule omvat not: `where not Exists (Employee)`.

Dit betekent dat alleen de namen uit de tabel Citizens die niet in Employees voorkomen in de nieuwe tabel worden geladen.

Houd er rekening mee dat hoewel er twee waarden voor Lucy in de tabel Citizens staan, slechts één in de resultaattabel wordt opgenomen. Wanneer u de eerste rij laadt met de waarde Lucy, wordt de waarde opgenomen in het veld Employee. Dus wanneer vervolgens de tweede rij wordt gecontroleerd, bestaat de waarde al.

Resultaten

| Employee | Address |
|---|---|
| Mary | London |
| Lucy | Madrid |

### Voorbeeld 5

In dit voorbeeld wordt getoond hoe u alle waarden kunt laden.

```
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;
```

Om alle waarden voor Lucy te kunnen ophalen, zijn er twee dingen gewijzigd:

- Een voorgaande lading van de tabel Employees was ingevoerd op de plaats waar Employee is gewijzigd in Name.
  `Load Employee As Name;`
- De Where-voorwaarde in Citizens is gewijzigd in:
  `not Exists (Name, Employee).`

Hierdoor worden velden voor Name en Employee gemaakt. Wanneer de tweede rij met Lucy wordt gecontroleerd, bestaat deze nog niet in Name.

Resultaten

| Employee | Address |
|---|---|
| Mary | London |
| Lucy | Madrid |
| Lucy | Paris |

## FieldIndex

**FieldIndex()** retourneert de positie van de veldwaarde **value** in het veld **field_name** (op volgorde van laden).

**Syntaxis:**

```
FieldIndex(field_name , value)
```

**Retourgegevenstypen:** geheel getal

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| field_name | Naam van het veld waarvoor de index is vereist. Bijvoorbeeld de kolom in een tabel. Moet worden opgegeven als een tekenreekswaarde. Dit betekent dat de veldnaam tussen enkele aanhalingstekens moet staan. |
| value | De waarde van het veld **field_name**. |

**Beperkingen:**

- Als **value** niet wordt gevonden tussen de veldwaarden van het veld **field_name**, wordt 0 geretourneerd.
- Het is niet toegestaan te sorteren op y-waarden in grafieken of op uitdrukkingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van de grafiek. Deze sorteeropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie gebruikt in een visualisatie of tabel, wordt de sortering van de visualisatie teruggezet naar de gesorteerde invoer van deze functie. Deze beperking is niet van toepassing op de bijbehorende scriptfunctie.

**Voorbeelden en resultaten:**

In de volgende voorbeelden wordt gebruikgemaakt van het veld: **First name** uit de tabel **Names**.

Voorbeelden en resultaten

| Voorbeelden | Resultaten |
|---|---|
| Voeg de voorbeeldgegevens toe aan uw app en voer deze uit. | De tabel **Names** is geladen, zoals in de steekproefgegevens. |
| Diagramfunctie: voeg in een tabel die de dimensie First name bevat het volgende in als een meting: | |

| Voorbeelden | Resultaten |
|---|---|
| `FieldIndex ('First name','John')` | 1, aangezien 'John' als eerste wordt weergegeven in de laadvolgorde van het veld **First name**. Houd er rekening mee dat **John** als nummer 2 zou worden weergegeven in een filtervak, omdat hierbij alfabetisch wordt gesorteerd en niet op laadvolgorde. |
| `FieldIndex ('First name','Peter')` | 4, aangezien **FieldIndex()** slechts één waarde retourneert, namelijk het eerste exemplaar in de laadvolgorde. |
| Scriptfunctie: gegeven dat de tabel **Names** is geladen, zoals in de steekproefgegevens: | |
| `John1:`<br><br>`Load FieldIndex('First name','John') as MyJohnPos`<br><br>`Resident Names;` | `MyJohnPos`=1, aangezien 'John' als eerste wordt weergegeven in de laadvolgorde van het veld **First name**. Houd er rekening mee dat **John** als nummer 2 zou worden weergegeven in een filtervak, omdat hierbij alfabetisch wordt gesorteerd en niet op laadvolgorde. |
| `Peter1:`<br><br>`Load FieldIndex('First name','Peter') as MyPeterPos`<br><br>`Resident Names;` | `MyPeterPos`=4, aangezien **FieldIndex()** slechts één waarde retourneert, namelijk het eerste exemplaar in de laadvolgorde. |

Gebruikte gegevens in voorbeeld:

```
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()** retourneert de waarde die is gevonden in positie **elem_no** van het veld **field_name** (op volgorde van laden).

**Syntaxis:**

```
FieldValue(field_name , elem_no)
```

**Retourgegevenstypen:** dubbele waarde

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| field_name | Naam van het veld waarvoor de waarde is vereist. Bijvoorbeeld de kolom in een tabel. Moet worden opgegeven als een tekenreekswaarde. Dit betekent dat de veldnaam tussen enkele aanhalingstekens moet staan. |
| elem_no | Het positienummer (elementnummer) van het veld, na de laadvolgorde, waarvoor de waarde wordt geretourneerd. Dit kan overeenkomen met de rij in een tabel, maar het is afhankelijk van de volgorde waaraan de elementen (rijen) worden geladen. |

**Beperkingen:**

- Als **elem_no** groter is dan het aantal veldwaarden, wordt NULL geretourneerd.
- Het is niet toegestaan te sorteren op y-waarden in grafieken of op uitdrukkingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van de grafiek. Deze sorteeropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie gebruikt in een visualisatie of tabel, wordt de sortering van de visualisatie teruggezet naar de gesorteerde invoer van deze functie. Deze beperking is niet van toepassing op de bijbehorende scriptfunctie.

Voorbeeld

## Load-script

Laad de volgende gegevens als een inline load in de editor voor laden van gegevens om het onderstaande voorbeeld te maken.

```
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;
```

### Een visualisatie maken

Maak een tabelvisualisatie in een Qlik Sense-werkblad. Voeg de velden **First name**, **MyPos1** en **MyPos2** toe aan de tabel.

### Resultaat

| First name | MyPos1 | MyPos2 |
|---|---|---|
| Jane | John | Jane |
| John | John | Jane |
| Mark | John | Jane |
| Peter | John | Jane |
| Sue | John | Jane |

### Uitleg

**FieldValue('First name','1')** resulteert in John als de waarde voor **MyPos1** voor alle voornamen omdat John als eerste verschijnt in de laadvolgorde van het veld **First name**. Houd er rekening mee dat John in een filtervak als nummer 2 van boven zou worden weergegeven, na Jane, omdat hierbij alfabetisch wordt gesorteerd en niet op laadvolgorde.

**FieldValue('First name','5')** resulteert in Jane als de waarde voor **MyPos2** voor alle voornamen omdat Jane als vijfde verschijnt in de laadvolgorde van het veld **First name**.

## FieldValueCount

**FieldValueCount()** is een functie voor **geheel getal** waarmee het aantal distinctieve waarden in een veld wordt gevonden.

Bij een gedeeltelijke lading kunnen waarden van de gegevens worden verwijderd, wat niet wordt weerspiegeld in het geretourneerde getal. Het geretourneerde getal komt overeen met alle afzonderlijke waarden die in de initiële lading of opeenvolgende gedeeltelijke ladingen zijn geladen.

*Het is niet toegestaan te sorteren op y-waarden in grafieken of op uitdrukkingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van de grafiek. Deze sorteeropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie gebruikt in een visualisatie of tabel, wordt de sortering van de visualisatie teruggezet naar de gesorteerde invoer van deze functie. Deze beperking is niet van toepassing op de bijbehorende scriptfunctie.*

**Syntaxis:**

```
FieldValueCount(field_name)
```

**Retourgegevenstypen:** geheel getal

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| **field_name** | Naam van het veld waarvoor de waarde is vereist. Bijvoorbeeld de kolom in een tabel. Moet worden opgegeven als een tekenreekswaarde. Dit betekent dat de veldnaam tussen enkele aanhalingstekens moet staan. |

**Voorbeelden en resultaten:**

In de volgende voorbeelden wordt gebruikgemaakt van het veld **First name** uit de tabel **Names**.

Voorbeelden en resultaten

| Voorbeelden | Resultaten |
|---|---|
| Voeg de voorbeeldgegevens toe aan uw app en voer deze uit. | De tabel **Names** is geladen, zoals in de steekproefgegevens. |
| Diagramfunctie: voeg in een tabel die de dimensie First name bevat het volgende in als een meting. | |
| `FieldValueCount('First name')` | 5 aangezien **Peter** tweemaal voorkomt. |
| `FieldValueCount('Initials')` | 6 aangezien **Initials** alleen distinctieve waarden heeft. |
| Scriptfunctie: gegeven dat de tabel **Names** is geladen, zoals in de steekproefgegevens: | |
| `FieldCount1:`<br><br>`Load FieldValueCount('First name') as MyFieldCount1`<br><br>`Resident Names;` | `MyFieldCount1=`5 aangezien 'Peter' tweemaal voorkomt. |
| `FieldCount2:`<br><br>`Load FieldValueCount('Initials') as MyInitialsCount1`<br><br>`Resident Names;` | `MyFieldCount1=`6 aangezien 'Initials' alleen distinctieve waarden heeft. |

Gegevens die worden gebruikt in voorbeelden:

```
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()** zoekt een tabel die al is geladen en retourneert de waarde van **field_name** die overeenkomt met het eerste exemplaar van de waarde **match_field_value** in het veld **match_field_name**. De tabel kan de huidige tabel zijn of een andere tabel die eerder is geladen.

**Syntaxis:**

```
LookUp(field_name, match_field_name, match_field_value [, table_name])
```

**Retourgegevenstypen:** dubbele waarde

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| **field_name** | Naam van het veld waarvoor de retourwaarde is vereist. De invoerwaarde moet worden aangegeven als een tekenreeks (bijvoorbeeld literal tussen aanhalingstekens). |
| **match_ field_name** | Naam van het veld waarin **match_field_value** moet worden opgezocht. De invoerwaarde moet worden aangegeven als een tekenreeks (bijvoorbeeld literal tussen aanhalingstekens). |
| **match_ field_value** | Waarde die moet worden opgezocht in het veld **match_field_name**. |
| **table_name** | Naam van de tabel waarin de waarde moet worden opgezocht. De invoerwaarde moet worden aangegeven als een tekenreeks (bijvoorbeeld literal tussen aanhalingstekens).<br><br>Als **table_name** wordt weggelaten, wordt de huidige tabel aangenomen. |

*Argumenten zonder aanhalingstekens verwijzen naar de huidige tabel. Als u naar andere tabellen wilt verwijzen, zet u een argument tussen enkele aanhalingstekens.*

**Beperkingen:**

De volgorde waarin de zoekopdracht wordt uitgevoerd is de laadvolgorde, tenzij de tabel het resultaat is van complexe bewerkingen, zoals joins, waarbij de volgorde niet goed is gedefinieerd. Zowel **field_name** als **match_field_name** moeten velden in dezelfde tabel zijn, aangegeven met **table_name**.

Als geen waarde wordt gevonden, wordt NULL geretourneerd.

Voorbeeld

### Load-script

Laad de volgende gegevens als een inline load in de editor voor laden van gegevens om het onderstaande voorbeeld te maken.

```
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;
```

### Een visualisatie maken

Maak een tabelvisualisatie in een Qlik Sense-werkblad. Voeg de velden **ProductID**, **InvoiceID**, **CustomerID**, **Units** en **CategoryID** toe aan de tabel.

### Resultaat

Resultaattabel

| 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 |

### Uitleg

In de steekproefgegevens wordt de functie **Lookup()** als volgt gebruikt:

```
Lookup('Category', 'ProductID', ProductID, 'ProductList')
```

De tabel **ProductList** wordt als eerste geladen.

De functie **Lookup()** wordt gebruikt voor het maken van de tabel **OrderData**. Hiermee wordt het derde argument opgegeven als **ProductID**. Dit is het veld waarvoor de waarde moet worden opgezocht in het tweede argument **'ProductID'** in de **ProductList**, zoals aangegeven door de enkele aanhalingstekens er omheen.

De functie retourneert de waarde voor '**Category**' (in de tabel **ProductList**), geladen als **CategoryID**.

Met de opdracht **drop** wordt de tabel **ProductList** verwijderd uit het gegevensmodel, omdat deze niet vereist is. Nu blijft de tabel **OrderData** over.

*De functie Lookup() is flexibel en kan toegang verkrijgen tot elke eerder geladen tabel. Deze functie is echter traag in vergelijking met de functie Applymap().*

**Zie ook:**

p *ApplyMap (page 1583)*

## NoOfRows

**NoOfRows()** retourneert het aantal rijen in het huidige kolomsegment in een tabel. Voor bitmapgrafieken wordt met **NoOfRows()** het aantal rijen in het equivalent van de standaard tabel voor de grafiek geretourneerd.

Als de tabel of het equivalent van de tabel meerdere verticale dimensies heeft, omvat het huidige kolomsegment alleen rijen met dezelfde waarden als de huidige rij in alle dimensiekolommen, met uitzondering van de kolom waarin de laatste dimensie wordt weergegeven in de onderlinge sorteervolgorde van de velden.

*Het is niet toegestaan te sorteren op y-waarden in grafieken of op uitdrukkingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van de grafiek. Deze sorteeropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie gebruikt in een visualisatie of tabel, wordt de sortering van de visualisatie teruggezet naar de gesorteerde invoer van deze functie.*

**Syntaxis:**

```
NoOfRows([TOTAL])
```

**Retourgegevenstypen:** geheel getal

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| TOTAL | Als de tabel eendimensionaal is of als de kwalificatie TOTAL wordt gebruikt als argument, is het huidige kolomsegment altijd gelijk aan de hele kolom. |

## Voorbeeld: diagramuitdrukking met NoOfRows

Voorbeeld - diagramuitdrukking

### Load-script

Laad de volgende gegevens als een inline-load in de editor voor laden van gegevens om de onderstaande voorbeelden van diagramuitdrukkingen te maken.

```
        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 '|');
```

### Diagramuitdrukking

Maak een tabelvisualisatie in een Qlik Sense-werkblad met **Region** en **SubRegion** als dimensies. Voeg `RowNo( ), NoOfRows()` en `NoOfRows(Total)` toe als metingen.

### Resultaat

| 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 |

| Region | SubRegion | RowNo() | NoOfRows() | NoOfRows (Total) |
|---|---|---|---|---|
| Asia | Eastern | 1 | 1 | 9 |
| Europe | Eastern | 1 | 3 | 9 |
| Europe | Northern | 2 | 3 | 9 |
| Eurrope | Western | 3 | 3 | 9 |
| Oceania | Australia | 1 | 1 | 9 |

### Uitleg

In dit voorbeeld wordt de sorteervolgorde bepaald door de eerste dimensie, Region. Dit heeft als gevolg dat elk kolomsegment bestaat uit een groep regio's met dezelfde waarde, bijvoorbeeld Afrika.

De kolom **RowNo()** toont de rijnummers voor elk kolomsegment, zo zijn er bijvoorbeeld twee rijen voor de regio Afrika. De rijnummering begint vervolgens opnieuw bij 1 voor het volgende kolomsegment, Americas.

De kolom **NoOfRows()** telt het aantal rijen in elk kolomsegment, zo heeft Europa bijvoorbeeld drie rijen in het kolomsegment.

In de kolom **NoOfRows(Total)** worden de dimensies genegeerd vanwege het argument `TOTAL` voor `NoOfRows()` en het aantal rijen in de tabel worden geteld.

Als de tabel is gesorteerd op de tweede dimensie, SubRegion, worden de kolomsegmenten gebaseerd op die dimensie, waardoor de rijnummers wijzigen voor elke SubRegion.

**Zie ook:**

p *RowNo - diagramfunctie (page 707)*

## Peek

**Peek()** retourneert de waarde van een veld in een tabel voor een rij die al is geladen. Het rijnummer kan worden opgegeven, net als de tabel. Als er geen rijnummer is opgegeven, wordt het laatst geladen record gebruikt.

De functie `Peek()` wordt het vaakst gebruikt om relevante grenzen in een eerder geladen tabel te vinden, oftewel de eerste waarde of laatste waarde van een specifiek veld. In de meeste gevallen wordt deze waarde opgeslagen in een variabele voor later gebruik, bijvoorbeeld als een voorwaarde in een do-while loop.

**Syntaxis:**

```
Peek(

field_name

[, row_no[, table_name ] ])
```

**Retourgegevenstypen:** dubbele waarde

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| field_name | Naam van het veld waarvoor de retourwaarde is vereist.De invoerwaarde moet worden aangegeven als een tekenreeks (bijvoorbeeld literal tussen aanhalingstekens). |
| row_no | De rij in de tabel die het vereiste veld aangeeft. Kan een uitdrukking zijn, maar moet worden herleid tot een geheel getal. 0 geeft de eerste record aan, 1 de tweede, enzovoort. Negatieve nummers geven de volgorde vanaf het eind van de tabel aan. -1 geeft de laatste gelezen record aan.<br><br>Als geen **row_no** wordt aangegeven, wordt -1 verondersteld. |
| table_name | Een label voor een tabel zonder dubbele punt aan het eind. Als geen **table_name** wordt vermeld, wordt de huidige tabel verondersteld. Als de tabelnaam buiten de **LOAD**-opdracht wordt gebruikt of als de tabelnaam verwijst naar een andere tabel, moet de **table_name** worden opgenomen. |

**Beperkingen:**

De functie kan alleen waarden van reeds geladen records retourneren. Dit betekent dat in de eerste record van een tabel een aanroep die -1 als `row_no` gebruikt NULL zal retourneren.

Voorbeelden en resultaten:

## Voorbeeld 1

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
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;
```

Resultaattabel

| 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 omdat `Peek('EmployeeCode',0, 'EmployeeDates')` de eerste waarde retourneert van EmployeeCode in de tabel EmployeeDates.

LastCode = 106 omdat `Peek('EmployeeCode',-1, 'EmployeeDates')` de laatste waarde retourneert van EmployeeCode in de tabel EmployeeDates.

Als de waarde van het argument **row_no** wordt vervangen, worden de waarden van de andere rijen in de tabel als volgt geretourneerd:

`Peek('EmployeeCode',2, 'EmployeeDates')` retourneert de derde waarde, 103, in de tabel als de FirstCode.

Houd er echter rekening mee dat als in deze voorbeelden de tabel niet wordt opgegeven als het derde argument **table_name**, de functie verwijst naar de huidige (in dit geval interne) tabel.

## Voorbeeld 2

Als u gegevens wilt bekijken die zich lager in een tabel bevinden, moet u dit in twee stappen doen: eerst laadt u de volledige tabel als een tijdelijke tabel en vervolgens sorteert u deze opnieuw met behulp van **Peek()**.

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
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.

Resultaattabel

| Id | Lijst | Waarde |
|---|---|---|
| 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 |

De opdracht **IF()** is samengesteld op basis van de tijdelijke tabel T1.
`Peek('ID')` verwijst naar het veld ID in de vorige rij in de huidige tabel T2.
`Peek('List')` verwijst naar het veld List in de vorige rij in de tabel T2, die momenteel wordt samengesteld terwijl de uitdrukking wordt geëvalueerd.

De opdracht wordt als volgt geëvalueerd:
Als de huidige waarde van ID gelijk is aan de vorige waarde van ID, wordt de waarde van Peek('List') aaneengeschakeld met de huidige waarde van Value weggeschreven. Anders wordt alleen de huidige waarde van Value weggeschreven.

Als Peek('List') al een aaneengeschakeld resultaat bevat, wordt het nieuwe resultaat van Peek('List') hier eveneens mee aaneengeschakeld.

*Let op de **Order by**-clausule. Deze geeft aan hoe de tabel wordt geordend (op ID in oplopende volgorde). Zonder deze clausule, wordt bij de functie Peek() gebruikgemaakt van de willekeurige volgorde van de interne tabel, hetgeen tot onvoorspelbare resultaten kan leiden.*

## Voorbeeld 3

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
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];
```

Resultaattabel

| 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 |

Het veld AmountMonthBefore bevat het aantal van de vorige maand.

Hier worden de parameters voor row_no en table_name weggelaten en worden de standaardwaarden gebruikt. In dit voorbeeld zijn de volgende functieaanroepen gelijk:

- Peek(Bedrag)
- Peek(Bedrag,-1)
- Peek(Bedrag,-1,'Bedragen')

Als u -1 gebruikt als row_no, wordt de waarde van de vorige rij gebruikt. Door deze waarde te vervangen, kunnen waarden van andere rijen in de tabel worden opgehaald:

Peek(Amount,2) retourneert de derde waarde in de tabel: 7.

## Voorbeeld 4

Gegevens moeten correct worden gesorteerd om de correcte resultaten te krijgen, maar helaas is dit niet altijd het geval. Bovendien kan de functie Peek() niet worden gebruikt om te verwijzen naar gegevens die nog niet zijn geladen. Door tijdelijke tabellen te gebruiken en gegevens door meerdere lagen te leiden, kunt u dergelijke problemen vermijden.

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
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;
```

### Uitleg

De initiële tabel wordt gesorteerd op maand, wat betekent dat de functie peek() in de meeste gevallen het aantal voor het verkeerde product retourneert. Daarom moet deze tabel opnieuw worden gestorteerd. Dit doet u door de gegevens door een tweede laag te leiden en een nieuwe tabel tmp2Amounts te maken. Let op de Order by-clausule. De records worden eerst gesorteerd op product, vervolgens op maand in oplopende volgorde.

De functie If() is vereist omdat AmountMonthBefore alleen moet worden berekend als de vorige rij de gegevens bevat voor hetzelfde product, maar dan van de vorige maand. Door het product van de huidige rij te vergelijken met het product van de vorige rij, kan deze voorwaarde worden gevalideerd.

Wanneer de tweede tabel wordt gemaakt, wordt de eerste tabel tmp1Amounts verwijderd met de opdracht Drop Table.

Tot slot gaan de gegevens door een derde laag, maar nu worden de maanden in omgekeerde volgorde gesorteerd. Op deze manier kan AmountMonthAfter ook worden berekend.

*De clausule Order by geeft aan wat de volgorde van de tabel is; zonder deze clausule, wordt bij de functie Peek() gebruikgemaakt van de willekeurige volgorde van de interne tabel, hetgeen tot onvoorspelbare resultaten kan leiden.*

**Resultaat**

Resultaattabel

| 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 | - |

## Voorbeeld 5

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
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;
```

**Resultaat**

Resultaattabel

| 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 |

**Uitleg**

De LOAD-instructie **Load *, rangesum(SumVal,peek('AccSumVal')) as AccSumVal** bevat een recursieve aanroep waarbij de vorige waarden worden toegevoegd aan de huidige waarde. Deze bewerking wordt gebruikt om een samenvoeging van waarden in het script te berekenen.

---

**Zie ook:**

## Previous

**Previous()** retourneert de waarde van de uitdrukking **expr** met gebruik van de gegevens uit de vorige invoerrecord die niet is genegeerd wegens een **where**-clausule. In de eerste record van een interne tabel retourneert de functie NULL.

**Syntaxis:**

```
Previous(expr)
```

**Retourgegevenstypen:** dubbele waarde

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| **expr** | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. De uitdrukking kan geneste functies **Previous()** bevatten om toegang te krijgen tot eerdere records. Gegevens worden rechtstreeks opgehaald uit de invoerbron. Zo kan ook worden verwezen naar velden die niet zijn geladen in Qlik Sense, en dus zelfs als deze niet zijn opgeslagen in de bijbehorende associatieve database. |

**Beperkingen:**

In de eerste record van een interne tabel retourneert de functie NULL.

**Voorbeeld:**

Voer het volgende in uw load-script in:

```
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 '|');
```

Door gebruik te maken van de functie **Previous()** in de opdracht **Load**, kunnen wij de huidige waarde van Sales vergelijken met de voorafgaande waarde en deze gebruiken in een derde veld, Increase.

Resultaattabel

| Maand | Sales. | Toename |
|---|---|---|
| 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 - diagramfunctie

**Top()** evalueert een uitdrukking in de eerste rij (bovenste rij) van een kolomsegment in een tabel. De rij waarvoor deze waarde wordt berekend is afhankelijk van de waarde van **offset**, indien aanwezig. De bovenste rij is de standaardwaarde. Voor andere diagrammen dan tabellen, wordt bij **Top()** de eerste rij van de huidige kolom in het equivalent van de standaard tabel in het diagram geëvalueerd.

**Syntaxis:**

```
Top([TOTAL] expr [ , offset [,count ]])
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **expr** | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |

| Argument | Beschrijving |
|---|---|
| **offset** | Als een **offset** of n groter dan 1 wordt opgegeven, wordt de evaluatie van de uitdrukking n rijen omlaag verplaatst onder de bovenste rij.<br><br>Als een negatief getal voor de beginwaarde wordt opgegeven, werkt de functie **Top** op dezelfde wijze als de functie **Bottom** met het overeenkomstige positieve getal als beginwaarde. |
| **count** | Als een derde parameter **count** groter dan 1 is aangegeven, retourneert de functie een bereik van **count** waarden. Een waarde voor elk van de laatste **count** rijen van het huidige kolomsegment. In deze vorm kan de functie worden gebruikt als argument voor een van de speciale bereikfuncties. *Bereikfuncties (page 1600)* |
| **TOTAL** | Als de tabel eendimensionaal is of als de kwalificatie **TOTAL** wordt gebruikt als argument, is het huidige kolomsegment altijd gelijk aan de hele kolom. |

*Een kolomsegment wordt gedefinieerd als een opeenvolgende subset van cellen die dezelfde waarden hebben voor de dimensies in de huidige sorteervolgorde. Interrecord-diagramfuncties worden berekend in het kolomsegment met uitzondering van de dimensie uiterst rechts in de equivalente strakke tabel. Als er slechts één dimensie is in het diagram, of als de kwalificatie TOTAL is opgegeven, wordt de uitdrukking geëvalueerd voor de volledige tabel.*

*Als de tabel of het equivalent van de tabel meerdere verticale dimensies heeft, omvat het huidige kolomsegment alleen rijen met dezelfde waarden als de huidige rij in alle dimensiekolommen, met uitzondering van de kolom waarin de laatste dimensie wordt weergegeven in de onderlinge sorteervolgorde van de velden.*

**Beperkingen:**

- Recursieve oproepen retourneren een NULL-waarde.
- Het is niet toegestaan te sorteren op y-waarden in grafieken of op uitdrukkingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van de grafiek. Deze sorteeropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie gebruikt in een visualisatie of tabel, wordt de sortering van de visualisatie teruggezet naar de gesorteerde invoer van deze functie.

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `Top( Sum(Sales) )` | Retourneert de waarde van `Sales` voor de eerste (bovenste) rij van een kolomsegment in een tabel. |

## Voorbeeld - Basisprincipes voor Top

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Year`
  - `Sales`

### Load-script

```
Example:
LOAD * INLINE [
Year, Sales
2019, 1000
2020, 1500
2021, 1200
2022, 1800
2023, 1700
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Year`

Maak de volgende metingen:

- `=Sum(Sales)`, om de jaaromzet te vinden.
- `=Top(Sum(Sales))`, om de verkoopwaarde voor het eerste jaar te vinden, dat is de eerste (bovenste) rij van een kolom in een tabel.
- `=Sum(Sales) - Top(Sum(Sales))`, om de variantie tussen de `Sales` waarde van de huidige rij en de waarde voor de eerste (bovenste) rij van een kolom te berekenen.
- `=Sum(Sales)-Top(Sum(Sales), 3)`, om de variantie tussen de verkoopwaarde van de huidige rij en de waarde van de 3e rij van boven.
- `=(Sales - Top(Sales)) / Top(Sales)`, om de verandering in `Sales` in vergelijking tot de eerste rij als een verhouding / percentage weer te geven. Om deze waarde als percentage weer te geven, selecteert u onder **GetalnotatieGetal > Eenvoudige opmaak > 12,34%**.

Sorteer de tabel in de dimensie `Year` in oplopende volgorde.

Resultatentabel

| 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% |

Als u de resultaten bekijkt, kunt u de `Sales` waarden in de huidige periode vergelijken met de bovenste (of eerste) periode.

- De meting `Sum(Sales) - Top(Sum(Sales))` retourneert de wijziging in de `Sales` waarden, waarbij de huidige periode wordt vergeleken met de eerste periode.
- De meting `Sum(Sales)-Top(Sum(Sales), 3)` retourneert de som van `Sales` voor de huidige rij minus de derde periode van boven.
- De meting `=(Sales - Top(Sales)) / Top(Sales)` retourneert de wijziging in `Sales` ten opzichte van de eerste periode als een percentage.

*Deze functie kan behalve in tabellen ook worden gebruikt in diagrammen, bijvoorbeeld in staafdiagrammen.*

**Doe het volgende:**

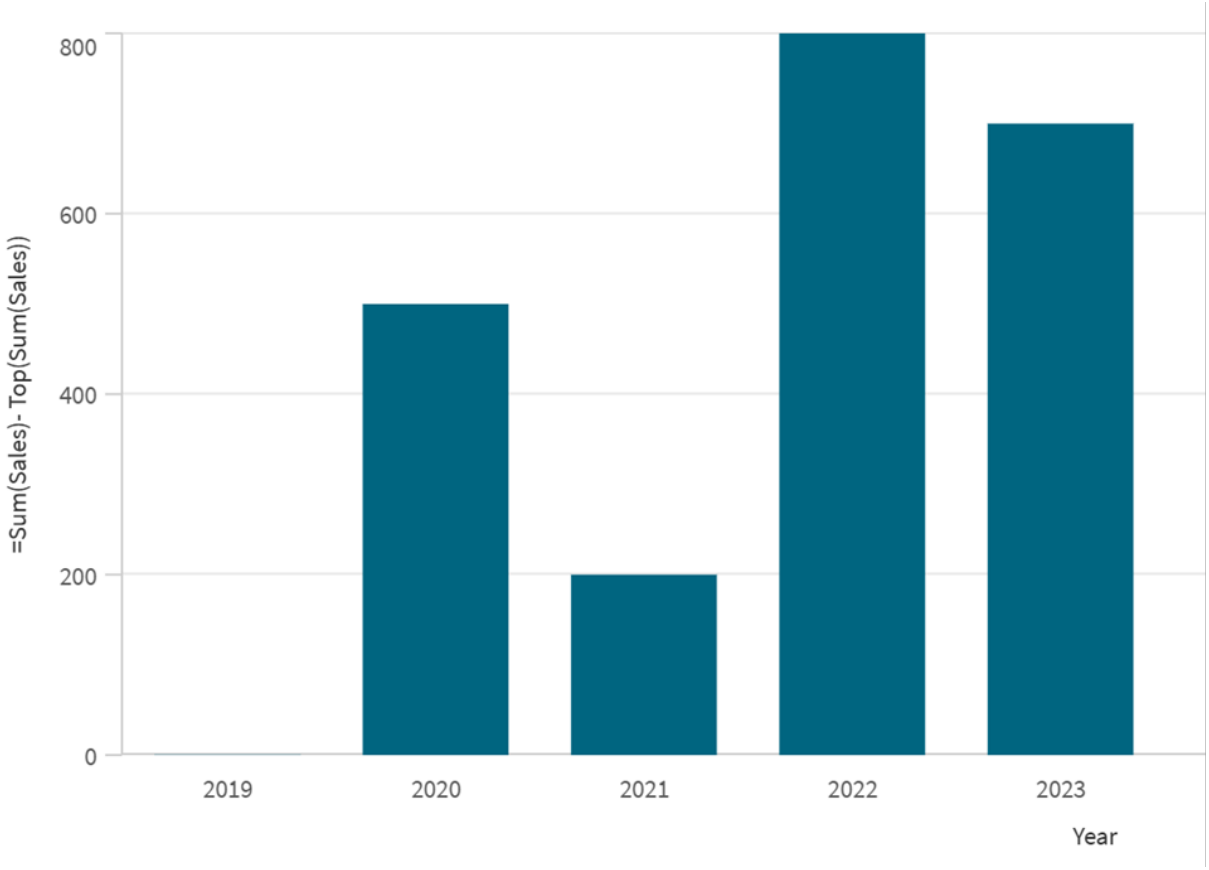
Maak een staafdiagram en voeg dit veld toe als dimensie:

- `Year`

Maak de volgende meting:

- `=Sum(Sales)- Top(Sum(Sales))`, om de variantie in verkoop tussen de huidige periode en de bovenste periode weer te geven.

*Staafdiagram met resultaten voor de functie Top*




Door de functie `Top` in een staafdiagram te gebruiken, krijgt u een visuele vergelijking van de variantie tussen de verkopen van de huidige periode en die van de bovenste (of eerste) periode.

## Voorbeeld 5 – Scenario met Top

### Diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:

- OrderID
- Product
- SalesValue
- OrderDate

### Load-script

```
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'
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- Product

Maak de volgende metingen:

- `=RangeAvg(Top(Sum(SalesValue),1,3))` om het gemiddelde van de drie bovenste rijen te retourneren.
- `=Sum(SalesValue) - RangeAvg (Top(Sum(SalesValue),1,3))`, om de variantie tussen de huidige rij en het gemiddelde van de 3 bovenste rijen te berekenen.
- `=Sum(SalesValue)` om de totale `SalesValue` voor de huidige rij te berekenen.

Resultatentabel

| Product | RangeAvg(Top(Sum (SalesValue),1,3)) | Sum(SalesValue) - RangeAvg (Top (Sum(SalesValue),1,3)) | Sum (SalesValue) |
|---|---|---|---|
| **Totals** | **416.67** | **1883.33** | **2300** |
| 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 |

De resultaten laten zien hoe u een combinatie van `RangeSum` en de functie `Top` kunt gebruiken om verkoopwaarden te analyseren en vergelijken.

## Voorbeeld - Diagramuitdrukking met gebruik van Top

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Twee gegevenstabellen genaamd `Monthnames` en `Sales2013`.

### Load-script

```
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 '|');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Customer`

Maak de volgende metingen:

- `=Sum(Sales)`, om de som van verkoop te berekenen.
- `=Top(Sum(Sales))`, om de som van de verkopen van de `Customer` in de bovenste rij te tonen.

- `=Sum(Sales)+Top(Sum(Sales))`, om de `Sum(Sales)` waarde van de huidige `Customer` toe te voegen aan de `Customer` in de bovenste rij.
- `=Sum(Sales)+Top(Sum(Sales), 3)`, om de `Sum(Sales)` waarde toe te voegen van de huidige `Customer` met de waarde voor de `Customer` die derde is van de bovenste rij.

Resultatentabel

| 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 |
| Canutilty | 683 | 587 | 1270 | 1366 |
| Divadip | 757 | 587 | 1344 | 1440 |

De meting `Top(Sum(Sales))` retourneert `587` voor alle rijen omdat dit de waarde van de bovenste rij is: `Astrida`.

Voor de meting `Sum(Sales)+Top(Sum(Sales))` retourneert de rij `Betacab 1126`, dit het resultaat van het optellen van de `Sum(Sales)` waarde voor `Betacab` en `Astrida`, de `Customer` in de bovenste rij.

Voor de meting `Sum(Sales)+Top(Sum(Sales), 3)`, is het argument `offset` ingesteld op `3`. De uitdrukking voegt de `Sum(Sales)` waarde voor de huidige rij en de derde rij van boven toe, `Canutility`. De waarde `1440` wordt bijvoorbeeld geretourneerd voor de klant `Divadip`. Dit is het resultaat van het optellen van de `Sum (Sales)` waarde voor `Divadip` en de `SumSales` voor `Canutility` (`757+683`).

**Zie ook:**

p *Bottom - diagramfunctie (page 1503)*
p *Above - diagramfunctie (page 1479)*
p *Sum - diagramfunctie (page 394)*
p *RangeAvg (page 1603)*
p *Bereikfuncties (page 1600)*

## SecondaryDimensionality - diagramfunctie

**SecondaryDimensionality()** retourneert het aantal rijen in een draaitabel met een niet-geaggregeerde inhoud, dat wil zeggen kolommen die geen subtotalen of samengevouwen aggregaten bevatten. Deze functie is het equivalent van de functie **dimensionality()** voor horizontale dimensies in draaitabellen.

**Syntaxis:**

```
SecondaryDimensionality( )
```

**Retourgegevenstypen:** geheel getal

**Beperkingen:**

- Tenzij deze wordt gebruikt in draaitabellen, retourneert de functie **SecondaryDimensionality** altijd de waarde 0.
- Het is niet toegestaan te sorteren op y-waarden in grafieken of op uitdrukkingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van de grafiek. Deze sorteeropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie gebruikt in een visualisatie of tabel, wordt de sortering van de visualisatie teruggezet naar de gesorteerde invoer van deze functie.

## Voorbeeld - Basisprincipes voor SecondaryDimensionality

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Date`
  - `Product`
  - `Category`
  - `Sales`

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe draaitabel en voeg dit veld toe als een rijdimensie:

- `Date`

Voeg deze velden toe als kolomdimensies:

- `Category`
- `Product`

Maak de volgende metingen:

- `=Sum(Sales)`, om de totale verkoop te berekenen.
- `=SecondaryDimensionality()` om het dimensiegetal te retourneren om het gebruik van deze functie te demonstreren.

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 |

De resultaten van de draaitabel tonen alleen de dimensie `Category`, weergegeven als een kolom. Om die reden retourneert de `SecondaryDimensionality` functie een waarde van 1.

Als u de dimensie `Category` uitbreidt, `Fruit`, zullen de resultaten van de tabel nu ook de waarden van de dimensie `Product`, `Apple` en `Banana` bevatten.

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 |

De functie `SecondaryDimensionality` retourneert nu een waarde van `2`, omdat er waarden worden weergegeven voor twee dimensies in de draaitabel: `Category` en `Product`.

## Voorbeeld - SecondaryDimensionality gebruiken om voorwaardelijke opmaak toe te passen in een diagramuitdrukking

Diagramuitdrukking

### Overzicht

Dit voorbeeld gebruikt de functie in een diagramuitdrukking in een draaitabel om voorwaardelijke celopmaak toe te passen, afhankelijk van welk dimensieniveau wordt weergegeven.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Year`
  - `Quarter`
  - `Product Category`
  - `Sales`

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe draaitabel en voeg dit veld toe als een rijdimensie:

- `Product Category`

Voeg deze velden toe als kolomdimensies:

- `Year`
- `Quarter`

Maak de volgende meting:

- `=Sum(Sales)`, om de totale verkoop te berekenen.

Voer in het deelvenster **Eigenschappen** de volgende uitdrukking in als de **Uitdrukking achtergrondkleur** voor de `Sum(Sales)`-meting:

```
IF(SecondaryDimensionality() = 1,
RGB(144, 238, 144),  // Green for Year-level
RGB(173, 216, 230),  // Blue for Quarter-level
)
```

De uitdrukking zal een groene achtergrond toepassen op de tabelcellen wanneer de functie `SecondaryDimensionality` 1 retourneert en een blauwe achtergrond wanneer de functie een ander resultaat retourneert.

Draaitabel met resultaten voor één dimensie: **Year** met groene celachtergrond

| Productcategorie | (+) 2023 |
|---|---|
| Electronics | 22000 |
| Furniture | 16000 |

De resultaten laten zien dat wanneer alleen de dimensie `Year` wordt weergegeven, dus de waarde `SecondaryDimensionality` 1 is, de cellen `Sum(Sales)` met een groene achtergrond verschijnen. Wanneer u de tabel uitbreidt om de `Sum(Sales)` per `Quarter` weer te geven, wordt de achtergrond nu blauw omdat er twee dimensies worden weergegeven, `Year` en `Quarter`, en de waarde `SecondaryDimensionality` is nu `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 - diagramfunctie

retourneert de waarde van een uitdrukking berekend aan de hand van de dimensiewaarden van een draaitabel die staan in de kolom voor de huidige kolom in een rijsegment in de draaitabel.

**Syntaxis:**

```
after([TOTAL] expr [, offset [, count ]])
```

> *Het is niet toegestaan te sorteren op y-waarden in grafieken of op uitdrukkingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van de grafiek. Deze sorteeropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie gebruikt in een visualisatie of tabel, wordt de sortering van de visualisatie teruggezet naar de gesorteerde invoer van deze functie.*

*Deze functie retourneert NULL in alle andere diagramtypen dan een draaitabel.*

Argumenten

| Argument | Beschrijving |
|---|---|
| **expr** | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| **offset** | Als een **offset** n groter dan 1 wordt opgegeven, wordt de evaluatie van de uitdrukking n kolommen verder naar rechts verplaatst, geteld vanaf de huidige kolom.<br><br>Als de beginwaarde 0 wordt opgegeven, wordt de uitdrukking geëvalueerd op de huidige kolom.<br><br>Als een negatief getal voor de beginwaarde wordt opgegeven, werkt de functie **After** op dezelfde wijze als de functie **Before** met het overeenkomstige positieve getal als beginwaarde. |
| **count** | Als een derde parameter **count** groter dan 1 wordt opgegeven, retourneert de functie een bereik van waarden. Eén waarde voor elke tabelkolom, met een maximum van **count**, naar rechts geteld vanaf de oorspronkelijke cel. |
| **TOTAL** | Als de tabel eendimensionaal is of als de kwalificatie **TOTAL** wordt gebruikt als argument, is het huidige kolomsegment altijd gelijk aan de hele kolom. |

Op de laatste kolom van een rijsegment wordt een NULL-waarde geretourneerd, omdat er geen kolom na deze kolom staat.

Als de draaitabel meerdere horizontale dimensies heeft, omvat het huidige rijsegment alleen kolommen met dezelfde waarden als de huidige kolom in alle dimensierijen, met uitzondering van de rij waarin de laatste horizontale dimensie wordt weergegeven in de onderlinge sorteervolgorde van de velden. De sorteervolgorde tussen velden voor horizontale dimensies in draaitabellen wordt simpelweg gedefinieerd door de volgorde van de dimensies van boven naar onderzijn.

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `After( Sum (Sales ))` | Retourneert de som van `Sales` van de volgende kolom direct rechts ten opzichte van de huidige kolom. |
| `After( Sum (Sales ), 2)` | Retourneert de som van `Sales` van de kolom die twee kolommen rechts van de huidige kolom staat. |
| `After( Total Sum( Sales ))` | Retourneert de totale som van `Sales` van de volgende kolom direct rechts ten opzichte van de huidige kolom. |
| `RangeAvg (After(Sum (x),1,3))` | Retourneert een gemiddelde van de drie resultaten van de functie `sum(x)`, berekend aan de hand van de drie kolommen direct rechts van de huidige kolom. |

## Voorbeeld - Basisprincipes voor After

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Year`
  - `Quarter`
  - `Sales`

### Load-script

```
Example:
Load * inline [
Year, Quarter, Sales
2023, Q1, 5000
2023, Q2, 4000
2023, Q3, 6000
2023, Q4, 7000
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe draaitabel en voeg dit veld toe als een rijdimensie:

- `Year`

Voeg dit veld toe als een kolomdimensie:

- `Quarter`

Maak de volgende metingen:

- `=Sum(Sales)`, om de som van verkoop te berekenen.
- `=After(Sum(Sales))`, om de som van de verkopen in het volgende kwartaal te tonen.

Resultatentabel

| Jaar | Q1 | | Q2 | | Q3 | | Q4 | |
|---|---|---|---|---|---|---|---|---|
| | Sum (Sales) | Na(Som (Verkoop)) | Sum (Sales) | Na(Som (Verkoop)) | Sum (Sales) | Na(Som (Verkoop)) | Sum (Sales) | Na(Som (Verkoop)) |
| 2023 | 5000 | 4000 | 4000 | 6000 | 6000 | 7000 | 7000 | - |

Als u naar de resultaten kijkt, kunt u zien hoe de functie `After` voor elke `Quarter` de waarde van de volgende kolom retourneert die direct rechts van de kolom `Sum(Sales)` staat. Bijvoorbeeld, in `Q1` is de waarde `Sum(Sales)` voor die periode `5000`. De uitvoer van de functie `After` voor `Q1` is `4000` omdat dit de waarde is van de kolom direct rechts van de `Sum(Sales)` voor `Q1`, namelijk de `Sum(Sales)` voor `Q2`.

## Voorbeeld 5 – Scenario met After

Diagramuitdrukking

### Overzicht

Een gegevensverzameling bevat verschillende jaren verkoopcijfers van verkochte eenheden bij een elektronicabedrijf. Het bedrijf wil de totale verkoop voor elk jaar presenteren en de verkoop per eenheid van jaar tot jaar vergelijken.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Year`
  - `Sales`

### Load-script

```
Example:
Load * inline [
Year, Sales
2021, 1350
2022, 1840
2023, 2250
2024, 1750
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe draaitabel en voeg dit veld toe als een kolomdimensie:

- `Year`

Maak de volgende metingen:

- `=Sum(Sales)`, om de som van verkoop te berekenen.
- `=Sum(Sales) - After(Sum(Sales))`, om het verschil in verkoop van jaar tot jaar te berekenen.

Resultatentabel

| Jaar | 2021 | | 2022 | | 2023 | | 2024 | |
|---|---|---|---|---|---|---|---|---|
| | Sum (Sales) | Som (Verkoop)-Na (Som (Verkoop)) | Sum (Sales) | Som (Verkoop)-Na (Som (Verkoop)) | Sum (Sales) | Sum (Verkoop)-After (Sum (Verkoop)) | Sum (Sales) | Sum (Verkoop)-After (Sum (Verkoop)) |
| 2023 | 1350 | -490 | 1840 | -410 | 2250 | 500 | 1750 | - |

De resultaten tonen de som van de verkoop voor elk jaar en het verschil tussen de verkoopcijfers van het ene jaar ten opzichte van het andere. Bijvoorbeeld, in `2022` bedroeg de `Sum(Sales)` `1840` . De uitvoer van de functie `After` geeft aan dat deze waarde `410` eenheden minder was dan de som van de verkopen voor `2023`.

## Voorbeeld - Geavanceerd scenario met After

Diagramuitdrukking

### Overzicht

Een gegevensverzameling bevat verkoopcijfers van verschillende jaren. Het bedrijf wil de totale omzet voor het huidige jaar en een voortschrijdend totaal van de omzet voor de komende 3 jaar presenteren.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `SalesYear`
  - `Sales`

### Load-script

```
Example:
Load * inline [
SalesYear, Sales
2021, 1300
2022, 1800
2023, 2000
2024, 2150
];
```

**Resultaten**

Laad de gegevens en open een werkblad. Maak een nieuwe draaitabel en voeg dit veld toe als een kolomdimensie:

- `SalesYear`

Maak de volgende metingen:

- `=Sum(Sales)`, om de som van verkoop te berekenen.
- `=RangeSum(After(Sum(Sales),1,3))`, om de som van verkoop voor de komende drie jaar te berekenen.

Resultatentabel

| 2021 | | 2022 | | 2023 | | 2024 | |
|---|---|---|---|---|---|---|---|
| Sum (Sales) | RangeSum (After(Sum (Verkoop),1, 3)) | Sum (Sales) | RangeSum (After(Sum (Omzet),1, 3)) | Sum (Sales) | RangeSum (After(Sum (Omzet),1, 3)) | Sum (Sales) | RangeSum (After(Sum (Omzet),1, 3)) |
| 1300 | 5950 | 1800 | 4150 | 2000 | 2150 | 2150 | 0 |

De resultaten laten zien hoe u de functie `After` kunt gebruiken om de som van de verkoopwaarden voor de volgende drie jaar te retourneren. Bijvoorbeeld, in `2021` is de jaarlijkse `Sum(Sales)` `1300` en de som van de verkopen voor de komende 3 jaar (`2022, 2023, 2024`) is `5950`. Er wordt een waarde van `0` geretourneerd voor `2024` omdat de gegevens voor toekomstige jaren niet beschikbaar zijn.

## Before - diagramfunctie

**Before()** retourneert de waarde van een uitdrukking berekend aan de hand van de dimensiewaarden van een draaitabel die staan in de kolom voor de huidige kolom in een rijsegment in de draaitabel.

**Syntaxis:**

```
Before([TOTAL] expr [, offset [, count]])
```

*Deze functie retourneert NULL in alle andere diagramtypen dan een draaitabel.*

*Het is niet toegestaan te sorteren op y-waarden in grafieken of op uitdrukkingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van de grafiek. Deze sorteeropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie gebruikt in een visualisatie of tabel, wordt de sortering van de visualisatie teruggezet naar de gesorteerde invoer van deze functie.*

Argumenten

| Argument | Beschrijving |
|---|---|
| **expr** | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| **offset** | Als een **offset** n groter dan 1 wordt opgegeven, wordt de evaluatie van de uitdrukking n kolommen verder naar links verplaatst, geteld vanaf de huidige kolom.<br><br>Als de beginwaarde 0 wordt opgegeven, wordt de uitdrukking geëvalueerd op de huidige kolom.<br><br>Als een negatief getal voor de beginwaarde wordt opgegeven, werkt de functie **Before** op dezelfde wijze als de functie **After** met het overeenkomstige positieve getal als beginwaarde. |
| **count** | Als een derde parameter **count** groter dan 1 wordt opgegeven, retourneert de functie een bereik van waarden. Eén waarde voor elke tabelkolom, met een maximum van **count**, naar links geteld vanaf de oorspronkelijke cel. |
| **TOTAL** | Als de tabel eendimensionaal is of als de kwalificatie **TOTAL** wordt gebruikt als argument, is het huidige kolomsegment altijd gelijk aan de hele kolom. |

Op de eerste kolom van een rijsegment wordt een NULL-waarde geretourneerd, daar er geen kolom voor deze kolom staat.

Als de draaitabel meerdere horizontale dimensies heeft, omvat het huidige rijsegment alleen kolommen met dezelfde waarden als de huidige kolom in alle dimensierijen, met uitzondering van de rij waarin de laatste horizontale dimensie wordt weergegeven in de onderlinge sorteervolgorde van de velden. De sorteervolgorde tussen velden voor horizontale dimensies in draaitabellen wordt simpelweg gedefinieerd door de volgorde van de dimensies van boven naar onderzijn.

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `Before( Sum (Sales ))` | Retourneert de som van `Sales` van de vorige kolom direct links ten opzichte van de huidige kolom. |
| `Before( Sum (Sales ), 2)` | Retourneert de som van `Sales` van de kolom die twee kolommen links van de huidige kolom staat. |
| `Before( Total Sum( Sales ))` | Retourneert de totale som van `Sales` van de vorige kolom direct links ten opzichte van de huidige kolom. |
| `RangeAvg ( Before(Sum (x),1,3))` | Rretourneert een gemiddelde van de drie resultaten van de functie `sum(x)`, berekend aan de hand van de drie kolommen direct links van de huidige kolom. |

## Voorbeeld - Basisprincipes voor Before

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Year`
  - `Quarter`
  - `Sales`

### Load-script

```
Example:
Load * inline [
Year, Quarter, Sales
2023, Q1, 5000
2023, Q2, 4000
2023, Q3, 6000
2023, Q4, 7000
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe draaitabel en voeg dit veld toe als een rijdimensie:

- `Year`

Voeg dit veld toe als een kolomdimensie:

- `Quarter`

Maak de volgende metingen:

- `=Sum(Sales)`, om de som van verkoop te berekenen.
- `=Before(Sum(Sales))`, om de som van de verkopen in het vorige kwartaal te tonen.

Resultatentabel

| Jaar | Q1 | | Q2 | | Q3 | | Q4 | |
|---|---|---|---|---|---|---|---|---|
| | Sum (Sales) | Before (Sum (Sales)) | Sum (Sales) | Before (Sum (Sales)) | Sum (Sales) | Before (Sum (Sales)) | Sum (Sales) | Before (Sum (Sales)) |
| 2023 | 5000 | - | 4000 | 5000 | 6000 | 4000 | 7000 | 6000 |

Als u naar de resultaten kijkt, kunt u zien hoe de functie `Before` voor elke `Quarter` de waarde van de vorige kolom retourneert die direct links van de kolom `Sum(Sales)` staat. In Q4 is de `Sum(Sales)` waarde voor die periode `7000`. De uitvoer van de functie `Before` voor `Q4` is `6000` omdat dit de waarde is van de kolom direct links van de `Sum(Sales)` voor `Q4`, namelijk de `Sum(Sales)` voor `Q3`.

## Voorbeeld - De functie Before gebruiken om de omzetgroei van jaar tot jaar te berekenen

Diagramuitdrukking

### Overzicht

Een gegevensverzameling bevat verkoopcijfers van verschillende jaren. Het bedrijf wil de totale verkoop voor elk jaar en de groei in verkoop op basis van de waarde van het voorgaande jaar presenteren.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Year`
  - `Sales`

### Load-script

```
Example:
Load * inline [
Year, Sales
2021, 1300
2022, 1800
2023, 2000
2024, 2100
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe draaitabel en voeg dit veld toe als een kolomdimensie:

- `Year`

Maak de volgende metingen:

- `=Sum(Sales)`, om de som van verkoop te berekenen.
- `=(Sum(Sales) - Before(Sum(Sales))) / Before(Sum(Sales))`, om de jaarlijkse omzetgroei te berekenen. Om deze waarde als percentage weer te geven, selecteert u onder **GetalnotatieGetal > Eenvoudige opmaak > 12,34%**.

Resultatentabel

| 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% |

De uitvoer van de functie `Before` geeft de jaarlijkse groei op basis van de verkoopcijfers van het vorige jaar. Bijvoorbeeld, de som van de verkopen voor `2022` was `38.46%` hoger dan de `Sum(Sales)` voor `2021`.

*U kunt ook de functie `After` gebruiken om dezelfde berekening van de omzetgroei in dit scenario uit te voeren. Als u een negatief getal voor de beginwaarde opgeeft in de functieparameters, werkt de functie `After` op dezelfde wijze als de functie `Before` met het overeenkomstige positieve getal als beginwaarde. De volgende uitdrukking met de functie `After` (met een negatieve offsetwaarde van `-1`) retourneert bijvoorbeeld dezelfde resultaten voor de jaarlijkse omzetgroei als de `Before` functie:*
*`=(Sum(Sales)-After(Sum(Sales),-1))/(After(Sum(Sales),-1))`.*

## First - diagramfunctie

**First()** retourneert de waarde van een uitdrukking berekend aan de hand van de dimensiewaarden van een draaitabel die staan in de eerste kolom van het huidige rijsegment in de draaitabel. Deze functie retourneert NULL in alle andere diagramtypen dan een draaitabel.

*Het is niet toegestaan te sorteren op y-waarden in grafieken of op uitdrukkingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van de grafiek. Deze sorteeropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie gebruikt in een visualisatie of tabel, wordt de sortering van de visualisatie teruggezet naar de gesorteerde invoer van deze functie.*

**Syntaxis:**

```
First([TOTAL] expr [, offset [, count]])
```

Argumenten

| Argument | Beschrijving |
|---|---|
| **expression** | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |

| Argument | Beschrijving |
|---|---|
| **offset** | Als een **offset** n groter dan 1 wordt opgegeven, wordt de evaluatie van de uitdrukking n kolommen verder naar rechts verplaatst, geteld vanaf de eerste kolom.<br>Als de beginwaarde 0 wordt opgegeven, wordt de uitdrukking geëvalueerd op de eerste kolom van het huidige rijsegment.<br><br>Als de beginwaarde 0 wordt opgegeven, wordt de uitdrukking geëvalueerd op de huidige rij.<br><br>Als een negatief getal voor de beginwaarde wordt opgegeven, werkt de functie **First** op dezelfde wijze als de functie **Last** met het overeenkomstige positieve getal als beginwaarde. |
| **count** | Als een derde parameter **count** groter dan 1 wordt opgegeven, retourneert de functie een bereik van waarden. Eén waarde voor elke tabelkolom, met een maximum van **count**, naar rechts geteld vanaf de oorspronkelijke cel. |
| **TOTAL** | Als de tabel eendimensionaal is of als de kwalificatie **TOTAL** wordt gebruikt als argument, is het huidige kolomsegment altijd gelijk aan de hele kolom. |

Als de draaitabel meerdere horizontale dimensies heeft, omvat het huidige rijsegment alleen kolommen met dezelfde waarden als de huidige kolom in alle dimensierijen, met uitzondering van de rij waarin de laatste horizontale dimensie wordt weergegeven in de onderlinge sorteervolgorde van de velden. De sorteervolgorde tussen velden voor horizontale dimensies in draaitabellen wordt simpelweg gedefinieerd door de volgorde van de dimensies van boven naar onderzijn.

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `First( Sum (Sales ))` | Retourneert de som van `Sales` van de eerste kolom van het huidige rijsegment. |
| `First( Sum (Sales ), 2)` | Retourneert de som van `Sales` die twee kolommen rechts van de huidige kolom in het huidige rijsegment staat. |
| `First( Total Sum( Sales ))` | Retourneert de totale som van `Sales` van de eerste kolom van het huidige rijsegment. |
| `RangeAvg (First(Sum (x),1,5))` | Retourneert een gemiddelde van de resultaten van de functie `sum(x)`, berekend aan de hand van de vijf meest linkse kolommen van het huidige rijsegment. |

## Voorbeeld - Basisprincipes voor First

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Year`
  - `Quarter`
  - `Sales`

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe draaitabel en voeg dit veld toe als een rijdimensie:

- `Year`

Voeg dit veld toe als een kolomdimensie:

- `Quarter`

Maak de volgende metingen:

- `=Sum(Sales)`, om de som van verkoop te berekenen.
- `=First(Sum(Sales))`, om de som van verkopen van de eerste kolom voor het huidige rijsegment te berekenen.

Resultatentabel

| Jaar | Q1 | | Q2 | | Q3 | | Q4 | |
|---|---|---|---|---|---|---|---|---|
| | Sum (Sales) | First (Sum (Sales)) | Sum (Sales) | First (Sum (Sales)) | Sum (Sales) | First (Sum (Sales)) | Sum (Sales) | First (Sum (Sales)) |
| 2023 | 5000 | 5000 | 4000 | 5000 | 6000 | 5000 | 7000 | 5000 |
| 2024 | 4500 | 4500 | 3250 | 4500 | 6500 | 4500 | 7250 | 4500 |

Als u naar de resultaten kijkt, kunt u zien hoe de functie `First` de som van de verkoopwaarden voor de eerste kolom van elk rijsegment voor elk kwartaal retourneert. Bijvoorbeeld, in `Q4` is de waarde `Sum(Sales)` voor die periode `7000`. De `First(Sum(Sales)` waarde voor `Q4` is `5000`, wat de waarde is voor `Sum(Sales)` in `Q1`, de eerste kolomwaarde van de huidige rij, `2023`.

## Voorbeeld 5 – Scenario met First

Diagramuitdrukking

### Overzicht

Een gegevensverzameling bevat verkoopcijfers per jaar voor verschillende producten over een periode van drie jaar. Het bedrijf wil de totale omzet voor het huidige jaar en de omzetgroei sinds het eerste jaar in de periode presenteren.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
    - `Date`
    - `Product`
    - `Sales`

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe draaitabel en voeg dit veld toe als een rijdimensie:

- `Product`

Voeg dit veld toe als een kolomdimensie:

- `Date`

Maak de volgende metingen:

- `=Sum(Sales)`, om de jaaromzet te berekenen.
- `=(Sum(Sales) - First(Sales)) / First(Sales)`, om de groei in de verkoop sinds het eerste jaar in de gegevensverzameling te berekenen. Om deze waarde als percentage weer te geven, selecteert u onder **GetalnotatieGetal > Eenvoudige opmaak > 12,34%**.

Resultatentabel

| Product | 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) |
| Appel | 100 | 0.00% | 110 | 10.00% | 150 | 50.00% |
| Banaan | 150 | 0.00% | 200 | 33.33% | 300 | 100.00% |
| Wortel | 120 | 0.00% | 80 | -33.33% | 160 | 33.33% |

De resultaten tonen de jaarlijkse verkoopcijfers voor elk jaar en het groeipercentage per jaar vergeleken met de verkoopwaarden van het eerste jaar. Bijvoorbeeld, de verkoop van `Apple` steeg van een initiële waarde van `100` in `2021` naar een waarde van `150` in `2023`, wat een stijging is van `50%`. De verkoop van `Carrot` daalde van `120` in `2021` naar `80` in `2022`, een daling van `33.33%`.

## Voorbeeld - Scenario met First met een reekssom om budgetcijfers te presenteren

Diagramuitdrukking

### Overzicht

Een gegevensverzameling bevat budgetcijfers per kwartaal voor een periode van twee jaar per productcategorie. Het bedrijf wil het budget voor het huidige kwartaal en het lopende totaal van het budget voor elk jaar presenteren.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `BudgetYear`
  - `BudgetQuarter`
  - `QuarterNo`
  - `BudgetAmount`
  - `Category`

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe draaotabel en voeg deze velden toe als dimensies:

- `BudgetYear`
- `Category`

Voeg dit veld toe als een kolomdimensie:

- `BudgetQuarter`

Maak de volgende metingen:

- `=Sum(BudgetAmount)`, om het verkoopbudget te berekenen.
- `=RangeSum(First(Sum(BudgetAmount), 1, QuarterNo))`, om het lopende totaal van het verkoopbudget voor elk kwartaal te berekenen door het bereik van het eerste kwartaal op te tellen tot `QuarterNo`, zoals geëvalueerd voor elk kwartaal.

Resultatentabel

| BudgetYear Category | Q1 | | Q2 | | Q3 | | Q4 | |
|---|---|---|---|---|---|---|---|---|
| | Sum (Budget Amount) | RangeSum(First (Sum (Budget Amount), 1, Quarter No)) | Sum (Budget Amount) | RangeSum(First (Sum (Budget Amount), 1, Quarter No)) | Sum (Budget Amount) | RangeSum(First (Sum (Budget Amount), 1, Quarter No)) | Sum (Budget Amount) | RangeSum(First (Sum (Budget Amount), 1, Quarter No)) |
| (-) 2023 | 7200 | 7200 | 8000 | 15200 | 9200 | 24400 | 11200 | 35600 |
| Online | 2100 | 2100 | 2500 | 4600 | 3100 | 7700 | 4100 | 11800 |
| Winkel | 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 |
| Winkel | 5300 | 5300 | 5800 | 11100 | 6300 | 17400 | 7400 | 24800 |

De resultaten tonen de budgetcijfers voor elk `Year` en elke `Category`, en het lopende totaal per `Quarter`. Bijvoorbeeld, in `Q2` van `2023`, is `QuarterNo 2` . Daarom wordt de functie-uitdrukking geëvalueerd als `RangeSum(First(Sum(BudgetAmount), 1, 2))`, die de kolommen `BudgetAmount` voor `Q1` (`7200`) en `Q2` (`8000`) bij elkaar optelt en de waarde `15200` geretourneerd. Het resultaat geeft het totaal weer van de kolommen `Sum(BudgetAmount)` voor `Q1` en `Q2`. In `2024`, voor de `Category Online`, is de `Sum(BudgetAmount)` voor `Q1` `2300` en voor `Q2` is dit `2800`. Daarom retourneert de `RangeSum` uitdrukking `5100`.

## Last - diagramfunctie

**Last()** retourneert de waarde van een uitdrukking berekend aan de hand van de dimensiewaarden van een draaitabel die staan in de laatste kolom van het huidige rijsegment in de draaitabel. Deze functie retourneert NULL in alle andere diagramtypen dan een draaitabel.

> *Het is niet toegestaan te sorteren op y-waarden in grafieken of op uitdrukkingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van de grafiek. Deze sorteeropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie gebruikt in een visualisatie of tabel, wordt de sortering van de visualisatie teruggezet naar de gesorteerde invoer van deze functie.*

**Syntaxis:**

```
Last([TOTAL] expr [, offset [, count]])
```

Argumenten

| Argument | Beschrijving |
|---|---|
| **expr** | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| **offset** | Als een **offset** n groter dan 1 wordt opgegeven, wordt de evaluatie van de uitdrukking n kolommen naar links verplaatst, geteld vanaf de laatste kolom.<br>Als de beginwaarde 0 wordt opgegeven, wordt de uitdrukking geëvalueerd op de laatste kolom van het huidige rijsegment.<br><br>Als een negatief getal voor de beginwaarde wordt opgegeven, werkt de functie **Last** op dezelfde wijze als de functie **First** met het overeenkomstige positieve getal als beginwaarde. |
| **count** | Als een derde parameter **count** groter dan 1 wordt opgegeven, retourneert de functie een bereik van waarden. Eén waarde voor elke tabelkolom, met een maximum van **count**, naar links geteld vanaf de oorspronkelijke cel. |
| **TOTAL** | Als de tabel eendimensionaal is of als de kwalificatie **TOTAL** wordt gebruikt als argument, is het huidige kolomsegment altijd gelijk aan de hele kolom. |

Als de draaitabel meerdere horizontale dimensies heeft, omvat het huidige rijsegment alleen kolommen met dezelfde waarden als de huidige kolom in alle dimensierijen, met uitzondering van de rij waarin de laatste horizontale dimensie wordt weergegeven in de onderlinge sorteervolgorde van de velden. De sorteervolgorde tussen velden voor horizontale dimensies in draaitabellen wordt simpelweg gedefinieerd door de volgorde van de dimensies van boven naar onderzijn.

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `Last( Sum (Sales ))` | Retourneert de som van `Sales` van de laatste kolom van het huidige rijsegment. |
| `Last( Sum (Sales ), 2)` | Retourneert de som van `Sales` van de kolom die twee kolommen links van de laatste kolom in het huidige rijsegment staat. |
| `Last( Total Sum( Sales ))` | Retourneert de totale som van `Sales` van de laatste kolom van het huidige rijsegment. |
| `RangeAvg (Last(Sum (x),1,5))` | Retourneert een gemiddelde van de resultaten van de functie `sum(x)`, berekend aan de hand van de vijf meest rechtse kolommen van het huidige rijsegment. |

## Voorbeeld - Basisprincipes voor Last

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Year`
  - `Quarter`
  - `Sales`

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe draaitabel en voeg dit veld toe als een rijdimensie:

- `Year`

Voeg dit veld toe als een kolomdimensie:

- `Quarter`

Maak de volgende metingen:

- `=Sum(Sales)`, om de som van verkoop te berekenen.
- `=Last(Sum(Sales))`, om de som van verkopen van de laatste kolom voor het huidige rijsegment te berekenen.

Resultatentabel

| Jaar | Q1 | | Q2 | | Q3 | | Q4 | |
|---|---|---|---|---|---|---|---|---|
| | Sum (Sales) | Last (Sum (Sales)) | Sum (Sales) | Last (Sum (Sales)) | Sum (Sales) | Last (Sum (Sales)) | Sum (Sales) | Last (Sum (Sales)) |
| 2023 | 5000 | 7000 | 4000 | 7000 | 6000 | 7000 | 7000 | 7000 |
| 2024 | 4500 | 7250 | 3250 | 7250 | 6500 | 7250 | 7250 | 7250 |

Als u naar de resultaten kijkt, kunt u zien dat de functie `Last` de som van de verkoopwaarden voor de laatste kolom van elk rijsegment voor elk kwartaal retourneert. Bijvoorbeeld, in `Q1` is de waarde `Sum(Sales)` voor die periode `5000`. De `Last(Sum(Sales)` waarde voor `Q1` is `7000`, wat de waarde is voor `Sum(Sales)` in `Q4`, de laatste kolomwaarde van de huidige rij, `2023`.

## Voorbeeld – Scenario met Last

Diagramuitdrukking

### Overzicht

Een gegevensverzameling bevat verkoopcijfers voor verschillende producten over een periode van drie maanden. Het bedrijf wil de totale omzet voor het huidige jaar en de omzetgroei in vergelijking met de laatste (of recentste) verkoopcijfers presenteren.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
    - `Date`
    - `Product`
    - `Sales`

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe draaitabel en voeg dit veld toe als een rijdimensie:

- `Product`

Voeg dit veld toe als een kolomdimensie:

- `Date`

Maak de volgende metingen:

- `=Sum(Sales)`, om de jaaromzet te berekenen.
- `=(Last(Sum(Sales)) - Sum(Sales)) / Sum(Sales)`, om de omzetgroei van de huidige periode tot de laatste periode te berekenen. Om deze waarde als percentage weer te geven, selecteert u onder **GetalnotatieGetal > Eenvoudige opmaak > 12,34%**.

Resultatentabel

| Product | 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) |
| Appel | 100 | 50.00% | 110 | 36.36% | 150 | 0.00% |
| Banaan | 150 | 100.00% | 200 | 50.00% | 300 | 0.00% |
| Wortel | 120 | 33.33% | 80 | 100.00% | 160 | 0.00% |

In dit voorbeeld kunt u zien hoe de verkoop is verlopen vanaf het begin van de verkoopperiode tot de laatste periode. In de eerste periode `2024-01-01` was de verkoop van `Apple` bijvoorbeeld `100` vergeleken met de laatste verkoopcijfers van `150` voor de periode `2024-03-01`. Dit wijst op een toename van `50%`.

## Voorbeeld - Last gebruiken om budgetvariantie te berekenen

Diagramuitdrukking

### Overzicht

Een gegevensverzameling bevat budgetcijfers per kwartaal voor een periode van twee jaar. Het bedrijf wil het totale budgetbedrag voor elk kwartaal presenteren en de variatie tussen het budgetbedrag voor de huidige periode en het gemiddelde kwartaalbudget voor het jaar.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Year`
  - `Quarter`
  - `Budget`

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe draaitabel en voeg dit veld toe als een rijdimensie:

- `Year`

Voeg dit veld toe als een kolomdimensie:

- `Quarter`

Maak de volgende metingen:

- `=Sum(Budget)`, om het budget voor het kwartaal te berekenen.
- `=Sum(Budget) - RangeAvg(Last(Budget, 1, 4))`, om het verschil te berekenen tussen het `Budget` bedrag voor de huidige periode vergeleken met de gemiddelde kwartaalbegroting voor het jaar.

Resultatentabel

| Jaar | Q1 | | Q2 | | Q3 | | Q4 | |
|---|---|---|---|---|---|---|---|---|
| | Sum (Budget) | Sum (Budget) - RangeAvg(Last (Budget, 1, 4)) | Sum (Budget) | Sum (Budget) - RangeAvg(Last (Budget, 1, 4)) | Sum (Budget) | Sum (Budget) - RangeAvg(Last (Budget, 1, 4)) | Sum (Budget) | Sum (Budget) - RangeAvg(Last (Budget, 1, 4)) |
| 2023 | 1200 | -50 | 1100 | -150 | 1300 | 50 | 1400 | 150 |
| 2024 | 1550 | 105 | 1230 | -215 | 1400 | -45 | 1600 | 155 |

Met de functie `Last` kunt u zien hoe het begrotingsbedrag per kwartaal varieert ten opzichte van de gemiddelde kwartaalbegroting. Bijvoorbeeld, in `2023` is de totale begrotingvoor het jaar `5000` (`1200 + 1100 + 1300 + 1400`), daarom is de gemiddelde kwartaalbegroting `1250`. Aangezien het begrotingsbedrag voor `Q11200` is, is het `50` lager vergeleken met de gemiddelde kwartaalbegroting van `1250`, terwijl de begrotingsvariantie voor `Q4150` hoger is dan de gemiddelde kwartaalbegroting.

## ColumnNo - diagramfunctie

**ColumnNo()** retourneert het nummer van de huidige kolom in het huidige rijsegment in een draaitabel. De eerste kolom is nummer 1.

**Syntaxis:**

```
ColumnNo([total])
```

Argumenten

| Argument | Beschrijving |
|---|---|
| TOTAL | Als de tabel eendimensionaal is of als de kwalificatie **TOTAL** wordt gebruikt als argument, is het huidige kolomsegment altijd gelijk aan de hele kolom. |

Als de draaitabel meerdere horizontale dimensies heeft, omvat het huidige rijsegment alleen kolommen met dezelfde waarden als de huidige kolom in alle dimensierijen, met uitzondering van de rij waarin de laatste horizontale dimensie wordt weergegeven in de onderlinge sorteervolgorde van de velden. De sorteervolgorde tussen velden voor horizontale dimensies in draaitabellen wordt simpelweg gedefinieerd door de volgorde van de dimensies van boven naar onderzijn.

*Het is niet toegestaan te sorteren op y-waarden in grafieken of op uitdrukkingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van de grafiek. Deze sorteeropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie gebruikt in een visualisatie of tabel, wordt de sortering van de visualisatie teruggezet naar de gesorteerde invoer van deze functie.*

Voorbeeld: diagramuitdrukking

| Voorbeeld | Resultaat |
|---|---|
| `IF( ColumnNo ()=1, 0, Sum (Sales) / Before (Sum(Sales)))` | Wanneer de `ColumnNo` gelijk is aan 1, zal het diagram null retourneren. Anders wordt het resultaat van `Sum(Sales) / Before(Sum(Sales)` geretourneerd. In dit voorbeeld retourneert het diagram waarden vanaf kolom 2, terwijl kolom 1 nul retourneert. |

### Voorbeeld - Basisprincipes voor ColumnNo

Diagramuitdrukking

**Overzicht**

Een gegevensverzameling bevat verkoopgegevens per kwartaal. Dit voorbeeld vermenigvuldigt de verkoopwaarde voor elke periode met het kolomnummer van de periode.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
    - `Period`
    - `Sales`

#### Load-script

```
Example:
Load * inline [
Period, Sales
Q1, 1000
Q2, 2000
Q3, 3000
Q4, 4000
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe draaitabel en voeg dit veld toe als een kolomdimensie:

- `Period`

Maak de volgende meting:

- `=Sum(Sales) * ColumnNo()`, om de som van de verkoopwaarden te berekenen, vermenigvuldigd met het kolomnummer, dat in dit voorbeeld het kwartaal is.

Resultatentabel

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| 1000 | 4000 | 9000 | 16000 |

De som van de verkoopwaarde voor elk kwartaal wordt vermenigvuldigd met de uitvoer van de functie `ColumnNo`. De waarde van de verkoop in `Q4` is bijvoorbeeld 4000 en de uitvoer van de functie `ColumnNo` retourneert `4` omdat `Q4` de vierde kolom is. Daarom vermenigvuldigt de metinguitdrukking `4000` met `4` en retourneert `16000`.

## Voorbeeld 5 – Scenario met ColumnNo

#### Diagramuitdrukking

#### Overzicht

Een gegevensverzameling bevat verkoopcijfers van elk kwartaal. Een bedrijf wil de verkoopgegevens voor Q1 en Q3 (de kolommen met oneven nummers) met 5% verhogen en de verkoop voor de overige kolommen (Q2 en Q4) met 10% verhogen.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Period`
  - `Sales`

### Load-script

```
Example:
Load * inline [
Period, Sales
Q1, 1000
Q2, 2000
Q3, 3000
Q4, 4000
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe draaitabel en voeg dit veld toe als een kolomdimensie:

- `Period`

Maak de volgende metingen:

- `=Sum(Sales)`, om de som van verkoop te berekenen.
- `=If(Odd(ColumnNo()), Amount * 1.05, Amount * 1.10)`, om de verkoopbedragen voor de kolommen met oneven nummers met 5% te verhogen en de overige kolommen met 10%.

Resultatentabel

| Q1 | | Q2 | | Q3 | | Q4 | |
|---|---|---|---|---|---|---|---|
| Sum (Sales) | If(Odd (ColumnNo ()), Amount * 1.05, Amount * 1.10) | Sum (Sales) | If(Odd (ColumnNo ()), Amount * 1.05, Amount * 1.10) | Sum (Sales) | If(Odd (ColumnNo ()), Amount * 1.05, Amount * 1.10) | Sum (Sales) | If(Odd (ColumnNo ()), Amount * 1.05, Amount * 1.10) |
| 1000 | 1050 | 2000 | 2200 | 3000 | 3150 | 4000 | 4400 |

De resultaten laten zien dat de verkoopwaarden voor de oneven genummerde kolommen, 1 en 3 (`Q1` en `Q3`), met 5% zijn gestegen, terwijl de overige even genummerde kolommen, 2 en 4 (`Q2` en `Q4`), met 10% zijn gestegen.

## Voorbeeld 5 – Geavanceerd scenario met ColumnNo

Diagramuitdrukking

### Overzicht

Dit voorbeeld gebruikt dezelfde gegevensverzameling als het vorige scenario om het cumulatieve totaal voor de waarden in elke periode weer te geven.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Period`
  - `Sales`

### Load-script

```
Example:
Load * inline [
Period, Sales
Q1, 1000
Q2, 2000
Q3, 3000
Q4, 4000
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe draaitabel en voeg dit veld toe als een kolomdimensie:

- `Period`

Maak de volgende meting:

- `=RangeSum(Before(Sum(Sales), 0, ColumnNo()))`, om het cumulatieve totaal te berekenen.

Resultatentabel

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| 1000 | 3000 | 6000 | 10000 |

Dit voorbeeld toont hoe u de functie `ColumnNo` kunt gebruiken om een lopend totaal te genereren. De volgende uitleg geeft meer details over de metinguitdrukking: `RangeSum(Before(Sum(Sales), 0, ColumnNo()))`.

`Sum(Sales)`: berekent de som van de verkopen voor de huidige cel in de draaitabel.

`Before(Sum(Sales), 0, ColumnNo())`: de functie `Before` haalt waarden op uit de voorgaande kolommen in de draaitabel.

- Het argument `Sum(Sales)` zorgt ervoor dat de functie de getotaliseerde `Sales` voor de voorgaande cellen ophaalt.
- De offsetwaarde van `0` specificeert de offset voor de kolom (die overblijft in het huidige rijsegment van de huidige kolom).
- `ColumnNo()` bepaalt hoeveel kolommen terug moeten gaan bij het ophalen van de gegevens. Het geeft dynamisch de positie van de huidige kolom weer (bijvoorbeeld, voor kolom 3 is `ColumnNo()` gelijk aan `3`, dus de functie kijkt terug over drie kolommen).

`RangeSum()`: de functie `RangeSum` berekent de som van het bereik van de opgegeven waarden. Het behandelt nullwaarden door ze als nul te behandelen. In deze uitdrukking genereert `RangeSum` effectief een cumulatief totaal door alle voorgaande `Sales` waarden die zijn opgehaald door de functie `Before()` bij elkaar op te tellen, evenals de verkoop van de huidige cel.

## NoOfColumns - diagramfunctie

**NoOfColumns()** retourneert het aantal kolommen in het huidige rijsegment in een draaitabel.

> *Het is niet toegestaan te sorteren op y-waarden in grafieken of op uitdrukkingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van de grafiek. Deze sorteeropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie gebruikt in een visualisatie of tabel, wordt de sortering van de visualisatie teruggezet naar de gesorteerde invoer van deze functie.*

**Syntaxis:**

```
NoOfColumns([total])
```

Argumenten

| Argument | Beschrijving |
|---|---|
| TOTAL | Als de tabel eendimensionaal is of als de kwalificatie TOTAL wordt gebruikt als argument, is het huidige kolomsegment altijd gelijk aan de hele kolom. |

In een draaitabel met meerdere horizontale dimensies, omvat het huidige rijsegment alleen kolommen met dezelfde waarden als de huidige kolom in alle dimensierijen, met uitzondering van de rij waarin de laatste horizontale dimensie wordt weergegeven in de onderlinge sorteervolgorde van de velden. De sorteervolgorde tussen velden voor horizontale dimensies in draaitabellen wordt simpelweg gedefinieerd door de volgorde van de dimensies van boven naar onderzijn.

Voorbeeld: diagramuitdrukking

| Voorbeeld | Resultaat |
|---|---|
| `IF(ColumnNo() > NoOfColumns() - 2, Sum(Sales), Null())` | Wanneer de `ColumnNo` groter is dan de `NoOfColumns` minus `2`, zal het diagram de waarde `Sum(Sales)` retourneren. Anders wordt, als `ColumnNo` kleiner is dan dat, `NULL` geretourneerd. In dit voorbeeld, vanaf de derde kolom, retourneert het diagram waarden terwijl kolom 1 en kolom 2 geen waarden retourneren. |

## Voorbeeld - Basisprincipes voor NoOfColumns

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Period`
  - `Sales`

### Load-script

```
Example:
Load * inline [
Period, Sales
Q1, 1000
Q2, 2000
Q3, 3000
Q4, 4000
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe draaitabel en voeg dit veld toe als een kolomdimensie:

- `Period`

Maak de volgende metingen:

- `=Sum(Sales)`, om de som van verkoop te berekenen.
- `=NoOfColumns()`, om het aantal kolommen te berekenen.

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 |

De resultaten tonen de som van de verkopen voor de periode `Q1` tot `Q4`. De functie `NoofColumns` retourneert een waarde van `4` omdat er `4` kolommen worden weergegeven voor het huidige rijsegment.

## Voorbeeld – Scenario met NoOfColumns

Diagramuitdrukking

### Overzicht

Een gegevensverzameling bevat verkoopcijfers van elk kwartaal. U wilt de verkoopwaarden voor alleen de laatste twee kwartalen (Q3 en Q4) in de draaitabel weergeven.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Product`
  - `Quarter`
  - `Sales`

### Load-script

```
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
];
```

**Resultaten**

Laad de gegevens en open een werkblad. Maak een nieuwe draaitabel en voeg dit veld toe als een rijdimensie:

- `Product`

Voeg dit veld toe als een kolomdimensie:

- `Quarter`

Maak de volgende meting:

- `=IF(ColumnNo() > NoOfColumns() - 2, Sum(Sales), Null())`, om de waarden voor alleen de laatste twee kolommen in de draaitabel weer te geven (`Q3` en `Q4`).

Results table

| Product | Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|---|
| Apple | - | - | 140 | 160 |
| Banana | - | - | 200 | 220 |
| Cherry | - | - | 110 | 150 |

De resultaten tonen alleen verkoopwaarden voor kolom 3 en 4, de laatste twee kwartalen van het jaar.

## 8.17 Logische functies

In dit gedeelte worden functies beschreven die logische bewerkingen behandelen. Alle functies kunnen zowel worden gebruikt in het load-script voor gegevens als in diagramuitdrukkingen.

**IsNum**

Retourneert -1 (True) als de uitdrukking kan worden geïnterpreteerd als een getal, anders 0 (False).

```
IsNum( expr )
```

**IsText**

Retourneert -1 (True) als de uitdrukking een tekstrepresentatie heeft, anders 0 (False).

```
IsText( expr )
```

*Zowel **IsNum** als **IsText** retourneert 0 als de uitdrukking NULL is.*

**Voorbeeld:**

In het volgende voorbeeld wordt een inline-tabel met gemengde tekst- en numerieke waarden galden en worden twee velden toegevoegd om te controleren of de waarde een numerieke waarde respectievelijk een tekstwaarde is.

```
Load *, IsNum(Value), IsText(Value)
Inline [
Value
23
Green
Blue
12
33Red];
```

De resulterende tabel ziet er als volgt uit:

Resulting table

| Value | IsNum(Value) | IsText(Value) |
|---|---|---|
| 23 | -1 | 0 |
| Green | 0 | -1 |
| Blue | 0 | -1 |
| 12 | -1 | 0 |
| 33Red | 0 | -1 |

# 8.18 Toewijzingsfuncties

In dit hoofdstuk komen functies voor het afhandelen van toewijzingstabellen aan de orde. Een toewijzingstabel kan worden gebruikt om veldwaarden en veldnamen tijdens de uitvoering van een script te vervangen.

Toewijzingsfuncties kunnen alleen worden gebruikt in het load-script voor gegevens.

## Overzicht toewijzingsfuncties

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

### ApplyMap

De scriptfunctie **ApplyMap** wordt gebruikt voor het toewijzen van de uitvoer van een uitdrukking aan een eerdere geladen toewijzingstabel.

```
ApplyMap ('mapname', expr [ , defaultexpr ] )
```

### MapSubstring

De scriptfunctie **MapSubstring** wordt gebruikt voor de toewijzing van onderdelen van een uitdrukking aan een eerder geladen toewijzingstabel. De toewijzing is hoofdlettergevoelig en niet-iteratief en subtekenreeksen worden van links naar rechts toegewezen.

```
MapSubstring ('mapname', expr)
```

## ApplyMap

De scriptfunctie **ApplyMap** wordt gebruikt voor het toewijzen van de uitvoer van een uitdrukking aan een eerdere geladen toewijzingstabel.

**Syntaxis:**

```
ApplyMap('map_name', expression [ , default_mapping ] )
```

**Retourgegevenstypen:** dual

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| map_name | De naam van een toewijzingstabel die eerder is gemaakt met de opdracht **mapping load** of **mapping select**. De naam moet tussen enkele, rechte aanhalingstekens staan.<br><br>*Als u deze functie gebruikt voor een variabele die is uitgebreid met een macro en verwijst naar een toewijzingstabel die niet bestaat, geeft de functie een fout weer en wordt er geen veld gemaakt.* |
| expression | De uitdrukking waarvan het resultaat moet worden toegewezen. |
| default_ mapping | Als de uitdrukking is opgegeven, wordt deze waarde gebruikt als standaardwaarde als de toewijzingstabel geen overeenkomstige waarde bevat voor expression. Als de uitdrukking niet is opgegeven, wordt de waarde van expression ongewijzigd geretourneerd. |

*Het uitvoerveld van ApplyMap mag niet dezelfde naam hebben als een van de invoervelden. Dit kan tot onverwachte resultaten leiden. Voorbeeld dat u niet moet gebruiken:* `ApplyMap ('Map', A) as A`.

**Voorbeeld:**

In dit voorbeeld laden wij een lijst met verkopers waarbij een landcode het land aangeeft waar zijn wonen. Wij gebruiken een tabel waarin een landcode wordt toegewezen aan een land ter vervanging van de landcode door de landnaam. Er zijn slechts drie landen gedefinieerd in de toewijzingstabel, andere landcode worden toegewezen aan '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';
```

De resulterende tabel (Verkopers) ziet er als volgt uit:

Resulting table

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

## MapSubstring

De scriptfunctie **MapSubstring** wordt gebruikt voor de toewijzing van onderdelen van een uitdrukking aan een eerder geladen toewijzingstabel. De toewijzing is hoofdlettergevoelig en niet-iteratief en subtekenreeksen worden van links naar rechts toegewezen.

**Syntaxis:**

```
MapSubstring('map_name', expression)
```

**Retourgegevenstypen:** tekenreeks

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| map_name | De naam van een toewijzingstabel die eerder is gelezen in een opdracht **mapping load** of **mapping select**. De naam moet tussen enkele, rechte aanhalingstekens staan.<br><br>*Als u deze functie gebruikt voor een variabele die is uitgebreid met een macro en verwijst naar een toewijzingstabel die niet bestaat, geeft de functie een fout weer en wordt er geen veld gemaakt.* |
| expression | De uitdrukking waarvan het resultaat door subtekenreeksen moet worden toegewezen. |

**Voorbeeld:**

In dit voorbeeld laden wij een lijst met productmodellen. Elk model heeft een reeks van kenmerken die worden beschreven via een samengestelde code. Door de toewijzingstabel met MapSubstring te gebruiken, kunnen wij de kenmerkcodes uitbreiden tot een beschrijving.

```
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';
```

De resulterende tabel ziet er als volgt uit:

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 Rekenkundige functies

In dit hoofdstuk komen functies voor rekenkundige constanten en Booleaanse waarden aan de orde. Deze functies hebben geen parameters, maar de haakjes zijn wel verplicht.

Alle functies kunnen zowel worden gebruikt in het load-script voor gegevens als in diagramuitdrukkingen.

### e

De functie retourneert het grondtal van de natuurlijke logaritmen, **e** ( 2,71828...).

```
e( )
```

### false

De functie retourneert een duale waarde met de tekstwaarde 'False' en numerieke waarde 0, die als logische waarde Onwaar in uitdrukkingen kan worden gebruikt.

```
false( )
```

### pi

De functie retourneert de waarde van π (3.14159...).

```
pi( )
```

### rand

De functie retourneert een willekeurig getal tussen 0 en 1. Hiermee kunnen steekproefgegevens worden gemaakt.

```
rand( )
```

**Voorbeeld:**

In dit voorbeeldscript wordt een tabel van 1000 records gemaakt met willekeurig geselecteerde tekens in hoofdletters, oftewel tekens in het bereik van 65 tot en met 91 (65+26).

```
Load
     Chr( Floor(rand() * 26) + 65) as UCaseChar,
     RecNo() as ID
     Autogenerate 1000;
```

### true

De functie retourneert een duale waarde met de tekstwaarde 'True' en numerieke waarde -1, die als logische waarde Waar in uitdrukkingen kan worden gebruikt.

```
true( )
```

## 8.20 NULL-functies

In dit hoofdstuk komen functies voor retournerende of detecterende NULL-waarden aan de orde.

Alle functies kunnen zowel worden gebruikt in het load-script voor gegevens als in diagramuitdrukkingen.

### Overzicht NULL-functies

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

### EmptyIsNull

De functie **EmptyIsNull** zet lege tekenreeksen om in NULL. Om die reden wordt NULL geretourneerd als de parameter een lege tekenreeks is, anders wordt de parameter geretourneerd.

```
EmptyIsNull (expr )
```

### IsNull

De functie **IsNull** test of de waarde van een uitdrukking NULL is. Als dit het geval is, wordt -1 (True) geretourneerd, anders 0 (False).

```
IsNull (expr )
```

### Null

De functie **Null** retourneert een null-waarde.

```
NULL( )
```

### EmptyIsNull

De functie **EmptyIsNull** zet lege tekenreeksen om in NULL. Om die reden wordt NULL geretourneerd als de parameter een lege tekenreeks is, anders wordt de parameter geretourneerd.

**Syntaxis:**

```
EmptyIsNull(exp )
```

**Retourgegevenstypen:** NULL als de parameter een lege tekenreeks is, anders wordt de parameter geretourneerd.

Argumenten

| Argument | Beschrijving |
|---|---|
| **expr** | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `EmptyIsNull(AdditionalComments)` | Deze uitdrukking wordt als NULL geretourneerd voor lege tekenreekswaarden van het veld `AdditionalComments` in plaats van lege tekenreeksen. Niet-lege tekenreeksen en getallen worden geretourneerd. |
| `EmptyIsNull(PurgeChar(PhoneNumber, ' -()'))` | Deze uitdrukking zorgt ervoor dat alle streepjes, spaties en haakjes uit het veld `PhoneNumber` worden verwijderd. Als alle tekens zijn verwijderd, retourneert de functie `EmptyIsNull` de lege tekenreeks als NULL; een leeg veld voor phone number is hetzelfde als geen phone number. |

## Voorbeeld - Basisprincipes voor EmptyIsNull

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `CustomerID`
  - `OrderID`

- Product
- Status
- PhoneNumber

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- CustomerID
- OrderID
- Product
- Status

Maak de volgende berekende dimensies:

- =EmptyIsNull(Status), om lege tekenreeksen in het veld Status te retourneren als NULL.
- =EmptyIsNull(PurgeChar(PhoneNumber, ' -()')), om alle streepjes, spaties en haakjes uit het veld PhoneNumber te verwijderen. Als alle tekens zijn verwijderd, retourneert de functie EmptyIsNull de lege tekenreeks als NULL; een leeg veld voor telefoonnummer is hetzelfde als geen telefoonnnummer.

*De volgende tabel bevat lege cellen om het gebruik van de functie EmptyIsNull te demonstreren.*

Results table

| CustomerID | OrderID | Product | Status | EmptyIsNull (Status) | EmptyIsNull(PurgeChar (PhoneNumber, ' -()')) |
|---|---|---|---|---|---|
| 1 | 1001 | Widget | | - | 0123456789 |
| 2 | 1002 | Gizmo | Open | Open | - |
| 3 | 1003 | Gadget | Closed | Closed | - |
| 4 | 1004 | Widget | | - | 6789888 |

In de kolom `Status` ziet u dat de records met lege waarden verschijnen als lege cellen met een standaard transparante achtergrondkleur. De uitvoer van de `EmptyIsNull(Status)` meting retourneert NULL voor deze lege waarden, zoals aangegeven door het streepjesteken (-) met grijze celachtergrond.

De tweede meting gebruikt de functie `EmptyIsNull` met de *PurgeChar (page 1835)* om alle streepjes (-), spaties en haakjes () uit het veld `PhoneNumber` te verwijderen, en retourneert vervolgens alle overgebleven lege tekenreeksen als NULL. De cellen met nulwaarden worden nu weergegeven met een streepje (-) en een grijze achtergrondopmaak.

## Voorbeeld - Nullwaarden vervangen door tekst en voorwaardelijke opmaak toevoegen

Diagramuitdrukking

### Overzicht

Een verkoopmanager wil verkoopgegevens analyseren en gemakkelijk de records identificeren die ontbrekende waarden bevatten.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `ID`
  - `Name`
  - `Age`
  - `Email`
  - `Sales`
  - `Region`
  - `OrderStatus`

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `ID`

Maak de volgende berekende dimensies:

- `=If(IsNull(EmptyIsNull(Age)), 'NULL', EmptyIsNull(Age))`, om alle lege tekenreeksen in het veld `Age` te retourneren als NULL, en vervolgens alle nulwaarden te identificeren en te vullen met de waarde `NULL` in plaats van een streepje (-).
- `=If(IsNull(EmptyIsNull(Email)), 'NULL', EmptyIsNull(Email))`, om alle lege tekenreeksen in het veld `Email` te retourneren als NULL, en vervolgens alle nulwaarden te identificeren en te vullen met de waarde `NULL` in plaats van een streepje (-).
- `=If(IsNull(EmptyIsNull(Sales)), 'NULL', EmptyIsNull(Sales))`, om alle lege tekenreeksen in het veld `Sales` te retourneren als NULL, en vervolgens alle nulwaarden te identificeren en te vullen met de waarde `NULL` in plaats van een streepje (-).
- `=If(IsNull(EmptyIsNull(Region)), 'NULL', EmptyIsNull(Region))`, om alle lege tekenreeksen in het veld `Region` te retourneren als NULL, en vervolgens alle nulwaarden te identificeren en te vullen met de waarde `NULL` in plaats van een streepje (-).
- `=If(IsNull(EmptyIsNull(OrderStatus)), 'NULL', EmptyIsNull(OrderStatus))`, om alle lege tekenreeksen in het veld `OrderStatus` te retourneren als NULL, en vervolgens alle nulwaarden te identificeren en te vullen met de waarde `NULL` in plaats van een streepje (-).

Stel voorwaardelijke celopmaak in om de achtergrondkleur te wijzigen in rood voor `NULL` waarden en groen voor alle andere waarden. Voer in het deelvenster **Eigenschappen** voor elke dimensie en meting de volgende uitdrukking in als de **Uitdrukking achtergrondkleur**:

- `ID` (dimensie): `=RGB(200,255,200)`
- `Age` (meting): `=If(IsNull(EmptyIsNull(Age)), RGB(255,200,200), RGB(200,255,200))`
- `Email` (meting): `=If(IsNull(EmptyIsNull(Email)), RGB(255,200,200), RGB(200,255,200))`
- `Sales` (meting): `=If(IsNull(EmptyIsNull(Sales)), RGB(255,200,200), RGB(200,255,200))`
- `Region` (meting): `=If(IsNull(EmptyIsNull(Region)), RGB(255,200,200), RGB(200,255,200))`
- `OrderStatus` (meting): `=If(IsNull(EmptyIsNull(OrderStatus)), RGB(255,200,200), RGB(200,255,200))`

Results table

| ID | If(IsNull(EmptyIsNull(Age)), 'NULL', EmptyIsNull(Age)) | If(IsNull(EmptyIsNull(Email)), 'NULL', EmptyIsNull(Email)) | If(IsNull(EmptyIsNull(Sales)), 'NULL', EmptyIsNull(Sales)) | If(IsNull(EmptyIsNull(Region)), 'NULL', EmptyIsNull(Region)) | If(IsNull(EmptyIsNull(OrderStatus)), 'NULL', EmptyIsNull(OrderStatus)) |
|---|---|---|---|---|---|
| 1 | 30 | john@email.com | 1000 | North | Active |

| ID | If(IsNull (EmptyIsNull (Age)), 'NULL', EmptyIsNull (Age)) | If(IsNull (EmptyIsNull (Email)), 'NULL', EmptyIsNull(Email)) | If(IsNull (EmptyIsNull (Sales)), 'NULL', EmptyIsNull (Sales)) | If(IsNull (EmptyIsNull (Region)), 'NULL', EmptyIsNull (Region)) | If(IsNull (EmptyIsNull (OrderStatus)), 'NULL', EmptyIsNull (OrderStatus)) |
|---|---|---|---|---|---|
| 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 |
| 6 | 35 | eva@email.com | 950 | North | NULL |
| 7 | 50 | NULL | NULL | South | Inactive |
| 8 | 42 | fiona@email.com | 1100 | East | Active |

De resultaten laten zien dat u door de functie `EmptyIsNull` te gebruiken en voorwaardelijke opmaak toe te voegen, gemakkelijk records met ontbrekende waarden kunt identificeren.

## IsNull

De functie **IsNull** test of de waarde van een uitdrukking NULL is. Als dit het geval is, wordt -1 (True) geretourneerd, anders 0 (False).

**Syntaxis:**

```
IsNull(expr )
```

*Een tekenreeks met een lengte van nul wordt niet beschouwd als NULL en **IsNull** retourneert hiervoor False.*

**Retourgegevenstypen:** Booleaanse waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **expr** | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |

Voorbeeld: diagramuitdrukking

| Voorbeeld | Resultaat |
|---|---|
| `IsNull(Productname)` | Retourneert `-1` (True) als `ProductName` een nullwaarde is, anders `0` (False). |

## Voorbeeld - Basisprincipes voor IsNull

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `ID`
  - `Value`

### Load-script

```
Example:
LOAD * Inline
[ID, Value
0,
1,NULL
2,-
3,378];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `ID`
- `Value`

Maak de volgende berekende dimensies:

- `=If(len(trim(Value))= 0 or Value='NULL' or Value='-', Null(), Value )`, om `NULL` te retourneren als het veld `Value` geen waarde, een streepje (-) of de waarde `NULL` heeft, anders wordt de waarde geretourneerd.
- `=If(IsNull(If(len(trim(Value))= 0 or Value='NULL' or Value='-', Null(), Value )), 'T', 'F')`, om de waarde `T` (True) of `F` (False) te retourneren, afhankelijk van of het veld `Value` een nullwaarde bevat of niet.

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 |

| 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') |
|---|---|---|---|
| 1 | NULL | - | T |
| 2 | - | - | T |
| 3 | 378 | 378 | F |

De kolom `Value` toont de gegevens zoals ze in de gegevensverzameling zijn ingevoerd. De eerste meting (kolom 3) retourneert `NULL` voor de eerste 3 rijen waarvan de waarde voldeed aan de voorwaarden van de `IF`-instructie: `len(trim(Value))= 0 or Value='NULL' or Value='-'`. De laatste rij geeft de ingevoerde waarde weer, `378`.

De tweede meting (kolom 4) neemt de waarde van de uitdrukking in kolom 3 en geeft `T` als `NULL` of `F` als niet `NULL` is. De resultaten laten zien dat de eerste 3 rijen nullwaarden zijn.

Load-script

In dit voorbeeld wordt een inline-tabel geladen met vier rijen, waarbij de eerste drie rijen niets, een streepje - of de tekst NULL in de kolom `Value` bevatten. Wij converteren deze waarde naar echte NULL-waarderepresentaties met de middelste voorafgaande **LOAD** via de functie **Null**.

Met de eerste voorafgaande **LOAD** wordt een veld toegevoegd dat controleert of de waarde NULL is, met behulp van de functie **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];
```

Hieronder ziet u de resulterende tabel. In de kolom `ValueNullConv` worden de `NULL`-waarden vertegenwoordigd door `-`.

Results table

| ID | Value | ValueNullConv | IsItNull |
|---|---|---|---|
| 0 | | - | T |
| 1 | NULL | - | T |
| 2 | - | - | T |
| 3 | 378 | 378 | F |

## Voorbeeld – Scenario met IsNull

Diagramuitdrukking

### Overzicht

Een gegevensverzameling met verkoopgegevens wordt geanalyseerd om te zien welke velden null-waarden hebben.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Een `NULLINTERPRET` variabele ingesteld op `NULL`.
- De volgende velden in de gegevenstabel:
  - `SalesID`
  - `Name`
  - `Age`
  - `Email`
  - `Region`
  - `Status`

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `SalesID`
- `Name`

Maak de volgende berekende dimensies:

- `=If(IsNull(Age), 'Age unavailable', Age)`, om het veld `Age` te vullen met de waarde `Age unavailable` als het `NULL` is.
- `=If(IsNull(Email), 'Email unavailable', Email)`, om het veld `Email` te vullen met de waarde `Email unavailable` als het `NULL` is.
- `=If(IsNull(Region), 'Region unavailable', Region)`, om het veld `Region` te vullen met de waarde `Region unavailable` als het `NULL` is.

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 |

De resultaten van de functie `IsNull`, die op null-waarden controleert en true teruggeeft als er null-waarden worden gevonden, geven aan dat verschillende velden in de gegevensverzameling null-waarden bevatten. Er werd bijvoorbeeld een null-waarde geïdentificeerd in de kolom `Age` voor `Charlie Lee`, en de uitvoer `Age unavailable` werd geretourneerd.

## NULL

De functie **Null** retourneert een null-waarde.

**Syntaxis:**

```
Null( )
```

**Retourgegevenstypen:** NULL

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `If(Product = 'A', Null(),Sum(Sales))` | Deze uitdrukking retourneert `NULL` voor `Product A`, anders retourneert het de som van de verkopen voor het product. |

## Voorbeeld - Basisprincipes voor Null

Diagramuitdrukking

### Overzicht

Er wordt een eenvoudige gegevensverzameling gebruikt om de verschillende toepassingen van de functie `Null` te demonstreren.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `ID`
  - `Value`

### Load-script

```
Example:
LOAD * Inline
[ID, Value
0,
1,NULL
2,-
3,378];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `ID`
- `Value`

Maak de volgende berekende dimensies:

- `=If(len(trim(Value))= 0 or Value='NULL' or Value='-', Null(), Value )`, om `NULL` te retourneren als het veld `Value` één van het volgende bevat: , geen waarde, een streepje (-) of de waarde `NULL` heeft, anders wordt de waarde geretourneerd.
- `=If(IsNull(If(len(trim(Value))= 0 or Value='NULL' or Value='-', Null(), Value )), 'T', 'F')`, om de waarde `T` (true) of `F` (false) te retourneren, afhankelijk van of het veld `Value` een null-waarde bevat of niet.

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 |

De kolom `Value` toont de gegevens zoals ze in de gegevensverzameling zijn ingevoerd. De eerste meting (kolom 3) gebruikt de functie `Null` om `NULL` te retourneren voor de waarden die voldoen aan de voorwaarden van de `IF`-instructie: `len(trim(Value))= 0 or Value='NULL' or Value='-'`. De eerste 3 rijen voldeden bijvoorbeeld aan de vereisten en retourneerden `NULL`. De laatste rij geeft de ingevoerde waarde weer, `378`.

De tweede meting (kolom 4) neemt de waarde in kolom 3 en geeft `T` als `NULL` of `F` als niet `NULL` is. De resultaten laten zien dat de eerste 3 rijen nullwaarden zijn.

Load-script

In dit voorbeeld wordt een inline-tabel geladen met vier rijen, waarbij de eerste drie rijen niets, een streepje - of de tekst NULL in de kolom `Value` bevatten. We converteren deze waarden naar echte NULL-waarderepresentaties.

De middelste voorafgaande **LOAD** voert de conversie uit met de functie `Null`.

Met de eerste voorafgaande **LOAD** wordt een veld toegevoegd dat controleert of de waarde NULL is, simpelweg voor illustratiedoeleinden in dit voorbeeld.

```
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];
```

Hieronder ziet u de resulterende tabel. In de kolom `ValueNullConv` worden de `NULL`-waarden vertegenwoordigd door `-`.

Results table

| ID | Value | ValueNullConv | IsItNull |
|---|---|---|---|
| 0 | | - | T |
| 1 | NULL | - | T |
| 2 | - | - | T |
| 3 | 386 | 386 | F |

## Voorbeeld - Waarden uitsluiten met behulp van Null

Diagramuitdrukking

### Overzicht

Verkoopgegevens worden geanalyseerd om de totale verkoop te bepalen en de verkoop die geen verband houdt met `Product A`.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Product`
  - `Sales`

### Load-script

```
Example:
LOAD * Inline [
Product, Sales
A, 100
B, 150
C, 200
D, 250
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Product`

Maak de volgende metingen:

- `=Sum(Sales)`, om de totale verkoop te berekenen.
- `=If(Product = 'A', Null(),Sum(Sales))`, om de totale verkoop te berekenen waarbij het `Product` niet `A` is.

Results table

| Product | Sum(Sales) | If(Product = 'A', Null(),Sum(Sales)) |
|---|---|---|
| A | 100 | - |
| B | 150 | 150 |
| C | 200 | 200 |
| D | 250 | 250 |

Dit voorbeeld toont een eenvoudige manier om de totale verkoop te berekenen en vervolgens de verkoop voor een specifiek product uit te sluiten met behulp van de functie `Null` om de waarde in te stellen op `NULL`. Bijvoorbeeld, in de tweede meting retourneerden alle waarden voor `Product A` `NULL` en werden ze uitgesloten.

## 8.21 Bereikfuncties

De bereikfuncties zijn functies die een reeks van waarden nemen en een enkele waarde als resultaat opleveren. Alle bereikfuncties kunnen zowel worden gebruikt in het load-script van gegevens als in diagramuitdrukkingen.

Zo kan een bereikfunctie in een visualisatie bijvoorbeeld een enkele waarde berekenen op basis van een interrecord-reeks. In het script voor het berekenen van waarden kan een bereikfunctie een enkele waarde berekenen op basis van een reeks van waarden in een interne tabel.

*Bereikfuncties vervangen de volgende algemene numerieke functies: **numsum**, **numavg**, **numcount**, **nummin** en **nummax**. Deze moeten nu als verouderd worden beschouwd.*

### Elementaire bereikfuncties

RangeMax

**RangeMax()** retourneert de hoogste numerieke waarden die binnen de uitdrukking of het veld worden gevonden.

```
RangeMax (first_expr[, Expression])
```

RangeMaxString

**RangeMaxString()** retourneert de laatste waarde die in de gesorteerde tekst van de uitdrukking of het veld wordt gevonden.

```
RangeMaxString (first_expr[, Expression])
```

RangeMin

**RangeMin()** retourneert de laagste numerieke waarden die binnen de uitdrukking of het veld worden gevonden.

```
RangeMin (first_expr[, Expression])
```

RangeMinString

**RangeMinString()** retourneert de eerste waarde die in de gesorteerde tekst van de uitdrukking of het veld wordt gevonden.

```
RangeMinString (first_expr[, Expression])
```

RangeMode

**RangeMode()** retourneert de meest voorkomende waarde (moduswaarde) in de uitdrukking of het veld.

```
RangeMode (first_expr[, Expression])
```

RangeOnly

**RangeOnly()** is een duale functie die een waarde retourneert als de uitdrukking één enkele unieke waarde oplevert. Als dat niet het geval is, retourneert de functie de waarde **NULL**.

```
RangeOnly (first_expr[, Expression])
```

RangeSum

**RangeSum()** retourneert de som van een reeks waarden. Alle niet-numerieke waarden worden behandeld als 0.

```
RangeSum (first_expr[, Expression])
```

## Bereikfuncties voor tellers

RangeCount

**RangeCount()** retourneert het aantal waarden, zowel tekst- als numerieke waarden, in de uitdrukking of het veld.

```
RangeCount (first_expr[, Expression])
```

RangeMissingCount

**RangeMissingCount()** retourneert het aantal niet-numerieke waarden (inclusief NULL) in de uitdrukking of het veld.

```
RangeMissingCount (first_expr[, Expression])
```

RangeNullCount

**RangeNullCount()** retourneert het aantal NULL-waarden in de uitdrukking of het veld.

```
RangeNullCount (first_expr[, Expression])
```

RangeNumericCount

**RangeNumericCount()** retourneert het aantal numerieke waarden in de uitdrukking of het veld.

```
RangeNumericCount (first_expr[, Expression])
```

RangeTextCount

**RangeTextCount()** retourneert het aantal tekstwaarden in de uitdrukking of het veld.

```
RangeTextCount (first_expr[, Expression])
```

## Statistische bereikfuncties

RangeAvg

**RangeAvg()** retourneert het gemiddelde van een bereik. De invoer van de functie is een waardebereik of een uitdrukking.

```
RangeAvg (first_expr[, Expression])
```

RangeCorrel

**RangeCorrel()** retourneert de correlatiecoëfficiënt voor twee gegevensverzamelingen. De correlatiecoëfficiënt is een maat voor de relatie tussen de gegevensverzamelingen.

```
RangeCorrel (x_values , y_values[, Expression])
```

RangeFractile

**RangeFractile()** retourneert de waarde die correspondeert met het n-de **fractile** (kwantiel) van een reeks getallen.

```
RangeFractile (fractile, first_expr[ ,Expression])
```

RangeKurtosis

**RangeKurtosis()** retourneert de waarde die correspondeert met de piekvorm van een reeks getallen.

```
RangeKurtosis (first_expr[, Expression])
```

RangeSkew

**RangeSkew()** retourneert de waarde die correspondeert met de scheefheid van een reeks getallen.

```
RangeSkew (first_expr[, Expression])
```

RangeStdev

**RangeStdev()** retourneert de standaardafwijking van een reeks getallen.

```
RangeStdev (expr1[, Expression])
```

## Financiële bereikfuncties

**RangeIRR**

**RangeIRR()** retourneert het interne opbrengstpercentage voor een reeks van cashflows die worden weergegeven door de invoerwaarden.

```
RangeIRR (value[, value][, Expression])
```

### RangeNPV

**RangeNPV()** retourneert de huidige nettowaarde van een investering op basis van een kortingspercentage en een reeks toekomstige periodieke betalingen (negatieve waarden) en inkomsten (positieve waarden). Het resultaat heeft de standaardgetalnotatie van **money**.

```
RangeNPV (discount_rate, value[, value][, Expression])
```

### RangeXIRR

**RangeXIRR()** retourneert het interne opbrengstpercentage (jaarlijks) voor een schema cashflows die niet noodzakelijk periodiek is. Gebruik voor de berekening van het interne opbrengstpercentage voor een reeks periodieke cashflows de functie **RangeIRR**.

```
RangeXIRR (values, dates[, Expression])
```

### RangeXNPV

**RangeXNPV()** retourneert de interne opbrengstwaarde voor een schema van cashflows (niet noodzakelijk periodiek) dat wordt vertegenwoordigd door getallenparen in de uitdrukkingen **pmt** en **date**. De kortingen voor alle betalingen zijn gebaseerd op een jaar van 365 dagen.

```
RangeXNPV (discount_rate, values, dates[, Expression])
```

---

**Zie ook:**

p *Interrecord-functies (page 1476)*

## RangeAvg

**RangeAvg()** retourneert het gemiddelde van een bereik. De invoer van de functie is een waardebereik of een uitdrukking.

**Syntaxis:**

```
RangeAvg(first_expr[, Expression])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

De argumenten van deze functie kunnen interrecordfuncties bevatten die op hun beurt een lijst met waarden opleveren.

Argumenten

| Argument | Beschrijving |
|---|---|
| first_expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| Expression | Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten. |

**Beperkingen:**

Als er geen numerieke waarde wordt gevonden, wordt NULL geretourneerd.

**Voorbeelden en resultaten:**

Scriptvoorbeelden

| Voorbeelden | Resultaten |
|---|---|
| `RangeAvg (1,2,4)` | retourneert 2,33333333 |
| `RangeAvg (1,'xyz')` | retourneert 1 |
| `RangeAvg (null( ), 'abc')` | Retourneert NULL |

**Voorbeeld:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
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
];
```

De resulterende tabel bevat de geretourneerde waarden van MyRangeAvg voor elk van de records in de tabel.

Resultaattabel

| RangeID | MyRangeAvg |
|---|---|
| 1 | 7 |
| 2 | 4 |
| 3 | 6 |
| 4 | 12.666 |
| 5 | 6.333 |
| 6 | 5 |

Voorbeeld met uitdrukking:

```
RangeAvg (Above(MyField),0,3))
```

Retourneert een glijdend gemiddelde van het resultaat van het bereik van drie waarden van **MyField** berekend op de huidige rij en de twee rijen boven de huidige rij. Doordat het derde argument wordt opgegeven als 3, retourneert de functie **Above()** drie waarden, met voldoende rijen erboven, die als invoer worden gebruikt voor de functie **RangeAvg()**.

Gegevens die worden gebruikt in voorbeelden:

*Schakel het sorteren van **MyField** uit om ervoor te zorgen dat het voorbeeld werkt zoals verwacht.*

Voorbeeldgegevens

| MyField | RangeAvg (Above (MyField,0,3)) | Comments |
|---|---|---|
| 10 | 10 | Omdat dit de bovenste rij is, bestaat het bereik uit slechts één waarde. |
| 2 | 6 | Er bevindt zich slechts één rij boven deze rij, dus is het bereik: 10,2. |
| 8 | 6.6666666667 | Het equivalent van RangeAvg(10,2,8) |
| 18 | 9.333333333 | - |
| 5 | 10. 333333333 | - |
| 9 | 10.6666666667 | - |

```
RangeTab:
LOAD * INLINE [
MyField
10
2
8
18
5
9
] ;
```

**Zie ook:**

p *Avg - diagramfunctie (page 464)*
p *Count - diagramfunctie (page 401)*

## RangeCorrel

**RangeCorrel()** retourneert de correlatiecoëfficiënt voor twee gegevensverzamelingen. De correlatiecoëfficiënt is een maat voor de relatie tussen de gegevensverzamelingen.

**Syntaxis:**

```
RangeCorrel(x_value , y_value[, Expression])
```

**Retourgegevenstypen:** numeriek

Gegevensreeksen moeten worden ingevoerd als (x,y) paren. Als u bijvoorbeeld twee reeksen gegevens, array 1 en array 2, wilt evalueren, waarbij array 1 = 2,6,9 en array 2 = 3,8,4, schrijft u `RangeCorrel (2,3,6,8,9,4)`, waarmee 0.269 wordt geretourneerd.

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| x-value, y-value | Elke waarde vertegenwoordigt een enkele waarde of een bereik van waarden die wordt geretourneerd door een interrecordfunctie met een derde optionele parameter. Elke waarde of bereik van waarden moet overeenkomen met een **x-value** of een bereik van **y-values**. |
| Expression | Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten. |

**Beperkingen:**

De functie vereist ten minste twee coördinaatparen.

Tekstwaarden, NULL-waarden en ontbrekende waarden geven NULL retour.

**Voorbeelden en resultaten:**

Voorbeelden van functies

| Voorbeelden | Resultaten |
|---|---|
| `RangeCorrel (2,3,6,8,9,4,8,5)` | Retourneert 0,2492. Deze functie kan in het script worden geladen of toegevoegd aan een visualisatie in de uitdrukkingseditor. |

**Voorbeeld:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
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 een tabel met ID1 als een dimensie en de meting RangeCorrel(x1,y1,x2,y2,x3,y3,x4,y4,x5,y5,x6,y6)), vindt de **RangeCorrel()** -functie de waarde van **Correl** over een bereik van zes x,y-paren, voor elk van de ID1-waarden.

Resultaattabel

| ID1 | MyRangeCorrel |
|---|---|
| 01 | -0.9517 |
| 02 | -0.5209 |
| 03 | -0.5209 |
| 04 | -0.1599 |

**Voorbeeld:**

```
XY:
LOAD recno() as RangeID, * Inline [
X|Y
2|3
6|8
9|4
8|5
](delimiter is '|');
```

In een tabel met RangeID als een dimensie en de meting: RangeCorrel(Below(X,0,4,BelowY,0,4)), de **RangeCorrel()** functie gebruikt de resultaten van de **Below()** functies, die, vanwege het feit dat het derde argument (count) op 4 is ingesteld, een bereik van vier x-y waarden produceren op basis van de geladen tabel XY.

Resultaattabel

| RangeID | MyRangeCorrel2 |
|---|---|
| 01 | 0.2492 |
| 02 | -0.9959 |
| 03 | -1.0000 |
| 04 | - |

De waarde voor RangeID 01 staat gelijk aan het handmatig invoeren van RangeCorrel(2,3,6,8,9,4,8,5). Voor de andere waarden van RangeID, zijn de series die worden geproduceerd door de Below() functie: (6,8,9,4,8,5), (9,4,8,5) en (8,5); de laatste hiervan produceert een null-resultaat.

**Zie ook:**

p *Correl - diagramfunctie (page 467)*

## RangeCount

**RangeCount()** retourneert het aantal waarden, zowel tekst- als numerieke waarden, in de uitdrukking of het veld.

**Syntaxis:**

```
RangeCount(first_expr[, Expression])
```

**Retourgegevenstypen:** geheel getal

**Argumenten:**

De argumenten van deze functie kunnen interrecordfuncties bevatten die op hun beurt een lijst met waarden opleveren.

Argumenten

| Argument | Beschrijving |
|---|---|
| first_expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden geteld. |
| Expression | Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden geteld. |

**Beperkingen:**

NULL-waarden worden niet meegeteld.

**Voorbeelden en resultaten:**

Voorbeelden van functies

| Voorbeelden | Resultaten |
|---|---|
| `RangeCount (1,2,4)` | retourneert 3 |
| `RangeCount (2,'xyz')` | retourneert 2 |
| `RangeCount (null( ))` | retourneert 0 |
| `RangeCount (2,'xyz', null())` | retourneert 2 |

**Voorbeeld:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
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

];
```

De resulterende tabel bevat de geretourneerde waarden van MyRangeCount voor elk van de records in de tabel.

Resultatentabel

| RangeID | MijnBereikaantal |
|---|---|
| 1 | 3 |
| 2 | 3 |
| 3 | 3 |
| 4 | 3 |
| 5 | 3 |
| 6 | 3 |

Voorbeeld met uitdrukking:

```
RangeCount (Above(MyField,1,3))
```

Retourneert het aantal waarden dat is opgenomen in de drie resultaten van **MyField**. Door het eerste argument van de functie **Above()** op te geven als 1 en het tweede argument als 3, worden de waarden geretourneerd van de eerste drie velden boven de huidige rij, waarbij er voldoende rijen zijn om als invoer te worden gebruikt voor de functie **RangeCount()**.

Gegevens die worden gebruikt in voorbeelden:

Voorbeeldgegevens

| MyField | RangeCount(Above(MyField,1,3)) |
|---|---|
| 10 | 0 |
| 2 | 1 |
| 8 | 2 |
| 18 | 3 |
| 5 | 3 |
| 9 | 3 |

Gegevens die worden gebruikt in voorbeelden:

```
RangeTab:
LOAD * INLINE [
MyField
10
2
8
18
5
9
] ;
```

**Zie ook:**

p *Count - diagramfunctie (page 401)*

## RangeFractile

**RangeFractile()** retourneert de waarde die correspondeert met het n-de **fractile** (kwantiel) van een reeks getallen.

> *RangeFractile() gebruikt lineaire interpolatie tussen dichtstbijzijnde rangnummers bij het berekenen van de fractiel.*

**Syntaxis:**

```
RangeFractile(fractile, first_expr[, Expression])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

De argumenten van deze functie kunnen interrecordfuncties bevatten die op hun beurt een lijst met waarden opleveren.

Argumenten

| Argument | Beschrijving |
|---|---|
| fractile | Een getal tussen 0 en 1 die correspondeert met de fractiel (kwantiel uitgedrukt als een fractie) die moet worden berekend. |
| first_expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| Expression | Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten. |

**Voorbeelden en resultaten:**

Voorbeelden van functies

| Voorbeelden | Resultaten |
|---|---|
| `RangeFractile (0.24,1,2,4,6)` | retourneert 1,72 |
| `RangeFractile(0.5,1,2,3,4,6)` | retourneert 3 |
| `RangeFractile (0.5,1,2,5,6)` | retourneert 3,5 |

**Voorbeeld:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
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
];
```

De resulterende tabel bevat de geretourneerde waarden van MyRangeFrac voor elk van de records in de tabel.

Resultaattabel

| RangeID | MyRangeFrac |
|---|---|
| 1 | 6 |
| 2 | 3 |
| 3 | 8 |
| 4 | 11 |
| 5 | 5 |
| 6 | 4 |

Voorbeeld met uitdrukking:

```
RangeFractile (0.5, Above(Sum(MyField),0,3))
```

In dit voorbeeld bevat de inter-recordfunctie **Above()** de optionele argumenten offset en count. Dit levert een bereik van resultaten op dat kan worden gebruikt als invoer voor elk van de bereikfuncties. In dit geval retourneert `Above(Sum(MyField),0,3)` de waarden van `MyField` voor de huidige rij en de twee rijen

erboven. Deze waarden vormen de invoer voor de functie **RangeFractile()**. Dus voor de onderste rij van de onderstaande tabel is dit het equivalent van `RangeFractile(0.5, 3,4,6)`, oftewel de berekening van de 0,5 fractiel voor de serie 3, 4 en 6. In de eerste twee rijen in de onderstaande tabel, wordt het aantal waarden in het bereik dienovereenkomstig verlaagd, in gevallen waarbij er geen rijen boven de huidige rij zijn. Andere inter-recordfuncties leveren vergelijkbare resultaten op.

Voorbeeldgegevens

| MyField | RangeFractile(0.5, Above(Sum(MyField),0,3)) |
|---|---|
| 1 | 1 |
| 2 | 1.5 |
| 3 | 2 |
| 4 | 3 |
| 5 | 4 |
| 6 | 5 |

Gegevens die worden gebruikt in voorbeelden:

```
RangeTab:
LOAD * INLINE [
MyField
1
2
3
4
5
6
] ;
```

**Zie ook:**

p *Above - diagramfunctie (page 1479)*
p *Fractile - diagramfunctie (page 471)*

## RangeIRR

**RangeIRR()** retourneert het interne opbrengstpercentage voor een reeks van cashflows die worden weergegeven door de invoerwaarden.

De interne rentabiliteit is het rentetarief dat wordt ontvangen voor een investering die bestaat uit betalingen (negatieve waarden) en inkomsten (positieve waarden) die zich in regelmatige perioden voordoen.

Deze functie gebruikt een vereenvoudigde versie van de Newton-methode voor het berekenen van de interne rentabiliteit (IRR).

**Syntaxis:**

```
RangeIRR(value[, value][, Expression])
```

**Retourgegevenstypen:** numeriek

Argumenten

| Argument | Beschrijving |
|---|---|
| value | Een enkele waarde of een enkel bereik van waarden zoals geretourneerd door een interrecordfunctie met een derde optionele parameter. Voor de functie is minimaal één positieve en één negatieve waarde vereist voor de berekening. |
| Expression | Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden worden genegeerd.

Voorbeeldtabel

| Voorbeelden | Resultaten |
|---|---|
| `RangeIRR(-70000,12000,15000,18000,21000,26000)` | retourneert 0,0866 |
| Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.<br>`RangeTab3:`<br>`LOAD *,`<br>`recno() as RangeID,`<br>`RangeIRR(Field1,Field2,Field3) as RangeIRR;`<br>`LOAD * INLINE [`<br>`Field1\|Field2\|Field3`<br>`-10000\|5000\|6000`<br>`-2000\|NULL\|7000`<br>`-8000\|'abc'\|8000`<br>`-1800\|11000\|9000`<br>`-5000\|5000\|9000`<br>`-9000\|4000\|2000`<br>`] (delimiter is '\|');` | De resulterende tabel bevat de geretourneerde waarden van RangeIRR voor elk van de records in de tabel.<br>RangeID RangeIRR<br>1 0.0639<br>2 0.8708<br>3 -<br>4 5.8419<br>5 0.9318<br>6 -0.2566 |

**Zie ook:**

p *Interrecord-functies (page 1476)*

## RangeKurtosis

**RangeKurtosis()** retourneert de waarde die correspondeert met de piekvorm van een reeks getallen.

**Syntaxis:**

```
RangeKurtosis(first_expr[, Expression])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

De argumenten van deze functie kunnen interrecordfuncties bevatten die op hun beurt een lijst met waarden opleveren.

Argumenten

| Argument | Beschrijving |
|---|---|
| first_expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| Expression | Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten. |

**Beperkingen:**

Als er geen numerieke waarde wordt gevonden, wordt NULL geretourneerd.

**Voorbeelden en resultaten:**

Voorbeelden van functies

| Voorbeelden | Resultaten |
|---|---|
| `RangeKurtosis (1,2,4,7)` | retourneert -0,28571428571429 |

**Zie ook:**

p *Kurtosis - diagramfunctie (page 479)*

## RangeMax

**RangeMax()** retourneert de hoogste numerieke waarden die binnen de uitdrukking of het veld worden gevonden.

**Syntaxis:**

```
RangeMax(first_expr[, Expression])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| first_expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| Expression | Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten. |

**Beperkingen:**

Als er geen numerieke waarde wordt gevonden, wordt NULL geretourneerd.

**Voorbeelden en resultaten:**

Voorbeelden van functies

| Voorbeelden | Resultaten |
|---|---|
| `RangeMax (1,2,4)` | retourneert 4 |
| `RangeMax (1,'xyz')` | retourneert 1 |
| `RangeMax (null( ), 'abc')` | Retourneert NULL |

**Voorbeeld:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
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
];
```

De resulterende tabel bevat de geretourneerde waarden van MyRangeMax voor elk van de records in de tabel.

Resultaattabel

| RangeID | MyRangeMax |
|---|---|
| 1 | 10 |

| RangeID | MyRangeMax |
|---|---|
| 2 | 7 |
| 3 | 8 |
| 4 | 18 |
| 5 | 9 |
| 6 | 9 |

Voorbeeld met uitdrukking:

```
RangeMax (Above(MyField,0,3))
```

Retourneert de maximale waarde in het bereik van drie waarden van **MyField** berekend op de huidige rij en de twee rijen boven de huidige rij. Doordat het derde argument wordt opgegeven als 3, retourneert de functie **Above()** drie waarden, met voldoende rijen erboven, die als invoer worden gebruikt voor de functie **RangeMax()**.

Gegevens die worden gebruikt in voorbeelden:

*Schakel het sorteren van **MyField** uit om ervoor te zorgen dat het voorbeeld werkt zoals verwacht.*

Voorbeeldgegevens

| MyField | RangeMax (Above(Sum(MyField),1,3)) |
|---|---|
| 10 | 10 |
| 2 | 10 |
| 8 | 10 |
| 18 | 18 |
| 5 | 18 |
| 9 | 18 |

Gegevens die worden gebruikt in voorbeelden:

```
RangeTab:
LOAD * INLINE [
MyField
10
2
8
18
5
9
] ;
```

## RangeMaxString

**RangeMaxString()** retourneert de laatste waarde die in de gesorteerde tekst van de uitdrukking of het veld wordt gevonden.

**Syntaxis:**

```
RangeMaxString(first_expr[, Expression])
```

**Retourgegevenstypen:** tekenreeks

**Argumenten:**

De argumenten van deze functie kunnen interrecordfuncties bevatten die op hun beurt een lijst met waarden opleveren.

Argumenten

| Argument | Beschrijving |
|---|---|
| first_expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| Expression | Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten. |

**Voorbeelden en resultaten:**

Voorbeelden van functies

| Voorbeelden | Resultaten |
|---|---|
| `RangeMaxString (1,2,4)` | retourneert 4 |
| `RangeMaxString ('xyz','abc')` | Retourneert 'xyz' |
| `RangeMaxString (5,'abc')` | Retourneert 'abc' |
| `RangeMaxString (null( ))` | Retourneert NULL |

Voorbeeld met uitdrukking:

```
RangeMaxString (Above(MaxString(MyField),0,3))
```

Retourneert het laatste (in alfabetische volgorde) van de drie resultaten van de functie **MaxString (MyField)** uitgevoerd op de huidige rij en twee rijen boven de huidige rij.

Gegevens die worden gebruikt in voorbeelden:

*Schakel het sorteren van **MyField** uit om ervoor te zorgen dat het voorbeeld werkt zoals verwacht.*

Voorbeeldgegevens

| MyField | RangeMaxString(Above(MaxString(MyField),0,3)) |
|---|---|
| 10 | 10 |
| abc | abc |
| 8 | abc |
| def | def |
| xyz | xyz |
| 9 | xyz |

Gegevens die worden gebruikt in voorbeelden:

```
RangeTab:
LOAD * INLINE [
MyField
10
'abc'
8
'def'
'xyz'
9
] ;
```

**Zie ook:**

p *MaxString - diagramfunctie (page 608)*

## RangeMin

**RangeMin()** retourneert de laagste numerieke waarden die binnen de uitdrukking of het veld worden gevonden.

**Syntaxis:**

```
RangeMin(first_expr[, Expression])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| first_expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| Expression | Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten. |

**Beperkingen:**

Als er geen numerieke waarde wordt gevonden, wordt NULL geretourneerd.

**Voorbeelden en resultaten:**

Voorbeelden van functies

| Voorbeelden | Resultaten |
|---|---|
| `RangeMin (1,2,4)` | retourneert 1 |
| `RangeMin (1,'xyz')` | retourneert 1 |
| `RangeMin (null( ), 'abc')` | Retourneert NULL |

**Voorbeeld:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
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
];
```

De resulterende tabel bevat de geretourneerde waarden van MyRangeMin voor elk van de records in de tabel.

Resultaattabel

| RangeID | MyRangeMin |
|---|---|
| 1 | 5 |
| 2 | 2 |
| 3 | 2 |
| 4 | 9 |
| 5 | 5 |
| 6 | 2 |

Voorbeeld met uitdrukking:

```
RangeMin (Above(MyField,0,3)
```

Retourneert de minimale waarde in het bereik van drie waarden van **MyField** berekend op de huidige rij en de twee rijen boven de huidige rij. Doordat het derde argument wordt opgegeven als 3, retourneert de functie **Above()** drie waarden, met voldoende rijen erboven, die als invoer worden gebruikt voor de functie **RangeMin()**.

Gegevens die worden gebruikt in voorbeelden:

Voorbeeldgegevens

| MyField | RangeMin(Above(MyField,0,3)) |
|---|---|
| 10 | 10 |
| 2 | 2 |
| 8 | 2 |
| 18 | 2 |
| 5 | 5 |
| 9 | 5 |

Gegevens die worden gebruikt in voorbeelden:

```
RangeTab:
LOAD * INLINE [
MyField
10
2
8
18
5
9
] ;
```

**Zie ook:**

p *Min - diagramfunctie (page 377)*

## RangeMinString

**RangeMinString()** retourneert de eerste waarde die in de gesorteerde tekst van de uitdrukking of het veld wordt gevonden.

**Syntaxis:**

```
RangeMinString(first_expr[, Expression])
```

**Retourgegevenstypen:** tekenreeks

**Argumenten:**

De argumenten van deze functie kunnen interrecordfuncties bevatten die op hun beurt een lijst met waarden opleveren.

Argumenten

| Argument | Beschrijving |
|---|---|
| first_expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| Expression | Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten. |

**Voorbeelden en resultaten:**

Voorbeelden van functies

| Voorbeelden | Resultaten |
|---|---|
| `RangeMinString (1,2,4)` | retourneert 1 |
| `RangeMinString ('xyz','abc')` | Retourneert 'abc' |
| `RangeMinString (5,'abc')` | retourneert 5 |
| `RangeMinString (null( ))` | Retourneert NULL |

Voorbeeld met uitdrukking:

```
RangeMinString (Above(MinString(MyField),0,3))
```

Retourneert het eerste (in alfabetische volgorde) van de drie resultaten van de functie **MinString(MyField)** uitgevoerd op de huidige rij en twee rijen boven de huidige rij.

Gegevens die worden gebruikt in voorbeelden:

*Schakel het sorteren van **MyField** uit om ervoor te zorgen dat het voorbeeld werkt zoals verwacht.*

Voorbeeldgegevens

| MyField | RangeMinString(Above(MinString(MyField),0,3)) |
|---|---|
| 10 | 10 |
| abc | 10 |
| 8 | 8 |
| def | 8 |
| xyz | 8 |
| 9 | 9 |

Gegevens die worden gebruikt in voorbeelden:

```
RangeTab:
LOAD * INLINE [
MyField
10
```

```
'abc'
8
'def'
'xyz'
9
] ;
```

**Zie ook:**

p *MinString - diagramfunctie (page 611)*

## RangeMissingCount

**RangeMissingCount()** retourneert het aantal niet-numerieke waarden (inclusief NULL) in de uitdrukking of het veld.

**Syntaxis:**

```
RangeMissingCount(first_expr[, Expression])
```

**Retourgegevenstypen:** geheel getal

**Argumenten:**

De argumenten van deze functie kunnen interrecordfuncties bevatten die op hun beurt een lijst met waarden opleveren.

Argumenten

| Argument | Beschrijving |
|---|---|
| first_expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden geteld. |
| Expression | Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden geteld. |

**Voorbeelden en resultaten:**

Voorbeelden van functies

| Voorbeelden | Resultaten |
|---|---|
| `RangeMissingCount (1,2,4)` | retourneert 0 |
| `RangeMissingCount (5,'abc')` | retourneert 1 |
| `RangeMissingCount (null( ))` | retourneert 1 |

Voorbeeld met uitdrukking:

```
RangeMissingCount (Above(MinString(MyField),0,3))
```

Retourneert het aantal niet-numerieke waarden in de drie resultaten van de functie **MinString(MyField)**, uitgevoerd op de huidige rij en twee rijen boven de huidige rij.

*Schakel het sorteren van **MyField** uit om ervoor te zorgen dat het voorbeeld werkt zoals verwacht.*

Voorbeeldgegevens

| MyField | RangeMissingCount (Above(MinString (MyField),0,3)) | Explanation |
|---|---|---|
| 10 | 2 | Retourneert 2, omdat er geen rijen boven deze rij staan, zodat 2 van de 3 waarden ontbreken. |
| abc | 2 | Retourneert 2, omdat er slechts 1 rij boven de huidige rij staat en de huidige rij niet-numeriek is ('abc'). |
| 8 | 1 | Retourneert 1, omdat 1 van de 3 rijen een niet-numerieke waarde bevat ('abc'). |
| def | 2 | Retourneert 2, omdat 2 van de 3 rijen niet-numerieke waarden bevatten ('def' en 'abc'). |
| xyz | 2 | Retourneert 2, omdat 2 van de 3 rijen niet-numerieke waarden bevatten (' xyz' en 'def'). |
| 9 | 2 | Retourneert 2, omdat 2 van de 3 rijen niet-numerieke waarden bevatten (' xyz' en 'def'). |

Gegevens die worden gebruikt in voorbeelden:

```
RangeTab:
LOAD * INLINE [
MyField
10
'abc'
8
'def'
'xyz'
9
] ;
```

**Zie ook:**

p *MissingCount - diagramfunctie (page 407)*

## RangeMode

**RangeMode()** retourneert de meest voorkomende waarde (moduswaarde) in de uitdrukking of het veld.

**Syntaxis:**

```
RangeMode(first_expr {, Expression})
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

De argumenten van deze functie kunnen interrecordfuncties bevatten die op hun beurt een lijst met waarden opleveren.

Argumenten

| Argument | Beschrijving |
|---|---|
| first_expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| Expression | Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten. |

**Beperkingen:**

Als er meerdere waarden zijn met de hoogste frequentie, wordt NULL geretourneerd.

**Voorbeelden en resultaten:**

Voorbeelden van functies

| Voorbeelden | Resultaten |
|---|---|
| `RangeMode (1,2,9,2,4)` | retourneert 2 |
| `RangeMode ('a',4,'a',4)` | Retourneert NULL |
| `RangeMode (null( ))` | Retourneert NULL |

**Voorbeeld:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
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

];
```

De resulterende tabel bevat de geretourneerde waarden van **MyRangeMode** voor elk van de records in de tabel.

Resultatentabel

| RangeID | MyRangMode |
|---|---|
| 1 | - |
| 2 | - |
| 3 | 8 |
| 4 | - |
| 5 | 5 |
| 6 | - |

Voorbeeld met uitdrukking:

```
RangeMode (Above(MyField,0,3))
```

Retourneert de meest voorkomende waarde in de drie resultaten van **MyField** uitgevoerd op de huidige rij en twee rijen boven de huidige rij. Doordat het derde argument wordt opgegeven als 3, retourneert de functie **Above()** drie waarden, met voldoende rijen erboven, die als invoer worden gebruikt voor de functie **RangeMode()**.

Gebruikte gegevens in voorbeeld:

```
RangeTab:
LOAD * INLINE [
MyField
10
2
8
18
5
9
] ;
```

*Schakel het sorteren van **MyField** uit om ervoor te zorgen dat het voorbeeld werkt zoals verwacht.*

Voorbeeldgegevens

| MyField | RangeMode(Above(MyField,0,3)) |
|---|---|
| 10 | Retourneert 10, omdat er geen bovenliggende rijen zijn, zodat de enkele waarde de waarde is die het meest voorkomt. |

| MyField | RangeMode(Above(MyField,0,3)) |
|---|---|
| 2 | - |
| 8 | - |
| 18 | - |
| 5 | - |
| 9 | - |

**Zie ook:**

p *Mode - diagramfunctie (page 382)*

## RangeNPV

**RangeNPV()** retourneert de huidige nettowaarde van een investering op basis van een kortingspercentage en een reeks toekomstige periodieke betalingen (negatieve waarden) en inkomsten (positieve waarden). Het resultaat heeft de standaardgetalnotatie van **money**.

Voor cashflows die niet noodzakelijkerwijs periodiek zijn, raadpleegt u *RangeXNPV (page 1639)*.

**Syntaxis:**

```
RangeNPV(discount_rate, value[,value][, Expression])
```

**Retourgegevenstypen:** numeriek

Argumenten

| Argument | Beschrijving |
|---|---|
| discount_rate | Het rentepercentage per periode. |
| value | Een betaling of inkomsten die aan het eind van elke periode plaatsvindt. Elke waarde kan een enkele waarde of een bereik van waarden zijn die is geretourneerd door een interrecordfunctie met een derde optionele parameter. |
| Expression | Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten. |

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden worden genegeerd.

| Voorbeelden | Resultaten |
|---|---|
| `RangeNPV(0.1,-10000,3000,4200,6800)` | retourneert 1188,44 |

| Voorbeelden | Resultaten |
|---|---|
| Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.<br>`RangeTab3:`<br>`LOAD *,`<br>`recno() as RangeID,`<br>`RangeNPV(Field1,Field2,Field3) as RangeNPV;`<br>`LOAD * INLINE [`<br>`Field1\|Field2\|Field3`<br>`10\|5\|-6000`<br>`2\|NULL\|7000`<br>`8\|'abc'\|8000`<br>`18\|11\|9000`<br>`5\|5\|9000`<br>`9\|4\|2000`<br>`] (delimiter is '\|');` | De resulterende tabel bevat de geretourneerde waarden van RangeNPV voor elk van de records in de tabel.<br>RangeID RangeNPV<br>1 $-49.13<br>2 $777.78<br>3 $98.77<br>4 $25.51<br>5 $250.83<br>6 $20.40 |

**Zie ook:**

p *Interrecord-functies (page 1476)*

## RangeNullCount

**RangeNullCount()** retourneert het aantal NULL-waarden in de uitdrukking of het veld.

**Syntaxis:**

```
RangeNullCount(first_expr [, Expression])
```

**Retourgegevenstypen:** geheel getal

**Argumenten:**

De argumenten van deze functie kunnen interrecordfuncties bevatten die op hun beurt een lijst met waarden opleveren.

Argumenten

| Argument | Beschrijving |
|---|---|
| first_expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| Expression | Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten. |

**Voorbeelden en resultaten:**

Voorbeelden van functies

| Voorbeelden | Resultaten |
|---|---|
| `RangeNullCount (1,2,4)` | retourneert 0 |
| `RangeNullCount (5,'abc')` | retourneert 0 |
| `RangeNullCount (null( ), null( ))` | retourneert 2 |

Voorbeeld met uitdrukking:

```
RangeNullCount (Above(Sum(MyField),0,3))
```

Retourneert het aantal NULL-waarden in de drie resultaten van de functie **Sum(MyField)** uitgevoerd op de huidige rij en twee rijen boven de huidige rij.

*Het kopiëren van **MyField** in het onderstaande voorbeeld resulteert niet in een NULL-waarde.*

Voorbeeldgegevens

| MyField | RangeNullCount(Above(Sum(MyField),0,3)) |
|---|---|
| 10 | Retourneert 2, omdat er geen rijen boven deze rij staan, zodat 2 van de 3 waarden ontbreken (=NULL). |
| 'abc' | Retourneert 1, omdat er slechts één rij boven de huidige rij staat, zodat één van de drie waarden ontbreekt (=NULL). |
| 8 | Retourneert 0 omdat één van de drie rijen een NULL-waarde is. |

Gegevens die worden gebruikt in voorbeelden:

```
RangeTab:
LOAD * INLINE [
MyField
10
'abc'
8
] ;
```

**Zie ook:**

p *NullCount - diagramfunctie (page 412)*

## RangeNumericCount

**RangeNumericCount()** retourneert het aantal numerieke waarden in de uitdrukking of het veld.

**Syntaxis:**

```
RangeNumericCount(first_expr[, Expression])
```

**Retourgegevenstypen:** geheel getal

**Argumenten:**

De argumenten van deze functie kunnen interrecordfuncties bevatten die op hun beurt een lijst met waarden opleveren.

Argumenten

| Argument | Beschrijving |
|---|---|
| first_expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| Expression | Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten. |

**Voorbeelden en resultaten:**

Voorbeelden van functies

| Voorbeelden | Resultaten |
|---|---|
| `RangeNumericCount (1,2,4)` | retourneert 3 |
| `RangeNumericCount (5,'abc')` | retourneert 1 |
| `RangeNumericCount (null( ))` | retourneert 0 |

Voorbeeld met uitdrukking:

```
RangeNumericCount (Above(MaxString(MyField),0,3))
```

Retourneert het aantal numerieke waarden in de drie resultaten van de functie **MaxString(MyField)** uitgevoerd op de huidige rij en twee rijen boven de huidige rij.

*Schakel het sorteren van **MyField** uit om ervoor te zorgen dat het voorbeeld werkt zoals verwacht.*

Voorbeeldgegevens

| MyField | RangeNumericCount(Above(MaxString(MyField),0,3)) |
|---|---|
| 10 | 1 |
| abc | 1 |
| 8 | 2 |
| def | 1 |
| xyz | 1 |
| 9 | 1 |

Gegevens die worden gebruikt in voorbeelden:

```
RangeTab:
LOAD * INLINE [
MyField
10
'abc'
8
def
xyz
9
] ;
```

**Zie ook:**

p *NumericCount - diagramfunctie (page 418)*

## RangeOnly

**RangeOnly()** is een duale functie die een waarde retourneert als de uitdrukking één enkele unieke waarde oplevert. Als dat niet het geval is, retourneert de functie de waarde **NULL**.

**Syntaxis:**

```
RangeOnly(first_expr[, Expression])
```

**Retourgegevenstypen:** duaal

**Argumenten:**

De argumenten van deze functie kunnen interrecordfuncties bevatten die op hun beurt een lijst met waarden opleveren.

| Argument | Beschrijving |
|---|---|
| first_expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| Expression | Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten. |

**Voorbeelden en resultaten:**

| Voorbeelden | Resultaten |
|---|---|
| `RangeOnly (1,2,4)` | Retourneert NULL |
| `RangeOnly (5,'abc')` | Retourneert NULL |
| `RangeOnly (null( ), 'abc')` | Retourneert 'abc' |
| `RangeOnly(10,10,10)` | retourneert 10 |

**Zie ook:**

p *Only - diagramfunctie (page 387)*

## RangeSkew

**RangeSkew()** retourneert de waarde die correspondeert met de scheefheid van een reeks getallen.

**Syntaxis:**

```
RangeSkew(first_expr[, Expression])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

De argumenten van deze functie kunnen interrecordfuncties bevatten die op hun beurt een lijst met waarden opleveren.

Argumenten

| Argument | Beschrijving |
|---|---|
| first_expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| Expression | Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten. |

**Beperkingen:**

Als er geen numerieke waarde wordt gevonden, wordt NULL geretourneerd.

**Voorbeelden en resultaten:**

Voorbeelden van functies

| Voorbeelden | Resultaten |
|---|---|
| `rangeskew (1,2,4)` | retourneert 0,93521952958283 |

| Voorbeelden | Resultaten |
|---|---|
| `rangeskew (above (SalesValue,0,3))` | Retourneert een glijdende scheefheid van het bereik van drie waarden dat wordt geretourneerd door de functie above() berekend op de huidige rij en de twee rijen boven de huidige rij. |

Gebruikte gegevens in voorbeeld:

Voorbeeldgegevens

| 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,<br>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
] ;
```

**Zie ook:**

p *Skew - diagramfunctie (page 514)*

## RangeStdev

**RangeStdev()** retourneert de standaardafwijking van een reeks getallen.

**Syntaxis:**

```
RangeStdev(first_expr[, Expression])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

De argumenten van deze functie kunnen interrecordfuncties bevatten die op hun beurt een lijst met waarden opleveren.

Argumenten

| Argument | Beschrijving |
|---|---|
| first_expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| Expression | Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten. |

**Beperkingen:**

Als er geen numerieke waarde wordt gevonden, wordt NULL geretourneerd.

**Voorbeelden en resultaten:**

Voorbeelden van functies

| Voorbeelden | Resultaten |
|---|---|
| `RangeStdev (1,2,4)` | retourneert 1,5275252316519 |
| `RangeStdev (null( ))` | Retourneert NULL |
| `RangeStdev (above (SalesValue),0,3))` | Retourneert een glijdende standaard van het bereik van drie waarden dat wordt geretourneerd door de functie above() berekend op de huidige rij en de twee rijen boven de huidige rij. |

Gebruikte gegevens in voorbeeld:

Voorbeeldgegevens

| 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,<br>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
] ;
```

**Zie ook:**

p *Stdev - diagramfunctie (page 517)*

## RangeSum

**RangeSum()** retourneert de som van een reeks waarden. Alle niet-numerieke waarden worden behandeld als 0.

**Syntaxis:**

```
RangeSum(first_expr[, Expression])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

De argumenten van deze functie kunnen interrecordfuncties bevatten die op hun beurt een lijst met waarden opleveren.

Argumenten

| Argument | Beschrijving |
|---|---|
| first_expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| Expression | Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten. |

**Beperkingen:**

De functie **RangeSum** behandelt alle niet-numerieke waarden als 0.

**Voorbeelden en resultaten:**

Voorbeelden

| Voorbeelden | Resultaten |
|---|---|
| `RangeSum (1,2,4)` | retourneert 7 |

| Voorbeelden | Resultaten |
|---|---|
| `RangeSum (5,'abc')` | retourneert 5 |
| `RangeSum (null( ))` | retourneert 0 |

**Voorbeeld:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
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
];
```

De resulterende tabel bevat de geretourneerde waarden van MyRangeSum voor elk van de records in de tabel.

Resultaattabel

| RangeID | MyRangeSum |
|---|---|
| 1 | 21 |
| 2 | 12 |
| 3 | 18 |
| 4 | 38 |
| 5 | 19 |
| 6 | 15 |

Voorbeeld met uitdrukking:

```
RangeSum (Above(MyField,0,3))
```

Retourneert de som van de drie waarden van **MyField)**: van de huidige rij en de twee rijen boven de huidige rij. Doordat het derde argument wordt opgegeven als 3, retourneert de functie **Above()** drie waarden, met voldoende rijen erboven, die als invoer worden gebruikt voor de functie **RangeSum()**.

Gegevens die worden gebruikt in voorbeelden:

*Schakel het sorteren van **MyField** uit om ervoor te zorgen dat het voorbeeld werkt zoals verwacht.*

Voorbeeldgegevens

| MyField | RangeSum(Above(MyField,0,3)) |
|---|---|
| 10 | 10 |
| 2 | 12 |
| 8 | 20 |
| 18 | 28 |
| 5 | 31 |
| 9 | 32 |

Gegevens die worden gebruikt in voorbeelden:

```
RangeTab:
LOAD * INLINE [
MyField
10
2
8
18
5
9
] ;
```

**Zie ook:**

p *Sum - diagramfunctie (page 394)*
p *Above - diagramfunctie (page 1479)*

## RangeTextCount

**RangeTextCount()** retourneert het aantal tekstwaarden in de uitdrukking of het veld.

**Syntaxis:**

```
RangeTextCount(first_expr[, Expression])
```

**Retourgegevenstypen:** geheel getal

**Argumenten:**

De argumenten van deze functie kunnen interrecordfuncties bevatten die op hun beurt een lijst met waarden opleveren.

Argument

| Argument | Beschrijving |
|---|---|
| first_expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| Expression | Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten. |

**Voorbeelden en resultaten:**

Voorbeelden van functies

| Voorbeelden | Resultaten |
|---|---|
| `RangeTextCount (1,2,4)` | retourneert 0 |
| `RangeTextCount (5,'abc')` | retourneert 1 |
| `RangeTextCount (null( ))` | retourneert 0 |

Voorbeeld met uitdrukking:

```
RangeTextCount (Above(MaxString(MyField),0,3))
```

Retourneert het aantal tekstwaarden in de drie resultaten van de functie **MaxString(MyField)** uitgevoerd op de huidige rij en twee rijen boven de huidige rij.

Gegevens die worden gebruikt in voorbeelden:

*Schakel het sorteren van **MyField** uit om ervoor te zorgen dat het voorbeeld werkt zoals verwacht.*

Voorbeeldgegevens

| 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 |

Gegevens die worden gebruikt in voorbeelden:

```
RangeTab:

LOAD * INLINE [

MyField

10

'abc'

8

null()

'xyz'

9

] ;
```

**Zie ook:**

p *TextCount - diagramfunctie (page 423)*

## RangeXIRR

**RangeXIRR()** retourneert het interne opbrengstpercentage (jaarlijks) voor een schema cashflows die niet noodzakelijk periodiek is. Gebruik voor de berekening van het interne opbrengstpercentage voor een reeks periodieke cashflows de functie **RangeIRR**.

De XIRR-functionaliteit van Qlik (de functies **XIRR()** en **RangeXIRR()**) gebruiken de volgende vergelijking, als oplossing voor de `Rate`-waarde, om de juiste XIRR-waarde te bepalen:

```
XNPV(Rate, pmt, date) = 0
```

De vergelijking wordt opgelost met een vereenvoudigde versie van de Newton-methode.

**Syntaxis:**

```
RangeXIRR(value, date{, value, date})
```

**Retourgegevenstypen:** numeriek

Argumenten

| Argument | Beschrijving |
|---|---|
| value | Een cashflow of een reeks cashflows die overeenkomen met een betalingsschema met datums. De reeks waarden moet ten minste een positieve en een negatieve waarde bevatten. |

| Argument | Beschrijving |
|---|---|
| date | Een betaaldatum of een schema van betaaldatums die corresponderen met de cashflowbetalingen. |

Als u werkt met deze functie, gelden de volgende beperkingen:

- Tekstwaarden, NULL-waarden en ontbrekende waarden worden genegeerd.
- De kortingen voor alle betalingen zijn gebaseerd op een jaar van 365 dagen.
- Voor deze functie is ten minste één geldige negatieve en één geldige positieve betaling vereist (met bijbehorende geldige datums). Als deze betalingen niet zijn opgegeven, wordt er een NULL-waarde geretourneerd.

De volgende onderwerpen kunnen u helpen bij het werken met deze functie:

- *RangeXNPV (page 1639)*: gebruik deze functie om de huidige nettowaarde te berekenen voor een schema cashflows die niet noodzakelijk periodiek is.
- *XIRR (page 440)*: de **XIRR()** functie berekent het interne opbrengstpercentage (jaarlijks) voor een schema cashflows (die niet noodzakelijk periodiek is).

*Binnen verschillende versies van Qlik Sense met clientbeheer bestaan er variaties op het onderliggende algoritme dat door deze functie wordt gebruikt. Voor meer informatie over recente updates van het algoritme, raadpleegt u het ondersteuningsartikel* XIRR-functie herstellen en bijwerken.

**Voorbeelden en resultaten:**

Voorbeelden en resultaten

| Voorbeelden | Resultaten |
|---|---|
| `RangeXIRR(-2500,'2008-01-01',2750,'2008-09-01')` | retourneert 0,1532 |

**Zie ook:**

p *RangeIRR (page 1612)*
p *RangeXNPV (page 1639)*
p *XIRR (page 440)*
≤ XIRR-functie herstellen en bijwerken

## RangeXNPV

**RangeXNPV()** retourneert de interne opbrengstwaarde voor een schema van cashflows (niet noodzakelijk periodiek) dat wordt vertegenwoordigd door getallenparen in de uitdrukkingen **pmt** en **date**. De kortingen voor alle betalingen zijn gebaseerd op een jaar van 365 dagen.

**Syntaxis:**

```
RangeXNPV(discount_rate, value, date{, value, date})
```

**Retourgegevenstypen:** numeriek

Argumenten

| Argument | Beschrijving |
|---|---|
| discount_rate | **discount_rate** is het jaartarief waarmee korting op betalingen wordt toegepast. |
| value | Een cashflow of een reeks cashflows die overeenkomen met een betalingsschema met datums. Elke waarde kan een enkele waarde of een bereik van waarden zijn die is geretourneerd door een interrecordfunctie met een derde optionele parameter. De reeks waarden moet ten minste een positieve en een negatieve waarde bevatten. |
| date | Een betaaldatum of een schema van betaaldatums die corresponderen met de cashflowbetalingen. |

Als u werkt met deze functie, gelden de volgende beperkingen:

- Tekstwaarden, NULL-waarden en ontbrekende waarden worden genegeerd.
- De kortingen voor alle betalingen zijn gebaseerd op een jaar van 365 dagen.

## Voorbeeld - script

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Financiële gegevens in een tabel met de naam `RangeTab3`/
- Het gebruik van de **RangeXNPV()** functie om de huidige nettowaarde te berekenen.

### Load-script

```
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 '|');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `RangeID`
- `RangeXNPV`

Resultatentabel

| RangeID | RangeXNPV |
|---|---|
| 1 | -$9.09 |
| 2 | -$0.00 |
| 3 | $13.64 |

## Voorbeeld - diagramuitdrukking

Load-script en diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Financiële gegevens in een tabel met de naam `RangeTab3`/
- Het gebruik van de **RangeXNPV()** functie om de huidige nettowaarde te berekenen.

### Load-script

```
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 '|');
```

### Resultaten

### Doe het volgende:

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg de volgende berekening als een meting toe.

```
=RangeXNPV(0.1, -2500,'2008-01-01',2750,'2008-09-01')
```

Resultatentabel

| =XIRR(Payments, Date) |
|---|
| $80.25 |

**Zie ook:**

p *XNPV (page 447)*

## 8.22 Relationele functies

Dit is een groep functies die eigenschappen van afzonderlijke dimensiewaarden in een diagram berekenen met behulp van eerder geaggregeerde getallen.

De functies zijn relationeel in de zin dat de functie-uitvoer niet alleen afhankelijk is van de waarde van het gegevenspunt zelf, maar ook van de relatie van de waarde met andere gegevenspunten. Een rangschikking kan bijvoorbeeld niet worden berekend zonder een vergelijking met andere dimensiewaarden.

Deze functies kunnen alleen worden gebruikt in diagramuitdrukkingen. Ze kunnen niet worden gebruikt in het load-script.

Een dimensie is vereist in het diagram, omdat deze de andere benodigde gegevenspunten voor de vergelijking definieert. Daarom is een relationele functie niet van betekenis in een diagram zonder dimensies (bijvoorbeeld een KPI-object).

### Rangordefuncties

> *De onderdrukking van nulwaarden wordt automatisch uitgeschakeld als deze functies worden gebruikt. NULL-waarden worden buiten beschouwing gelaten.*

Rank

**Rank()** evalueert de rijen van het diagram in de uitdrukking en geeft voor elke rij de relatieve positie van de waarde van de dimensie die in de uitdrukking wordt geëvalueerd weer. Bij de evaluatie van de uitdrukking vergelijkt de functie het resultaat met het resultaat van de andere rijen die het huidige kolomsegment bevatten, en retourneert de rangschikking van de huidige rij in het segment.

```
Rank - diagramfunctie([TOTAL [<fld {, fld}>]] expr[, mode[, fmt]])
```

HRank

**HRank()** evalueert de uitdrukking en vergelijkt het resultaat met het resultaat van de andere kolommen die het huidige rijsegment van een draaitabel bevatten. De functie retourneert vervolgens de rangschikking van de huidige kolom binnen het segment.

```
HRank - diagramfunctie([TOTAL] expr[, mode[, fmt]])
```

### Clusterfuncties

KMeans2D

De eigenschapgroep **Sitelicentie** bevat eigenschappen die gerelateerd zijn aan de licentie voor het Qlik Sense-systeem. Alle velden zijn verplicht en mogen niet leeg zijn.

Sitelicentie-eigenschappen

| Eigenschapnaam | Beschrijving |
|---|---|
| **Naam eigenaar** | De gebruikersnaam van de Qlik Sense-producteigenaar. |
| **Organisatie eigenaar** | De naam van de organisatie waarvan de Qlik Sense-producteigenaar lid is. |
| **Serienummer** | Het serienummer dat aan de Qlik Sense-software is toegewezen. |
| **Controlenummer** | Het controlenummer dat aan de Qlik Sense-software is toegewezen. |
| **Toegang tot LEF** | De License Enabler File (LEF) die aan de Qlik Sense-software is toegewezen. |

**KMeans2D()** evalueert de rijen van het diagram door K-means clustering toe te passen. Voor elke rij van het diagram wordt de cluster-id weergegeven van het cluster waaraan dit gegevenspunt is toegewezen. De kolommen die worden gebruikt door het clustering-algoritme worden bepaald door de parameter coordinate_1 respectievelijk coordinate_2. Dit zijn beide aggregraties. De parameter num_clusters bepaalt het aantal clusters dat wordt gemaakt. Gegevens kunnen optioneel worden genormaliseerd door de normparameter.

```
KMeans2D - diagramfunctie(num_clusters, coordinate_1, coordinate_2 [, norm])
```

KMeansND

**KMeansND()** evalueert de rijen van het diagram door K-means clustering toe te passen. Voor elke rij van het diagram wordt de cluster-id weergegeven van het cluster waaraan dit gegevenspunt is toegewezen. De kolommen die worden gebruikt door het clustering-algoritme worden bepaald door de parameter coordinate_1 en coordinate_2 etc. tot en met de n-kolommen Dit zijn beide aggregaties. De parameter num_clusters bepaalt het aantal clusters dat wordt gemaakt.

```
KMeansND - diagramfunctie(num_clusters, num_iter, coordinate_1, coordinate_2
[,coordinate_3 [, ...]])
```

KMeansCentroid2D

**KMeansCentroid2D()** evalueert de rijen van het diagram door K-means clustering toe te passen. Voor elke rij van het diagram wordt de gewenste coördinaat weergegeven van het cluster waaraan dit gegevenspunt is toegewezen. De kolommen die worden gebruikt door het clustering-algoritme worden bepaald door de parameter coordinate_1 respectievelijk coordinate_2. Dit zijn beide aggregraties. De parameter num_clusters bepaalt het aantal clusters dat wordt gemaakt. Gegevens kunnen optioneel worden genormaliseerd door de normparameter.

```
KMeansCentroid2D - diagramfunctie(num_clusters, coordinate_no, coordinate_1,
coordinate_2 [, norm])
```

KMeansCentroidND

**KMeansCentroidND()** evalueert de rijen van het diagram door K-means clustering toe te passen. Voor elke rij van het diagram wordt de gewenste coördinaat weergegeven van het cluster waaraan dit gegevenspunt is toegewezen. De kolommen die worden gebruikt door het clustering-algoritme worden bepaald door de parameter coordinate_1 respectievelijk coordinate_2. Dit zijn beide aggregaties. De parameter num_clusters bepaalt het aantal clusters dat wordt gemaakt.

```
KMeansCentroidND - diagramfunctie(num_clusters, num_iter, coordinate_no,
coordinate_1, coordinate_2 [,coordinate_3 [, ...]])
```

## Ontledingsfuncties voor tijdreeksen

STL_Trend

**STL_Trend** is een splitsingsfunctie voor tijdreeksen. Samen met **STL_Seasonal** en **STL_Residual** wordt deze functie gebruikt om een tijdreeks te splitsen in seizoensgebonden-, trendgebonden- en restcomponenten. Binnen de context van het STL-algoritme wordt de splitsingsfunctie voor tijdreeksen gebruikt om zowel terugkerende seizoensgebonden patronen te identificeren op basis van invoergegevens en andere parameters. De functie **STL_Trend** identificeert een algemene trend, ongeacht de seizoensgebonden patronen of cycli, op basis van de gegevens van tijdreeksen.

```
STL_Trend - diagramfunctie(target_measure, period_int [,seasonal_smoother
[,trend_smoother]])
```

STL_Seasonal

**STL_Seasonal** is een splitsingsfunctie voor tijdreeksen. Samen met **STL_Trend** en **STL_Residual** wordt deze functie gebruikt om een tijdreeks te splitsen in seizoensgebonden-, trendgebonden- en restcomponenten. Binnen de context van het STL-algoritme wordt de splitsingsfunctie voor tijdreeksen gebruikt om zowel terugkerende seizoensgebonden patronen te identificeren op basis van invoergegevens en andere parameters. De functie **STL_Seasonal** kan een seizoensgebonden patroon binnen een tijdsreeks identificeren, die wordt gescheiden van de algemene trend binnen de gegevens.

```
STL_Seasonal - diagramfunctie(target_measure, period_int [,seasonal_smoother
[,trend_smoother]])
```

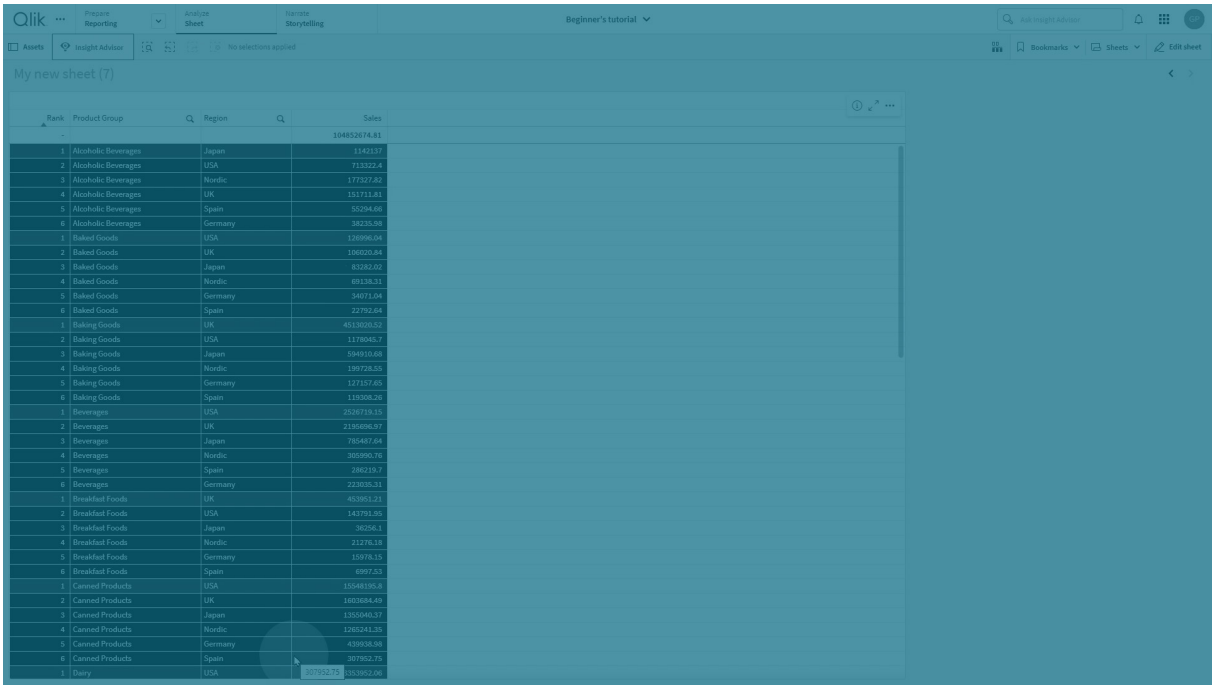
STL_Residual

**STL_Residual** is een splitsingsfunctie voor tijdreeksen. Samen met **STL_Seasonal** en **STL_Trend** wordt deze functie gebruikt om een tijdreeks te splitsen in seizoensgebonden-, trendgebonden- en restcomponenten. Binnen de context van het STL-algoritme wordt de splitsingsfunctie voor tijdreeksen gebruikt om zowel terugkerende seizoensgebonden patronen te identificeren op basis van invoergegevens en andere parameters. Bij het uitvoeren van deze bewerking zal een deel van de variatie binnen de invoergegevens niet passen binnen het seizoensgebonden of trendgebonden component en als overig component worden aangeduid. De diagramfunctie **STL_Residual** registreert dit deel van de berekening.

```
STL_Residual - diagramfunctie(target_measure, period_int [,seasonal_smoother
[,trend_smoother]])
```

## Rank - diagramfunctie

**Rank()** evalueert de rijen van het diagram in de uitdrukking en geeft voor elke rij de relatieve positie van de waarde van de dimensie die in de uitdrukking wordt geëvalueerd weer. Bij de evaluatie van de uitdrukking vergelijkt de functie het resultaat met het resultaat van de andere rijen die het huidige kolomsegment bevatten, en retourneert de rangschikking van de huidige rij in het segment.




*Kolomsegmenten*

Voor andere diagrammen dan tabellen wordt het huidige kolomsegment gedefinieerd zoals deze wordt weergegeven in het equivalent van de strakke tabel.

### Syntaxis:

```
Rank([TOTAL] expr[, mode[, fmt]])
```

**Retourgegevenstypen:** dubbele waarde

### Argumenten:

Argumenten

| Argument | Beschrijving |
|---|---|
| expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| mode | Geeft de numerieke representatie aan van het resultaat van de functie. |
| fmt | Geeft de tekstrepresentatie aan van het resultaat van de functie. |
| TOTAL | Als het diagram eendimensionaal is of als de uitdrukking wordt voorafgegaan door de kwalificatie **TOTAL**, wordt de functie geëvalueerd voor de hele kolom. Als de tabel of het equivalent van de tabel meerdere verticale dimensies heeft, omvat het huidige kolomsegment alleen rijen met dezelfde waarden als de huidige rij in alle dimensiekolommen, met uitzondering van de kolom waarin de laatste dimensie wordt weergegeven in de onderlinge sorteervolgorde van de velden. |

De rangorde wordt geretourneerd als een dubbele waarde. Als elke rij een unieke rangschikking heeft, is dit een geheel getal tussen 1 en het aantal rijen in het huidige kolomsegment.

Als meerdere rijen dezelfde rangschikking hebben, kan de tekstuele en numerieke representatie worden bestuurd met de parameters **mode** en **fmt**.

**mode**

Het tweede argument, **mode**, kan de volgende waarden hebben:

**mode**-voorbeelden

| Waarde | Beschrijving |
|---|---|
| 0 (standaard) | Als alle rangnummers in de delende groep lager zijn dan de middenwaarde van de volledige rangschikking, krijgen alle rijen het laagste positienummer in de delende groep.<br><br>Als alle rangnummers in de delende groep hoger zijn dan de middenwaarde van de volledige rangschikking, krijgen alle rijen het hoogste rangnummer in de delende groep.<br><br>Als rangnummers in de delende groep zowel lager als hoger zijn dan de middenwaarde van de volledige rangschikking, krijgen alle rijen de waarde die overeenkomt met het gemiddelde van het hoogste en laagste rangnummer in het volledige kolomsegment. |
| 1 | Laagste rangnummer in alle rijen. |
| 2 | Gemiddelde rangnummer in alle rijen. |
| 3 | Hoogste rangnummer in alle rijen. |
| 4 | Laagste rangnummer in eerste rij, vervolgens opgehoogd met één voor elke rij. |

**fmt**

Het derde argument, **fmt**, kan de volgende waarden hebben:

**fmt**-voorbeelden

| Waarde | Beschrijving |
|---|---|
| 0 (standaard) | Lage waarde **-** hoge waarde in alle rijen (bijvoorbeeld 3 - 4). |
| 1 | Lage waarde in alle rijen. |
| 2 | Lage waarde in eerste rij, leeg in alle volgende rijen. |

De volgorde van rijen voor **mode** 4 en **fmt** 2 wordt bepaald door de sorteervolgorde van de diagramdimensies.

**Voorbeelden en resultaten:**

Maak twee visualisaties op basis van de dimensies Product en Sales en nog één op basis van Product en UnitSales. Voeg metingen toe zoals weergegeven in de volgende tabel.

Voorbeelden van rangorde

| Voorbeelden | Resultaten |
|---|---|
| Voorbeeld 1. Maak een tabel met de dimensies `Customer` en `Sales` en de meting `Rank(Sales)` | Het resultaat is afhankelijk van de sorteervolgorde van de dimensies. Als de tabel wordt gesorteerd op Customer, bevat de tabel alle waarden van Sales voor Astrida, daarna Betacab enzovoort. De resultaten voor Rank(Sales) geven 10 aan voor de waarde 12 van Sales, 9 voor de waarde 13 van Sales enzovoort, waarbij de rangordewaarde 1 wordt geretourneerd voor de waarde 78 van Sales. Het volgende kolomsegment begint met Betacab, waarvoor de eerste waarde van Sales in het segment 12 bedraagt. De rangordewaarde van Rank(Sales) hiervoor is gegeven als 11.<br><br>Als de tabel wordt gesorteerd op Sales, bestaan de kolomsegmenten uit de waarden van Sales en de bijbehorende Customer. Aangezien er twee waarden 12 zijn voor Sales (voor Astrida en Betacab), is de waarde van Rank (Sales) voor dat kolomsegment 1-2, voor elke waarde van Customer. Dit komt doordat er twee waarden van Customer zijn voor de waarde 12 van Sales. Als er 4 waarden waren geweest, zou het resultaat 1-4 zijn, voor alle rijen. Dit geeft aan hoe het resultaat eruitziet voor de standaardwaarde (0) van het argument fmt. |
| Voorbeeld 2. Vervang de dimensie Customer door Product en voeg de meting toe `Rank(Sales,1,2)` | Dit geeft 1 als resultaat in de eerste rij van elk kolomsegment en laat alle andere rijen leeg omdat de argumenten **mode** en **fmt** zijn ingesteld op respectievelijk 1 en 2. |

Resultaten voor voorbeeld 1, waarbij de tabel is gesorteerd op Customer:

Resultatentabel

| Customer | Sales | Rank(Sales) |
|---|---|---|
| Astrida | 12 | 10 |
| Astrida | 13 | 9 |
| Astrida | 20 | 8 |
| Astrida | 22 | 7 |
| Astrida | 45 | 6 |

| Customer | Sales | Rank(Sales) |
|---|---|---|
| Astrida | 46 | 5 |
| Astrida | 60 | 4 |
| Astrida | 65 | 3 |
| Astrida | 70 | 2 |
| Astrida | 78 | 1 |
| Betcab | 12 | 11 |

Resultaten voor voorbeeld 1, waarbij de tabel is gesorteerd op Sales:

Resultatentabel

| 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 |

Gegevens die worden gebruikt in voorbeelden:

```
ProductData:

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|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 '|');
```

**Zie ook:**

p *Sum - diagramfunctie (page 394)*

## HRank - diagramfunctie

**HRank()** evalueert de uitdrukking en vergelijkt het resultaat met het resultaat van de andere kolommen die het huidige rijsegment van een draaitabel bevatten. De functie retourneert vervolgens de rangschikking van de huidige kolom binnen het segment.

**Syntaxis:**

```
HRank([ TOTAL ] expr [ , mode [, fmt ] ])
```

**Retourgegevenstypen:** dual

> *Deze functie werkt alleen in draaitabellen. In alle andere diagramtypen levert de functie NULL op.*

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| expr | De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. |
| mode | Geeft de numerieke representatie aan van het resultaat van de functie. |
| fmt | Geeft de tekstrepresentatie aan van het resultaat van de functie. |

| Argument | Beschrijving |
|---|---|
| TOTAL | Als het diagram eendimensionaal is of als de uitdrukking wordt voorafgegaan door de kwalificatie **TOTAL**, wordt de functie geëvalueerd voor de hele kolom. Als de tabel of het equivalent van de tabel meerdere verticale dimensies heeft, omvat het huidige kolomsegment alleen rijen met dezelfde waarden als de huidige rij in alle dimensiekolommen, met uitzondering van de kolom waarin de laatste dimensie wordt weergegeven in de onderlinge sorteervolgorde van de velden. |

Als de draaitabel eendimensionaal is of als de uitdrukking wordt voorafgegaan door de kwalificatie **total**, is het huidige rijsegment altijd gelijk aan de hele rij. Als de draaitabel meerdere horizontale dimensies heeft, omvat het huidige rijsegment alleen kolommen met dezelfde waarden als de huidige kolom in alle dimensierijen, met uitzondering van de rij waarin de laatste horizontale dimensie wordt weergegeven in de onderlinge sorteervolgorde van de velden.

De rangschikking wordt geretourneerd als een dubbele waarde. Als elke kolom een unieke rangschikking heeft, is dit een heel getal tussen 1 en het aantal kolommen in het huidige rijsegment.

Als meerdere kolommen dezelfde rangschikking hebben, kan de tekstuele en numerieke representatie worden bestuurd met de parameters **mode** en **format**.

Het tweede argument, **mode**, geeft de numerieke representatie aan van het resultaat van de functie:

**mode**-voorbeelden

| Waarde | Beschrijving |
|---|---|
| 0 (standaard) | Als alle rangnummers in de delende groep lager zijn dan de middenwaarde van de volledige rangschikking, krijgen alle kolommen het laagste positienummer in de delende groep.<br><br>Als alle rangnummers in de delende groep hoger zijn dan de middenwaarde van de volledige rangschikking, krijgen alle kolommen het hoogste rangnummer in de delende groep.<br><br>Als rangnummers in de delende groep zowel lager als hoger zijn dan de middenwaarde van de volledige rangschikking, krijgen alle rijen de waarde die overeenkomt met het gemiddelde van het hoogste en laagste rangnummer in het volledige kolomsegment. |
| 1 | Laagste positienummer van alle kolommen in de groep. |
| 2 | Gemiddelde positienummer van alle kolommen in de groep. |
| 3 | Hoogste positienummer van alle kolommen in de groep. |
| 4 | Laagste positienummer in eerste kolom, vervolgens opgehoogd met één voor elke kolom in de groep. |

Het derde argument, **format**, geeft de tekstrepresentatie aan van het resultaat van de functie:

**format**-voorbeelden

| Waarde | Beschrijving |
|---|---|
| 0 (standaard) | Lage waarde**&' - '&**hoge waarde in alle kolommen (bijv. 3 - 4). |
| 1 | Laagste waarde van alle kolommen in de groep. |
| 2 | Laagste waarde in de eerste kolom, leeg in alle volgende kolommen in de groep. |

De volgorde van kolommen voor **mode** 4 en **format** 2 wordt bepaald door de sorteervolgorde van de diagramdimensies.

**Voorbeelden:**

```
HRank( sum( Sales ))

HRank( sum( Sales ), 2 )

HRank( sum( Sales ), 0, 1 )
```

## Optimaliseren met k-means: een voorbeeld uit de praktijk

In het volgende voorbeeld wordt een voorbeeld uit de praktijk getoond waarbij functies voor K-means en middenpunten op een gegevensverzameling zijn toegepast. Met behulp van de functie K-means worden gegevenspunten gescheiden in clusters met dezelfde eigenschappen. De clusters worden compacter en verder uitgesplitst als het K-means-algoritme is toegepast op een configureerbaar aantal iteraties.

K-means wordt gebruikt in veel sectoren en gebruikscases. Enkele voorbeelden van gebruikscases waarbij clustering is toegepast zijn klantsegmentatie, fraudedetectie, het voorspellen van klantverloop, het afstemmen van stimuleringen voor klanten, het identificeren van cybercriminaliteit en optimalisatie van bezorgroutes. Het K-means clustering algoritme wordt steeds meer gebruikt door ondernemingen die patronen willen herkennen en hun diensten willen optimaliseren.

### Qlik Sense Functies voor K-means en middenpunten

Qlik Sense biedt twee functies voor K-means waarmee gegevenspunten op basis van overeenkomstigheid in clusters worden gegroepeerd. Zie *KMeans2D - diagramfunctie (page 1660)* en *KMeansND - diagramfunctie (page 1675)*. De functie **KMeans2D** accepteert twee dimensies en werkt goed voor het visualiseren van gegevens met behulp van een **spreidingsplot**. De functie **KMeansND** accepteert meer dan twee dimensies. Het is vrij eenvoudig om een 2D-resultaat op standaarddiagrammen toe te passen. In het volgende voorbeeld wordt K-means toegepast op een **spreidingsplot** met behulp van twee dimensies. K-means clustering kan worden gevisualiseerd door middel van een uitdrukking of een dimensie zoals in dit voorbeeld wordt getoond.

Middenpuntfuncties in Qlik Sense bepalen het rekenkundig gemiddelde van alle gegevenspunten in het cluster en identificeren een centraal punt, oftewel een middenpunt voor dat cluster. Voor elke diagramrij (of record) toont de middenpuntfunctie de coördinaat van het cluster waaraan dit gegevenspunt is toegewezen. Zie *KMeansCentroid2D - diagramfunctie (page 1690)* en *KMeansCentroidND - diagramfunctie (page 1691)*.

## Overzicht gebruikscase en voorbeeld

In het volgende voorbeeld wordt u door een gesimuleerd scenario uit de praktijk geleid. Een textielbedrijf in de staat New York, in de VS, moet de kosten verlagen door de leveringskosten te minimaliseren. Ze kunnen dit doen door magazijnen te verplaatsen naar locaties dichter bij hun distributeurs. Het bedrijf maakt gebruik van 118 distributeurs binnen de staat New York. In het volgende voorbeeld wordt gesimuleerd hoe een operationeel manager de distributeurs kan opdelen in vijf regio's met behulp van de functie K-means en vervolgens vijf optimale magazijnlocaties kan bepalen met behulp van de middenpuntfunctie. Het doel is om coördinaten te ontdekken die gebruikt kunnen worden om de vijf centrale magazijnlocaties te identificeren.

## De gegevensverzameling

De gegevensverzameling is gebaseerd op willekeurig gegenereerde namen en adressen in de staat New York met nauwkeurige lengte- en breedtecoördinaten. De dataset bevat de volgende tien kolommen: id, voornaam, achternaam, telefoon, adres, stad, staat, postcode, breedtegraad, lengtegraad. De gegevensverzameling is hieronder beschikbaar als een bestand dat u lokaal kunt downloaden en vervolgens kunt uploaden naar Qlik Sense of uitgelijnd voor de editor voor laden van gegevens. De app die wordt aangemaakt heet *Distributeurs K-means en middenpunt* en het eerste werkblad in de app heet *Clusteranalyse distributie*.

Selecteer de volgende link om het bestand met voorbeeldgegevens te downloaden: DistributorData.csv

*Distributor-gegevensverzameling: inline lading voor de editor voor het laden van gegevens in Qlik Sense (page 1658)*

Titel: DistributorData

Totaal aantal records: 118

## De KMeans2D-functie toepassen

In dit voorbeeld wordt de configuratie van een diagram van een **spreidingsplot** weergegeven met behulp van de gegevensverzameling *DistributorData*, de **KMeans2D**-functie is toegepast en het diagram is gekleurd op basis van dimensie.

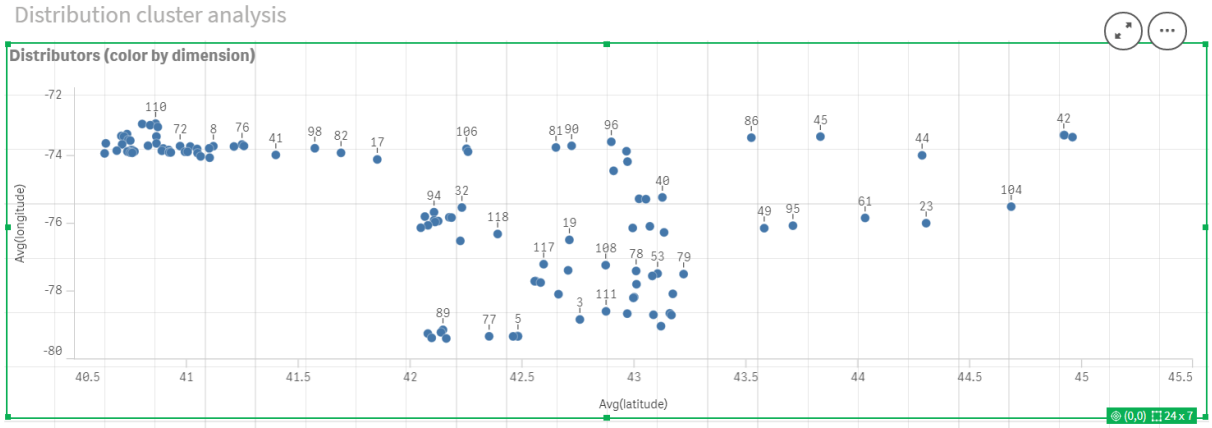
Houd er rekening mee dat de Qlik Sense K-meansfunctie ondersteuning biedt voor automatische clustering met behulp van de methode diepteverschil (DeD). Als een gebruiker het aantal clusters instelt op 0, wordt een optimaal aantal clusters voor die gegevensverzameling bepaalt. Voor dit voorbeeld is er een variabele gemaakt voor het argument **num_clusters** (raadpleeg *KMeans2D - diagramfunctie (page 1660)* voor de syntaxis). Het gewenste aantal clusters (k=5) wordt opgegeven per variabele.

1. Er wordt een **spreidingsplot**-diagram naar het werkblad gesleept. Dit wordt *Distributeurs (per dimensie)* genoemd.
2. Er wordt een **variabele** gemaakt om het aantal clusters op te geven. De **variabele** krijgt de naam *vDistClusters*. Voor de variabele wordt voor **Definitie** *=5* ingevoerd.
3. De **gegevens**configuratie voor het diagram:
    a. Onder **Dimensies** is *id* geselecteerd voor het veld voor **Bel**. *Cluster-id* is ingevoerd voor het **label**.

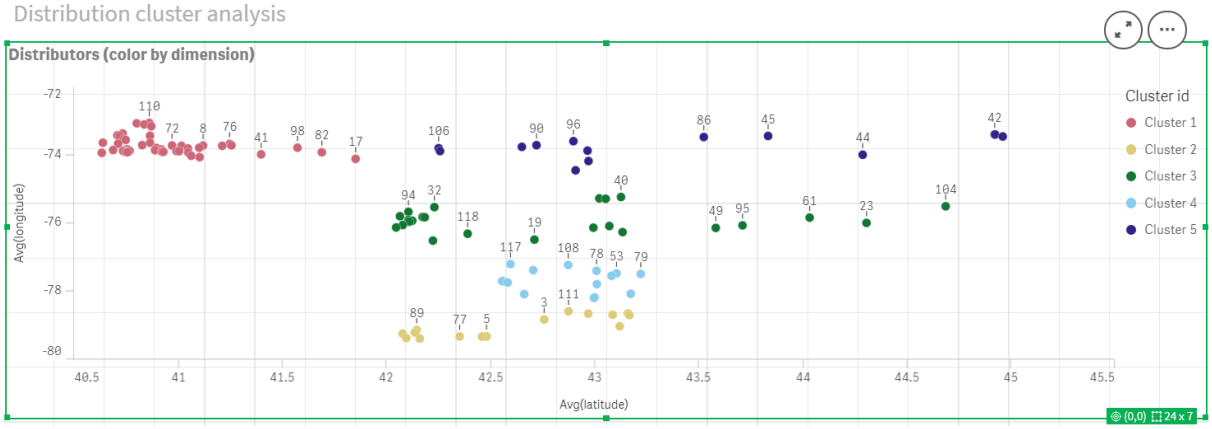
b. Onder **Metingen**, is *Avg([latitude])* de uitdrukking voor de **X-as**.
c. Onder **Metingen**, is *Avg([longitude])* de uitdrukking voor de **Y-as**.

4. Configuratie voor **Uiterlijk**:
   a. Onder **Kleuren en legenda** is **Aangepast** geselecteerd voor **Kleuren**.
   b. **Per dimensie** is geselecteerd voor het kleuren van het diagram.
   c. De volgende uitdrukking is ingevoerd: *=pick(aggr(KMeans2D(vDistClusters,only (latitude),only(longitude)),id)+1, 'Cluster 1', 'Cluster 2', 'Cluster 3', 'Cluster 4', 'Cluster 5')*
   d. Het selectievakje **Kleuren bij selectie behouden** is geselecteerd.

*Spreidingsplot vóór K-means kleuren per dimensie is toegepast*




*Spreidingsplot na K-means kleuren per dimensie is toegepast*




## Een tabel toevoegen: *Distributeurs*

Het kan handig zijn om een tabel bij de hand te hebben voor snelle toegang tot relevante gegevens. Het **spreidingsplot**-diagram toont weliswaar de *id's*, maar ter referentie is een tabel met de namen van de overeenkomstige distributeurs toegevoegd.

1. Er wordt een **strakke tabel** met de naam *Distributeurs* naar het werkblad gesleept, waaraan de volgende **Kolommen** (dimensies) zijn toegevoegd: *id*, *first_name* en *last_name*.

*Tabel: Namen distributeurs*

**Distributors**

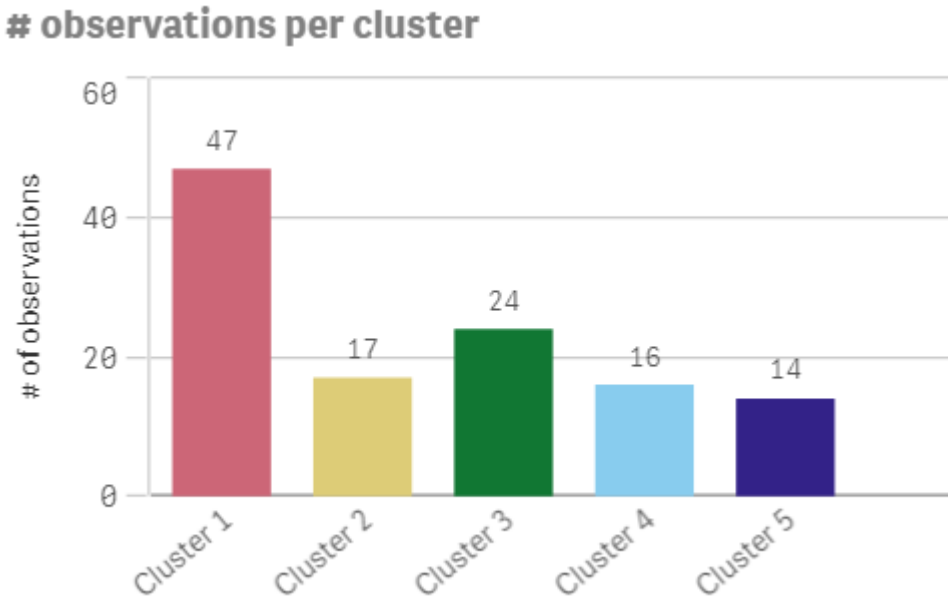
| id | first_name | last_name |
|---|---|---|
| 1 | Kaiya | Snow |
| 2 | Dean | Roy |
| 3 | Eden | Paul |
| 4 | Bryanna | Higgins |
| 5 | Elisabeth | Lee |
| 6 | Skylar | Robinson |
| 7 | Cody | Bailey |
| 8 | Dario | Sims |
| 9 | Deacon | Hood |

## Toevoegen van een **staafdiagram**: *# observaties per cluster*

Voor het scenario magazijndistributie is het nuttig te weten hoeveel distributeurs door elk magazijn zullen worden bediend. Daarom wordt er een **staafdiagram** aangemaakt, dat meet hoeveel distributeurs aan elk cluster zijn toegewezen.

1. Er wordt een **staafdiagram** naar het werkblad gesleept. Het diagram krijgt de naam: *# observaties per cluster*.
2. **Gegevens**configuratie voor het **staafdiagram**:
    a. Er wordt een **dimensie** met het label *Clusters* toegevoegd (het label kan worden toegevoegd nadat de uitdrukking is toegepast). De volgende uitdrukking wordt ingevoerd: *=pick(aggr(KMeans2D(vDistClusters,only(latitude),only(longitude)),id)+1, 'Cluster 1', 'Cluster 2', 'Cluster 3', 'Cluster 4', 'Cluster 5')*
    b. De **Meting** met het label *# observaties* wordt toegevoegd. De volgende uitdrukking wordt ingevoerd: *=count(aggr(KMeans2D(vDistClusters,only(latitude),only(longitude)),id))*
3. Configuratie voor **Uiterlijk**:
    a. Onder **Kleuren en legenda** is **Aangepast** geselecteerd voor **Kleuren**.
    b. **Per dimensie** is geselecteerd voor het kleuren van het diagram.
    c. De volgende uitdrukking wordt ingevoerd: *=pick(aggr(KMeans2D(vDistClusters,only(latitude),only(longitude)),id)+1, 'Cluster 1', 'Cluster 2', 'Cluster 3', 'Cluster 4', 'Cluster 5')*
    d. Het selectievakje **Kleuren bij selectie behouden** is geselecteerd.
    e. **Legenda tonen** is uitgeschakeld.
    f. Onder **Presentatie** staat **Waardelabels** ingesteld op **Auto**.
    g. Onder **X-as**: **Clusters** is **Alleen labels** geselecteerd.

*Staafdiagram: # observaties per cluster*




## Toepassen van de **Centroid2D**-functie

Er wordt een tweede tabel toegevoegd voor de **Centroid2D**-functie waarmee de coördinaten voor potentiële magazijnlocaties worden bepaald. Deze tabel toont de centrale locatie (middenpuntwaarden) voor de vijf geïdentificeerde distributeurgroepen.

1. Sleep een **Tabel** naar het werkblad, noem het *Cluster middenpunten* en voeg de volgende kolommen toe:
    a. Er wordt een **dimensie** met het label *Clusters* toegevoegd. De volgende uitdrukking wordt ingevoerd: *=pick(aggr(KMeans2D(vDistClusters,only(latitude),only (longitude)),id)+1,'Warehouse 1','Warehouse 2','Warehouse 3','Warehouse 4','Warehouse 5'*)
    b. Er wordt een **Meting** met het label *breedtegraad (D1)* toegevoegd. De volgende uitdrukking wordt ingevoerd:*=only(aggr(KMeansCentroid2D(vDistClusters,0,only(latitude),only (longitude)),id))*
    Let op: de parameter **coordinate_no** komt overeen met de eerste dimensie(0). In dit geval wordt de dimensie *breedtegraad* uitgezet tegen de x-as. Als we met de **CentroidND**-functie zouden werken en er zouden tot zes dimensies zijn, dan zouden deze parametergegevens om het even welke van zes waarden kunnen zijn: 0, 1, 2, 3, 4 of 5.
    c. Er wordt een **Meting** met het label *lengtegraad (D2)* toegevoegd. De volgende uitdrukking wordt ingevoerd: *=only(aggr(KMeansCentroid2D(vDistClusters,1,only(latitude),only (longitude)),id))*
    De parameter **coordinate_no** in deze uitdrukking komt overeen met de tweede dimensie(1). De dimensie *lengtegraad* wordt uitgezet tegen de y-as.

*Tabel: Berekeningen middenpuntcluster*

**Cluster centroids**

| Clusters | latitude (D1) | longitude (D2) |
|---|---|---|
| **Totals** | - | - |
| 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 |

## Toewijzen van middenpunten

De volgende stap bestaat uit het toewijzen van de middenpunten. Het is aan de ontwikkelaar van de app of hij de visualisatie op afzonderlijke werkbladen wenst te plaatsen.

1. Er wordt een **kaart** met de naam *Toewijzen van middenpunten* naar het werkblad gesleept.
2. In de sectie **Lagen**. **Laag toevoegen** wordt geselecteerd, vervolgens wordt **Puntlaag** geselecteerd.
    a. Het **Veld** *id* wordt geselecteerd en het **Label** voor de *Dist id's* wordt toegevoegd.
    b. In de sectie **Locatie** wordt het selectievakje voor de **velden Lengte- en breedtegraad** geselecteerd.
    c. Voor **Breedtegraad** wordt het veld *breedtegraad* geselecteerd.
    d. Voor **Lengtegraad** wordt het veld *lengtegraad* geselecteerd.
    e. In de sectie **Grootte en vorm** wordt **Bel** geselecteerd voor **Vorm** en de **Grootte** wordt naar wens verkleind met de schuif.
    f. In de sectie **Kleuren** wordt **Enkele kleur** geselecteerd en blauw wordt geselecteerd voor de **Kleur** en grijs voor de **Omtrek** (ook deze keuzes mag u zelf bepalen).
3. In de sectie **Lagen** wordt een tweede **Puntlaag** toegevoegd door eerst **Laag toevoegen** en dan **Puntlaag** te selecteren.
    a. De volgende uitdrukking wordt ingevoerd: *=aggr(KMeans2D(vDistClusters,only (latitude),only(longitude)),id)*
    b. Het **Label** *Clusters* wordt toegevoegd.
    c. In de sectie **Locatie** wordt het selectievakje voor de **velden Lengte- en breedtegraad** geselecteerd.
    d. Voor de **Breedtegraad**, die in dit geval tegen de x-as wordt uitgezet, wordt de volgende uitdrukking toegevoegd: *=aggr(KMeansCentroid2D(vDistClusters,0,only(latitude),only (longitude)),id)*
    e. Voor de **Lengtegraad** die in dit geval langs de y-as wordt uitgezet, wordt de volgende uitdrukking toegevoegd: *=aggr(KMeansCentroid2D(vDistClusters,1,only(latitude),only (longitude)),id)*

f. In de sectie **Grootte en vorm** wordt **Driehoek** geselecteerd voor **Vorm** en de **Grootte** wordt naar wens verkleind met de schuif.

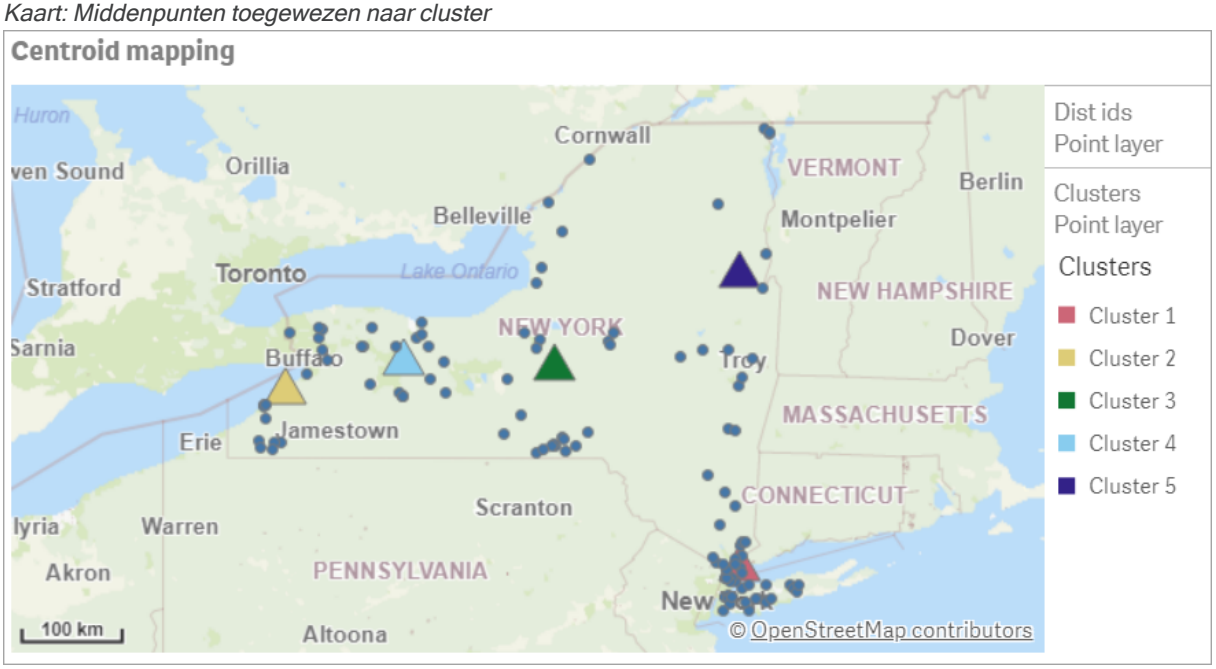
g. Onder **Kleuren en legenda** is **Aangepast** geselecteerd voor **Kleuren**.

h. **Per dimensie** is geselecteerd voor het kleuren van het diagram. De volgende uitdrukking wordt ingevoerd: *=pick(aggr(KMeans2D(vDistClusters,only(latitude),only(longitude)),id)+1, 'Cluster 1', 'Cluster 2', 'Cluster 3', 'Cluster 4', 'Cluster 5')*

i. De dimensie krijgt het label *Clusters*.

4. In **Kaartinstellingen** wordt **Adaptief** geselecteerd voor **Projectie**. **Metrisch** wordt geselecteerd voor **Meeteenheden.**

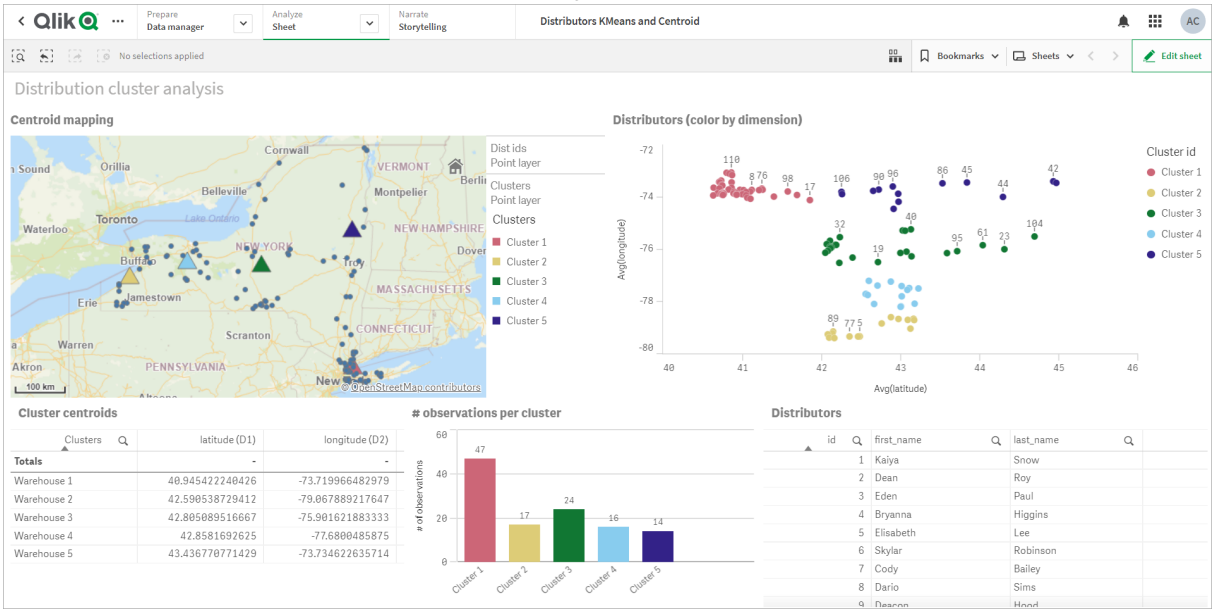
*Kaart: Middenpunten toegewezen naar cluster*




## Conclusie

Met behulp van de K-means-functie zijn in dit praktijkscenario distributeurs gesegmenteerd in soortgelijke groepen of clusters op basis van hun overeenkomsten; in dit geval de mate van onderlinge nabijheid. Op deze clusters is de middenpuntfunctie toegepast om vijf kaartcoördinaten te bepalen. Deze coördinaten geven een eerste centrale locatie aan voor de bouw of de vestiging van magazijnen. De middenpuntfunctie wordt toegepast op het **kaart**diagram, zodat app-gebruikers kunnen zien waar de middenpunten zich ten opzichte van de omliggende clustergegevenspunten bevinden. De resulterende coördinaten staan voor potentiële locaties van magazijnen die ervoor kunnen zorgen dat de leveringskosten voor distributeurs in de staat New York kunnen worden gedrukt.

*App: Voorbeeld van een K-means- en middenpuntanalyse*




**Distributor-gegevensverzameling: inline lading voor de editor voor het laden van gegevens 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,Cooley,(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,Brooklynn,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,Baldinsville,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,Kennedi,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 - diagramfunctie

**KMeans2D()** evalueert de rijen van het diagram door K-means clustering toe te passen. Voor elke rij van het diagram wordt de cluster-id weergegeven van het cluster waaraan dit gegevenspunt is toegewezen. De kolommen die worden gebruikt door het clustering-algoritme worden bepaald door de parameter

coordinate_1 respectievelijk coordinate_2. Dit zijn beide aggregraties. De parameter num_clusters bepaalt het aantal clusters dat wordt gemaakt. Gegevens kunnen optioneel worden genormaliseerd door de normparameter.

**KMeans2D** retourneert één waarde per gegevenspunt. De geretourneerde waarde is een dubbele waarde en is het gehele getal dat overeenkomt met het cluster waaraan elk gegevenspunt is toegewezen.

**Syntaxis:**

```
KMeans2D(num_clusters, coordinate_1, coordinate_2 [, norm])
```

**Retourgegevenstypen:** dubbele waarde

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| num_clusters | Het geheel getal dat het aantal clusters aangeeft. |
| coordinate_1 | De aggregatie die de eerste coördinaat berekent, meestal de x-as van de verdelingsplot die op basis van het diagram kan worden gemaakt. De aanvullende parameter, coordinate_2, berekent de tweede coördinaat. |
| norm | De optionele normalisatiemethode die is toegepast op gegevensverzamelingen voordat K-means clustering wordt toegepast.<br>Mogelijke waarden:<br>0 of 'geen' voor geen normalisatie<br>1 of 'zscore' for z-score normalisatie<br>2 of 'minmax' voor min-max normalisatie<br>Als er geen parameter is opgegeven of de opgegeven parameter is onjuist, wordt er geen normalisatie toegepast.<br>De Z-score normaliseert gegevens op basis van de gemiddelde- en standaardafwijking. De Z-score zorgt er niet voor dat elke functie dezelfde schaal toebedeeld krijgt, maar het is een betere manier om aan te gaan met uitschieters dan de min-max-methode.<br>De min-max normalisatie zorgt ervoor dat functies dezelfde schaal toebedeeld krijgen door gebruik te maken van de minimum- en maximumwaarden van elk gegevenspunt en elk gegevenspunt opnieuw te berekenen. |

Voorbeeld: diagramuitdrukking

In dit voorbeeld maken we een verdelingsplot met behulp van de gegevensset *Iris* en gebruiken vervolgens KMeans om de gegevens te kleuren op uitdrukking.

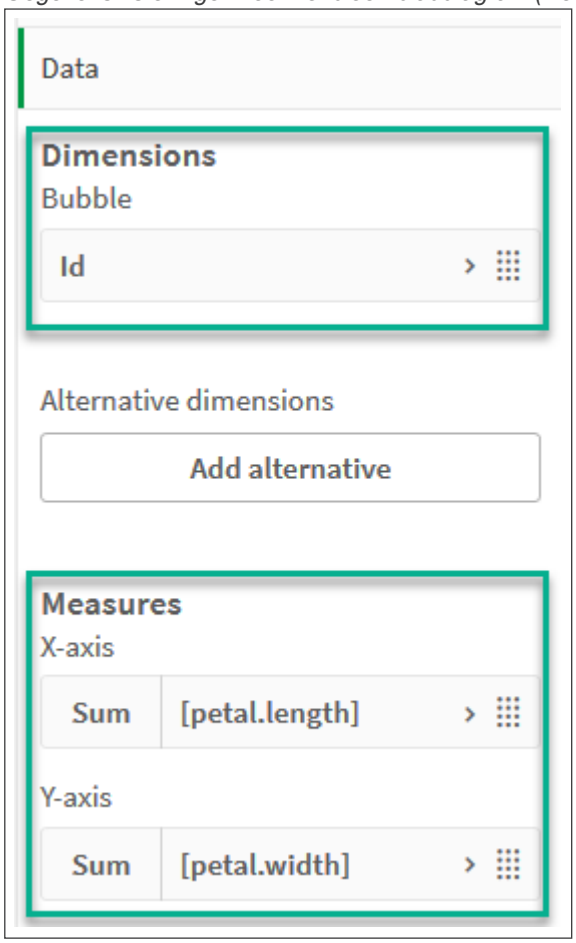
Bovendien maken we een variabele voor het argument *num_clusters* en gebruiken vervolgens een invoervak voor de variabele om het aantal clusters te wijzigen.

De gegevensset *Iris* is in verschillende indelingen openbaar toegankelijk. We hebben de gegevens ter beschikking gesteld als een uitgelijnde tabel die geladen kan worden met behulp van de editor voor laden van gegevens in Qlik Sense. Houd er rekening mee dat we in dit voorbeeld de kolom *id* hebben toegevoegd.

Nadat we de gegevens in Qlik Sense hebben geladen, gaan we als volgt te werk:

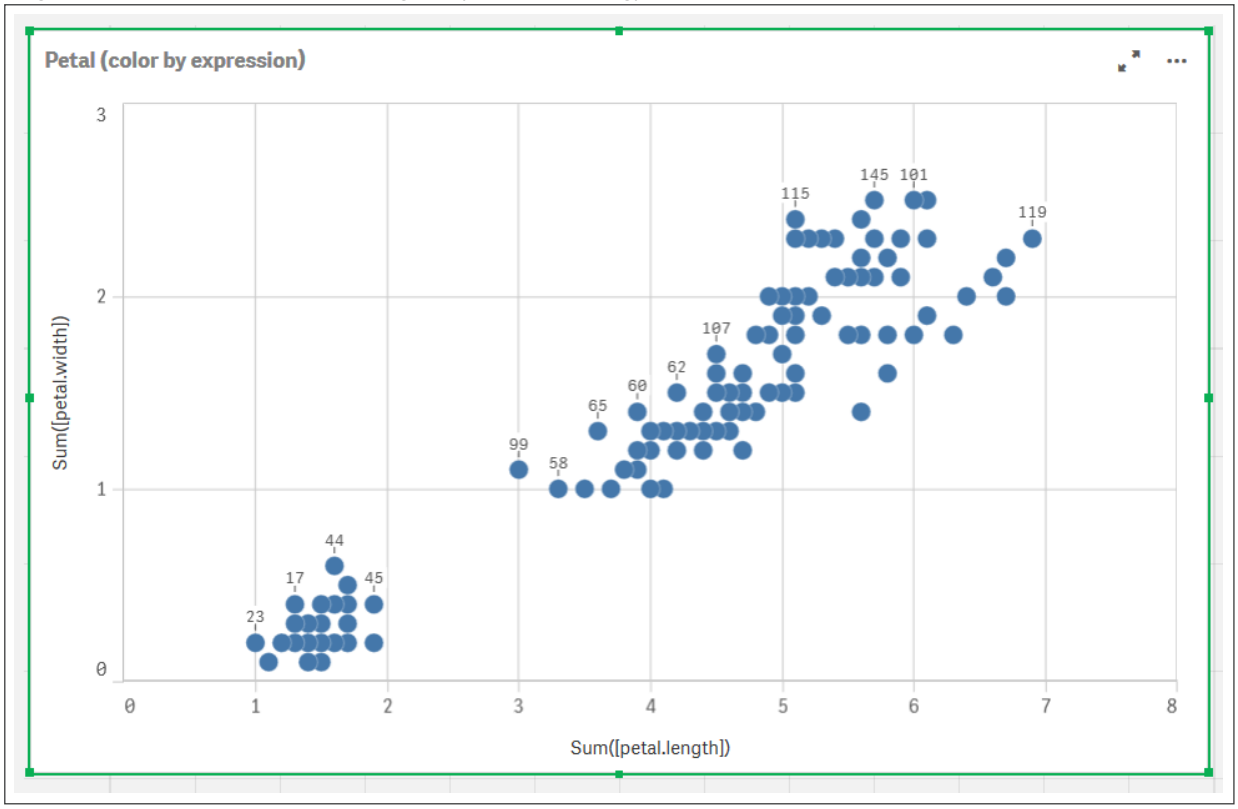
1. Sleep een **verdelingsplot** op een nieuw werkblad. Geef het *bloembladdiagram (kleuruitdrukking)* een naam.
2. Maak een variabele om het aantal clusters op te geven. Voor de variabele **Naam** voert u *KmeansPetalClusters* in. Voor de variabele **Definitie** voert u *=2* in.
3. Configureer **Uiterlijk** voor het diagram:
    i. Onder **Dimensies** kiest u *id* voor het veld voor **Bel**. Voer de cluster-id in voor het label.
    ii. Onder **Metingen** kiest u *Som([lengte.bloemenblad])* voor de uitdrukking voor **X-as**.
    iii. Onder **Metingen** kiest u *Som([breedte.bloemenblad])* voor de uitdrukking voor **Y-as**.

    *Gegevensinstellingen voor het bloembladdiagram (kleuruitdrukking)*




De gegevenspunten worden in het diagram geplot.

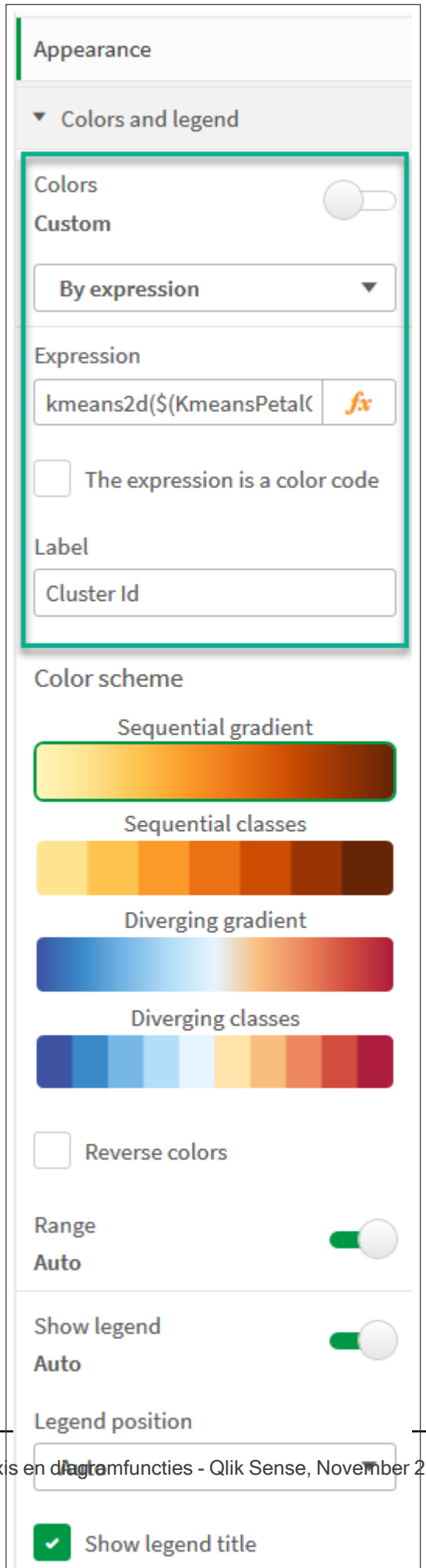
*Gegevenspunten in het bloembladdiagram (kleuruitdrukking)*




4. Configureer **Uiterlijk** voor het diagram:
    i. Onder **Kleuren en legenda** kiest u **Aangepast** voor **Kleuren**.
    ii. Kleur het diagram **Op uitdrukking**.
    iii. Voer het volgende in voor **Uitdrukking**: *kmeans2d($(KmeansPetalClusters), Sum ([petal.length]), Sum([petal.width]))*
    Houd er rekening mee dat *KmeansPetalClusters* de variabele is die we hebben ingesteld op *2*.
    U kunt ook het volgende invoeren: *kmeans2d(2, Sum([petal.length]), Sum([petal.width]))*
    iv. Schakel het selectievakje uit voor **De uitdrukking is een kleurcode**.

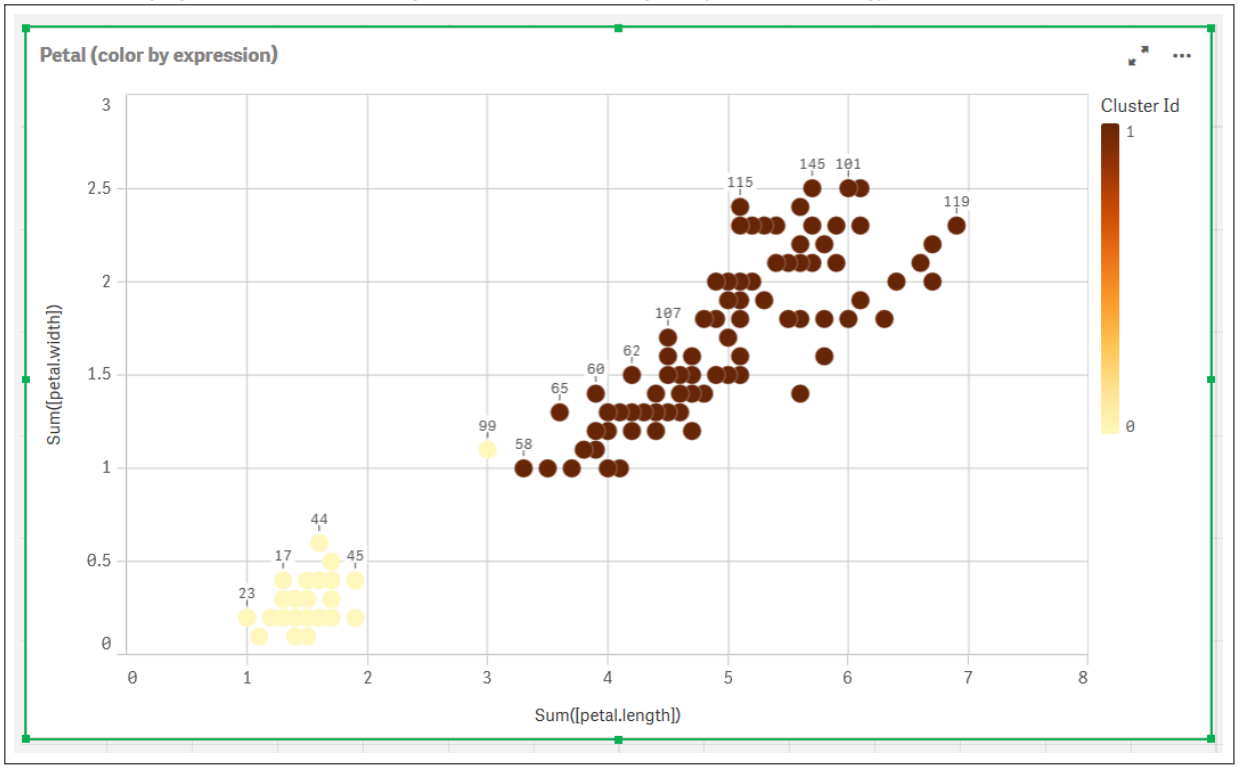
v. Voer het volgende in voor **Label**: *Cluster-ID*

*Instellingen voor het uiterlijk van het bloembladdiagram (kleuruitdrukking)*


Appearance
Colors and legend
Colors
Custom
By expression
Expression
kmeans2d($(KmeansPetal(
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
Auto
Show legend title

De twee clusters van het diagram worden gekleurd door de uitdrukking KMeans.

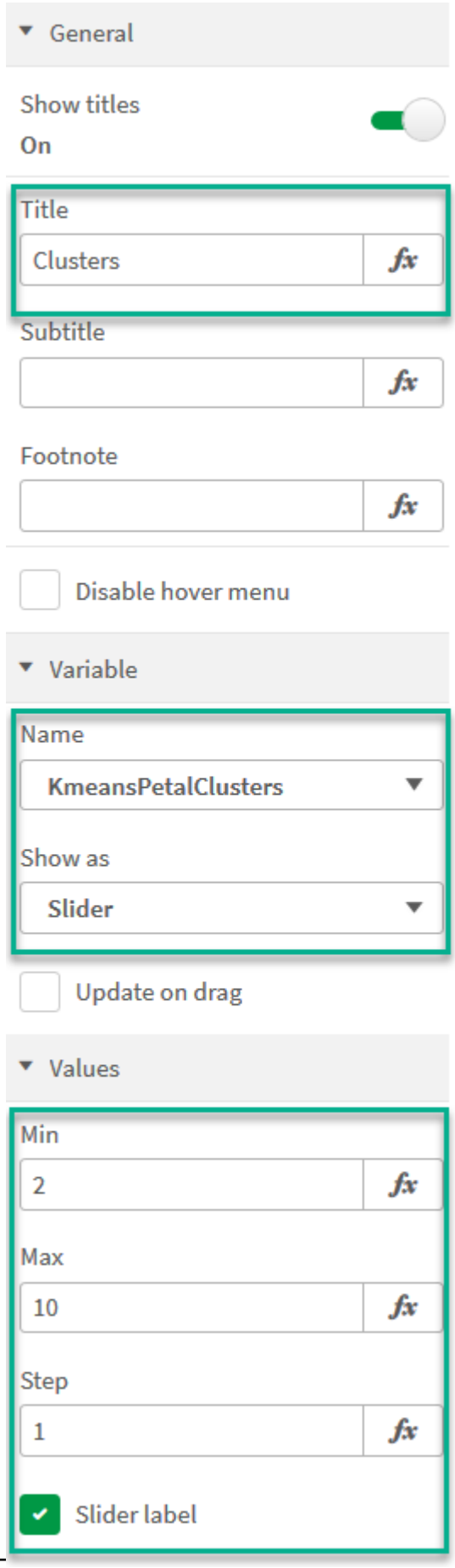
*Clusters die zijn gekleurd op uitdrukking in een bloembladdiagram (kleuruitdrukking)*




5. Voeg een **invoerveld voor variabelen** toe voor het aantal clusters.
    i. Kies **Qlik Dashboard-bundle** onder **Aangepaste objecten** in het **bedrijfsmiddelen**venster. Als we geen toegang hadden tot de dashboard-bundle, konden we nog steeds het aantal clusters wijzigen met behulp van de variabele die we aangemaakt hebben, of rechtstreeks als geheel getal in de uitdrukking.
    ii. Sleep een **invoervak voor variabelen** naar het werkblad.
    iii. Klik onder **Uiterlijk** op **Algemeen**.
    iv. Voer het volgende in voor **Titel**: *Clusters*
    v. Klik op **Variabele**.
    vi. Kies de volgende variabele voor **Naam**: *KmeansPetalClusters*.
    vii. Kies **Schuifregelaar** voor **Tonen als**.

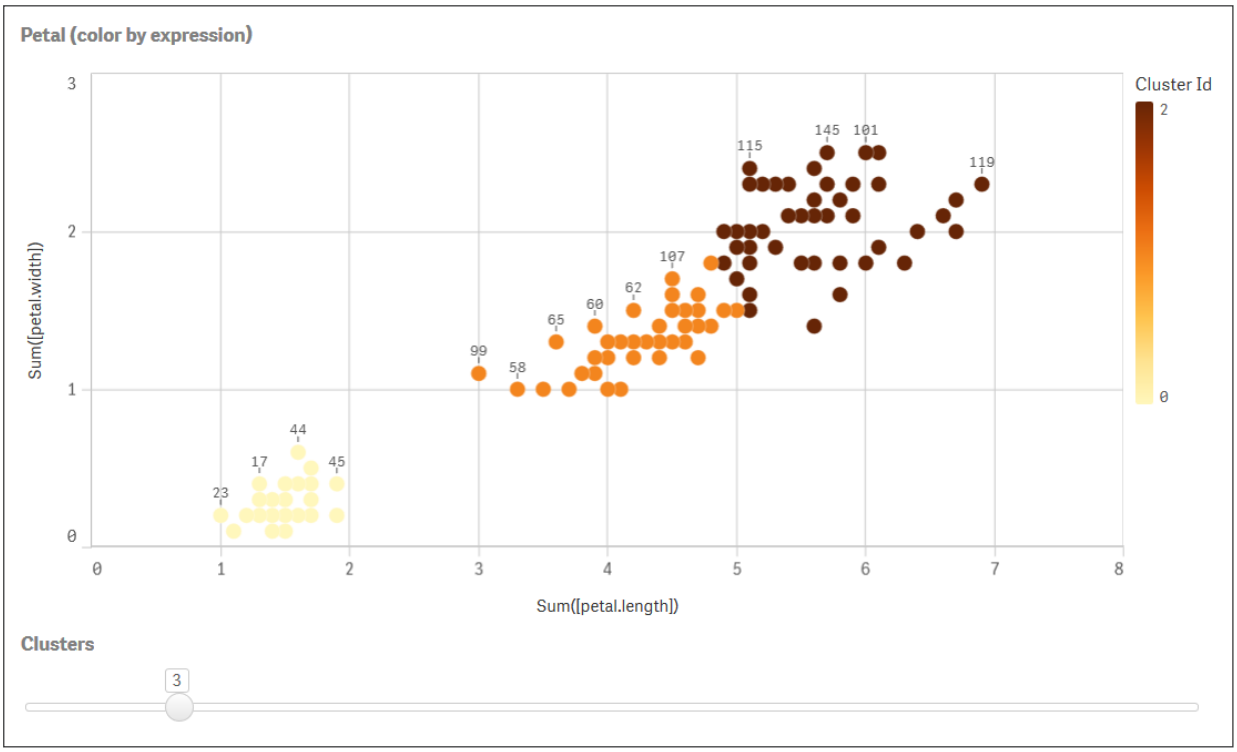
viii. Kies **Waarden** en configureer de instellingen zoals vereist.

*Uiterlijk voor het clusters invoervak voor variabelen*


General
Show titles
On
Title
Clusters
Subtitle
Footnote
Disable hover menu
Variable
Name
KmeansPetalClusters
Show as
Slider
Update on drag
Values
Min
2
Max
10
Step
1
Slider label

Als we klaar zijn met bewerken, kunnen we het aantal clusters wijzigen met behulp van de schuifregelaar in het invoerveld voor de *Clusters*-variabele.

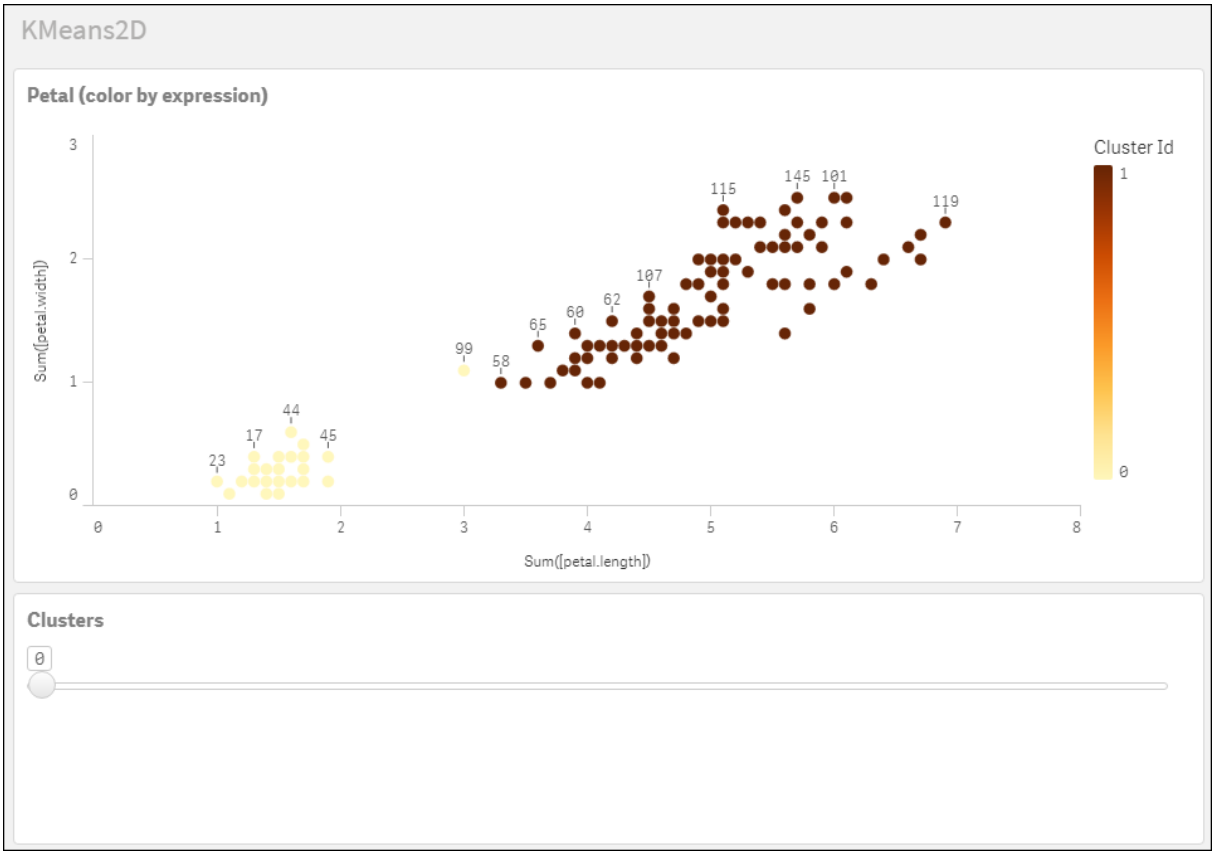
*Clusters die zijn gekleurd op uitdrukking in een bloembladdiagram (kleuruitdrukking)*




### Automatische clustering

De functies voor **KMeans** bieden ondersteuning voor automatische clustering met behulp van de methode diepteverschil (DeD). Als een gebruiker het aantal clusters instelt op 0, wordt een optimaal aantal clusters voor die gegevensverzameling bepaalt. Houd er rekening mee dat een geheel getal voor het aantal clusters (*k*) niet expliciet wordt geretourneerd, maar wel wordt berekend binnen het KMeans-algoritme. Als bijvoorbeeld 0 is opgegeven in de functie voor de waarde van *KmeansPetalClusters* of is ingesteld via een invoervak voor variabelen, worden clustertoewijzingen automatisch berekend voor de gegevensverzameling die is gebaseerd op een optimaal aantal clusters.

*De diepteverschilmethode KMeans depth bepaalt het optimale aantal clusters als (k) is ingesteld op 0*




**Iris gegevensverzameling: inline lading voor de editor voor het laden van gegevens 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 - diagramfunctie

**KMeansND()** evalueert de rijen van het diagram door K-means clustering toe te passen. Voor elke rij van het diagram wordt de cluster-id weergegeven van het cluster waaraan dit gegevenspunt is toegewezen. De kolommen die worden gebruikt door het clustering-algoritme worden bepaald door de parameter coordinate_1 en coordinate_2 etc. tot en met de n-kolommen Dit zijn beide aggregaties. De parameter num_clusters bepaalt het aantal clusters dat wordt gemaakt.

**KMeansND** retourneert één waarde per gegevenspunt. De geretourneerde waarde is een dubbele waarde en is het gehele getal dat overeenkomt met het cluster waaraan elk gegevenspunt is toegewezen.

**Syntaxis:**

```
KMeansND(num_clusters, num_iter, coordinate_1, coordinate_2 [,coordinate_3 [,
...]])
```

**Retourgegevenstypen:** dual

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| num_clusters | Het geheel getal dat het aantal clusters aangeeft. |
| num_iter | Het aantal iteraties voor clusters met opnieuw geïnitialiseerde middenpunten. |
| coordinate_1 | De aggregatie die de eerste coördinaat berekent, meestal de x-as (van de verdelingsplot die op basis van het diagram kan worden gemaakt). De aanvullende parameters berekenen de tweede, derde en vierde coördinaat, etc. |

Voorbeeld: Diagramuitdrukking

In dit voorbeeld maken we een verdelingsplot met behulp van de gegevensset *Iris* en gebruiken vervolgens KMeans om de gegevens te kleuren op uitdrukking.

Bovendien maken we een variabele voor het argument *num_clusters* en gebruiken vervolgens een invoervak voor de variabele om het aantal clusters te wijzigen.

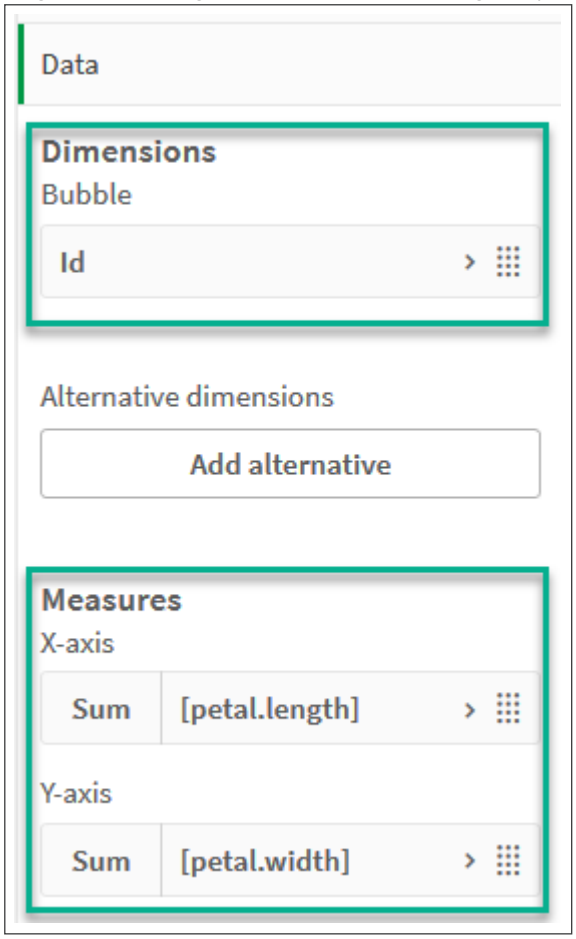
Bovendien maken we een variabele voor het argument *num_iter* en gebruiken vervolgens een tweede invoervak voor de variabele om het aantal iteraties te wijzigen.

De gegevensset *Iris* is in verschillende indelingen openbaar toegankelijk. We hebben de gegevens ter beschikking gesteld als een uitgelijnde tabel die geladen kan worden met behulp van de editor voor laden van gegevens in Qlik Sense. Houd er rekening mee dat we in dit voorbeeld de kolom *id* hebben toegevoegd.

Nadat we de gegevens in Qlik Sense hebben geladen, gaan we als volgt te werk:

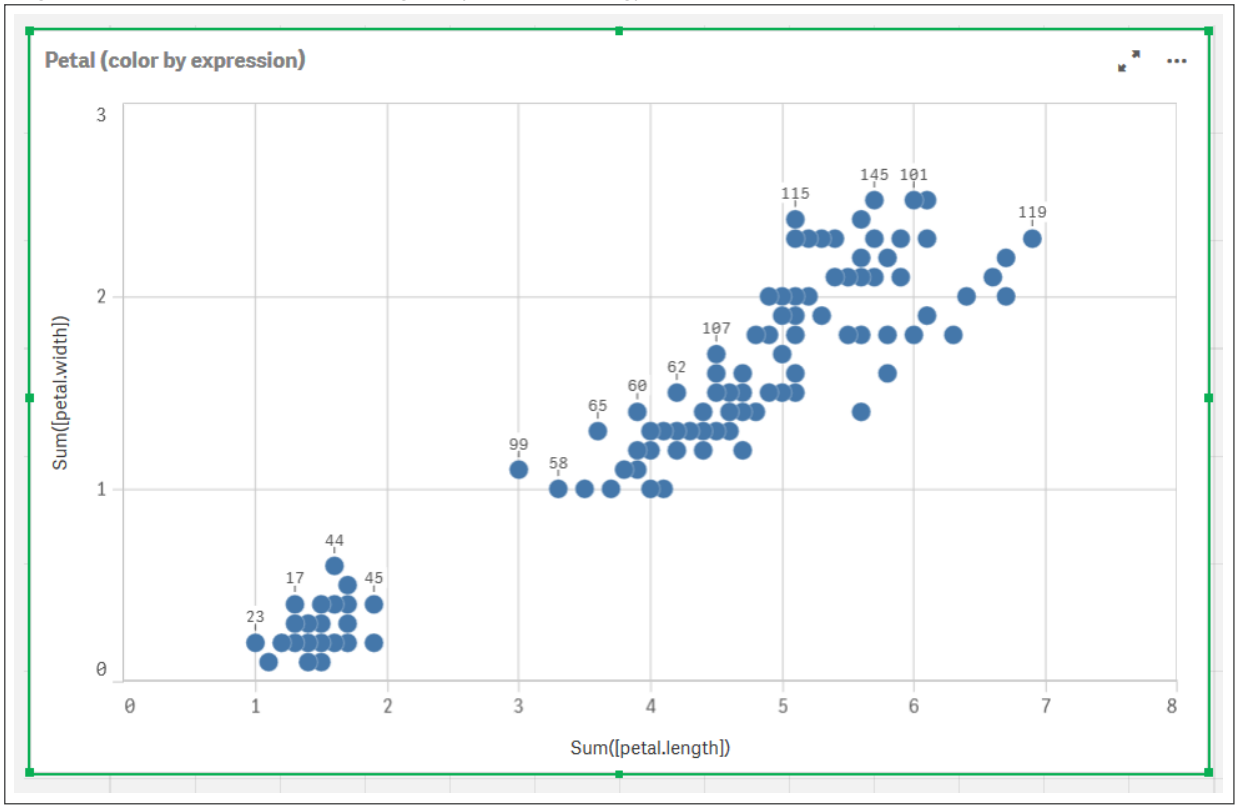
1. Sleep een **verdelingsplot** op een nieuw werkblad. Geef het *bloembladdiagram (kleuruitdrukking)* een naam.
2. Maak een variabele om het aantal clusters op te geven. Voor de variabele **Naam** voert u *KmeansPetalClusters* in. Voor de variabele **Definitie** voert u *=2* in.
3. Maak een variabele om het aantal iteraties op te geven. Voer voor de variabele **Naam** *KMeansNumberIterations* in. Voer voor de variabele **Definitie** *=1* in.
4. Configureer **Uiterlijk** voor het diagram:
    i. Onder **Dimensies** kiest u *id* voor het veld voor **Bel**. Voer de cluster-id in voor het label.
    ii. Onder **Metingen** kiest u *Som([lengte.bloemenblad])* voor de uitdrukking voor **X-as**.
    iii. Onder **Metingen** kiest u *Som([breedte.bloemenblad])* voor de uitdrukking voor **Y-as**.

*Gegevensinstellingen voor het bloembladdiagram (kleuruitdrukking)*




De gegevenspunten worden in het diagram geplot.

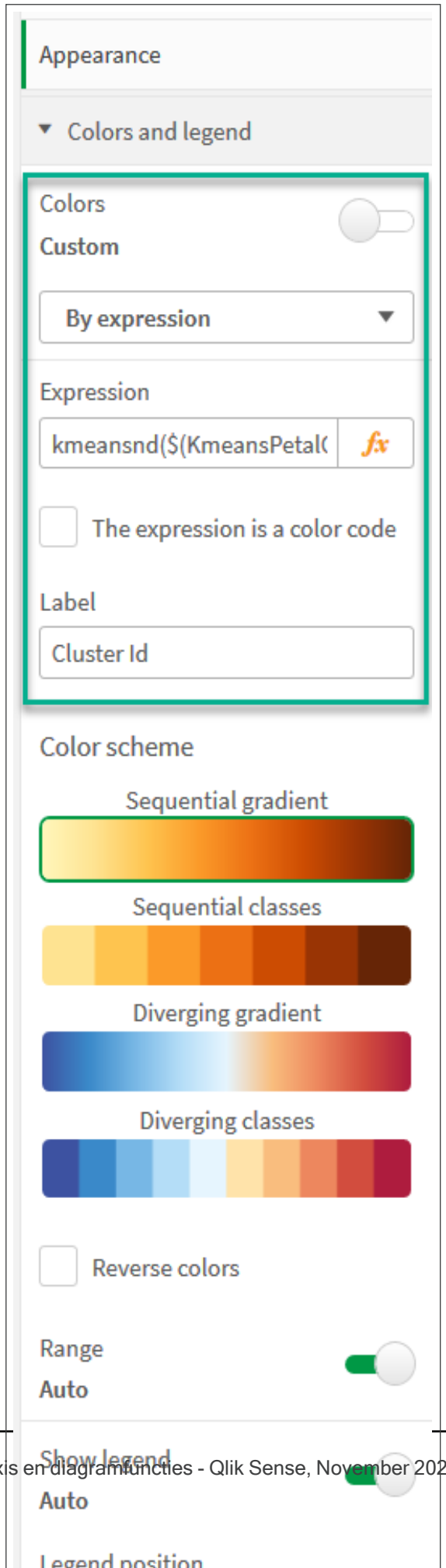
*Gegevenspunten in het bloembladdiagram (kleuruitdrukking)*




5. Configureer **Uiterlijk** voor het diagram:
    i. Onder **Kleuren en legenda** kiest u **Aangepast** voor **Kleuren**.
    ii. Kleur het diagram **Op uitdrukking**.
    iii. Voer het volgende in voor **Uitdrukking**: *kmeansnd ($(KmeansPetalClusters),$(KmeansNumberIterations), Sum([petal.length]), Sum ([petal.width]),Sum([sepal.length]), Sum([sepal.width]))*
    Houd er rekening mee dat *KmeansPetalClusters* de variabele is die we hebben ingesteld op *2*. We stellen de variabele *KmeansNumberIterations* in op *1*.
    U kunt ook het volgende invoeren: *kmeans2d(2, Sum([petal.length]), Sum([petal.width]))*
    iv. Schakel het selectievakje uit voor **De uitdrukking is een kleurcode**.

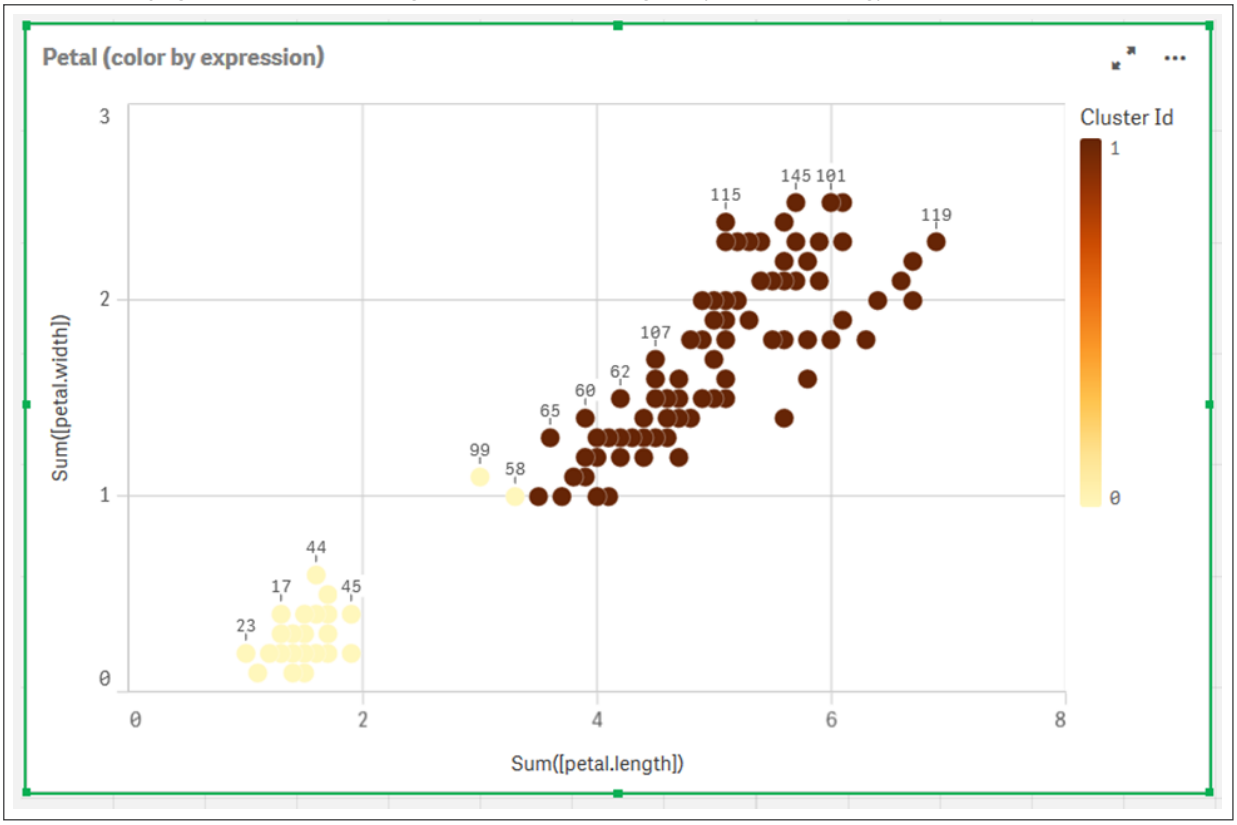
v. Voer het volgende in voor **Label**: *Cluster-id*

*Instellingen voor het uiterlijk van het bloembladdiagram (kleuruitdrukking)*


Appearance
Colors and legend
Colors
Custom
By expression
Expression
kmeansnd($(KmeansPetal(
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

De twee clusters van het diagram worden gekleurd door de uitdrukking KMeans.

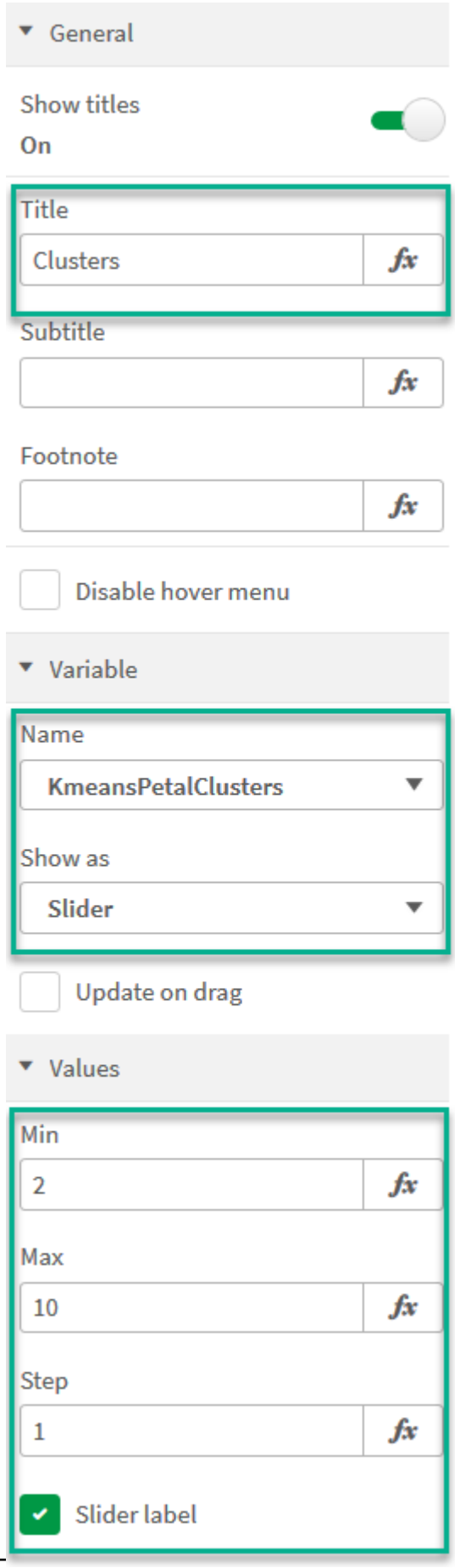
*Clusters die zijn gekleurd op uitdrukking in een bloembladdiagram (kleuruitdrukking)*




6. Voeg een **invoerveld voor variabelen** toe voor het aantal clusters.
    i. Kies **Qlik Dashboard-bundle** onder **Aangepaste objecten** in het **bedrijfsmiddelen**venster. Als we geen toegang hadden tot de dashboard-bundle, konden we nog steeds het aantal clusters wijzigen met behulp van de variabele die we aangemaakt hebben, of rechtstreeks als geheel getal in de uitdrukking.
    ii. Sleep een **invoervak voor variabelen** naar het werkblad.
    iii. Klik onder **Uiterlijk** op **Algemeen**.
    iv. Voer het volgende in voor **Titel**: *Clusters*
    v. Klik op **Variabele**.
    vi. Kies de volgende variabele voor **Naam**: *KmeansPetalClusters*.
    vii. Kies **Schuifregelaar** voor **Tonen als**.

viii. Kies **Waarden** en configureer de instellingen zoals vereist.

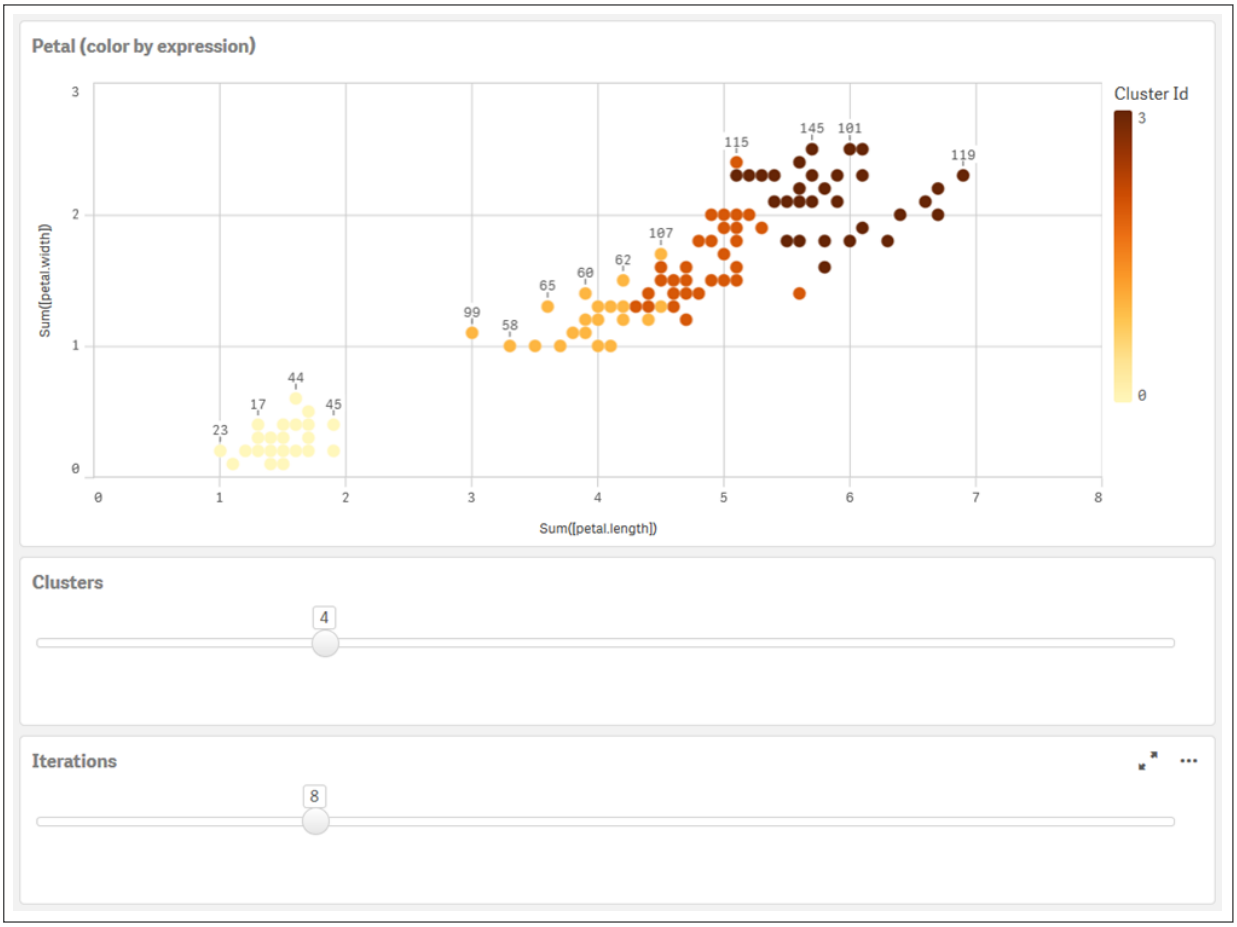
*Uiterlijk voor het clusters invoervak voor variabelen*


General
Show titles
On
Title
Clusters
Subtitle
Footnote
Disable hover menu
Variable
Name
KmeansPetalClusters
Show as
Slider
Update on drag
Values
Min
2
Max
10
Step
1
Slider label

7. Voeg een **Invoervak voor variabelen** toe voor het aantal iteraties.
    i. Sleep een **Invoervak voor variabelen** naar het werkblad.
    ii. Onder **Uiterlijk** kiest u **Algemeen**.
    iii. Voer het volgende in voor **Titel**: *Iteraties*
    iv. Onder **Uiterlijk** kiest u **Variabele**.
    v. Kies de volgende variabele onder **Naam**: *KmeansNumberIterations*.
    vi. Configureer de extra instellingen zoals vereist,

    We kunnen nu het aantal clusters en iteraties wijzigen met de schuifbalken in de invoervakken voor variabelen.

    *Clusters die zijn gekleurd op uitdrukking in een bloembladdiagram (kleuruitdrukking)*

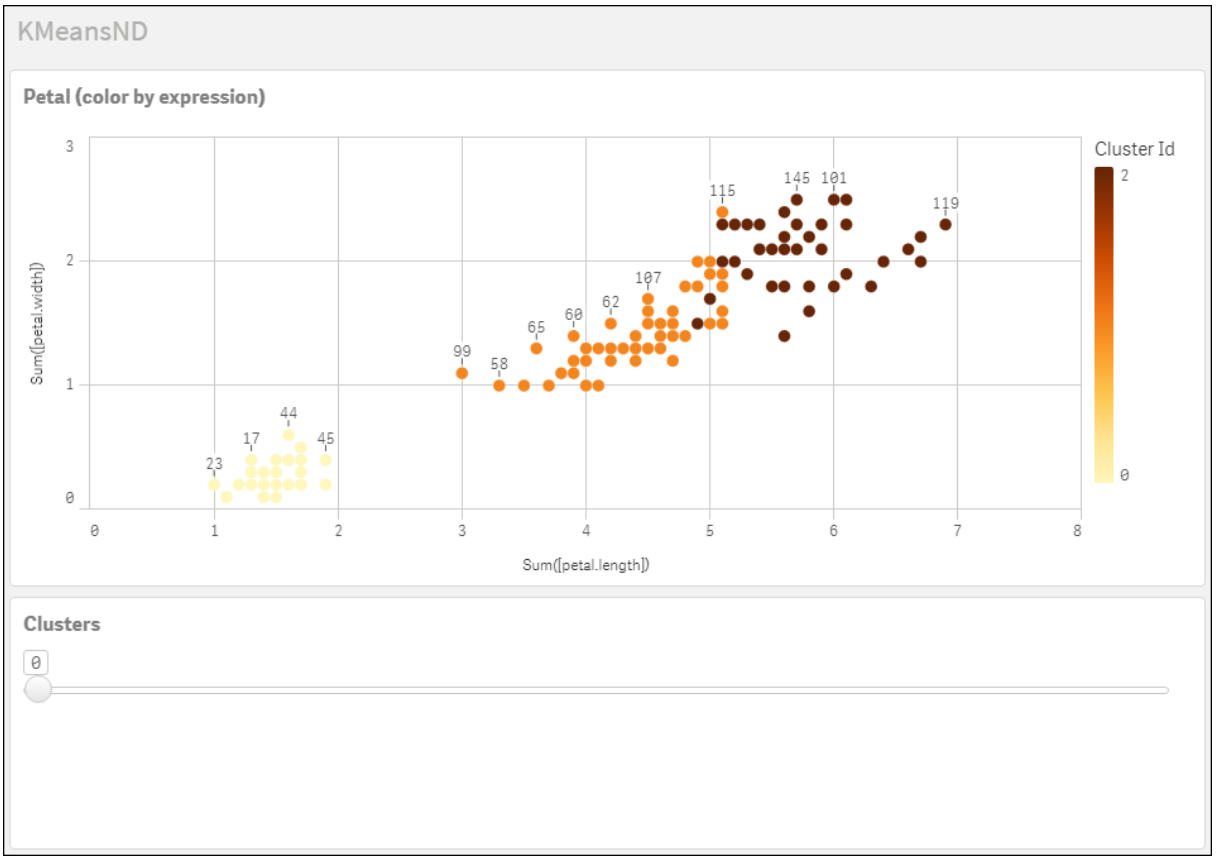



### Automatische clustering

De functies voor **KMeans** bieden ondersteuning voor automatische clustering met behulp van de methode diepteverschil (DeD). Als een gebruiker het aantal clusters instelt op 0, wordt een optimaal aantal clusters voor die gegevensverzameling bepaald. Houd er rekening mee dat een geheel getal voor het aantal clusters (*k*) niet expliciet wordt geretourneerd, maar wel wordt berekend binnen het KMeans-algoritme. Als bijvoorbeeld 0 is opgegeven in de functie voor de waarde van *KmeansPetalClusters* of is ingesteld via een

invoervak voor variabelen, worden clustertoewijzingen automatisch berekend voor de gegevensverzameling die is gebaseerd op een optimaal aantal clusters. Op basis van de Iris-gegevensverzameling, als 0 is geselecteerd voor het aantal clusters, bepaalt het algoritme (automatische clustering) een optimaal aantal clusters (3) voor deze gegevensverzameling.

*De diepteverschilmethode KMeans depth bepaalt het optimale aantal clusters als (k) is ingesteld op 0*




**Iris-gegevensverzameling: Uitgelijnde lading voor editor voor laden van gegevens 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 - diagramfunctie

**KMeansCentroid2D()** evalueert de rijen van het diagram door K-means clustering toe te passen. Voor elke rij van het diagram wordt de gewenste coördinaat weergegeven van het cluster waaraan dit gegevenspunt is toegewezen. De kolommen die worden gebruikt door het clustering-algoritme worden bepaald door de parameter coordinate_1 respectievelijk coordinate_2. Dit zijn beide aggregraties. De parameter num_clusters bepaalt het aantal clusters dat wordt gemaakt. Gegevens kunnen optioneel worden genormaliseerd door de normparameter.

**KMeansCentroid2D** retourneert één waarde per gegevenspunt. De geretourneerde waarde is een dubbele waarde en is een van de coördinaten van de positie die overeenkomt met het middenpunt van het cluster waaraan elk gegevenspunt is toegewezen.

**Syntaxis:**

```
KMeansCentroid2D(num_clusters, coordinate_no, coordinate_1, coordinate_2 [, norm])
```

**Retourgegevenstypen:** dubbele waarde

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| num_clusters | Het geheel getal dat het aantal clusters aangeeft. |

| Argument | Beschrijving |
|---|---|
| coordinate_no | Het gewenste aantal coördinaten van de zwaartepunten (die bijvoorbeeld overeenkomen met de x-, y-, of z-as). |
| coordinate_1 | De aggregatie die de eerste coördinaat berekent, meestal de x-as van de verdelingsplot die op basis van het diagram kan worden gemaakt. De aanvullende parameter, coordinate_2, berekent de tweede coördinaat. |
| norm | De optionele normalisatiemethode die is toegepast op gegevensverzamelingen voordat K-means clustering wordt toegepast.<br><br>Mogelijke waarden:<br><br>0 of 'geen' voor geen normalisatie<br><br>1 of 'zscore' for z-score normalisatie<br><br>2 of 'minmax' voor min-max normalisatie<br><br>Als er geen parameter is opgegeven of de opgegeven parameter is onjuist, wordt er geen normalisatie toegepast.<br><br>De Z-score normaliseert gegevens op basis van de gemiddelde- en standaardafwijking. De Z-score zorgt er niet voor dat elke functie dezelfde schaal toebedeeld krijgt, maar het is een betere manier om aan te gaan met uitschieters dan de min-max-methode.<br><br>De min-max normalisatie zorgt ervoor dat functies dezelfde schaal toebedeeld krijgen door gebruik te maken van de minimum- en maximumwaarden van elk gegevenspunt en elk gegevenspunt opnieuw te berekenen. |

### Automatische clustering

De functies voor **KMeans** bieden ondersteuning voor automatische clustering met behulp van de methode diepteverschil (DeD). Als een gebruiker het aantal clusters instelt op 0, wordt een optimaal aantal clusters voor die gegevensverzameling bepaalt. Houd er rekening mee dat een geheel getal voor het aantal clusters (*k*) niet expliciet wordt geretourneerd, maar wel wordt berekend binnen het KMeans-algoritme. Als bijvoorbeeld 0 is opgegeven in de functie voor de waarde van *KmeansPetalClusters* of is ingesteld via een invoervak voor variabelen, worden clustertoewijzingen automatisch berekend voor de gegevensverzameling die is gebaseerd op een optimaal aantal clusters.

## KMeansCentroidND - diagramfunctie

**KMeansCentroidND()** evalueert de rijen van het diagram door K-means clustering toe te passen. Voor elke rij van het diagram wordt de gewenste coördinaat weergegeven van het cluster waaraan dit gegevenspunt is toegewezen. De kolommen die worden gebruikt door het clustering-algoritme worden bepaald door de parameter coordinate_1 respectievelijk coordinate_2. Dit zijn beide aggregaties. De parameter num_clusters bepaalt het aantal clusters dat wordt gemaakt.

**KMeansCentroidND** retourneert één waarde per gegevenspunt. De geretourneerde waarde is een dubbele waarde en is een van de coördinaten van de positie die overeenkomt met het middenpunt van het cluster waaraan elk gegevenspunt is toegewezen.

**Syntaxis:**

```
KMeansCentroidND((num_clusters, num_iter, coordinate_no, coordinate_1,
coordinate_2 [,coordinate_3 [, ...]])
```

**Retourgegevenstypen:** dual

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| num_clusters | Het geheel getal dat het aantal clusters aangeeft. |
| num_iter | Het aantal iteraties voor clusters met opnieuw geïnitialiseerde middenpunten. |
| coordinate_no | Het gewenste aantal coördinaten van de zwaartepunten (die bijvoorbeeld overeenkomen met de x-, y-, of z-as). |
| coordinate_1 | De aggregatie die de eerste coördinaat berekent, meestal de x-as (van de verdelingsplot die op basis van het diagram kan worden gemaakt). De aanvullende parameters berekenen de tweede, derde en vierde coördinaat, etc. |

### Automatische clustering

De functies voor **KMeans** bieden ondersteuning voor automatische clustering met behulp van de methode diepteverschil (DeD). Als een gebruiker het aantal clusters instelt op 0, wordt een optimaal aantal clusters voor die gegevensverzameling bepaalt. Houd er rekening mee dat een geheel getal voor het aantal clusters (*k*) niet expliciet wordt geretourneerd, maar wel wordt berekend binnen het KMeans-algoritme. Als bijvoorbeeld 0 is opgegeven in de functie voor de waarde van *KmeansPetalClusters* of is ingesteld via een invoervak voor variabelen, worden clustertoewijzingen automatisch berekend voor de gegevensverzameling die is gebaseerd op een optimaal aantal clusters.

## STL_Trend - diagramfunctie

**STL_Trend** is een splitsingsfunctie voor tijdreeksen. Samen met **STL_Seasonal** en **STL_Residual** wordt deze functie gebruikt om een tijdreeks te splitsen in seizoensgebonden-, trendgebonden- en restcomponenten. Binnen de context van het STL-algoritme wordt de splitsingsfunctie voor tijdreeksen gebruikt om zowel terugkerende seizoensgebonden patronen te identificeren op basis van invoergegevens en andere parameters. De functie **STL_Trend** identificeert een algemene trend, ongeacht de seizoensgebonden patronen of cycli, op basis van de gegevens van tijdreeksen.

De drie STL-functies zijn gerelateerd aan de inputgegevens op basis van een simpele som:

**STL_Trend** + **STL_Seasonal** + **STL_Residual** = invoergegevens

STL (uitsplitsing op basis van seizoenen en trends met behulp van Loess) gebruikt technieken voor het gladstrijken van gegevens en door middel van de invoerparameters kan een gebruiker de periodiciteit wijzigen van de uit te voeren berekeningen. De periodiciteit bepaalt hoe de tijddimensie van de invoergegevens (een meting) wordt gesegmenteerd in de analyse.

**STL_Trend** moet minstens een input-statistiek (`target_measure`) en een geheel getal voor `period_int` bevatten en produceert een waarde met drijvende komma. De invoergegevens hebben de vorm van een aggregatie die varieert binnen de tijdsdimensie. U kunt ook waarden opnemen voor de `seasonal_smoother` en `trend_smoother` om het gladheidsalgoritme te wijzigen.

U kunt deze functie gebruiken door hem direct in te voeren in de uitdrukkingseditor voor een diagram.

**Syntaxis:**

```
STL_Trend(target_measure, period_int [,seasonal_smoother [,trend_smoother]])
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **target_measure** | De meting voor het uitsplitsen in seizoensgebonden en trendgebonden componenten. Dit moet een meting zijn zoals Sum(Sales) of Sum(Passengers) die variëren op basis van de tijdsdimensie.<br><br>Dit mag geen constante waarde zijn. |
| **period_int** | De periodiciteit van de gegevensverzameling. Deze parameter is een gehele waarde die het aantal discrete stappen bevat van één periode, of seizoenscyclus, van het signaal.<br><br>Als de tijdsreeks bijvoorbeeld is gesegmenteerd in één sectie voor elk kwartaal van het jaar, moet u de **period_int** instellen op 4 om de periodiciteit te definiëren als Jaar. |
| **seasonal_smoother** | Lengte van de seizoensgebonden gladheidsfactor. Dit moet een oneven geheel getal zijn. De seizoensgebonden gladheidsfactor gebruikt gegevens voor een specifieke fase binnen seizoensfluctuaties over een aantal perioden. Eén discrete stap van de tijdsdimensie van elke periode wordt gebruikt. De seizoensgebonden gladheidsfactor geeft het aantal perioden dat voor de gladheid wordt gebruikt.<br><br>Als de tijdsdimensie wordt gesegmenteerd op maand en de periode Jaar (12) is, wordt het seizoenscomponent berekend zodat elke maand van elk jaar wordt berekend op basis van gegevens van dezelfde maand, zowel in dat jaar als in voorgaande jaren. De waarde **seasonal_smoother** is het aantal jaren dat voor gladheid wordt gebruikt. |
| **trend_smoother** | Lengte van de trendgebonden gladheidsfactor. Dit moet een oneven geheel getal zijn. De trendgladheid gebruikt dezelfde tijdschaal als de parameter **period_int** en de waarde is het aantal granulaten dat wordt gebruikt voor het gladstrijken.<br><br>Als een tijdreeks bijvoorbeeld is gesegmenteerd per maand, dan is de trendgladheid het aantal maanden dat wordt gebruikt voor het gladstrijken. |

De diagramfunctie **STL_Trend** wordt vaak gebruikt in combinatie met de volgende functies:

Gerelateerde functies

| Functie | Interactie |
|---|---|
| *STL_Seasonal - diagramfunctie (page 1694)* | Dit is de functie die wordt gebruikt voor het berekenen van de seizoenscomponent van een tijdreeks. |
| *STL_Residual - diagramfunctie (page 1696)* | Bij het uitsplitsen van een invoergegevens in seizoens- en trendcomponenten past een deel van de variatie van de meting niet binnen een of twee van de hoofdcomponenten. De functie **STL_Residual** berekent dit deel van de uitsplitsing. |

Ga voor een tutorial met een volledig voorbeeld dat laat zien hoe u deze functie gebruikt naar *Tutorial: Uitsplitsing van tijdreeksen in Qlik Sense (page 1698)*.

## STL_Seasonal - diagramfunctie

**STL_Seasonal** is een splitsingsfunctie voor tijdreeksen. Samen met **STL_Trend** en **STL_Residual** wordt deze functie gebruikt om een tijdreeks te splitsen in seizoensgebonden-, trendgebonden- en restcomponenten. Binnen de context van het STL-algoritme wordt de splitsingsfunctie voor tijdreeksen gebruikt om zowel terugkerende seizoensgebonden patronen te identificeren op basis van invoergegevens en andere parameters. De functie **STL_Seasonal** kan een seizoensgebonden patroon binnen een tijdsreeks identificeren, die wordt gescheiden van de algemene trend binnen de gegevens.

De drie STL-functies zijn gerelateerd aan de inputgegevens op basis van een simpele som:

**STL_Trend** + **STL_Seasonal** + **STL_Residual** = invoergegevens

STL (uitsplitsing op basis van seizoenen en trends met behulp van Loess) gebruikt technieken voor het gladstrijken van gegevens en door middel van de invoerparameters kan een gebruiker de periodiciteit wijzigen van de uit te voeren berekeningen. De periodiciteit bepaalt hoe de tijddimensie van de invoergegevens (een meting) wordt gesegmenteerd in de analyse.

**STL_Seasonal** moet minstens een input-statistiek (`target_measure`) en een geheel getal voor `period_int` bevatten en produceert een waarde met een drijvende komma. De invoergegevens hebben de vorm van een aggregatie die varieert binnen de tijdsdimensie. U kunt ook waarden opnemen voor de `seasonal_smoother` en `trend_smoother` om het gladheidsalgoritme te wijzigen.

U kunt deze functie gebruiken door hem direct in te voeren in de uitdrukkingseditor voor een diagram.

**Syntaxis:**

```
STL_Seasonal(target_measure, period_int [,seasonal_smoother [,trend_
smoother]])
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **target_measure** | De meting voor het uitsplitsen in seizoensgebonden en trendgebonden componenten. Dit moet een meting zijn zoals Sum(Sales) of Sum(Passengers) die variëren op basis van de tijdsdimensie.<br><br>Dit mag geen constante waarde zijn. |
| **period_int** | De periodiciteit van de gegevensverzameling. Deze parameter is een gehele waarde die het aantal discrete stappen bevat van één periode, of seizoenscyclus, van het signaal.<br><br>Als de tijdsreeks bijvoorbeeld is gesegmenteerd in één sectie voor elk kwartaal van het jaar, moet u de **period_int** instellen op 4 om de periodiciteit te definiëren als Jaar. |
| **seasonal_smoother** | Lengte van de seizoensgebonden gladheidsfactor. Dit moet een oneven geheel getal zijn. De seizoensgebonden gladheidsfactor gebruikt gegevens voor een specifieke fase binnen seizoensfluctuaties over een aantal perioden. Eén discrete stap van de tijdsdimensie van elke periode wordt gebruikt. De seizoensgebonden gladheidsfactor geeft het aantal perioden dat voor de gladheid wordt gebruikt.<br><br>Als de tijdsdimensie wordt gesegmenteerd op maand en de periode Jaar (12) is, wordt het seizoenscomponent berekend zodat elke maand van elk jaar wordt berekend op basis van gegevens van dezelfde maand, zowel in dat jaar als in voorgaande jaren. De waarde **seasonal_smoother** is het aantal jaren dat voor gladheid wordt gebruikt. |
| **trend_smoother** | Lengte van de trendgebonden gladheidsfactor. Dit moet een oneven geheel getal zijn. De trendgladheid gebruikt dezelfde tijdschaal als de parameter **period_int** en de waarde is het aantal granulaten dat wordt gebruikt voor het gladstrijken.<br><br>Als een tijdreeks bijvoorbeeld is gesegmenteerd per maand, dan is de trendgladheid het aantal maanden dat wordt gebruikt voor het gladstrijken. |

De diagramfunctie **STL_Seasonal** wordt vaak gebruikt in combinatie met de volgende functies:

Gerelateerde functies

| Functie | Interactie |
|---|---|
| *STL_Trend - diagramfunctie (page 1692)* | Dit is de functie die wordt gebruikt voor het berekenen van de trendcomponent van een tijdreeks. |
| *STL_Residual - diagramfunctie (page 1696)* | Bij het uitsplitsen van een invoergegevens in seizoens- en trendcomponenten past een deel van de variatie van de meting niet binnen een of twee van de hoofdcomponenten. De functie **STL_Residual** berekent dit deel van de uitsplitsing. |

Ga voor een tutorial met een volledig voorbeeld dat laat zien hoe u deze functie gebruikt naar *Tutorial: Uitsplitsing van tijdreeksen in Qlik Sense (page 1698)*.

## STL_Residual - diagramfunctie

**STL_Residual** is een splitsingsfunctie voor tijdreeksen. Samen met **STL_Seasonal** en **STL_Trend** wordt deze functie gebruikt om een tijdreeks te splitsen in seizoensgebonden-, trendgebonden- en restcomponenten. Binnen de context van het STL-algoritme wordt de splitsingsfunctie voor tijdreeksen gebruikt om zowel terugkerende seizoensgebonden patronen te identificeren op basis van invoergegevens en andere parameters. Bij het uitvoeren van deze bewerking zal een deel van de variatie binnen de invoergegevens niet passen binnen het seizoensgebonden of trendgebonden component en als overig component worden aangeduid. De diagramfunctie **STL_Residual** registreert dit deel van de berekening.

De drie STL-functies zijn gerelateerd aan de inputgegevens op basis van een simpele som:

**STL_Trend** + **STL_Seasonal** + **STL_Residual** = invoergegevens

STL (uitsplitsing op basis van seizoenen en trends met behulp van Loess) gebruikt technieken voor het gladstrijken van gegevens en door middel van de invoerparameters kan een gebruiker de periodiciteit wijzigen van de uit te voeren berekeningen. De periodiciteit bepaalt hoe de tijddimensie van de invoergegevens (een meting) wordt gesegmenteerd in de analyse.

Bij de splitsing van tijdreeksen wordt over het algemeen gekeken naar seizoensgebonden en algemene variaties in de gegevens, daarom wordt de informatie in het overige component als minst belangrijk beschouwt. Een onregelmatig of periodiek restcomponent kan echter helpen om problemen in de berekening te identificeren, zoals incorecte instellingen voor periodiciteit.

Voor **STL_Residual** gelden minimale invoergegevens (`target_measure`) en een gehele waarde voor de `period_int`, waarbij een waarde met drijvende komma wordt geretourneerd. De invoergegevens hebben de vorm van een aggregatie die varieert binnen de tijdsdimensie. U kunt ook waarden opnemen voor de `seasonal_smoother` en `trend_smoother` om het gladheidsalgoritme te wijzigen.

U kunt deze functie gebruiken door hem direct in te voeren in de uitdrukkingseditor voor een diagram.

**Syntaxis:**

```
STL_Residual(target_measure, period_int [,seasonal_smoother [,trend_
smoother]])
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **target_measure** | De meting voor het uitsplitsen in seizoensgebonden en trendgebonden componenten. Dit moet een meting zijn zoals Sum(Sales) of Sum(Passengers) die variëren op basis van de tijdsdimensie.<br><br>Dit mag geen constante waarde zijn. |
| **period_int** | De periodiciteit van de gegevensverzameling. Deze parameter is een gehele waarde die het aantal discrete stappen bevat van één periode, of seizoenscyclus, van het signaal.<br><br>Als de tijdsreeks bijvoorbeeld is gesegmenteerd in één sectie voor elk kwartaal van het jaar, moet u de **period_int** instellen op 4 om de periodiciteit te definiëren als Jaar. |
| **seasonal_smoother** | Lengte van de seizoensgebonden gladheidsfactor. Dit moet een oneven geheel getal zijn. De seizoensgebonden gladheidsfactor gebruikt gegevens voor een specifieke fase binnen seizoensfluctuaties over een aantal perioden. Eén discrete stap van de tijdsdimensie van elke periode wordt gebruikt. De seizoensgebonden gladheidsfactor geeft het aantal perioden dat voor de gladheid wordt gebruikt.<br><br>Als de tijdsdimensie wordt gesegmenteerd op maand en de periode Jaar (12) is, wordt het seizoenscomponent berekend zodat elke maand van elk jaar wordt berekend op basis van gegevens van dezelfde maand, zowel in dat jaar als in voorgaande jaren. De waarde **seasonal_smoother** is het aantal jaren dat voor gladheid wordt gebruikt. |
| **trend_smoother** | Lengte van de trendgebonden gladheidsfactor. Dit moet een oneven geheel getal zijn. De trendgladheid gebruikt dezelfde tijdschaal als de parameter **period_int** en de waarde is het aantal granulaten dat wordt gebruikt voor het gladstrijken.<br><br>Als een tijdreeks bijvoorbeeld is gesegmenteerd per maand, dan is de trendgladheid het aantal maanden dat wordt gebruikt voor het gladstrijken. |

De diagramfunctie **STL_Residual** wordt vaak gebruikt in combinatie met de volgende functies:

Gerelateerde functies

| Functie | Interactie |
|---|---|
| *STL_Seasonal - diagramfunctie (page 1694)* | Dit is de functie die wordt gebruikt voor het berekenen van de seizoenscomponent van een tijdreeks. |
| *STL_Trend - diagramfunctie (page 1692)* | Dit is de functie die wordt gebruikt voor het berekenen van de trendcomponent van een tijdreeks. |

Ga voor een tutorial met een volledig voorbeeld dat laat zien hoe u deze functie gebruikt naar *Tutorial: Uitsplitsing van tijdreeksen in Qlik Sense (page 1698)*.

## Tutorial: Uitsplitsing van tijdreeksen in Qlik Sense

In deze tutorial tonen we hoe drie diagramfuncties te gebruiken om een tijdreeks uit te splitsen met behulp van het STL-algoritme.

We maken in deze tutorial gebruik van tijdreeksgegevens voor het aantal passagiers dat elke maand met een bepaalde luchtvaartmaatschappij vliegt, om de functionaliteit van het STL-algoritme aan te tonen. De diagramfuncties **STL_Trend**, **STL_Seasonal**, and **STL_Residual** worden gebruikt om de diagrammen te maken. Meer informatie over de uitsplitsing van tijdreeksen in Qlik Sense, vindt u in *Ontledingsfuncties voor tijdreeksen (page 1644)*.

### Een app maken

Maak eerst een nieuwe app aan en importeer daarin de gegevensverzameling.

Download deze gegevensverzameling:

Tutorial: Uitsplitsing van tijdreeksen

Dit bestand bevat gegevens over het maandelijkse aantal passagiers van een luchtvaartmaatschappij.

**Doe het volgende:**

1. Klik in de hub op **Nieuwe app maken**.
2. Open de app en zet het bestand *Tutorial - Time series decomposition.csv* erin.

### De gegevens voorbereiden en uploaden

Als u ervoor wilt zorgen dat Qlik Sense het veld YearMonth correct interpreteert, moet u met behulp van Gegevensbeheer het veld aanmerken als een datumveld, geen veld met een tekenreeks. Deze stap wordt meestal automatisch uitgevoerd, maar in dit voorbeeld verschijnen de datums in een ongebruikelijke

indeling, namelijk *JJJJ-MM*.

1. Selecteer de tabel in Gegevensbeheer en klik op .
2. Selecteer het veld *YearMonth*, klik op en stel het **Veldtype** in op **Datum**.
3. Voer bij **Invoerindeling**de waarde *JJJJ-MM* in.
4. Voer onder **Weergave-indeling** de waarde *JJJJ-MM* in en klik op **OK**.
   In het veld verschijnt nu het kalenderpictogram.
5. Klik op **Gegevens laden**.

U kunt nu de STL-functies gebruiken om uw gegevens visueel weer te geven.

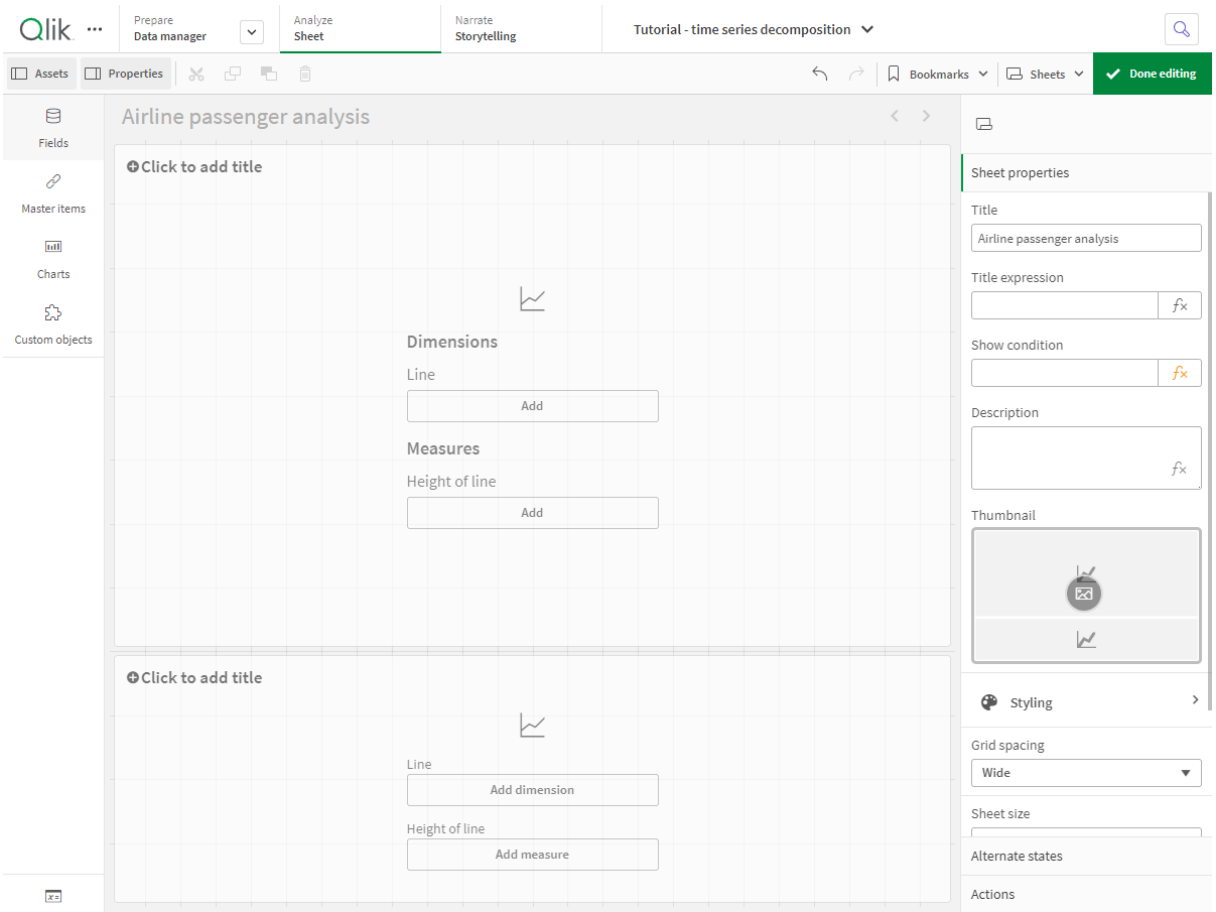
## Diagrammen aanmaken

Nu gaat u twee lijndiagrammen aanmaken om de functionaliteit te zien van de diagramfuncties **STL_Trend**, **STL_Seasonal** en **STL_Residual**.

Open een nieuw werkblad en geef het een titel.

Voeg twee lijndiagrammen toe aan het werkblad. Pas de afmetingen en positie van de diagrammen aan volgens de onderstaande afbeelding.

*Qlik Sense raster-omtrek van blanco app-werkblad*



### Eerste lijndiagram: tendensen en seizoensgebaseerde componenten

**Doe het volgende:**

1. Voeg de titel *Seizoenen en tendensen* toe aan het eerste lijndiagram.
2. Voeg *YearMonth* als dimensie toe en geef dit het label *Datum*.
3. Voeg de volgende meting toe en geeft dit het label *Passagiers per maand*:
   *=Sum(Passengers)*
4. Ga naar **Datums** vouw de meting *Passagiers per maand* open en klik op **Trendlijn toevoegen**.
5. Stel het **Type** in op **Lineair**.
   U gaat deze trendlijn vergelijken met het gladgemaakte resultaat van de trendcomponent.
6. Voeg de volgende meting toe om de trendcomponent te tekenen en geef deze de label *Trend*:
   *=STL_Trend(SUM(Passengers), 12)*
7. Voeg vervolgens de volgende meting toe om het seizoensgebaseerde component te tekenen en geef deze het label *seizoensgebaseerde*:
   *=STL_Seasonal(SUM(Passengers), 12)*
8. Ga naar **Uiterlijk** > **Presentatie** en stel **Schuifbalk** in op **Geen**.
9. Behoud de standaardkleuren of verander ze indien gewenst.

### Tweede lijndiagram: restcomponent

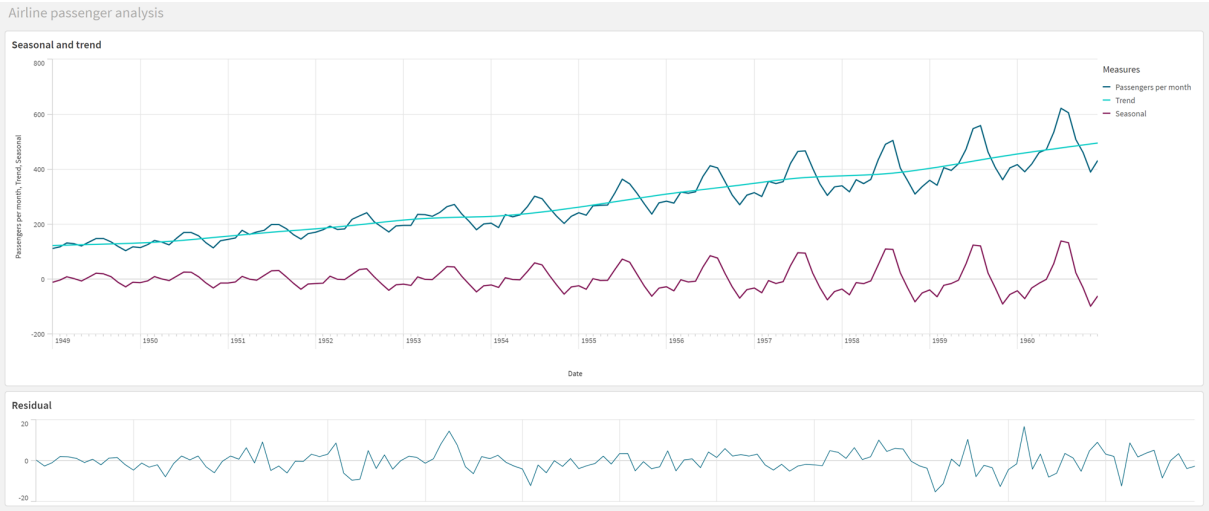
Configureer vervolgens het tweede lijndiagram. Deze visualisatie toont het restcomponent van de tijdreeksen.

**Doe het volgende:**

1. Sleep een lijndiagram naar het werkblad. Geef het de titel *Restwaarde*.
2. Voeg *Datum* toe als dimensie.
3. Voeg de volgende meting toe en geef deze het label *Restwaarde*:
   *=STL_Residual(SUM(Passengers), 12)*
4. Ga naar **Uiterlijk** > **Presentatie** en stel **Schuifbalk** in op **Geen**.

Uw werkblad moet er nu uitzien zoals het onderstaande werkblad.

*Qlik Sense-werkblad voor analyse luchtvaartpassagiers*




## De gegevens interpreteren en toelichten

Met de STL-diagramfuncties kunt u een aantal conclusies trekken uit de gegevens van uw tijdreeksen.

### Trendcomponent

De statistische informatie in de trendcomponent is niet seizoensgebaseerde. Zo ziet u vlotter de algemene, eenmalige schommelingen in de loop van de tijd. In vergelijking met de rechte, lineaire trendlijn voor *Passagiers per maand* geeft de STL-trendcomponent de wijzigende tendensen beter weer. Deze component vertoont duidelijke verschillen maar blijft toch heel duidelijk leesbaar. Het gladmakende gedrag van het STL-algoritme helpt daarbij.

De daling in het aantal passagiers, die duidelijk te zien is in de STL-trendgrafiek, is een teken van de economische impact van de recessies van de jaren 50 van de vorige eeuw.

### seizoensgebaseerde component

De seizoensgebaseerde component zonder trend isoleerde de wederkerende schommelingen in de tijdreeksen en verwijderde de algemene trendinformatie uit dat deel van de analyse. We zijn begonnen met een dataset van jaar/maand-aggregaties. Deze data betekenen impliciet dat we ze segmenteren in stukken van één maand. Door een periodewaarde van 12 in te stellen, stellen we het diagram in om modellen te maken van seizoensgebaseerde patronen in cycli van één jaar (twaalf maanden).

De gegevens vertonen een wederkerend seizoensgebaseerde patroon van een stijging in passagiers tijdens de zomermaanden, gevolgd door een daling in de wintermaanden. Dit komt overeen met het idee dat de zomer een populaire tijd is om op vakantie te gaan en te reizen. We zien ook dat deze seizoensgebaseerde cycli in de loop van de tijdreeksen aanzienlijk breder worden.

### Restcomponent

Het diagram voor het restcomponent toont alle informatie die niet in de trend en niet in de seizoensgebaseerde uitsplitsing is opgenomen. De restcomponent omvat statistische ruis, maar kan ook wijzen op een foute instelling van de argumenten van de STL-trend en van de seizoensgebaseerde functie. Als er periodieke schommelingen te zien zijn in het restcomponent van het signaal of als de

weergegeven informatie duidelijk niet willekeurig is, dan is dat meestal een teken dat de tijdreeks informatie bevat die momenteel niet in de seizoensgebaseerde of trendcomponenten opgenomen is. In dit geval moet u teruggaan naar uw definities van elk functieargument en mogelijk de periodiciteit ervan aanpassen.

### Waarden van gladheidsfactoren

We hebben geen waarden voor de trend- en seizoensgebaseerde gladheidsfactoren opgegeven, en daarom gebruikt de functie de standaardwaarden voor deze parameters. In Qlik Sense produceren de standaardgladheidswaarden van het STL-algoritme effectieve resultaten. Deze argumenten kunnen daarom in de uitdrukkingen meestal worden weggelaten.

*Als u de seizoens- of trendgebonden gladmakersargumenten instelt op 0 in een van de drie STL-functies, gebruikt het algoritme de standaardwaarden in plaats van de waarde 0.*

De trendgebonden gladmakerswaarde gebruikt de dimensie die in het diagram is opgegeven. In het veld *JaarMaand* staan de gegevens per maand; daarom is de waarde van de trendgladmaker het aantal maanden. De seizoensgebaseerde gladmaker geeft de ingestelde periodiciteit weer. In dit geval hebben we één periode ingesteld als twaalf maanden (één jaar), dus de waarde van de seizoensgebaseerde gladmaker is het aantal jaren. Dit klinkt verwarrend, maar in feite betekent het dat we een aantal seizoenen moeten analyseren om seizoensgebonden patronen te vinden. Deze waarde is de seizoensgebaseerde gladmaker.

### Andere nuttige informatie

De seizoensgebaseerde cycli verbreden in de loop van de tijd, en dat betekent dat een meer geavanceerde analytische aanpak gebruik kan maken van algoritmische functies om een uitsplitsing in veelvouden te maken. In de praktijk kan in Qlik Sense een eenvoudige meting van relatieve breedte worden aangemaakt door de seizoensgebaseerde component te verdelen door de trendgebaseerde component. Zo merken we dat de zomerpieken van elke cyclus in de loop van de tijd relatief breder worden. De breedte van het winterse laagseizoen wordt in de loop van de tijd echter niet groter.

## 8.23 Statistische verdelingsfuncties

De statistische verdelingsfuncties retourneren de waarschijnlijkheid van verschillende mogelijke uitkomsten voor een opgegeven invoervariabele. U kunt deze functies gebruiken om de mogelijke waarden van uw gegevenspunten te berekenen.

De drie groepen van de hieronder beschreven statistische verdelingsfuncties worden via de Cephes-functiebibliotheek in Qlik Sense geïmplementeerd. Zie ≤ Cephes library voor referenties en meer informatie over gebruikte algoritmen, nauwkeurigheid enzovoort. De Cephes-functiebibliotheek wordt met toestemming gebruikt.

- De waarschijnlijkheidsfuncties berekenen de waarschijnlijkheid vanaf het punt in de verdeling die door de opgegeven waarde wordt gegeven.

    - De frequentiefuncties worden gebruikt voor discrete verdelingen.
    - De dichtheidsfuncties worden gebruikt voor continue functies.
- De dist-functies berekenen de gecumuleerde waarschijnlijkheid van de verdeling vanaf het punt in de verdeling die door de opgegeven waarde wordt gegeven.
- De Inv-functies berekenen de waarde, gegeven de gecumuleerde waarschijnlijkheid van de verdeling.

Alle functies kunnen zowel worden gebruikt in het load-script voor gegevens als in diagramuitdrukkingen.

## Overzicht van statistische verdelingsfuncties

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

### BetaDensity

`BetaDensity()` retourneert de waarschijnlijkheid van de bèta-verdeling.

```
BetaDensity (value, alpha, beta)
```

### BetaDist

`BetaDist()` retourneert de gecumuleerde waarschijnlijkheid van de bèta-verdeling.

```
BetaDist (value, alpha, beta)
```

### BetaInv

`BetaINV()` retourneert het omgekeerde van de gecumuleerde waarschijnlijkheid van de bèta-verdeling.

```
BetaInv (prob, alpha, beta)
```

### BinomDist

`BinomDist()` retourneert de gecumuleerde waarschijnlijkheid van de binomiale verdeling.

```
BinomDist (value, trials, trial_probability)
```

### BinomFrequency

`BinomFrequency()` retourneert de binomiale waarschijnlijkheidsverdeling.

```
BinomFrequency (value, trials, trial_probability)
```

### BinomInv

`BinomInv()` retourneert het omgekeerde van de gecumuleerde waarschijnlijkheid van de binomiale verdeling.

```
BinomInv (prob, trials, trial_probability)
```

### ChiDensity

`ChiDensity()` retourneert de eenzijdige waarschijnlijkheid van de $chi^2$-verdeling. De $chi^2$-dichtheidsfunctie is gekoppeld aan een $chi^2$-test.

```
ChiDensity (value, degrees_freedom)
```

### ChiDist

`ChiDist()` retourneert de eenzijdige waarschijnlijkheid van de chi$^2$-verdeling. De chi$^2$-verdeling is gekoppeld aan een chi$^2$-test.

```
ChiDist (value, degrees_freedom)
```

### ChiInv

`ChiInv()` retourneert het omgekeerde van de eenzijdige waarschijnlijkheid van de chi$^2$-verdeling.

```
ChiInv (prob, degrees_freedom)
```

### FDensity

`FDensity()` retourneert de waarschijnlijkheid van de `F`-verdeling.

```
FDensity (value, degrees_freedom1, degrees_freedom2)
```

### FDist

`FDist()` retourneert de gecumuleerde waarschijnlijkheid van de `F`-verdeling.

```
FDist (value, degrees_freedom1, degrees_freedom2)
```

### FInv

`FInv()` retourneert het omgekeerde van de gecumuleerde waarschijnlijkheid van de `F`-verdeling.

```
FInv (prob, degrees_freedom1, degrees_freedom2)
```

### GammaDensity

GammaDensity() retourneert de waarschijnlijkheid van de gamma-verdeling.

```
GammaDensity (value, k, θ)
```

### GammaDist

`GammaDist()` retourneert de gecumuleerde waarschijnlijkheid van de gamma-verdeling.

```
GammaDist (value, k, θ)
```

### GammaInv

`GammaInv()` retourneert het omgekeerde van de gecumuleerde waarschijnlijkheid van de gamma-verdeling.

```
GammaInv (prob, k, θ)
```

### NormDist

`NormDist()` retourneert de cumulatieve normale verdeling van het opgegeven gemiddelde en de opgegeven standaarddeviatie. Als mean = 0 en standard_dev = 1, retourneert de functie de standaard normale verdeling.

```
NormDist (value, mean, standard_dev)
```

### NormInv

`NormInv()` retourneert het omgekeerde van de normale cumulatieve verdeling van het opgegeven gemiddelde en de opgegeven standaarddeviatie.

```
NormInv (prob, mean, standard_dev)
```

### PoissonDist

`PoissonDist()` retourneert de gecumuleerde waarschijnlijkheid van de poisson-verdeling.

```
PoissonDist (value, mean)
```

### PoissonFrequency

`PoissonFrequency()` retourneert de poisson-waarschijnlijkheidsverdeling.

```
PoissonFrequency (value, mean)
```

### PoissonInv

`PoissonInv()` retourneert het omgekeerde van de gecumuleerde waarschijnlijkheid van de poisson-verdeling.

```
PoissonInv (prob, mean)
```

### TDensity

`TDensity()` retourneert de waarde van de dichtheid `t` van de student, waarbij een numerieke waarde een berekende waarde van `t` is, waarvan de waarschijnlijk moet worden berekend.

```
TDensity (value, degrees_freedom, tails)
```

### TDist

`TDist()` retourneert de waarschijnlijkheid van de `t`-verdeling van de student, waarbij een numerieke waarde een berekende waarde van `t` is, waarvan de waarschijnlijk moet worden berekend.

```
TDist (value, degrees_freedom, tails)
```

### TInv

`TInv()` retourneert de waarde `t` van de `t`-verdeling van de student als functie van de waarschijnlijkheid en de vrijheidsgraden.

```
TInv (prob, degrees_freedom)
```

---

**Zie ook:**

p *Statistische aggregatiefuncties (page 456)*

## BetaDensity

`BetaDensity()` retourneert de waarschijnlijkheid van de bèta-verdeling.

**Syntaxis:**

```
BetaDensity(value, alpha, beta)
```

**Retourgegevenstypen:** getal

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De waarde waarvan u de verdeling wilt bepalen. De waarde moet tussen 0 en 1 liggen. |
| alpha | Een positief getal definieert de eerste vormparameter. Het is de exponent van de toevalsvariabele. |
| beta | Een positief getal definieert de tweede vormparameter. Geeft het aantal vrijheidsgraden van de noemer aan. |

## BetaDist

`BetaDist()` retourneert de gecumuleerde waarschijnlijkheid van de bèta-verdeling.

**Syntaxis:**

```
BetaDist(value, alpha, beta)
```

**Retourgegevenstypen:** getal

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De waarde waarvan u de verdeling wilt bepalen. De waarde moet tussen 0 en 1 liggen. |
| alpha | Een positief getal definieert de eerste vormparameter. Het is de exponent van de toevalsvariabele. |
| beta | Een positief getal definieert de tweede vormparameter. Dit is de exponent die de vorm van de verdeling bepaalt. |

Deze functie is op de volgende manier gerelateerd aan de functie `BetaInv`:
`If prob = BetaDist(value, alpha, beta), then BetaInv(prob, alpha, beta) = value`

## BetaInv

`BetaINV()` retourneert het omgekeerde van de gecumuleerde waarschijnlijkheid van de bèta-verdeling.

**Syntaxis:**

```
BetaInv(prob, alpha, beta)
```

**Retourgegevenstypen:** getal

Argumenten

| Argument | Beschrijving |
|---|---|
| prob | Een waarschijnlijkheid die is verbonden met de bèta-waarschijnlijkheidsverdeling. Dit moet een waarde zijn tussen 0 en 1. |
| alpha | Een positief getal definieert de eerste vormparameter. Het is de exponent van de toevalsvariabele. |
| beta | Een positief getal definieert de tweede vormparameter. Dit is de exponent die de vorm van de verdeling bepaalt. |

Deze functie is op de volgende manier gerelateerd aan de functie `BetaDist`:
`If prob = BetaDist(value, alpha, beta), then BetaInv(prob, alpha, beta) = value`

## BinomDist

`BinomDist()` retourneert de gecumuleerde waarschijnlijkheid van de binomiale verdeling.

**Syntaxis:**

```
BinomDist(value, trials, trial_probability)
```

**Retourgegevenstypen:** getal

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De waarde waarvan u de verdeling wilt bepalen. De waarde moet een geheel getal zijn dat niet kleiner is dan 0 en niet groter is dan het aantal proeven. |
| trials | Een positief getal dat het aantal proeven weergeeft. |
| trial_ probability | De kans op succes voor elke proef. Dit is altijd een getal tussen 0 en 1. |

Deze functie is op de volgende manier gerelateerd aan de functie `BinomInv`:
`If prob = BinomDIST(value, trials, trial_probability), then BinomInv(prob, trials, trial_ probability) = value`

## BinomFrequency

`BinomFrequency()` retourneert de binomiale waarschijnlijkheidsverdeling.

**Syntaxis:**

```
BinomFrequency(value, trials, trial_probability)
```

**Retourgegevenstypen:** getal

Argumenten

| Argument | Beschrijving |
|---|---|
| `value` | De waarde waarvan u de verdeling wilt bepalen. De waarde moet een geheel getal zijn dat niet kleiner is dan 0 en niet groter is dan het aantal proeven. |
| `trials` | Een positief getal dat het aantal proeven weergeeft |
| `trial_probability` | De kans op succes voor elke proef. Dit is altijd een getal tussen 0 en 1. |

## BinomInv

`BinomInv()` retourneert het omgekeerde van de gecumuleerde waarschijnlijkheid van de binomiale verdeling.

**Syntaxis:**

```
BinomInv(prob, trials, trial_probability)
```

**Retourgegevenstypen:** getal

Argumenten

| Argument | Beschrijving |
|---|---|
| `prob` | Een waarschijnlijkheid die is verbonden met de binomiale waarschijnlijkheidsverdeling. Dit moet een waarde zijn tussen 0 en 1. |
| `trials` | Een positief getal dat het aantal proeven weergeeft. |
| `trial_probability` | De kans op succes voor elke proef. Dit is altijd een getal tussen 0 en 1. |

Deze functie is op de volgende manier gerelateerd aan de functie `BinomDist`:
`If prob = BinomDist(value, trials, trial_probability), then BinomInv(prob, trials, trial_probability) = value`

## ChiDensity

`ChiDensity()` retourneert de eenzijdige waarschijnlijkheid van de chi$^2$-verdeling. De chi$^2$-dichtheidsfunctie is gekoppeld aan een chi$^2$-test.

**Syntaxis:**

```
ChiDensity(value, degrees_freedom)
```

**Retourgegevenstypen:** getal

Argumenten

| Argument | Beschrijving |
|---|---|
| `value` | De waarde waarvan u de verdeling wilt bepalen. De waarde mag niet negatief zijn. |
| `degrees_freedom` | Een positief geheel getal waarmee het aantal vrijheidsgraden van de teller wordt aangegeven. |

## ChiDist

`ChiDist()` retourneert de eenzijdige waarschijnlijkheid van de chi$^2$-verdeling. De chi$^2$-verdeling is gekoppeld aan een chi$^2$-test.

**Syntaxis:**

```
CHIDIST(value, degrees_freedom)
```

**Retourgegevenstypen:** getal

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De waarde waarvan u de verdeling wilt bepalen. De waarde mag niet negatief zijn. |
| degrees_freedom | Een positief geheel getal waarmee het aantal vrijheidsgraden wordt aangegeven. |

Deze functie is op de volgende manier gerelateerd aan de functie **ChiInv**:
`If prob = CHIDIST(value,df), then CHIINV(prob, df) = value`

**Beperkingen:**

Alle argumenten moeten numeriek zijn, anders wordt een NULL-waarde geretourneerd.

Voorbeelden en resultaten:

| Voorbeeld | Resultaat |
|---|---|
| CHIDIST( 8, 15) | retourneert 0,9238 |

## ChiInv

`ChiInv()` retourneert het omgekeerde van de eenzijdige waarschijnlijkheid van de chi$^2$-verdeling.

**Syntaxis:**

```
CHIINV(prob, degrees_freedom)
```

**Retourgegevenstypen:** getal

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| prob | Een waarschijnlijkheid die is verbonden met de chi$^2$-verdeling. Dit moet een waarde zijn tussen 0 en 1. |
| degrees_ freedom | Een geheel getal waarmee het aantal vrijheidsgraden wordt aangegeven. |

Deze functie is op de volgende manier gerelateerd aan de functie **ChiDist**:
`If prob = CHIDIST(value,df), then CHIINV(prob, df) = value`

**Beperkingen:**

Alle argumenten moeten numeriek zijn, anders wordt een NULL-waarde geretourneerd.

Voorbeelden en resultaten:

| Voorbeeld | Resultaat |
|---|---|
| CHIINV(0.9237827, 15 ) | retourneert 8,0000 |

## FDensity

`FDensity()` retourneert de waarschijnlijkheid van de F-verdeling.

**Syntaxis:**

```
FDensity(value, degrees_freedom1, degrees_freedom2)
```

**Retourgegevenstypen:** getal

Argumenten

| Argument | Beschrijving |
|---|---|
| `value` | De waarde waarvan u de verdeling wilt bepalen. De waarde mag niet negatief zijn. |
| `degrees_ freedom1` | Een positief geheel getal waarmee het aantal vrijheidsgraden van de teller wordt aangegeven. |
| `degrees_ freedom2` | Een positief geheel getal waarmee het aantal vrijheidsgraden van de noemer wordt aangegeven. |

## FDist

`FDist()` retourneert de gecumuleerde waarschijnlijkheid van de F-verdeling.

**Syntaxis:**

```
FDist(value, degrees_freedom1, degrees_freedom2)
```

**Retourgegevenstypen:** getal

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De waarde waarvan u de verdeling wilt bepalen. De waarde mag niet negatief zijn. |
| degrees_ freedom1 | Een positief geheel getal waarmee het aantal vrijheidsgraden van de teller wordt aangegeven. |
| degrees_ freedom2 | Een positief geheel getal waarmee het aantal vrijheidsgraden van de noemer wordt aangegeven. |

Deze functie is op de volgende manier gerelateerd aan de functie **FInv**:
`If prob = FDIST(value, df1, df2), then FINV(prob, df1, df2) = value`

**Beperkingen:**

Alle argumenten moeten numeriek zijn, anders wordt een NULL-waarde geretourneerd.

Voorbeelden en resultaten:

| Voorbeeld | Resultaat |
|---|---|
| FDIST(15, 8, 6) | retourneert 0,0019 |

## FInv

`FInv()` retourneert het omgekeerde van de gecumuleerde waarschijnlijkheid van de `F`-verdeling.

**Syntaxis:**

```
FInv(prob, degrees_freedom1, degrees_freedom2)
```

**Retourgegevenstypen:** getal

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| prob | Een waarschijnlijkheid die is verbonden met de F-waarschijnlijkheidsverdeling. Dit moet een waarde zijn tussen 0 en 1. |
| degrees_ freedom | Een geheel getal waarmee het aantal vrijheidsgraden wordt aangegeven. |

Deze functie is op de volgende manier gerelateerd aan de functie **FDist**:
`If prob = FDIST(value, df1, df2), then FINV(prob, df1, df2) = value`

**Beperkingen:**

Alle argumenten moeten numeriek zijn, anders wordt een NULL-waarde geretourneerd.

Voorbeelden en resultaten:

| Voorbeeld | Resultaat |
|---|---|
| FINV( 0.0019369, 8, 6) | retourneert 15,0000 |

## GammaDensity

GammaDensity() retourneert de waarschijnlijkheid van de gamma-verdeling.

**Syntaxis:**
```
GammaDensity(value, k, θ)
```

**Retourgegevenstypen:** getal

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De waarde waarvan u de verdeling wilt bepalen. De waarde mag niet negatief zijn. |
| k | Een positief getal dat de tweede vormparameter definieert. |
| θ | Een positief getal dat de tweede schaalparameter definieert. |

## GammaDist

`GammaDist()` retourneert de gecumuleerde waarschijnlijkheid van de gamma-verdeling.

**Syntaxis:**
```
GammaDist(value, k, θ)
```

**Retourgegevenstypen:** getal

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De waarde waarvan u de verdeling wilt bepalen. De waarde mag niet negatief zijn. |
| k | Een positief getal dat de tweede vormparameter definieert. |
| θ | Een positief getal dat de tweede schaalparameter definieert. |

Deze functie is op de volgende manier gerelateerd aan de functie `GammaINV`:
`If prob = GammaDist(value, k, θ), then GammaInv(prob, k, θ) = value`

## GammaInv

`GammaInv()` retourneert het omgekeerde van de gecumuleerde waarschijnlijkheid van de gamma-verdeling.

**Syntaxis:**

```
GammaInv(prob, k, θ)
```

**Retourgegevenstypen:** getal

Argumenten

| Argument | Beschrijving |
|---|---|
| `prob` | Een waarschijnlijkheid die is verbonden met de gamma-waarschijnlijkheidsverdeling. Dit moet een waarde zijn tussen 0 en 1. |
| `k` | Een positief getal dat de tweede vormparameter definieert. |
| `θ` | Een positief getal dat de tweede schaalparameter definieert. |

Deze functie is op de volgende manier gerelateerd aan de functie `GammaDist`:
`If prob = GammaDist(value, k, θ), then GammaInv(prob, k, θ) = value`

## NormDist

`NormDist()` retourneert de cumulatieve normale verdeling van het opgegeven gemiddelde en de opgegeven standaarddeviatie. Als mean = 0 en standard_dev = 1, retourneert de functie de standaard normale verdeling.

**Syntaxis:**

```
NORMDIST(value, [mean], [standard_dev], [cumulative])
```

**Retourgegevenstypen:** getal

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De waarde waarvan u de verdeling wilt bepalen. |
| mean | Optionele waarde waarmee het rekenkundige gemiddelde voor de verdeling wordt aangegeven.<br><br>Als u geen waarde voor dit argument aangeeft, is de standaardwaarde 0. |

| Argument | Beschrijving |
|---|---|
| standard_dev | Optionele positieve waarde waarmee de standaardafwijking van de verdeling wordt aangegeven.<br><br>Als u geen waarde voor dit argument aangeeft, is de standaardwaarde 1. |
| cumulative | U kunt er optioneel voor kiezen om een standaard normale verdeling of een cumulatieve verdeling te gebruiken.<br><br>0 = standaard normale verdeling<br><br>1 = cumulatieve verdeling (standaard) |

Deze functie is op de volgende manier gerelateerd aan de functie **NormInv**:
`If prob = NORMDIST(value, m, sd), then NORMINV(prob, m, sd) = value`

**Beperkingen:**

Alle argumenten moeten numeriek zijn, anders wordt een NULL-waarde geretourneerd.

Voorbeelden en resultaten:

| Voorbeeld | Resultaat |
|---|---|
| NORMDIST( 0.5, 0, 1) | retourneert 0,6915 |

## NormInv

`NormInv()` retourneert het omgekeerde van de normale cumulatieve verdeling van het opgegeven gemiddelde en de opgegeven standaarddeviatie.

**Syntaxis:**

```
NORMINV(prob, mean, standard_dev)
```

**Retourgegevenstypen:** getal

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| prob | Een waarschijnlijkheid die is verbonden met de normale verdeling. Dit moet een waarde zijn tussen 0 en 1. |
| mean | Een waarde waarmee het rekenkundige gemiddelde voor de verdeling wordt aangegeven. |
| standard_dev | Een positieve waarde waarmee de standaarddeviatie van de verdeling wordt aangegeven. |

Deze functie is op de volgende manier gerelateerd aan de functie **NormDist**:
`If prob = NORMDIST(value, m, sd), then NORMINV(prob, m, sd) = value`

**Beperkingen:**

Alle argumenten moeten numeriek zijn, anders wordt een NULL-waarde geretourneerd.

Voorbeelden en resultaten:

| Voorbeeld | Resultaat |
|---|---|
| NORMINV( 0.6914625, 0, 1 ) | retourneert 0,5000 |

## PoissonDist

`PoissonDist()` retourneert de gecumuleerde waarschijnlijkheid van de poisson-verdeling.

**Syntaxis:**

```
PoissonDist(value, mean)
```

**Retourgegevenstypen:** getal

Argumenten

| Argument | Beschrijving |
|---|---|
| `value` | De waarde waarvan u de verdeling wilt bepalen. De waarde mag niet negatief zijn. |
| `mean` | Een positief getal dat de gemiddelde uitkomst definieert. |

Deze functie is op de volgende manier gerelateerd aan de functie `PoissonInv`:
`If prob = PoissonDist(value, mean), then PoissonInv(prob, mean) = value`

## PoissonFrequency

`PoissonFrequency()` retourneert de poisson-waarschijnlijkheidsverdeling.

**Syntaxis:**

```
PoissonFrequency(value, mean)
```

**Retourgegevenstypen:** getal

Argumenten

| Argument | Beschrijving |
|---|---|
| `value` | De waarde waarvan u de verdeling wilt bepalen. De waarde mag niet negatief zijn. |
| `mean` | Een positief getal dat de gemiddelde uitkomst definieert. |

## PoissonInv

`PoissonInv()` retourneert het omgekeerde van de gecumuleerde waarschijnlijkheid van de poisson-verdeling.

**Syntaxis:**

```
PoissonInv(prob, mean)
```

**Retourgegevenstypen:** getal

Argumenten

| Argument | Beschrijving |
|---|---|
| `prob` | Een waarschijnlijkheid die is verbonden met de poisson-waarschijnlijkheidsverdeling. Dit moet een waarde zijn tussen 0 en 1. |
| `mean` | Een positief getal dat de gemiddelde uitkomst definieert. |

Deze functie is op de volgende manier gerelateerd aan de functie `PoissonDIST`:
`If prob = PoissonDist(value, mean), then PoissonInv(prob, mean) = value`

## TDensity

`TDensity()` retourneert de waarde van de dichtheid `t` van de student, waarbij een numerieke waarde een berekende waarde van `t` is, waarvan de waarschijnlijk moet worden berekend.

**Syntaxis:**

```
TDensity(value, degrees_freedom)
```

**Retourgegevenstypen:** getal

Argumenten

| Argument | Beschrijving |
|---|---|
| `value` | De waarde waarvan u de verdeling wilt bepalen. De waarde mag niet negatief zijn. |
| `degrees_freedom` | Een positief geheel getal waarmee het aantal vrijheidsgraden wordt aangegeven. |

## TDist

`TDist()` retourneert de waarschijnlijkheid van de `t`-verdeling van de student, waarbij een numerieke waarde een berekende waarde van `t` is, waarvan de waarschijnlijk moet worden berekend.

**Syntaxis:**

```
TDist(value, degrees_freedom, tails)
```

**Retourgegevenstypen:** getal

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| value | De waarde waarvan u de verdeling wilt bepalen. De waarde mag niet negatief zijn. |
| degrees_freedom | Een positief geheel getal waarmee het aantal vrijheidsgraden wordt aangegeven. |
| tails | Moet 1 (eenzijdige verdeling) of 2 (tweezijdige verdeling) zijn. |

Deze functie is op de volgende manier gerelateerd aan de functie **TInv**:
`If prob = TDIST(value, df ,2), then TINV(prob, df) = value`

**Beperkingen:**

Alle argumenten moeten numeriek zijn, anders wordt een NULL-waarde geretourneerd.

Voorbeelden en resultaten:

| Voorbeeld | Resultaat |
|---|---|
| TDIST(1, 30, 2) | retourneert 0,3253 |

## TInv

`TInv()` retourneert de waarde `t` van de `t`-verdeling van de student als functie van de waarschijnlijkheid en de vrijheidsgraden.

**Syntaxis:**

```
TINV(prob, degrees_freedom)
```

**Retourgegevenstypen:** getal

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| prob | Een tweezijdige waarschijnlijkheid, verbonden met de t-verdeling. Dit moet een waarde zijn tussen 0 en 1. |
| degrees_ freedom | Een geheel getal waarmee het aantal vrijheidsgraden wordt aangegeven. |

**Beperkingen:**

Alle argumenten moeten numeriek zijn, anders wordt een NULL-waarde geretourneerd.

Deze functie is op de volgende manier gerelateerd aan de functie **TDist**:
`If prob = TDIST(value, df ,2), then TINV(prob, df) = value.`

Voorbeelden en resultaten:

| Voorbeeld | Resultaat |
|---|---|
| TINV(0.3253086, 30 ) | retourneert 1,0000 |

# 8.24 Tekenreeksfuncties

In dit hoofdstuk worden functies voor het werken met en manipuleren van tekenreeksen beschreven.

Alle functies kunnen zowel worden gebruikt in het load-script voor gegevens als in diagramuitdrukkingen, behalve **Evaluate**, dat alleen kan worden gebruikt in het load-script voor gegevens.

## Overzicht van tekenreeksfuncties

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

### Capitalize

De **Capitalize()** functie converteert het eerste teken van elk woord in een teksttekenreeks naar hoofdletters en converteert alle andere tekens naar kleine letters.

```
Capitalize (text)
```

### Chr

**Chr()** retourneert het Unicode-teken dat overeenkomt met het ingevoerde gehele getal.

```
Chr (int)
```

### CountRegEx

**CountRegEx()** retourneert het aantal keren dat het opgegeven reguliere expressiepatroom voorkomt in de tekst van de invoertekenreeks. Als er geen overeenkomst wordt gevonden, wordt 0 geretourneerd.

```
CountRegEx (text, regex)
```

### Evaluate

**Evaluate()** bepaalt of de tekenreeks van de invoertekst kan worden geëvalueerd als een geldige Qlik Sense-uitdrukking en retourneert, als dat inderdaad het geval is, de waarde van de uitdrukking als een tekenreeks. Als de invoertekenreeks geen geldige uitdrukking is, wordt NULL geretourneerd.

```
Evaluate (expression_text)
```

### ExtractRegEx

**ExtractRegEx()** extraheert tekst uit een tekenreeksuitdrukking met behulp van het opgegeven reguliere expressiepatroon. De functie retourneert een nulwaarde als er geen overeenkomsten worden gevonden.

```
ExtractRegEx (text, regex [, field_no])
```

### ExtractRegExGroup

**ExtractRegExGroup()** extraheert tekst uit een tekenreeksuitdrukking met behulp van het samengestelde reguliere expressiepatroon dat is opgegeven. Geef bij het gebruik van de functie de groep op die binnen de samengestelde regex moet worden gebruikt. De functie retourneert een nulwaarde als er geen overeenkomsten worden gevonden.

```
ExtractRegExGroup (text, regex, group [, field_no])
```

### FindOneOf

**FindOneOf()** doorzoekt een tekenreeks om de positie te vinden van enig teken uit de reeks van opgegeven tekens. De positie van het N-de exemplaar van een teken uit de zoekset wordt geretourneerd, waarbij N de optionele derde parameter van de functie is. Als er geen derde parameter wordt opgegeven, wordt het eerste exemplaar geretourneerd. Als geen waarde wordt gevonden, wordt **0** geretourneerd.

```
FindOneOf (text, char_set[, count])
```

### Hash128

**Hash128()** retourneert een 128-bits hashwaarde van de gecombineerde invoerwaarden van de uitdrukking. Het resultaat is een tekenreeks van 22 tekens.

```
Hash128 (expr{, expression})
```

### Hash160

**Hash160()** retourneert een 160-bits hashwaarde van de gecombineerde invoerwaarden van de uitdrukking. Het resultaat is een tekenreeks van 27 tekens.

```
Hash160 (expr{, expression})
```

### Hash256

**Hash256()** retourneert een 256-bits hashwaarde van de gecombineerde invoerwaarden van de uitdrukking. Het resultaat is een tekenreeks van 43 tekens.

```
Hash256 (expr{, expression})
```

### Index

**Index()** doorzoekt een tekenreeks om de beginpositie te vinden van de n-de keer dat een opgegeven subtekenreeks voorkomt. Een optioneel derde argument levert de waarde van n, die 1 is als het argument wordt weggelaten. Bij een negatieve waarde wordt gezocht vanaf het einde van de tekenreeks. De nummering van de posities in de tekenreeks loopt van links naar rechts en begint bij **1**.

```
Index (text, substring[, count])
```

### IndexRegEx

**IndexRegEx()** doorzoekt de invoertekenreeks en geeft de startpositie van de n-de herhaling van het gespecificeerde reguliere expressiepatroon. Een optioneel derde argument **count** levert de waarde van n, die 1 is als het argument wordt weggelaten. De nummering van de posities in de tekenreeks loopt van links naar rechts en begint bij 1. Als er geen overeenkomst wordt gevonden, retourneert de functie 0.

```
IndexRegEx (text, regex [, count])
```

### IndexRegExGroup

**ExtractRegExGroup()** extraheert tekst uit een tekenreeksuitdrukking met behulp van het samengestelde reguliere expressiepatroon dat is opgegeven. Geef bij het gebruik van de functie de groep op die binnen de samengestelde regex moet worden gebruikt. De functie retourneert een nulwaarde als er geen overeenkomsten worden gevonden.

```
IndexRegExGroup (text, regex, group [, count])
```

### IsJson

**IsJson()** controleert of een opgegeven string geldige JSON-gegevens (JavaScript Object Notation) bevat. U kunt ook een specifiek JSON-gegevenstype valideren.

```
IsJson (json [, type])
```

### IsRegEx

**IsRegEx()** retourneert of de opgegeven tekst een geldige reguliere expressie is.

```
IsRegEx (expr [, debug)
```

### JsonGet

**JsonGet()** retourneert het pad van een JSON-gegevensreeks (JavaScript Object Notation). De JSON-gegevens moeten een geldige JSON-string zijn, maar kunnen extra spaties of nieuwe regels bevatten.

```
JsonGet (json, path)
```

### JsonSet

**JsonSet()** wijzigt een string die JSON-gegevens (JavaScript Object Notation) bevat. Het kan een JSON-waarde instellen of invoegen met de nieuwe locatie die is opgegeven door het pad. De JSON-gegevens moeten een geldige JSON zijn, maar kunnen extra spaties of nieuwe regels bevatten.

```
JsonSet(json, path, value)
```

### KeepChar

**KeepChar()** retourneert een tekenreeks die bestaat uit tekens van de eerste rekenreeks die overeenkomen met de tekens in de tweede tekensreeks. Deze functie is hoofdlettergevoelig.

```
KeepChar (text, keep_chars)
```

### Left

**Left()** retourneert een tekenreeks die bestaat uit de eerste (meest linker) van de invoertekenreeks, waarbij het aantal tekens wordt bepaald door het tweede argument.

```
Left (text, count)
```

### Len

**Len()** retourneert de lengte van de invoertekenreeks.

```
Len (text)
```

### LevenshteinDist

**LevenshteinDist()** retourneert de afstand van Levenshtein tussen twee tekenreeksen. Het wordt gedefinieerd als het aantal bewerkingen van één teken (invoegingen, verwijderingen of vervangingen) dat is vereist om een tekenreeks te wijzigen. De functie kan worden gebruikt om fuzzy tekenreeksen te vergelijken.

```
LevenshteinDist (text1, text2)
```

### Lower

**Lower()** converteert alle tekens in de invoertekenreeks naar kleine letters.

```
Lower (text)
```

### LTrim

**LTrim()** retourneert de invoertekenreeks, ontdaan van beginspaties.

```
LTrim (text)
```

### MatchRegEx

**MatchRegEx()** vergelijkt de invoertekenreeks met een of meer gespecificeerde reguliere expressiepatronen en retourneert de numerieke locatie van de overeenkomende reguliere expressiepatronen. Als er geen overeenkomsten worden gevonden, retourneert de functie 0. Deze functie zoekt alleen naar exacte overeenkomsten.

```
MatchRegEx (text, regex1 [ , regex2,...regexN])
```

### Mid

**Mid()** retourneert het gedeelte van de invoertekenreeks dat begint op de positie van het teken dat is gedefinieerd door het tweede argument, 'start', en retourneert het aantal tekens dat is gedefinieerd door het derde argument, 'count'. Als 'count' wordt weggelaten, wordt de rest van de invoertekenreeks geretourneerd. Het eerste teken in de invoertekenreeks heeft het nummer 1.

```
Mid (text, start[, count])
```

### Ord

**Ord()** returneert de numerieke (ASCII of Unicode) waarde van het eerste teken van een tekenreeks. Deze functie is handig om tekenreeksen te evalueren of te vergelijken op basis van hun onderliggende tekencodes, bijvoorbeeld bij het sorteren of filteren van tekenreeksen met niet-standaard tekens.

```
Ord (text)
```

### PurgeChar

**PurgeChar()** retourneert een tekenreeks die bestaat uit de tekens die zijn opgenomen in de invoertekenreeks ('text'), behalve de tekens die voorkomen in het tweede argument ('remove_chars').

```
PurgeChar (text, remove_chars)
```

### Repeat

**Repeat()** vormt een tekenreeks die bestaat uit de invoertekenreeks die het aantal keren wordt herhaald dat wordt gedefinieerd door het tweede argument.

```
Repeat (text[, repeat_count])
```

### Replace

**Replace()** retourneert een tekenreeks nadat alle treffers van een bepaalde subtekenreeks in de tekenreeks zijn vervangen door een andere subtekenreeks. De functie is niet-recursief en wordt van links naar rechts uitgevoerd.

```
Replace (text, from_str, to_str)
```

### ReplaceRegEx

**ReplaceRegEx()** retourneert een tekenreeks na het vervangen van één of meer overeenkomsten tussen een invoertekenreeks en een gespecificeerd regulier expressiepatroon. De tekst die de overeenkomende tekst vervangt, wordt opgegeven in het argument **to_str**. De functie wordt van links naar rechts uitgevoerd, maar als u een negatieve **occurrence** waarde opgeeft, leest de functie van rechts naar links.

```
ReplaceRegEx (text, regex, to_str [, occurrence])
```

### ReplaceRegExGroup

**ReplaceRegExGroup()** retourneert een tekenreeks na het vervangen van één of meer overeenkomsten tussen een invoertekenreeks en het samengestelde reguliere expressiepatroon dat is opgegeven. De tekst die de overeenkomende tekst vervangt, wordt opgegeven in het argument **to_str**. De functie wordt van links naar rechts uitgevoerd, maar als u een negatieve **occurrence** waarde opgeeft, leest de functie van rechts naar links.

```
ReplaceRegExGroup (text, regex, to_str, group [, occurrence])
```

### Right

**Right()** retourneert een tekenreeks die bestaat uit de laatste tekens (meest rechts) van de invoertekenreeks, waarbij het aantal tekens wordt bepaald door het tweede argument.

```
Right (text, count)
```

### RTrim

**RTrim()** retourneert de invoertekenreeks, ontdaan van eindspaties.

```
RTrim (text)
```

### SubField

**Subfield()** wordt gebruikt voor het extraheren van onderdelen van subtekenreeksen van een bovenliggend tekenreeksveld, waarbij de oorspronkelijke recordvelden bestaat uit twee of meer onderdelen met een scheidingsteken ertussen.

```
SubField (text, delimiter[, field_no ])
```

### SubFieldRegEx

**SubFieldRegEx()** extraheert tekst uit een tekenreeksuitdrukking en gebruikt het opgegeven reguliere expressiepatroon als scheidingsteken. De functie retourneert een nulwaarde als er geen overeenkomsten worden gevonden.

```
SubFieldRegEx (text, regex_delimiter [, field_no])
```

#### SubStringCount

**SubstringCount()** retourneert het aantal keren dat de opgegeven subtekenreeks voorkomt in de tekst van de invoertekenreeks. Als er geen overeenkomst wordt gevonden, wordt 0 geretourneerd.

```
SubStringCount (text, substring)
```

#### TextBetween

**TextBetween()** retourneert de tekst in de invoertekenreeks die tussen de tekens staat die zijn opgegeven als scheidingstekens.

```
TextBetween (text, delimiter1, delimiter2[, n])
```

#### Trim

**Trim()** retourneert de invoertekenreeks, ontdaan van alle begin- en eindspaties.

```
Trim (text)
```

#### Upper

**Upper()** converteert alle tekens in de invoertekenreeks naar hoofdletters voor alle teksttekens in de uitdrukking. Cijfers en symbolen worden genegeerd.

```
Upper (text)
```

## Werken met reguliere expressies in load-scripts en diagramuitdrukkingen

Verbeter uw analyses en gegevensvoorbereiding door reguliere expressies (regex) te gebruiken in load-scripts voor gegevens en in diagramuitdrukkingen. Reguliere expressies bieden geavanceerde, flexibele en krachtige opties voor het vinden en vervangen van op tekst gebaseerde gegevens. Om met reguliere expressies te werken in Qlik Sense, gebruikt u de beschikbare script- en diagramfuncties.

### Werken met reguliere expressies

Reguliere expressies (regex) bieden uitgebreide opties voor geavanceerde patroonherkenning. Schrijf een reguliere expressie om een bepaald tekstpatroon op te geven dat u wilt identificeren. U wilt bijvoorbeeld e-mailadressen of web-URL's vinden binnen grotere bestanden in natuurlijke taal.

Er zijn veel verschillende formaten waarin reguliere expressies geschreven kunnen worden, elk met hun eigen syntaxisregels. De beschikbare functies voor reguliere expressies gebruiken de syntaxis van Perl reguliere expressies.

Om reguliere expressies te gebruiken in Qlik Sense, gebruikt u de beschikbare regex script- en diagramfuncties. Het gebruik van deze functies veronderstelt dat u een basiskennis hebt van hoe reguliere expressies werken. Voor meer informatie en voorbeelden raadpleegt u *Beschikbare functies en voorbeelden (page 1724)*.

### Toepassingen

U kunt reguliere expressies gebruiken voor:

- Informatie extraheren en scheiden uit tekst die meerdere afzonderlijke componenten bevat. Bijvoorbeeld e-mailadressen, telefoonnummers, web-URL's en meer.
- Opmaak van gegevens standaardiseren.
- Tekst zoeken en/of vervangen.

Met de beschikbare regexfuncties kunt u het volgende doen in load-scripts:

- Gegevens transformeren en laden voor gebruik in analyses.
- Valideer gegevens om ervoor te zorgen dat ze de opmaak en andere nalevingsnormen volgen.
- Persoonlijk identificeerbare informatie (PII) identificeren, maskeren of wijzigen in een vorm die geschikt is voor consumenten van uw analyse-inhoud.
- Laad gegevens selectief in apps op basis van of ze al dan niet overeenkomen met specifieke tekstpatronen.

Gebruikssituaties voor regexfuncties in diagramuitdrukkingen zijn vergelijkbaar met die voor load-scripts. Door de functies in diagramuitdrukkingen te gebruiken, kunt u ook:

- Transformeer gegevens direct, voorwaardelijk of statisch, afhankelijk van de behoeften en interacties van de gebruiker.
- Gebruik de functies om labels, kleurinstellingen en andere diagram-specifieke eigenschappen te definiëren.

## Prestatie-overwegingen:

De functies voor reguliere expressies die beschikbaar zijn in load-scripts en diagramuitdrukkingen worden het best gebruikt voor geavanceerde tekenreekstransformaties waarbij flexibiliteit een vereiste is. Het gebruik van regexfuncties kost meer middelen dan de alternatieven zonder regex. Voor meer eenvoudige gegevenstransformaties kunt u andere beschikbare script- en diagramfuncties gebruiken, inclusief tekenreeksfuncties. Zie *Tekenreeksfuncties (page 1718)* voor een lijst met beschikbare tekenreeksfuncties.

Wanneer u met grote gegevensvolumes werkt, kunnen regexbewerkingen om te zoeken naar overeenkomsten van grote of negatieve **n** waarden verminderde prestaties veroorzaken. In de context van de beschikbare regexfuncties komt de waarde **n** overeen met functieargumenten zoals **occurrence**, **count** en **field_no**.

## Beschikbare functies en voorbeelden


- *CountRegEx (page 1730)*
- *ExtractRegEx (page 1740)*
- *ExtractRegExGroup (page 1746)*
- *IndexRegEx (page 1777)*
- *IndexRegExGroup (page 1781)*
- *IsRegEx (page 1790)*
- *MatchRegEx (page 1824)*
- *ReplaceRegEx (page 1845)*

- *ReplaceRegExGroup (page 1851)*
- *SubFieldRegEx (page 1870)*

## Capitalize

De **Capitalize()** functie converteert het eerste teken van elk woord in een teksttekenreeks naar hoofdletters en converteert alle andere tekens naar kleine letters.

**Syntaxis:**

```
Capitalize(text)
```

**Retourgegevenstypen:** tekenreeks

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De tekenreeks die moet worden geëvalueerd. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `Capitalize( 'star trek' )` | Retourneert `Star Trek` |
| `Capitalize( 'AA bb cC Dd' )` | Retourneert `Aa Bb Cc Dd` |

### Voorbeeld - Basisprincipes voor Capitalize

Diagramuitdrukking

**Overzicht**

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `TextToCapitalize`.

**Load-script**

```
Example:
Load * inline [
TextToCapitalize
50 Cent
a-ha
BIGBANG
girl in red
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `TextToCapitalize`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=Capitalize(TextToCapitalize)` om te berekenen welke tekens u met een hoofdletter moet schrijven.

Resultatentabel

| TextToCapitalize | Capitalize(TextToCapitalize) |
|---|---|
| 50 Cent | 50 Cent |
| a-ha | A-Ha |
| BIGBANG | Bigbang |
| girl in red | Girl In Red |

Er is geen wijziging in de eerste rij gegevensverzamelingen omdat de eerste reeks tekens met een cijfer begint en de tweede al een hoofdletter is.

In de tweede rij worden het eerste en derde teken met een hoofdletter geschreven.

In de derde rij wordt alleen de eerste letter met een hoofdletter geschreven, en alle volgende letters met een kleine letter.

In de vierde rij wordt de eerste letter van elk woord met een hoofdletter geschreven.

### Load-script

De volgende code laat zien hoe u de functie in een load-script kunt gebruiken.

```
Load
String,
Capitalize(String)
Inline
[String
rHode iSland
washingTon d.C.
new york];
```

Resultatentabel

| Tekenreeks | Capitalize(String) |
|---|---|
| rHode iSland | Rhode Island |
| washingTon d.C. | Washington D.C. |
| new york | New York |

## Voorbeeld - Scenario met Capitalize

Diagramuitdrukking

### Overzicht

Een gegevensverzameling met klantnamen wordt ingevoerd in verschillende formaten, zoals allemaal kleine letters, allemaal hoofdletters, of een combinatie van hoofdletters en kleine letters. De functie `Capitalize` wordt gebruikt om de gegevens op te schonen zodat de voor- en achternaam van de klant de juiste hoofdletters krijgen wanneer ze op het dashboard worden weergegeven.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `CustomerName`.

### Load-script

```
Example:
Load * inline [
CustomerName
john smith
JANE DOE
miCHAel Brown
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `CustomerName`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=Capitalize(CustomerName)` om de klantnamen met een hoofdletter te schrijven.

Resultatentabel

| CustomerName | Capitalize(CustomerName) |
|---|---|
| JANE DOE | Jane Doe |
| john smith | John Smith |
| miCHAel Brown | Michael Brown |

Vergelijk de uitvoer van de functie `Capitalize` met de oorspronkelijke `CustomerName` tekenreekswaarden die in het script geladen zijn. U kunt zien hoe de functie met succes de namen heeft geconverteerd naar

het juiste gebruik van hoofdletters en kleine letters. Het weergeven van klantnamen met gestandaardiseerd hoofdlettergebruik verbetert het uiterlijk, de consistentie en de leesbaarheid van het dashboard.

# Chr

**Chr()** retourneert het Unicode-teken dat overeenkomt met het ingevoerde gehele getal.

**Syntaxis:**

```
Chr(int)
```

**Retourgegevenstypen:** tekenreeks

Argumenten

| Argument | Beschrijving |
|---|---|
| **int** | Het geheel getal om te evalueren. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| Chr(65) | Retourneert de tekenreeks `A` |
| Chr(163) | Retourneert de tekenreeks `£` |
| Chr(35) | Retourneert de tekenreeks `#` |

## Voorbeeld - Basisprincipes voor Chr

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `InputInt`. Het veld bevat de gehele waarde die moet worden geëvalueerd.

### Load-script

```
Example:
Load * inline [
InputInt
65
163
35
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `InputInt`

Maak de volgende meting:

- `=Chr(InputInt)`, om de waarden in `InputInt` te berekenen en een Unicode-teken te retourneren.

Resultatentabel

| InputInt | Chr(InputInt) |
|---|---|
| 35 | # |
| 65 | A |
| 163 | £ |

De eerste rij retourneert de tekenreeks `#`, aangezien dat het Unicode-teken is dat overeenkomt met de waarde 35.

De tweede rij retourneert de tekenreeks `A`, aangezien dat het Unicode-teken is dat overeenkomt met de waarde 65.

De derde rij retourneert de tekenreeks `£`, aangezien dat het Unicode-teken is dat overeenkomt met de waarde 163.

## Voorbeeld – Scenario met Chr

Diagramuitdrukking

#### Overzicht

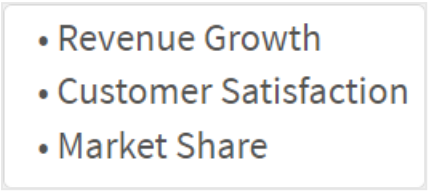
Dit voorbeeld laat zien hoe u de functie `Chr` kunt gebruiken om een lijst met opsommingstekens te maken van de teksttekenreeksen in een tekstobject.

#### Resultaten

Open een werkblad. Maak een nieuw tekstobject en voeg de volgende code toe aan een meting:

- `= Chr(149) & ' Revenue Growth' & Chr(10) & Chr(149) & ' Customer Satisfaction' & Chr(10) & Chr(149) & ' Market Share'`

*Uitvoer van de metinguitdrukking in tekstobject*




De uitvoer wordt op de volgende manier afgeleid :

- `Chr(149)` geeft een opsommingsteken (•).
- `Chr(10)` voegt een regeleinde (`newline`) in tussen de items.
- De ampersand (`&`) is een aaneengeschakelde tekenreeksoperator die de afzonderlijke stukken tekst aaneenschakelt tot één enkele teksttekenreeks.

## CountRegEx

**CountRegEx()** retourneert het aantal keren dat het opgegeven reguliere expressiepatroom voorkomt in de tekst van de invoertekenreeks. Als er geen overeenkomst wordt gevonden, wordt 0 geretourneerd.

Deze functie voert regexbewerkingen uit die hoofdlettergevoelig zijn. U kunt eventueel ook de variant **CountRegExI()** gebruiken om hoofdletteronafhankelijke regexbewerkingen uit te voeren.

**Syntaxis:**

```
CountRegEx (text, regex)
```

**Retourgegevenstypen:** numeriek

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De tekst van de invoertekenreeks waarin u wilt zoeken naar een reguliere expressie. |
| **regex** | Het reguliere expressiepatroon om naar te zoeken. |

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `CountRegEx('abc123','[a-z]')` | Retourneert `3`. |
| `CountRegEx('abc123','[a-z][0-9]')` | Retourneert `1`. |
| `CountRegEx('abc123','[0-9]')` | Retourneert `3`. |
| `CountRegEx('ABC','[a-z]{3}')` | Retourneert `0`. |
| `CountRegExI('ABC','[a-z]{3}')` | Retourneert `1`. Omdat de variant **CountRegExI()** wordt gebruikt, zijn zoekopdrachten in tekst niet hoofdlettergevoelig. |

## Wanneer gebruiken

Voorbeelden van gebruiksscenario's voor **CountRegEx()** zijn onder andere:

- Het tellen van het aantal keren dat een bepaald tekstpatroon in een tekenreeks voorkomt. U kunt bijvoorbeeld zoeken naar voorbeelden van e-mailadressen, telefoonnummers en andere informatie in documenten en e-mailberichten.
- Valideren of gegevens voldoen aan een specifieke syntaxisvereiste. U kunt bijvoorbeeld fouten bij het invoeren van gegevens identificeren.

## Voorbeeld 1 - load-script om geldige jaarbereiken te tellen

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een container met de naam `Vehicles` die een lijst met voertuigen bevat, samen met de staat en het modeljaar van elk voertuig.
- Voor elke record in `Vehicles`, een berekening om te bepalen of het jaarbereik voor het voertuig voldoet aan de verwachte syntaxis. Geldige jaarbereiken zijn 1990, 2000, 2010 en 2020. Deze berekening wordt uitgevoerd met de functie **CountRegEx()**.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `Vehicle_ID`
- `Valid_Year_Data`

Resultatentabel

| Vehicle_ID | Valid_Year_Data |
|---|---|
| 1 | 0 |
| 2 | 1 |
| 3 | 1 |
| 4 | 0 |
| 5 | 1 |
| 6 | 1 |
| 7 | 0 |
| 8 | 1 |
| 9 | 0 |
| 10 | 1 |

U kunt ook een KPI aan het werkblad toevoegen en deze meting toevoegen:

```
Sum(Valid_Year_Data)
```

Dit diagram toont het totale aantal records met geldige *Model_YearRange* waarden.

*De functie **MatchRegEx()** is ook nuttig voor dit soort validatiescenario's.*

## Voorbeeld 2 - load-script om het aantal e-mailadressen in teksttekenreeksen te tellen

### Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een container met de naam `Employees` met gegevens over de werknemers van een bedrijf. Alle e-mailadressen voor elke werknemer staan in een enkele tekenreeks.
- Het maken van een veld, `NoOfEmails`, om te berekenen hoeveel e-mailadressen de invoertekenreeks voor elke werknemer bevat.

#### Load-script

```
Employees:
```

```
Load CountRegExI(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');
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- ID
- Emails
- NoOfEmails

Resultatentabel

| ID | Emails | NoOfEmails |
|---|---|---|
| 1 | john_doe@test.com john_michael_doe@example.com 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 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 |

## Voorbeeld 3 - diagramuitdrukking om exemplaren van een tekenreeks te tellen

Diagramuitdrukking en resultaten

#### Overzicht

Open de Editor voor laden van gegevens en voeg het volgende load-script toe aan een nieuwe sectie.

Het load-script bevat de tekst van een aantal bedrijfscorrespondenties die naar werknemers van een organisatie zijn gestuurd. De gegevens zijn al ontdaan van alle persoonlijk identificeerbare informatie (PII), behalve de bedrijfsnaam en het e-mailadres dat elke correspondentie heeft verzonden. De inhoud van elk bericht is samengevoegd tot een enkele doorlopende tekenreeks. Met andere woorden, alle opmaak- en newline-tekens zijn verwijderd.

Onze vereisten:

- Identificeer welke berichten een tekenreeks bevatten die het e-mailadres *TestCompanyNameCorrespondence@test.com* bevat, samen met een specifieke inleiding tot dit adres die aangeeft dat dit de afzender was.
- We willen flexibiliteit in welke introducties we kunnen identificeren, maar we willen ook een aantal duidelijke beperkingen hebben. De inleidingen `Sender` en `Sent by` worden beide als geldig beschouwd. Om dit doel te bereiken, kunnen regexpatronen worden gebruikt.
- Hoofdlettergevoeligheid van e-mailadressen speelt geen rol. Daarom moet de zoekopdracht hoofdletterongevoelig zijn.

### Load-script

```
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 |);
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `ID`
- `MessageBody`

Voeg de volgende meting toe:

```
=CountRegExI(MessageBody,'(Sender.*TestCompanyNameCorrespondence@test.com)|(Sent by.*TestCompanyNameCorrespondence@test.com)')
```

Stel in de eigenschappen van de meting **de functie Totalen** in op **Sum**. Dit configureert de totalenrij bovenaan de tabel om het totale aantal records te tellen die bijdragen aan de telling.

De meting telt het aantal overeenkomsten voor het gespecificeerde regexpatroon binnen de berichtinhoud. In het regexpatroon geeft `.*` een willekeurig aantal tekens aan, behalve newline-tekens. Het patroon houdt ook rekening met variatie in de manier waarop het e-mailadres wordt ingevoerd: zowel `Sender` als `Sent by` zijn geldige overeenkomsten. De **CountRegExI()** variant van de functie zorgt voor hoofdletterongevoelige zoekopdrachten.

Resultatentabel

| ID | MessageBody | =CountRegExI(MessageBody,'(Sender.*TestCompanyNameCorrespondence@test.com)\|(Sent by.*TestCompanyNameCorrespondence@test.com)') |
|---|---|---|
| **Totals** | - | **3** |
| 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! | 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 |

| ID | MessageBody | =CountRegExl(MessageBody,'(Sender.*TestCompanyNameCorrespondence@test.com)\|(Sent by.*TestCompanyNameCorrespondence@test.com)') |
|---|---|---|
| 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 | 1 |

In de vijf berichten zijn er drie die overeenkomen met het gespecificeerde regexpatroon. Elk bericht bevat één overeenkomst.

*De functie **MatchRegEx()** is ook nuttig voor dit soort validatiescenario's.*

## Evaluate

**Evaluate()** bepaalt of de tekenreeks van de invoertekst kan worden geëvalueerd als een geldige Qlik Sense-uitdrukking en retourneert, als dat inderdaad het geval is, de waarde van de uitdrukking als een tekenreeks. Als de invoertekenreeks geen geldige uitdrukking is, wordt NULL geretourneerd.

**Syntaxis:**

```
Evaluate(expression_text)
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De tekenreeks die moet worden geëvalueerd. |

*Deze tekenreeksfunctie kan niet in diagramuitdrukkingen worden gebruikt.*

Functievoorbeeld en resultaten

| Functievoorbeeld | Resultaat |
|---|---|
| `Evaluate( 5 * 8 )` | Retourneert 40 |

## Voorbeeld - Basisprincipes voor Evaluate

Load-script

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `ExpressionText`.
- Een voorgaande lading die de functie `Evaluate` gebruikt om het veld `ExpressionText` te evalueren en twee nieuwe velden te retourneren: `Evaluated` en `Evaluatedx4`.

### Load-script

```
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
0123456.551247999457895236485934646 9
Today()
Bob
];
```

### Resultaten

De volgende velden worden in de gegevenstabel geladen:

- `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 |
| 0123456.55124799945789523648593 46469 | 0123456.55124799945789523648593 46469 | 493826.204992 |
| Today() | 11/5/2024 | 182404 |
| Bob | - | - |

De uitvoer van de functie `Evaluate` retourneert waarden voor alle `ExpressionText` tekenreeksen, behalve de laatste rij. De tekenreeks `Bob` kan niet worden geëvalueerd, dus de functie retourneert geen waarde - NULL. Alle rijen zijn geëvalueerd, bijvoorbeeld de tweede rij voegt `5` en `3` toe om `8` te retourneren. Sommige van de `ExpressionText` tekenreeksen retourneren de tekenreeks in de huidige staat omdat de tekenreeks al evalueert naar een getal. Bijvoorbeeld, de tekenreeks `0123456789012345678` evalueert naar hetzelfde getal.

Hoewel Qlik Sense een precisielimiet van 14 cijfers voor numerieke waarden heeft, evalueerde de functie `Evaluate` tekenreeksen van 18 cijfers als een getal en paste vermenigvuldiging toe binnen het script. Dit is handig voor het verwerken van zeer grote getallen in het script.

## Voorbeeld – Scenario met Evaluate

Load-script

**Overzicht**

Een gegevensverzameling met verkoopgegevens bevat producten, prijzen en kortingen. Dit voorbeeld leidt de gereduceerde prijs voor de producten af.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Product`
  - `Price`
  - `Discount %`
- Een voorafgaande lading die de waarden `Price` en `Discount %` neemt en een extra veld met de naam `DiscountedPrice` berekent. Dit nieuwe veld gebruikt de functie `Evaluate` om de prijs minus de korting te berekenen.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `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 |

De uitvoer van de functie `Evaluate` laat zien hoe u de functie kunt gebruiken om waarden te berekenen.

## ExtractRegEx

**ExtractRegEx()** extraheert tekst uit een tekenreeksuitdrukking met behulp van het opgegeven reguliere expressiepatroon. De functie retourneert een nulwaarde als er geen overeenkomsten worden gevonden.

Deze functie voert regexbewerkingen uit die hoofdlettergevoelig zijn. U kunt eventueel ook de variant **ExtractRegExI()** gebruiken om hoofdletteronafhankelijke regexbewerkingen uit te voeren.

**Syntaxis:**

```
ExtractRegEx (text, regex [, field_no])
```

**Retourgegevenstypen:** tekenreeks

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | Tekenreeksuitdrukking die de tekst bevat die in de retourwaarde moet worden geëxtraheerd. |
| **regex** | De reguliere expressie om te gebruiken voor het extraheren van tekst. |
| **field_no** | Het nummer van de overeenkomst die moet worden uitgepakt. Dit is handig als er meerdere overeenkomsten voor de reguliere expressie in de tekst kunnen worden gevonden. Geef bijvoorbeeld de waarde `4` op om de vierde overeenkomst te extraheren. Dit is een optioneel argument. Geef een negatieve waarde op om de volgorde van de overeenkomsten om te keren.<br><br>Het volgende is van toepassing ongeacht of de functie wordt gebruikt in een load-script of diagramuitdrukking:<br><br>• Als **field_no** positief is, zal de functie één waarde retourneren, die de overeenkomsten van links naar rechts identificeert.<br>• Als **field_no** negatief is, zal de functie één waarde retourneren, die de overeenkomsten van rechts naar links identificeert.<br><br>Wanneer u de functie in een load-script gebruikt:<br><br>• Als u de functie **ExtractRegEx()** in een **LOAD**-instructie gebruikt en **field_no** is weggelaten, genereert de functie evenveel records als er overeenkomsten zijn.<br>• Als meerdere velden worden geladen met **ExtractRegEx()** en bij geen darvan een **field_no** argument is opgegeven, worden de cartesische producten van alle combinaties gemaakt.<br><br>Wanneer u de functie in een diagramuitdrukking gebruikt:<br><br>• Als **field_no** is weggelaten, is de standaardwaarde `1`. |

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `ExtractRegEx('a,b c;1 2,3','[a-z]\|[0-9]', 1)` | Retourneert `a` wat de eerste gevonden overeenkomst is. |
| `ExtractRegEx('a,b c;1 2,3','[a-z]\|[0-9]', 2)` | Retourneert `b` wat de tweede gevonden overeenkomst is. |
| `ExtractRegEx('a,b c;1 2,3','[a-z]\|[0-9]', 4)` | Retourneert `1` wat de vierde gevonden overeenkomst is. |
| `ExtractRegEx ('abc','a\|b',-1)` | Retourneert `b`. Als er een negatieve waarde wordt opgegeven, worden de overeenkomsten in omgekeerde volgorde geteld. |
| `ExtractRegEx('A,B C;1 2,3','[a-c]\|[4-9]')` | Retourneert een nulwaarde, omdat **ExtractRegEx()** hoofdlettergevoelig is, en de hoofdletterongevoelige variant niet is gebruikt. |
| `ExtractRegExI('A,B C;1 2,3','[a-c]\|[4-9]')` | Retourneert `A`, omdat de hoofdletterongevoelige variant **ExtractRegExI()** is gebruikt. |

## Wanneer gebruiken

U kunt **Extract RegEx()** gebruiken om informatie te extraheren die u wilt isoleren uit gegevens die mogelijk ook andere informatie bevatten (bijvoorbeeld vrije tekst of JSON-tekenreeksen). Bijvoorbeeld:

- Extraheer e-mailadressen, telefoonnummers, rekeningnummers en andere informatie uit tekst.
- Extraheer numerieke waarden uit tekst (bijvoorbeeld valuta).
- Standaardiseer de opmaak van tekst of numerieke gegevens.

## Voorbeeld 1 - load-script om e-mailadressen uit JSON te extraheren

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een tabel `OrderDetails` met een veld `OrderJson`.
- Het veld `OrderJson` bevat een JSON-tekenreeks met bestelgegevens. Opmaak, zoals inspringen en regeleinden, is verwijderd uit de JSON. De bestelgegevens bevatten e-mailadressen die we in afzonderlijke waarden willen uitpakken.
- Een nieuwe tabel `EmailAddresses`, die wordt geladen na `OrderDetails`. In de `EmailAddresses` **LOAD**-instructie laden we het veld `OrderJson` uit de tabel `OrderDetails` en maken we een nieuw berekend veld, `Email`, dat de geëxtraheerde e-mailadressen bevat. `OrderDetails` wordt dan verwijderd.

Onze vereisten:

- E-mailadressen in een specifiek domein: `example.com`.
- E-mailadressen die een willekeurig aantal alfanumerieke tekens en een gedefinieerde reeks speciale tekens bevatten.

### Load-script

```
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;
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Email`

Resultatentabel

| Email |
|---|
| amalia.craig@example.com |
| jane.doe@example.com |
| john.doe@example.com |
| ken.roberts@example.com |

Door geen argumentwaarde op te geven voor **field_no**, worden alle overeenkomsten automatisch in afzonderlijke waarden geëxtraheerd.

## Voorbeeld 2 - load-script om numerieke waarden uit valutabedragen te extraheren

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een tabel `Sales` met verkoopgegevens.
- In de tabel `Sales` staat een veld met de naam `Amount`, dat transactiebedragen in USD bevat. Deze bedragen worden uitgedrukt als tekenreeksen, die valutasymbolen en codes bevatten.
- De definitie van een aangepast veld binnen `Sales`, `OrderAmountNumeric_USD`, dat elk bedrag in een zuiver numerieke waarde extraheert. Deze gegevens moeten in numerieke vorm zijn, zodat we ze in diagrammen kunnen gebruiken.

### Load-script

```
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
`;
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `OrderID`
- `Amount`
- `OrderAmountNumeric_USD`

Resultatentabel

| 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 |
| 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 |

## Voorbeeld 3 - diagramuitdrukkingen om e-mailadressen uit JSON te extraheren

### Diagramuitdrukkingen en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een tabel `OrderDetails` met een veld `OrderJson`.
- Het veld `OrderJson` bevat een JSON-tekenreeks met bestelgegevens. Opmaak, zoals inspringen en regeleinden, is verwijderd uit de JSON. De bestelgegevens bevatten e-mailadressen die we in afzonderlijke waarden willen uitpakken.

Onze vereisten:

- E-mailadressen in een specifiek domein: `example.com`.
- E-mailadressen die een willekeurig aantal alfanumerieke tekens en een gedefinieerde reeks speciale tekens bevatten.

### Load-script

```
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');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een tabel en voeg de volgende uitdrukkingen toe als berekende dimensies:

- `=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)`

Resultatentabel

| =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) |
|---|---|---|---|
| john.doe@example.com | jane.doe@example.com | amalia.craig@example.com | ken.roberts@example.com |

## ExtractRegExGroup

**ExtractRegExGroup()** extraheert tekst uit een tekenreeksuitdrukking met behulp van het samengestelde reguliere expressiepatroon dat is opgegeven. Geef bij het gebruik van de functie de groep op die binnen de samengestelde regex moet worden gebruikt. De functie retourneert een nulwaarde als er geen overeenkomsten worden gevonden.

Als u de functie **ExtractRegExGroup()** in een **LOAD**-instructie gebruikt en **field_no** is weggelaten, dan zal de functie meerdere records retourneren. Als meerdere velden worden geladen met **ExtractRegExGroup ()**, worden de cartesische producten van alle combinaties gemaakt.

Deze functie voert regexbewerkingen uit die hoofdlettergevoelig zijn. U kunt eventueel ook de variant **ExtractRegExGroupI()** gebruiken om hoofdletteronafhankelijke regexbewerkingen uit te voeren.

**Syntaxis:**

```
ExtractRegExGroup (text, regex, group [, field_no])
```

**Retourgegevenstypen:** tekenreeks

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | Tekenreeksuitdrukking die de tekst bevat die in de retourwaarde moet worden geëxtraheerd. |
| **regex** | De reguliere expressie om te gebruiken voor het extraheren van tekst. |
| **group** | Het nummer van de groep, in het geval van een samengestelde reguliere expressie.<br><br>Als de reguliere expressie slechts één groep bevat, gebruik dan in plaats daarvan de functie **ExtractRegEx()**. Als alternatief kunt u **ExtractRegExGroup()** gebruiken met een **group** waarde van `0`.<br><br>U kunt een negatieve **group** waarde opgeven om van rechts naar links naar overeenkomsten te zoeken. |
| **field_no** | Het nummer van de overeenkomst die moet worden uitgepakt. Dit is handig als er meerdere overeenkomsten voor de reguliere expressie in de tekst kunnen worden gevonden. Geef bijvoorbeeld de waarde `4` op om de vierde overeenkomst te extraheren.<br><br>Dit is een optioneel argument. De standaardinstelling is 1 als deze niet is opgegeven. |

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `ExtractRegExGroup('abc123 def456','([a-z]+) ([0-9]+)',1)` | Retourneert `abc` (eerste groep van eerste overeenkomst). |

| Voorbeeld | Resultaat |
|---|---|
| `ExtractRegExGroup('abc123 def456','([a-z]+) ([0-9]+)',1,2)` | Retourneert `def` (eerste groep van tweede overeenkomst). |
| `ExtractRegExGroup('abc123 def456','([a-z]+) ([0-9]+)',2)` | Retourneert `123` (tweede groep van eerste overeenkomst). |
| `ExtractRegExGroup('abc123 def456','([a-z]+) ([0-9]+)',2,2)` | Retourneert `456` (tweede groep van tweede overeenkomst). |

## Wanneer gebruiken

U kunt **Extract RegEx()** gebruiken om informatie te extraheren die u wilt isoleren uit gegevens die mogelijk ook andere informatie bevatten (bijvoorbeeld vrije tekst of JSON-tekenreeksen). Bijvoorbeeld:

- Extraheer e-mailadressen, telefoonnummers, rekeningnummers en andere informatie uit tekst.
- Extraheer numerieke waarden uit tekst (bijvoorbeeld valuta).
- Standaardiseer de opmaak van tekst of numerieke gegevens.

## Voorbeeld 1 - load-script om transactiecodes te parseren

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een tabel met de naam `Transactions`, die informatie over transacties bevat. Bepaalde details worden vastgelegd met transactiecodes met behulp van de volgende syntaxis: *Transactiejaar-Transactiebron* (d.w.z. online aankoop of aankoop in de winkel)-*Gekoppeld distributiecentrum*
- Het aanmaken van verschillende nieuwe velden om elk detail te extraheren, waarbij telkens dezelfde samengestelde reguliere expressie wordt hergebruikt.

Onze vereisten:

- Het jaar kan een willekeurige combinatie van vier cijfers zijn.
- De waarden *ONLINE* en *INSTORE* zijn de enige aanvaardbare waarden voor transactiebron.
- Het distributiecentrum moet precies vijf nummers hebben.

### Load-script

```
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
`;
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `RecordID`
- `TransactionCode`
- `TransactionCode_Unparsed`
- `TransactionYear`
- `TransactionSource`
- `TransactionDC`

Resultatentabel

| 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 |

Deze resultaten laten zien hoe u met het argument **group** een enkele regex kunt hergebruiken voor meerdere bewerkingen. Het veld `TransactionCode_Unparsed`, dat een **group** waarde van 0 gebruikt, biedt in dit geval geen extra waarde, maar wordt hier weergegeven om de functie te demonstreren.

## Voorbeeld 2 - load-script om zakelijke contactgegevens te extraheren

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een tabel met de naam `BusinessContactInfo`, die informatie over bedrijven bevat die gevonden werd op de website van elk bedrijf.
- `ContactInfo` is een veld dat vrije tekst bevat, inclusief e-mailadressen en telefoonnummers van bedrijven.
- Het aanmaken van verschillende nieuwe velden om elk detail te extraheren, waarbij telkens dezelfde samengestelde reguliere expressie wordt hergebruikt.

Onze vereisten:

- E-mailadressen moeten voldoen aan een specifieke reeks vereisten en syntaxis.
- Telefoonnummers moeten NANP-telefoonnummers met 10 cijfers zijn. We willen toestaan dat het netnummer tussen haakjes komt te staan en op bepaalde punten spaties en koppeltekens toestaan.

### Load-script

```
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');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `CompanyName`
- `CompanyEmail`
- `CompanyPhoneNum`

Resultatentabel

| 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 |

Dezelfde samengestelde reguliere expressie wordt hergebruikt om verschillende informatie op te halen. Het argument **group** specificeert welke van de twee groepen van de reguliere expressie gezocht moet worden, en het argument **field_no** specificeert welke algemene overeenkomst (over de hele zoekreeks) we willen vinden.

De **ExtractRegEGroupxI()** variant van de functie zorgt voor hoofdletterongevoelige zoekopdrachten.

## Voorbeeld 3 - load-script om componenten uit ISBN-codes te extraheren

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Het aanmaken van een variabele, `ISBN_RegEx`, om de reguliere expressie op te slaan die we voor alle extracties willen gebruiken.
- Een set ISBN-codes voor boeken die een boekwinkel bij verschillende leveranciers wil bestellen.

We moeten het volgende uit elke ISBN-code extraheren:

- EAN
- Groep
- Registrant
- Publicatie
- Controlesom

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `Supplier`
- `EAN`
- `Group`
- `Registrant`
- `Publication`
- `Checksum`

Resultatentabel

| 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 |

## Voorbeeld 4 - diagramuitdrukkingen om zakelijke contactgegevens te extraheren (met vergelijking ExtractRegEx() )

Diagramuitdrukkingen en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een tabel met de naam `BusinessContactInfo`, die informatie over bedrijven bevat die gevonden werd op de website van elk bedrijf.
- `ContactInfo` is een veld dat vrije tekst bevat, inclusief e-mailadressen en telefoonnummers van bedrijven. We willen elk e-mailadres en telefoonnummer extraheren met diagramuitdrukkingen.

Onze vereisten:

- E-mailadressen moeten voldoen aan een specifieke reeks vereisten en syntaxis.
- Telefoonnummers moeten NANP-telefoonnummers met 10 cijfers zijn. We willen toestaan dat het netnummer tussen haakjes komt te staan en op bepaalde punten spaties en koppeltekens toestaan.

### Load-script

```
BusinessContactInfo:

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');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `CompanyName`

Voeg de volgende berekende dimensies toe aan de tabel:

- `=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)`
- `=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)`

Resultatentabel

| CompanyName | =ExtractRegExGroupI(ContactInfo, '([a-zA-Z0-9!#$%^&*-_+=~{\|}\V."]+@[a-zA-Z0-9!#$%^&*-_+=~{\|}\V."]{1,50}\.[a-zA-Z0-9!#$%^&*\-_+=~{\|}\V."]{1,50})\|(\({0,1}[0-9]{3}\){0,1}[ -]*[0-9]{3}[ -]*[0-9]{4})',1,1) | =ExtractRegExGroupI(ContactInfo, '([a-zA-Z0-9!#$%^&*-_+=~{\|}\V."]+@[a-zA-Z0-9!#$%^&*-_+=~{\|}\V."]{1,50}\.[a-zA-Z0-9!#$%^&*\-_+=~{\|}\V."]{1,50})\|(\({0,1}[0-9]{3}\){0,1}[ -]*[0-9]{3}[ -]*[0-9]{4})',2,2) |
|---|---|---|
| Company A | Company1@example.com | (123) 456-7890 |
| Company B | company2@test.com | 0123456790 |
| Company C | company3@placeholder.com | 234-567-8901 |

Dezelfde samengestelde reguliere expressie wordt hergebruikt om verschillende informatie op te halen. Het argument **group** specificeert welke van de twee groepen van de reguliere expressie gezocht moet worden, en het argument **field_no** specificeert welke algemene overeenkomst (over de hele zoekreeks) we willen vinden.

De **ExtractRegEGroupI()** variant van de functie zorgt voor hoofdletterongevoelige zoekopdrachten.

## Voorbeeld 5 - URL-parseren (met **ExtractRegEx()** vergelijking)

Diagramuitdrukkingen en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een tabel met de naam `Correspondence`, die tekst van een e-mailbericht bevat in een veld met de naam `EmailBody`.
- Inhoud van e-mailberichten die web-URL's bevatten.

We willen de tweede URL, indien aanwezig, uitpakken in de volgende componenten:

- Volledige URL
- Protocol
- Domein
- Pad

We willen één reguliere expressie gebruiken om alle extracties te regelen.

### Load-script

```
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');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `ID`

Voeg deze berekende dimensies toe:

- *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)`
- *URL 2 Protocol*:
  `=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)`
- *URL 2 Domein*:
  `=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)`
- *URL 2 Pad*:
  `=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)`

Resultatentabel

| Id | URL 2 | URL 2 Protocol | URL 2 Domein | URL 2 Pad |
|---|---|---|---|---|
| 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()** retourneert de hele URL, terwijl **ExtractRegExGroup()** individuele delen van de URL retourneert, die overeenkomen met de **group** waarde die we hebben gebruikt. De record met een `ID` waarde van 2 bevat slechts één URL, dus hiervoor worden geen gegevens geretourneerd.

Dezelfde reguliere expressie wordt in alle diagramuitdrukkingen gebruikt om verschillende stukjes informatie te retourneren. Een uitsplitsing van de groepen die in de reguliere expressie zijn gedefinieerd, is als volgt.

Groepen in reguliere expressie

| Groepsnummer | Reguliere expressie | URL-gedeelte |
|---|---|---|
| 1 | ([a-zA-Z0-9]+) | Protocol |
| 2 | (([a-zA-Z0-9]+\.)*([a-zA-Z0-9]+)\.([a-zA-Z0-9]+)) | Domein (groepen 3, 4 en 5) |
| 3 | ([a-zA-Z0-9]+\.)* | Hoofddomein |
| 4 | ([a-zA-Z0-9]+) | Topniveaudomein |
| 5 | ([a-zA-Z0-9]+) | Pad (groepen 7, 8, 9 en 10) (optioneel) |
| 6 | ((\/[a-zA-Z0-9]+)*(\/([a-zA-Z0-9]+)(\.[a-zA-Z0-9]+)?))? | Submap |
| 7 | (\/[a-zA-Z0-9]+)* | Bestand (groepen 9 en 10) |
| 8 | (\/([a-zA-Z0-9]+)(\.[a-zA-Z0-9]+)?) | Bestand |
| 9 | ([a-zA-Z0-9]+) | Bestandsnaam |
| 10 | (\.[a-zA-Z0-9]+)? | Bestandsextensie (optioneel) |

## FindOneOf

**FindOneOf()** doorzoekt een tekenreeks om de positie te vinden van enig teken uit de reeks van opgegeven tekens. De positie van het N-de exemplaar van een teken uit de zoekset wordt geretourneerd, waarbij N de optionele derde parameter van de functie is. Als er geen derde parameter wordt opgegeven, wordt het eerste exemplaar geretourneerd. Als geen waarde wordt gevonden, wordt **0** geretourneerd.

**Syntaxis:**

```
FindOneOf(text, char_set[, count])
```

**Retourgegevenstypen:** geheel getal

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De oorspronkelijke tekenreeks. |
| **char_set** | Een reeks tekens om naar te zoeken in **text**. |
| **count** | Definieert naar welk exemplaar van de tekens moet worden gezocht. Bijvoorbeeld, bij een waarde van 2 wordt naar het tweede exemplaar gezocht. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `FindOneOf( 'my example text string', 'et%s' )` | Retourneert `4` omdat `e` het vierde teken is in de voorbeeldtekenreeks. |
| `FindOneOf( 'my example text string', 'et%s', 3 )` | Retourneert `12` omdat wordt gezocht naar een van de tekens `e`, `t`, `%` of `s` en `t` het derde exemplaar is op positie `12` van de voorbeeldtekenreeks. |
| `FindOneOf( 'my example text string', '¤%&' )` | Retourneert `0` omdat geen van de tekens `¤`, `%` of `&` voorkomt in de voorbeeldtekenreeks. |

## Voorbeeld - Basisprincipes voor FindOneOf

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Een veld in de gegevenstabel genaamd `Phrase`, dat de originele teksttekenreeks bevat die verwerkt moeten worden.

### Load-script

```
Example:
Load *
Inline
[Phrase
Many tiny beads
For a very long time
Has the potential for growth
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Phrase`

Maak de volgende berekende dimensies:

- `=FindOneOf(Phrase, 'e', 1)`
- `=FindOneOf(Phrase, 'y', 1)`
- `=FindOneOf(Phrase, 'ey', 1)`
- `=FindOneOf(Phrase, 'e', 2)`

- `=FindOneOf(Phrase, 'y', 2)`
- `=FindOneOf(Phrase, 'ey', 2)`

Resultatentabel

| Phrase | FindOneOf (Phrase, 'e', 1) | FindOneOf (Phrase, 'y', 1) | FindOneOf (Phrase, 'ey', 1) | FindOneOf (Phrase, 'e', 2) | FindOneOf (Phrase, 'y', 2) | FindOneOf (Phrase, 'ey', 2) |
|---|---|---|---|---|---|---|
| For a very long time | 8 | 10 | 8 | 20 | 0 | 10 |
| Has the potential for growth | 7 | 0 | 7 | 12 | 0 | 12 |
| Many tiny beads | 12 | 4 | 4 | 0 | 9 | 9 |

Voor elke berekende dimensie geeft de uitvoer de positie van de tekens uit de zoekset voor elke zin. Bijvoorbeeld, de letter `e` komt voor het eerst voor op positie `8`, `7`, en `12` voor respectievelijk elke zin. De tweede keer dat de letter `e` in elke zin voorkomt, is op positie `20`, `12` en `0` (niet gevonden). Zo komt de letter '`e` of `y`' voor het eerst voor op positie `8`, `7` en `4`, voor respectievelijk elke zin. Merk op dat in de zin `For a very long time`, de eerste vermelding van '`e` of `y`' wordt gevonden op positie `8`, een `e`, en de tweede vermelding op positie `10`, een `y`.

### Load-script

De volgende code laat zien hoe u de functie in een load-script kunt gebruiken.

```
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, ¤%&,1
];
```

Resultatentabel

| InputText | SearchFor | Exemplaar | FindOneOf_Matches |
|---|---|---|---|
| mijn voorbeeldtekstreeks | et%s | 1 | 4 |
| mijn voorbeeldtekstreeks | et%s | 3 | 12 |
| mijn voorbeeldtekstreeks | ¤%& | 1 | 0 |

## Voorbeeld - FindOneOf scenario

Diagramuitdrukking

### Overzicht

In dit voorbeeld wordt de functie `FindOneOf` gebruikt om opmerkingen van klanten in een gegevensverzameling van klantenorders te analyseren en orders te identificeren waarvoor mogelijk actie moet worden ondernomen. Elke bestelling heeft een veld `CustomerComment` waar klanten notities of opmerkingen over hun bestellingen kunnen achterlaten. Door de opmerkingen te analyseren, kunt u specifieke trefwoorden of tekens identificeren, zoals '!', '@', '#', die op urgentie of speciale verzoeken kunnen duiden.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
    - `TicketID` het identificatienummer van het ticket
    - `CustomerComment` de oorspronkelijke tekenreeks

### Load-script

```
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!
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `TicketID`
- `CustomerComment`

Maak de volgende berekende dimensies:

- `=FindOneOf(CustomerComment, '!,@,#')`
- `=If(FindOneOf(CustomerComment, '!,@,#')>0,'High Priority','Normal Priority')`

Resultatentabel

| TicketID | Klantopmerking | FindOneOf (Klantopmerking, '!,@,#') | Als(VindEenVan(KlantOpmerking, '!,@,#')>0,'Hoge Prioriteit','Normale Prioriteit') |
|---|---|---|---|
| 1 | Ik heb deze bestelling ASAP nodig! | 23 | High Priority |
| 2 | Gelieve mijn bestelling @12345 te bevestigen. | 25 | High Priority |
| 3 | Kunt u mijn bestelling bijwerken? | 0 | Normale prioriteit |
| 4 | Ik heb een vraag over #korting. | 25 | High Priority |
| 5 | Hartelijk dank! | 10 | High Priority |

Vergelijk de uitvoer van de functie `FindOneOf` met de oorspronkelijke `CustomerComment` tekenreekswaarden die in het script geladen zijn. Met behulp van de waarden `!,@,#`, gaf de functie met succes de positie terug waar die tekens voorkomen in de tekenreeks met commentaar. De laatste maatregel gebruikt een voorwaardelijke `If` opdracht om records te identificeren die een van de zoektekens bevatten en deze toe te wijzen als `High Priority`. Als geen van de zoektekens `!,@,#` wordt gevonden, bijvoorbeeld rij 3, dan wordt het ticket geclassificeerd als `Normal Priority`.

## Hash128

**Hash128()** retourneert een 128-bits hashwaarde van de gecombineerde invoerwaarden van de uitdrukking. Het resultaat is een tekenreeks van 22 tekens. Hashwaarden zijn handig voor het maskeren van persoonlijke identificeerbare informatie (PII), zoals klantnamen, burgerservicenummers of rekeningnummers.

**Syntaxis:**

```
Hash128(expr{, expression})
```

**Retourgegevenstypen:** tekenreeks

Argumenten

| Argument | Beschrijving |
|---|---|
| **expr** | De tekenreeks die moet worden geëvalueerd. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `Hash128( 'abc', 'xyz', '123' )` | Retourneert `MA&5]6+3=:>:>G%S<U*S2+` |

| Voorbeeld | Resultaat |
| --- | --- |
| `Hash128 ( Region, Year, Month )`<br><br>Note: `Region`, `Year`, and `Month` are table fields. | Retourneert `G7*=6GKPJ(Z+)^KM?<$'A+` |

## Voorbeeld - Basisprincipes voor Hash128

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden worden gebruikt om een 128-bits hashwaarde te maken:
  - `Region`
  - `Year`
  - `Month`

### Load-script

```
Example:
Load *
inline [
Region, Year, Month
abc, xyz, 123
EU, 2022, 01
UK, 2022, 02
US, 2022, 02
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `Region`
- `Year`
- `Month`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=Hash128(Region, Year, Month)` om de 128-bits hashwaarde te berekenen.

Resultatentabel

| Regio | Jaar | Maand | 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 |

Voor elke rij worden de hashstrings voor `Region`, `Year`, en `Month` samengevoegd en wordt een tekenreeks van 22 tekens geretourneerd.

### Load-script

De volgende code laat zien hoe u de functie in een load-script kunt gebruiken.

```
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 ];
```

Resultatentabel

| Regio | Jaar | Maand | 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 |

## Voorbeeld - Hash128 scenario met duplicaten

### Diagramuitdrukking

#### Overzicht

Dit voorbeeld maakt een unieke hashwaarde voor elke productvermelding. De hashwaarde dient als unieke identificatie, waardoor het bedrijf duplicaten effectief kan opsporen en beheren.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:

- TransactionID
- ProductID
- ProductName
- SupplierID

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- Transaction ID
- ProductID
- ProductName
- SupplierID

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=Hash128(ProductID, ProductName, SupplierID)` om een sleutel van 22 tekens te berekenen.

Resultatentabel

| TransactionID | ProductID | ProductName | SupplierID | Hash128(ProductID, ProductName, SupplierID) |
|---|---|---|---|---|
| 1 | 101 | Widget A | S001 | CY`&^ (N,E/#`TJT4&]/7N% |
| 2 | 102 | Widget B | S002 | O7(1;V1%MG;Z+A [/H)G5&0 |
| 3 | 101 | Widget A | S001 | CY`&^ (N,E/#`TJT4&]/7N% |

Vergelijk de uitvoer van de functie `Hash128`. Merk op dat de eerste en derde `TransactionID` records dubbele hashwaarden hebben omdat de combinatie van de tekenreeksen `ProductID`, `ProductName` en `SupplierID` hetzelfde zijn voor deze rijen.

## Voorbeeld - Hash128 scenario om gegevens te anonimiseren

Diagramuitdrukking

### Overzicht

Een bedrijf dat financiële diensten verleent, verwerkt gevoelige informatie over klanten, zoals creditcardnummers, burgerservicenummers en namen van klanten. Om de privacy te beschermen en naleving van voorschriften zoals de Algemene verordening gegevensbescherming (AVG) of de Health Insurance Portability and Accountability Act (HIPAA) te garanderen, moeten ze gevoelige gegevens anonimiseren wanneer ze rapporten genereren voor analyse. Ze moeten echter ook consistentie behouden, zodat ze dezelfde klant in verschillende gegevensverzamelingen kunnen volgen zonder de identiteit van de klant te onthullen.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `CustomerID`
  - `CustomerName`
  - `SSN`

### Load-script

```
Example:
Load *
inline [
CustomerID, CustomerName, SSN
101, John Doe, 123-45-6789|
102, Jane Smith, 987-65-4321
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `CustomerID`
- `CustomerName`
- `SSN`

Maak de volgende berekende dimensies:

- `=Hash128(CustomerName)`
- `=Hash128(SSN)`

Resultatentabel

| CustomerID | CustomerName | SSN | Hash128 (CustomerName) | Hash128(SSN) |
|---|---|---|---|---|
| 101 | John Doe | 123-45-6789\| | GMD (*B9*!PQ&MTY2@$4A\+ | N& [PSH:I$M?OF<5/9LV&N# |
| 102 | Jane Smith | 987-65-4321 | EP@9]GSI)5G\I*,B?H,C?) | HCFXMR'2=\3WF6?_ \6LN;$ |

De uitvoer genereert unieke sleutels die de volgende voordelen hebben:

- Privacy van gegevens: de oorspronkelijke gevoelige gegevens zijn niet direct zichtbaar, maar de hashwaarde blijft consistent in verschillende records. Dit maakt analyse mogelijk, zoals het aggregeren van klantactiviteit, zonder de werkelijke identiteit van de klant prijs te geven.
- Naleving: door gevoelige velden te hashen, zorgt het bedrijf ervoor dat de privacyregels worden nageleefd.
- Beveiliging: de hashfunctie produceert een uitvoer met een vaste grootte die niet omkeerbaar is, en voegt zo een extra beschermingslaag toe voor gevoelige gegevens.

## Hash160

**Hash160()** retourneert een 160-bits hashwaarde van de gecombineerde invoerwaarden van de uitdrukking. Het resultaat is een tekenreeks van 27 tekens. Hashwaarden zijn handig voor het maskeren van persoonlijke identificeerbare informatie (PII), zoals klantnamen, burgerservicenummers of rekeningnummers.

**Syntaxis:**

```
Hash160(expr{, expression})
```

**Retourgegevenstypen:** tekenreeks

Argumenten

| Argument | Beschrijving |
|---|---|
| **expr** | De tekenreeks die moet worden geëvalueerd. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `Hash160( 'abc', 'xyz', '123' )` | Retourneert `MA&5]6+3=:>;>G%S<U*S2I:`=X*` |
| `Hash160( Region, Year, Month )`<br><br>Note: `Region`, `Year`, and `Month` are table fields. | Retourneert `G7*=6GKPJ (Z+)^KM?<$'AI.)?U$` |

## Voorbeeld - Basisprincipes voor Hash160

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden worden gebruikt om een 160-bits hashwaarde te maken:
  - `Region`
  - `Year`
  - `Month`

### Load-script

```
Example:
Load *
inline [
Region, Year, Month
abc, xyz, 123
EU, 2022, 01
UK, 2022, 02
US, 2022, 02
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `Region`
- `Year`
- `Month`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=Hash160(Region, Year, Month)` om de 160-bits hashwaarde te berekenen.

Resultatentabel

| Regio | Jaar | Maand | 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 |

Voor elke rij worden de hashtekenreeksen voor `Region`, `Year`, en `Month` samengevoegd en wordt een tekenreeks van 27 tekens geretourneerd.

### Load-script

De volgende code laat zien hoe u de functie in een load-script kunt gebruiken.

```
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 ];
```

Resultatentabel

| Regio | Jaar | Maand | 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 |

## Voorbeeld - Hash160 scenario met duplicaten

### Diagramuitdrukking

#### Overzicht

Dit voorbeeld maakt een unieke hashwaarde voor elke productvermelding. De hashwaarde dient als unieke identificatie, waardoor het bedrijf duplicaten effectief kan opsporen en beheren.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `TransactionID`
  - `ProductID`
  - `ProductName`
  - `SupplierID`

#### Load-script

```
Example:
Load *
inline [
TransactionID, ProductID, ProductName, SupplierID
```

```
1, 101, Widget A, S001
2, 102, Widget B, S002
3, 101, Widget A, S001
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `TransactionID`
- `ProductID`
- `ProductName`
- `SupplierID`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=Hash160(ProductID, ProductName, SupplierID)` om een sleutel van 27 tekens te berekenen.

Resultatentabel

| TransactionID | ProductID | ProductName | SupplierID | Hash160(ProductID, ProductName, SupplierID) |
|---|---|---|---|---|
| 1 | 101 | Widget A | S001 | CY`&^ (N,E/#`TJT4&]/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&]/7N1Q+*T3 |

Vergelijk de uitvoer van de functie `Hash160`. Merk op dat de eerste en derde `TransactionID` records dubbele hashwaarden hebben omdat de combinatie van de tekenreeksen `ProductID`, `ProductName` en `SupplierID` hetzelfde zijn voor deze rijen.

## Voorbeeld - Scenario met hash160 om gegevens te anonimiseren

Diagramuitdrukking

#### Overzicht

Een bedrijf dat financiële diensten verleent, verwerkt gevoelige informatie over klanten, zoals creditcardnummers, burgerservicenummers en namen van klanten. Om de privacy te beschermen en naleving van voorschriften zoals de Algemene verordening gegevensbescherming (AVG) of de Health Insurance Portability and Accountability Act (HIPAA) te garanderen, moeten ze gevoelige gegevens anonimiseren wanneer ze rapporten genereren voor analyse. Ze moeten echter ook consistentie behouden, zodat ze dezelfde klant in verschillende gegevensverzamelingen kunnen volgen zonder de identiteit van de klant te onthullen.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `CustomerID`
  - `CustomerName`
  - `SSN`

### Load-script

```
Example:
Load *
inline [
CustomerID, CustomerName, SSN
101, John Doe, 123-45-6789|
102, Jane Smith, 987-65-4321
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `CustomerID`
- `CustomerName`
- `SSN`

Maak de volgende berekende dimensies:

- `=Hash160(CustomerName)`
- `=Hash160(SSN)`

Resultatentabel

| CustomerID | CustomerName | SSN | Hash160(CustomerName) | Hash160(SSN) |
|---|---|---|---|---|
| 101 | John Doe | 123-45-6789\| | GMD (*B9*!PQ&MTY2@$4A\KCL?[0 | N& [PSH:I$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? |

De uitvoer genereert unieke sleutels die de volgende voordelen hebben:

- Privacy van gegevens: de oorspronkelijke gevoelige gegevens zijn niet direct zichtbaar, maar de hashwaarde blijft consistent in verschillende records. Dit maakt analyse mogelijk, zoals het aggregeren van klantactiviteit, zonder de werkelijke identiteit van de klant prijs te geven.
- Naleving: door gevoelige velden te hashen, zorgt het bedrijf ervoor dat de privacyregels worden nageleefd.

- Beveiliging: de hashfunctie produceert een uitvoer met een vaste grootte die niet omkeerbaar is, en voegt zo een extra beschermingslaag toe voor gevoelige gegevens.

## Hash256

**Hash256()** retourneert een 256-bits hashwaarde van de gecombineerde invoerwaarden van de uitdrukking. Het resultaat is een tekenreeks van 43 tekens. Hashwaarden zijn handig voor het maskeren van persoonlijke identificeerbare informatie (PII), zoals klantnamen, burgerservicenummers of rekeningnummers.

**Syntaxis:**

```
Hash256(expr{, expression})
```

**Retourgegevenstypen:** tekenreeks

Argumenten

| Argument | Beschrijving |
|---|---|
| **expr** | De tekenreeks die moet worden geëvalueerd. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `Hash256( 'abc', 'xyz', '123' )` | Retourneert `MA&5]6+3=:>;>G%S<U*S2I:`=X*A.IO*8N\%Y7Q;YEJ` |
| `Hash256( Region, Year, Month )`<br><br>Note: `Region`, `Year`, and `Month` are table fields. | Retourneert `G7*=6GKPJ(Z+)^KM?<$'AI.)?U$#X2RB[:OZP=+Z`F:` |

### Voorbeeld - Basisprincipes voor Hash256

Diagramuitdrukking

**Overzicht**

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden worden gebruikt om een 256-bits hashwaarde te maken:
  - `Region`
  - `Year`
  - `Month`

### Load-script

```
Example:
Load *
inline [
Region, Year, Month
abc, xyz, 123
EU, 2022, 01
UK, 2022, 02
US, 2022, 02
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `Region`
- `Year`
- `Month`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=Hash256(Region, Year, Month)` om de 256-bits hashwaarde te berekenen.

Resultatentabel

| Regio | Jaar | Maand | 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 |

Voor elke rij worden de hashtekenreeksen voor `Region`, `Year`, en `Month` samengevoegd en wordt een tekenreeks van 43 tekens geretourneerd.

### Load-script

De volgende code laat zien hoe u de functie in een load-script kunt gebruiken.

```
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 ];
```

Resultatentabel

| Regio | Jaar | Maand | 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 |

## Voorbeeld - Scenario met hash256 met duplicaten

Diagramuitdrukking

### Overzicht

Dit voorbeeld maakt een unieke hashwaarde voor elke productvermelding. De hashwaarde dient als unieke identificatie, waardoor het bedrijf duplicaten effectief kan opsporen en beheren.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `TransactionID`
  - `ProductID`
  - `ProductName`
  - `SupplierID`

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `Transaction ID`
- `ProductID`
- `ProductName`
- `SupplierID`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=Hash256(ProductID, ProductName, SupplierID)` om een sleutel van 43 tekens te berekenen.

Resultatentabel

| 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] |

Vergelijk de uitvoer van de functie `Hash256`. Merk op dat de eerste en derde `TransactionID` records dubbele hashwaarden hebben omdat de combinatie van de tekenreeksen `ProductID`, `ProductName` en `SupplierID` hetzelfde zijn voor deze rijen.

## Voorbeeld - Scenario met hash256 om gegevens te anonimiseren

Diagramuitdrukking

### Overzicht

Een bedrijf dat financiële diensten verleent, verwerkt gevoelige informatie over klanten, zoals creditcardnummers, burgerservicenummers en namen van klanten. Om de privacy te beschermen en naleving van voorschriften zoals de Algemene verordening gegevensbescherming (AVG) of de Health Insurance Portability and Accountability Act (HIPAA) te garanderen, moeten ze gevoelige gegevens anonimiseren wanneer ze rapporten genereren voor analyse. Ze moeten echter ook consistentie behouden, zodat ze dezelfde klant in verschillende gegevensverzamelingen kunnen volgen zonder de identiteit van de klant te onthullen.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `CustomerID`
  - `CustomerName`
  - `SSN`

### Load-script

```
Example:
Load *
inline [
CustomerID, CustomerName, SSN
101, John Doe, 123-45-6789|
102, Jane Smith, 987-65-4321
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `CustomerID`
- `CustomerName`
- `SSN`

Maak de volgende berekende dimensies:

- `=Hash256(CustomerName)`
- `=Hash256(SSN)`

Resultatentabel

| CustomerID | CustomerName | SSN | Hash256(CustomerName) | Hash256(SSN) |
|---|---|---|---|---|
| 101 | John Doe | 123-45-6789\| | GMD(*B9*!PQ&MTY2@$4A\KCL?[0@T%D@8C#6&UR8/95 | N&[PSH:I$M?OF<5/9LV&N*;`CZ((X'"9Y2FKJC6W04[ |
| 102 | Jane Smith | 987-65-4321 | EP@9]GSI)5G\I*,B?H,C?A$%VG:SL#6I>Q1:L<0O7UW | HCFXMR'2=\3WF6?_\6LN;.*6:K?Y:%MZ#*PP]WI`YS< |

De uitvoer genereert unieke sleutels die de volgende voordelen hebben:

- Privacy van gegevens: de oorspronkelijke gevoelige gegevens zijn niet direct zichtbaar, maar de hashwaarde blijft consistent in verschillende records. Dit maakt analyse mogelijk, zoals het aggregeren van klantactiviteit, zonder de werkelijke identiteit van de klant prijs te geven.
- Naleving: door gevoelige velden te hashen, zorgt het bedrijf ervoor dat de privacyregels worden nageleefd.
- Beveiliging: de hashfunctie produceert een uitvoer met een vaste grootte die niet omkeerbaar is, en voegt zo een extra beschermingslaag toe voor gevoelige gegevens.

## Index

**Index()** doorzoekt een tekenreeks om de beginpositie te vinden van de n-de keer dat een opgegeven subtekenreeks voorkomt. Een optioneel derde argument levert de waarde van n, die 1 is als het argument wordt weggelaten. Bij een negatieve waarde wordt gezocht vanaf het einde van de tekenreeks. De nummering van de posities in de tekenreeks loopt van links naar rechts en begint bij **1**.

**Syntaxis:**

```
Index(text, substring[, count])
```

**Retourgegevenstypen:** geheel getal

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De oorspronkelijke tekenreeks. |
| **substring** | Een reeks van tekens om naar te zoeken in **text**.<br><br>*Als de subtekenreeks niet bestaat in de tekst, retourneert Index 0.* |
| **count** | Definieert naar welk exemplaar van **substring** moet worden gezocht. Bijvoorbeeld, bij een waarde van 2 wordt naar het tweede exemplaar gezocht. Een negatieve waarde begint vanaf het einde van de tekenreeks. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `Index( 'abcdefg', 'cd' )` | Retourneert `3` |
| `Index( 'abcdabcd', 'b', 2 )` | Retourneert `6` (het tweede exemplaar van `b`) |
| `Index( 'abcdabcd', 'b',-2 )` | Retourneert `2` (het tweede exemplaar van `b` te beginnen vanaf het einde) |
| `Left( Date, Index( Date,'-') -1 )` where `Date = 1997-07-14` | Retourneert `1997` |
| `Mid( Date, Index( Date, '-', 2 ) -2, 2 )` where `Date = 1997-07-14` | Retourneert `07` |
| `Index( 'abc', 'x' )` | Retourneert `0` (`x` bestaat niet in de tekenreeks `abc`). |
| `Index( 'abc', 'a', 2 )` | Retourneert `0` (er is geen 2e exemplaar van `a`) |

## Voorbeeld - Basisprincipes voor Index

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `InputString`.

### Load-script

```
Example:
Load * inline [
InputString
abcdefg
abcdabcd ];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `InputString`

Maak de volgende berekende dimensies:

- `=Index(InputString, 'cd')`
- `=Index(InputString, 'e')`
- `=Index(InputString, 'b', -1)`

Resultatentabel

| InputString | Index(InputString, 'cd') | Index(InputString, 'e') | Index(InputString, 'b', -1) |
|---|---|---|---|
| abcdabcd | 3 | 0 | 6 |
| abcdefg | 3 | 5 | 2 |

De eerste rij met de tekenreeks `abcdabcd` retourneert:

- `3`–het eerste exemplaar van `cd`
- `0`–want `e` is niet gevonden in de tekenreeks
- `6`–het eerste exemplaar van `b` vanaf het einde van de tekenreeks

De tweede rij met de tekenreeks `abcdefg` retourneert:

- 3—het eerste exemplaar van `cd`
- 5—het eerste exemplaar van `e`
- 2—het eerste exemplaar van `b` vanaf het einde van de tekenreeks

*Een negatieve `count` parameter scant de tekenreeks in omgekeerde volgorde. De positie die wordt geretourneerd, wordt echter altijd van links naar rechts aangegeven, zelfs als er in omgekeerde richting wordt gescand.*

### Load-script

De volgende code laat zien hoe u de functie in een load-script kunt gebruiken.

```
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 ];
```

Resultatentabel

| Tekenreeks | Index_CD | Index_B | Index_B2 |
|---|---|---|---|
| abcdefg | 3 | 2 | 2 |
| abcdabcd | 3 | 2 | 6 |

## Voorbeeld – Scenario met Index

### Diagramuitdrukking

#### Overzicht

Een klantenservicecentrum heeft een gegevensverzameling die onderwerpregels van e-mailberichten van klanten bevat. Het ondersteuningscentrum wil deze e-mails categoriseren op basis van de trefwoorden die in de onderwerpregel voorkomen, bijvoorbeeld `Refund`, `Complaint`, en `Technical Issue`. De onderwerpregels worden opgeslagen in een veld in de gegevensverzameling met de naam `EmailSubject`.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `EmailSubject`.

#### Load-script

```
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
];
```

**Resultaten**

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `EmailSubject`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=If(Index(EmailSubject, 'Refund') > 0, 'Refund', 'Other')` om te berekenen welke tekst van het e-mailonderwerp betrekking heeft op terugbetalingen.

Resultaten

| EmailSubject | If(Index(EmailSubject, 'Refund') > 0, 'Refund', 'Other') |
|---|---|
| Complaint about delivery service | Other |
| Follow-up on Refund Request | Refund |
| Request for a Refund - Order 12345 | Refund |
| Technical Issue not resolved | Other |
| Technical Issue with Product XYZ | Other |

De uitvoer van de functie `Index` heeft met succes alle onderwerpregels geïdentificeerd die de tekenreeks `Refund` bevatten. Dit is een voorbeeld van hoe u gegevens kunt analyseren en categoriseren met behulp van geneste functies en meerwaarde kunt bieden aan gebruikers.

## IndexRegEx

**IndexRegEx()** doorzoekt de invoertekenreeks en geeft de startpositie van de n-de herhaling van het gespecificeerde reguliere expressiepatroon. Een optioneel derde argument **count** levert de waarde van n, die 1 is als het argument wordt weggelaten. De nummering van de posities in de tekenreeks loopt van links naar rechts en begint bij 1. Als er geen overeenkomst wordt gevonden, retourneert de functie 0.

Deze functie voert regexbewerkingen uit die hoofdlettergevoelig zijn. U kunt eventueel ook de variant **IndexRegExI()** gebruiken om hoofdletteronafhankelijke regexbewerkingen uit te voeren.

**Syntaxis:**

```
IndexRegEx (text, regex [, count])
```

**Retourgegevenstypen:** geheel getal

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De tekst van de invoertekenreeks waarin u wilt zoeken naar een reguliere expressie. |
| **regex** | De reguliere expressie die moet worden gebruikt om de invoertekenreeks te doorzoeken. |
| **count** | Het nummer van de overeenkomst. Dit is handig als er meerdere overeenkomsten voor de reguliere expressie in de tekst kunnen worden gevonden. Geef bijvoorbeeld de waarde `4` op om de positie van de vierde overeenkomst te extraheren.<br><br>Dit is een optioneel argument. De standaardinstelling is 1 als deze niet is opgegeven. U kunt een negatieve waarde opgeven om van rechts naar links naar overeenkomsten te zoeken. |

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `IndexRegEx('abc123','[a-z][0-9]+')` | Retourneert `3` (de startpositie van de eerste overeenkomst). |
| `IndexRegEx('abc123','[a-z][0-9]+',2)` | Retourneert `0` (regex heeft geen tweede overeenkomst). |
| `IndexRegEx('ABC123','[a-z][0-9]+')` | Retourneert `0`, omdat **IndexRegEx()** hoofdlettergevoelig is. |
| `IndexRegExI('ABC123','[a-z][0-9]+')` | Retourneert `3`. De hoofdletterongevoelige variant van de functie, **IndexRegExI()**, wordt gebruikt. |

## Wanneer gebruiken

Gebruiksscenario's voor deze functie zijn onder andere:

- Identificeren waar specifieke tekstpatronen voorkomen binnen grotere stukken tekst. U wilt bijvoorbeeld weten waar een e-mailadrespatroon wordt gebruikt in een reeks lange e-mailberichten.
- **IndexRegEx()** is vooral nuttig voor geavanceerde gegevensverwerking en wordt vaak gebruikt in de eerste stap van een langere, complexere transformatie. Het is meestal eenvoudiger om problemen op te lossen met andere regexfuncties zoals **ExtractRegEx()**, **MatchRegEx()**, en **CountRegEx()**, maar er kunnen momenten zijn waarop **IndexRegEx()** oplossingen biedt die deze functies niet kunnen bieden.

## Voorbeeld 1 - load-script om boeken op basis van ISBN te identificeren

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een tabel `ISBN` bevat de tekst van een e-mailbericht in een veld met de naam `EmailBody`.

We willen de posities van alle 13-cijferige ISBN-codes uit de natuurlijke taal in het e-mailbericht vinden.

### Load-script

```
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/';
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `ISBNCode1`
- `ISBNCode2`
- `ISBNCode3`

Resultatentabel

| ISBNCode1 | ISBNCode2 | ISBNCode3 |
|---|---|---|
| 301 | 487 | 588 |

## Voorbeeld 2 - diagramuitdrukking om contactpersonen met secundaire telefoonnummers te identificeren

Diagramuitdrukking en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een tabel met de naam `BusinessContactInfo`, die informatie over bedrijven bevat die gevonden werd op de website van elk bedrijf.
- `ContactInfo` is een veld dat telefoonnummers van bedrijven bevat. We willen bepalen welke bedrijven meerdere telefoonnummers beschikbaar hebben.

Onze vereisten:

- Telefoonnummers moeten NANP-telefoonnummers met 10 cijfers zijn.
- We willen toestaan dat het netnummer tussen haakjes staat.
- We willen een enkele spatie of koppelteken toestaan tussen elk segment van een telefoonnummer.

### Load-script

```
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');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `CompanyName`

Voeg deze berekende dimensies toe aan de tabel:

```
=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)')
```

Resultatentabel

| 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` heeft één secundair telefoonnummer. `Company B` heeft twee secundaire telefoonnummers. `Company C` heeft drie secundaire telefoonnummers.

*De functie **MatchRegEx()** is ook nuttig voor dit soort validatiescenario's.*

## IndexRegExGroup

**IndexRegExGroup()** doorzoekt de invoertekenreeks en geeft de startpositie van de n-de herhaling van het samengestelde reguliere expressiepatroon dat is opgegeven. Een optioneel vierde argument **count** levert de waarde van n, die 1 is als het argument wordt weggelaten. De nummering van de posities in de tekenreeks loopt van links naar rechts en begint bij 1. Als er geen overeenkomst wordt gevonden, retourneert de functie 0.

Deze functie voert regexbewerkingen uit die hoofdlettergevoelig zijn. U kunt eventueel ook de variant **IndexRegExGroupI()** gebruiken om hoofdletteronafhankelijke regexbewerkingen uit te voeren.

**Syntaxis:**

```
IndexRegExGroup (text, regex, group [, count])
```

**Retourgegevenstypen:** geheel getal

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De tekst van de invoertekenreeks waarin u wilt zoeken naar een reguliere expressie. |
| **regex** | De reguliere expressie die moet worden gebruikt om de invoertekenreeks te doorzoeken. |
| **group** | Het nummer van de groep, in het geval van een samengestelde reguliere expressie.<br><br>Een **group** waarde van `0` retourneert de index van de volledige regex. Als de reguliere expressie echter alleen de index van de gehele overeenkomst moet retourneren, gebruikt u in plaats daarvan de functie **IndexRegEx()**.<br><br>U kunt een negatieve **group** waarde opgeven om van rechts naar links naar overeenkomsten te zoeken. |

| Argument | Beschrijving |
|---|---|
| **count** | Het nummer van de overeenkomst. Dit is handig als er meerdere overeenkomsten voor de reguliere expressie in de tekst kunnen worden gevonden. Geef bijvoorbeeld de waarde `4` op te zoeken naarde positie van de vierde overeenkomst.<br><br>Dit is een optioneel argument. De standaardinstelling is 1 als deze niet is opgegeven. U kunt een negatieve waarde opgeven om van rechts naar links naar overeenkomsten te zoeken. |

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `IndexRegExGroup ('abc123','([a-z])([0-9]+)',0)` | Retourneert `3` (startpositie van volledige regex). |
| `IndexRegExGroup ('abc123','([a-z])([0-9]+)',1)` | Retourneert `3` (startpositie van de eerste groep). |
| `IndexRegExGroup ('abc123','([a-z])([0-9]+)',2)` | Retourneert `4`, dit is de startpositie van de tweede groep. In het regexpatroon van de invoer komt de tekenreeks `([0-9]+)` overeen met de tweede groep. |
| `IndexRegExGroup ('ABC123','([a-z])([0-9]+)',1)` | Retourneert `0`, omdat **IndexRegExGroup()** hoofdlettergevoelig is. |
| `IndexRegExGroupI ('ABC123','([a-z])([0-9]+)',1)` | Retourneert `3`. De hoofdletterongevoelige variant van de functie, **IndexRegExI()**, wordt gebruikt. |

## Wanneer gebruiken

Gebruiksscenario's voor deze functie zijn onder andere:

- Identificeren waar specifieke tekstpatronen voorkomen binnen grotere stukken tekst. U wilt bijvoorbeeld weten waar het domein van een e-mailadres wordt gebruikt in een reeks lange e-mailberichten.
- **IndexRegExGroup()** is vooral nuttig voor geavanceerde gegevensverwerking en wordt vaak gebruikt in de eerste stap van een langere, complexere transformatie. Het is meestal eenvoudiger om problemen op te lossen met andere regexfuncties zoals **ExtractRegExGroup()**, **MatchRegEx()**, en **CountRegEx()**, maar er kunnen momenten zijn waarop **IndexRegExGroup()** oplossingen biedt die deze functies niet kunnen bieden.

## Voorbeeld 1 - load-script om posities van ISBN-componenten te vinden

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Het aanmaken van een variabele, `ISBN_RegEx` om de reguliere expressie op te slaan die we willen gebruiken.
- Een set ISBN-codes voor boeken die een boekwinkel bij verschillende leveranciers wil bestellen.

Voor elke ISBN-code moeten we de startpositie van elk van de volgende extraheren:

- EAN
- Groep
- Registrant
- Publicatie
- Controlesom

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `Supplier`
- `EAN`
- `Group`
- `Registrant`
- `Publication`
- `Checksum`

Resultatentabel

| 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 dit geval bevat elk boek componenten op dezelfde posities.

## Voorbeeld 2 - diagramuitdrukkingen om URL-posities te vinden (met vergelijking IndexRegEx() )

Diagramuitdrukkingen en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een tabel met de naam `Correspondence`, die tekst van een e-mailbericht bevat in een veld met de naam `EmailBody`.
- Inhoud van e-mailberichten die web-URL's bevatten.

We willen de positie van de tweede URL vinden, indien aanwezig, evenals de posities van de volgende componenten:

- Domein
- Pad

We willen één reguliere expressie gebruiken om alle bewerkingen uit te voeren.

### Load-script

```
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');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- ID

Voeg deze berekende dimensies toe:

- *URL 2 Positie*:
  `=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)`
- *URL 2 Domein Positie*:
  `=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)`
- *URL 2 Pad Positie*:
  `=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)`

Resultatentabel

| ID | URL 2 Positie | URL 2 Domein Positie | URL 2 Pad Positie |
|---|---|---|---|
| 1 | 274 | 281 | 295 |

| ID | URL 2 Positie | URL 2 Domein Positie | URL 2 Pad Positie |
|---|---|---|---|
| 2 | 0 | 0 | 0 |
| 3 | 298 | 306 | 327 |

**IndexRegEx()** retourneert de startpositie van de hele URL, terwijl **IndexRegExGroup()** startposities van de individuele delen van de URL retourneert, die overeenkomen met de **group** waarde die we hebben gebruikt. De record met een `ID` waarde van 2 bevat slechts één URL, dus worden er waarden van 0 geretourneerd.

Dezelfde reguliere expressie wordt in alle diagramuitdrukkingen gebruikt om verschillende stukjes informatie te retourneren. Een uitsplitsing van de groepen die in de reguliere expressie zijn gedefinieerd, is als volgt.

Groepen in reguliere expressie

| Groepsnummer | Reguliere expressie | URL-gedeelte |
|---|---|---|
| 1 | ([a-zA-Z0-9]+) | Protocol |
| 2 | (([a-zA-Z0-9]+\.)*([a-zA-Z0-9]+)\.([a-zA-Z0-9]+)) | Domein (groepen 3, 4 en 5) |
| 3 | ([a-zA-Z0-9]+\.)* | Hoofddomein |
| 4 | ([a-zA-Z0-9]+) | Topniveaudomein |
| 5 | ([a-zA-Z0-9]+) | Pad (groepen 7, 8, 9 en 10) (optioneel) |
| 6 | ((\/[a-zA-Z0-9]+)*(\/([a-zA-Z0-9]+)(\.[a-zA-Z0-9]+)?))? | Submap |
| 7 | (\/[a-zA-Z0-9]+)* | Bestand (groepen 9 en 10) |
| 8 | (\/([a-zA-Z0-9]+)(\.[a-zA-Z0-9]+)?) | Bestand |
| 9 | ([a-zA-Z0-9]+) | Bestandsnaam |
| 10 | (\.[a-zA-Z0-9]+)? | Bestandsextensie (optioneel) |

## IsJson

**IsJson()** controleert of een opgegeven string geldige JSON-gegevens (JavaScript Object Notation) bevat. U kunt ook een specifiek JSON-gegevenstype valideren.

**Syntaxis:**

```
value IsJson(json [, type])
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **json** | Tekenreeks om te testen. Deze kan extra spaties of nieuwe regels bevatten. |
| **type** | Optioneel argument dat het JSON-gegevenstype aangeeft dat getest moet worden.<br>• 'value' (default)<br>• 'object'<br>• 'array'<br>• 'string'<br>• 'number'<br>• 'Boolean'<br>• null |

Voorbeeld: geldige en ongeldige diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `IsJson( 'null' )` | Retourneert `-1 (true)` |
| `IsJson( '"abc"', 'value' )` | Retourneert `-1 (true)` |
| `IsJson( '"abc"', 'string' )` | Retourneert `-1 (true)` |
| `IsJson( 123, 'number' )` | Retourneert `-1 (true)` |
| `IsJson( 'text' )` | Retourneert `0 (false)`, `'text'` is geen geldige JSON-waarde |
| `IsJson( '"text"', 'number' )` | Retourneert `0 (false)`, `'"text"'` is geen geldig JSON-getal |
| `IsJson( '"text"', 'text' )` | Retourneert `0 (false)`, `'text'` is geen geldig JSON-type |

## Voorbeeld - Basisprincipes voor IsJson

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `API_Response`.
- Een afgeleid veld genaamd `ID`. Dit veld gebruikt de functie `Rec()` om de invoerrecords op te sommen.

### Load-script

```
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"}'
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `ID`
- `API_Response`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=IsJson(API_Response)` om te berekenen of de waarden in `API_Response` geldige JSON zijn.

Resultatentabel

| Id | API_Response | IsJson(API_Response) |
|---|---|---|
| 1 | {"id": 1, "name": "Alice"} | -1 (true) |
| 2 | {"id": 2, "name": "Bob"} | -1 (true) |
| 3 | {invalid json string} | 0 (false) |
| 4 | {"id": 4, "name": "Charlie"} | -1 (true) |
| 5 | {"id": 5, name: "David"} | 0 (false) |

De uitvoer retourneert `-1`, of true, voor die waarden met een geldige JSON-syntaxis.

De uitvoer retourneert `0`, of false, voor de volgende id-records (`3` en `5`) met ongeldige JSON:

- `{invalid json string}` –De teksttekenreeks vertegenwoordigt geen geldige JSON-antwoordstructuur.
- `{"id": 5, name: "David"}` –Het tweede sleutelwaardepaar mist aanhalingstekens rond de sleutel `"name"`.

## Voorbeeld – IsJson scenario

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `OrderDetails`.

### Load-script

```
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 }'
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `OrderDetails`

Maak de volgende metingen:

- `IsJson( OrderDetails)` om te berekenen of de waarden in `OrderDetails` geldige JSON zijn.
- `IsJson( JsonGet ( OrderDetails, '/items/price' ), 'number' )`, met behulp van de functie `JsonGet` haalt de JSON-tekst voor de `price` sleutel op en valideert dat de waarde van `price` een getal is.

Resultatentabel

| OrderDetails | IsJson (OrderDetails) | IsJson( JsonGet ( OrderDetails, '/items/price' ), 'number' ) |
|---|---|---|
| { "order_id": "12345", "customer": { "name": "John Doe", "email": "john.doe@example.com"}, "items": {"product": "Laptop", "quantity": 2, "price": 1200 }, "total_price": 2400 } | -1 (true) | -1 (true) |

De eerste meting retourneert `-1 (true)` omdat `OrderDetails` een geldige JSON-syntaxis bevat.

De tweede meting retourneert `-1 (true)` omdat de waarde van de sleutel `price` een geldig getal is, `1200`.

# IsRegEx

**IsRegEx()** retourneert of de opgegeven tekst een geldige reguliere expressie is.

Deze functie voert regexbewerkingen uit die hoofdlettergevoelig zijn. U kunt eventueel ook de variant **IsRegExI()** gebruiken om hoofdletteronafhankelijke regexbewerkingen uit te voeren.

**Syntaxis:**

```
IsRegEx (expr [, debug])
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **expr** | Reguliere expressie die de te evalueren reguliere expressie bevat. |
| **debug** | Optioneel argument. Als voor dit argument een waarde wordt opgegeven, retourneert de expressie een teksthint die aangeeft hoe een ongeldige reguliere expressie kan worden opgelost. |

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `IsRegEx('[a-z]')` | Retourneert `-1` (true). |
| `IsRegEx(']0-1[', 1)` | Dit voorbeeld geeft een foutbericht met een hint om de ongeldige reguliere expressie `]0-1[` te corrigeren. |
| `IsRegEx('[a-z')` | Retourneert `0` (false). Er wordt geen foutbericht weergegeven omdat het argument **debug** niet is opgegeven. |

## Wanneer gebruiken

U kunt **IsRegEx()** gebruiken om te testen of de reguliere expressies die u schrijft geldig zijn in Qlik Sense. U kunt bijvoorbeeld een invoertabel in uw app laden met invoertekst en een evaluatie van de geldigheid van de tekst als regex.

## Voorbeeld 1 - geldigheid van regex in gegevensmodel laden

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een veld, `reg_exp`, dat invoertekenreeksen bevat die al dan niet geldige reguliere expressies kunnen zijn.
- Twee berekende velden om te controleren of elke waarde in `reg_exp` een geldige reguliere expressie is of niet:
    - `validity_no_hint`: Geeft `-1` (true) weer als de regex geldig is, en 0 (false) als deze ongeldig is.
    - `validity_with_hint`: Geeft -1 (true) weer als de regex geldig is. Als de regex ongeldig is, geeft een automatisch gegenereerd foutbericht problemen met de invoer aan, met mogelijke hints om het probleem op te lossen.

### Load-script

```
[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)
`;
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `reg_exp`
- `validity_no_hint`
- `validity_with_hint`

Resultatentabel

| reg_exp | validity_no_hint | validity_with_hint |
|---|---|---|
| ([a-z)(0-9)+ | 0 | -1 |
| [a-z]+ | -1 | Hinttekst (automatisch gegenereerd) |
| \$\|¢(0-9) | -1 | Hinttekst (automatisch gegenereerd) |
| abcdefg\|[0-7 | 0 | -1 |

## Voorbeeld 2 - alleen geldige regex in gegevensmodel laden

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een veld, `reg_exp`, dat invoertekenreeksen bevat die al dan niet geldige reguliere expressies kunnen zijn. Dit veld staat in een tabel met de naam `Regular Expressions`.
- Een **where**-clausule die alleen records in het gegevensmodel laadt als de invoer een geldige reguliere expressie is.

### Load-script

```
[Regular Expressions]:

Load
*
Inline
`
reg_exp
[a-z]+
([a-z)(0-9)+
abcdefg|[0-7
\$|¢(0-9)
`
WHERE
IsRegEx(reg_exp)=True()
;
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `reg_exp`

Resultaten tabel

| reg_exp |
| --- |
| [a-z]+ |
| \$\|¢(0-9) |

`[a-z]+` en `\$|¢(0-9)` zijn geldige reguliere expressies, dus zijn ze geladen in het gegevensmodel. `([a-z)(0-9)+` en `abcdefg|[0-7` zijn geen geldige reguliere expressies, dus zijn ze niet geladen in het gegevensmodel.

## Voorbeeld 3: diagramuitdrukkingen

Diagramuitdrukkingen en resultaten

### Overzicht

In dit voorbeeld laden we een gegevenstabel met invoertekenreeksen en voegen deze toe aan een tabel, samen met twee berekende dimensies die de geldigheid van de regex evalueren. De twee berekende dimensies geven dezelfde informatie op verschillende manieren weer.

Open de Editor voor laden van gegevens en voeg het volgende load-script toe aan een nieuwe sectie.

Het load-script bevat een veld, `reg_exp`, dat invoertekenreeksen bevat die al dan niet geldige reguliere expressies kunnen zijn. Dit veld staat in een tabel met de naam `Regular Expressions`.

### Load-script

```
[Regular Expressions]:

Load
*
Inline
`
reg_exp
[a-z]+
([a-z)(0-9)+
abcdefg|[0-7
\$|¢(0-9)
`;
```

### Resultaten

### Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel. Voeg `reg_exp` toe als een dimensie.
2. Voeg vervolgens de volgende berekende dimensie toe:
   `=IsRegEx(reg_exp)`
   Dit is een basis diagramuitdrukking die `-1` (true) weergeeft als de invoer een geldige reguliere expressie is, en `0` (false) als dat niet zo is.
3. Voeg een andere berekende dimensies toe:
   `=if(IsRegEx(reg_exp), 'Valid regex', 'Invalid regex')`
   Dit is een voorwaardelijke expressie die meer details geeft over de geldigheid van de invoer.

Resultatentabel

| reg_exp | =IsRegEx(reg_exp) | =if(IsRegEx(reg_exp), 'Valid regex', 'Invalid regex') |
|---|---|---|
| ([a-z)(0-9)+ | 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 aggregeert en schakelt JSON-gegevens aaneen.

**Syntaxis:**

```
JsonArray([DISTINCT][TOTAL] value [, sort_weight])
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| DISTINCT | Als het woord **DISTINCT** voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd. |
| TOTAL | Als het woord **TOTAL** voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd. |
| value | Invoerveld. De uitdrukking die de gegevens bevat die moeten worden gesorteerd. De eerste (laagste) waarde van **sort_weight** wordt gevonden, op basis waarvan de overeenkomstige waarde van de uitdrukking **value** wordt bepaald. Als u een minteken vóór **sort_weight** plaatst, retourneert de functie de laatste (hoogste) gesorteerde waarde. |
| sort_weight | Invoerveld. De uitdrukking die de gegevens bevat die moeten worden gesorteerd. De eerste (laagste) waarde van **sort_weight** wordt gevonden, op basis waarvan de overeenkomstige waarde van de uitdrukking **value** wordt bepaald. Als u een minteken vóór **sort_weight** plaatst, retourneert de functie de laatste (hoogste) gesorteerde waarde. |

**Voorbeeld:**

Het volgende load-script wordt geladen en aggregeert gegevens met **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 '|');
```

Dit resulteert in de volgende tabel met gegevens:

Resultaten

| Id | Json | JsonDistinct | JsonSorted |
|---|---|---|---|
| 1 | [123] | [123] | [123] |
| 2 | ["json text","normal text"] | ["json text","normal text"] | ["normal text","json text"] |
| 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 and text",""] | [1,"2 and text",""] | [1,"2 and text",""] |

## JsonGet

**JsonGet()** retourneert het pad van een JSON-gegevensreeks (JavaScript Object Notation). De JSON-gegevens moeten een geldige JSON-string zijn, maar kunnen extra spaties of nieuwe regels bevatten.

**Syntaxis:**

```
value JsonGet(json, path)
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **json** | Tekenreeks met JSON-gegevens. |
| **path** | Het pad moet zijn gespecificeerd volgens ≤ RFC 6901. Dit maakt zoeken naar eigenschappen in json-gegevens mogelijk zonder gebruik van complexe subtekenreeks- of indexfuncties. |

Voorbeeld: geldige en ongeldige diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `JsonGet( '{"a":{"xyz":"XYZ"},"b":[123,"abc","ABC"]}', '' )` | Retourneert `{"a":{"xyz":"XYZ"},"b":[123,"abc","ABC"]}` |
| `JsonGet( '{"a":{"xyz":"XYZ"},"b":[123,"abc","ABC"]}', '/a' )` | Retourneert `{"xyz":"XYZ"}` |
| `JsonGet( '{"a":{"xyz":"XYZ"},"b":[123,"abc","ABC"]}', '/a/xyz' )` | Retourneert `"XYZ"` |
| `JsonGet( '{"a":{"xyz":"XYZ"},"b":[123,"abc","ABC"]}', '/b' )` | Retourneert `[123,"abc","ABC"]` |
| `JsonGet( '{"a":{"xyz":"XYZ"},"b":[123,"abc","ABC"]}', '/b/0' )` | Retourneert `123` |
| `JsonGet( '{"a":{"xyz":"XYZ"},"b":[123,"abc","ABC"]}', '/b/1' )` | Retourneert `abc` |
| `JsonGet( '{"a":{"xyz":"XYZ"},"b":[123,"abc","ABC"]}', '/b/2' )` | Retourneert `ABC` |
| `JsonGet( '{"a":"b"}','/b' )` | Retourneert `null`. Het pad verwijst niet naar een geldig deel van de JSON-gegevens. |
| `JsonGet( '{"a"}','/a' )` | Retourneert `null`. De JSON-gegevens zijn geen geldige JSON (lid `"a"` bevat geen waarde). |

## Voorbeeld - Basisprincipes voor JsonGet

Diagramuitdrukking

### Overzicht

U ontvangt JSON-gegevens met bestelgegevens als een teksttekenreeks in een enkele regel. U wilt de volgende informatie ophalen:

- prijs van het item en validatie dat de waarde een getal is
- e-mailadres klant
- totale prijs (prijs * hoeveelheid)

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `OrderDetails`.

### Load-script

```
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 }'
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `OrderDetails`

Maak de volgende metingen:

- `IsJson( JsonGet ( OrderDetails, '/items/price' ), 'number' )`, in combinatie met de functie `IsJson` haalt de JSON-tekst voor `price` op en valideert dat de waarde van `price` een getal is.
- `JsonGet( OrderDetails, '/customer/email' )` om het e-mailadres van de klant op te halen met behulp van het juiste pad.
- `JsonGet( OrderDetails, '/total_price' ), )` om de totaalprijs op te halen met behulp van het juiste pad.

Resultatentabel

| 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 |

De eerste meting retourneert `-1 true` omdat de `price` sleutel een waarde heeft van `1200`, wat een getal is.

De tweede meting retourneert het e-mailadres van de klant, `john.doe@example.com`, van `OrderDetails` met behulp van het juiste pad naar de `email` sleutel in de JSON-tekenreeks.

De derde meting retourneert `2400`, wat de totale prijs van de bestelling uit het veld `OrderDetails` is, waarbij het juiste pad naar de `total_price` sleutel in de JSON-tekenreeks wordt gebruikt.

## JsonObject

JsonObject wordt gebruikt om JSON-objecten te bouwen.

*U kunt een willekeurig aantal naam-waardeparen toevoegen met JsonObject.*

**Syntaxis:**

```
JsonObject(name, value [name, value, ...])
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| json | Tekenreeks met JSON-gegevens. |
| naam | De naam van het veld in JSON-indeling. |
| waarde | De tekenreekswaarde in JSON-indeling. |

**Voorbeeld:**

Met het volgende load-script worden gegevens geladen en ingesteld als een JSON-object.

```
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
];
```

Dit resulteert in de volgende tabel met JSON-gegevens:

```
{"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()** wijzigt een string die JSON-gegevens (JavaScript Object Notation) bevat. Het kan een JSON-waarde instellen of invoegen met de nieuwe locatie die is opgegeven door het pad. De JSON-gegevens moeten een geldige JSON zijn, maar kunnen extra spaties of nieuwe regels bevatten.

*U kunt een willekeurig aantal naam-waardeparen toevoegen met JsonSet.*

**Syntaxis:**

```
value JsonSet(json, path, value)
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| **json** | Tekenreeks met JSON-gegevens. |
| path | Het pad moet zijn gespecificeerd volgens ≤ RFC 6901. Dit maakt opbouwen van eigenschappen in json-gegevens mogelijk zonder gebruik van complexe subtekenreeks- of indexfuncties en aaneenschakeling. |
| **value** | De nieuwe tekenreekswaarde in JSON-indeling. |

Voorbeeld: geldige en ongeldige diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `JsonSet( '{}','/a','"b"' )` | Retourneert `{"a":"b"}` |
| `JsonSet( '[]','/0','"x"' )` | Retourneert `["x"]` |
| `JsonSet( '"abc"','','123' )` | Retourneert `123` |
| `JsonSet( '"abc"','/x','123' )` | Retourneert `null`. Het pad verwijst niet naar een geldig deel van de JSON-gegevens. |
| `JsonSet( '{"a":{"b":"c"}}','a/b','"x"' )` | Retourneert `null`. Het pad is ongeldig. |
| `JsonSet( '{"a":"b"}','/a','abc' )` | Retourneert `null`. De waarde is geen geldige JSON. Een tekenreeks moet tussen aanhalingstekens worden geplaatst. |

## Voorbeeld - Basisprincipes voor JsonSet

Diagramuitdrukking

**Overzicht**

JSON-gegevens worden ontvangen als een teksttekenreeks in een enkele regel. U wilt de prijs van het artikel en de totaalprijs van de bestelling dienovereenkomstig bijwerken.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `OrderDetails`.

**Load-script**

```
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 }'
];
```

**Resultaten**

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `OrderDetails`

Maak de volgende meting:

- `JsonSet(JsonSet(OrderDetails,'/items/price',1100),'/total_price',2200)` om de prijs in te stellen op `1100` en de totaalprijs op `2200` met behulp van het juiste pad naar de respectieve sleutels in de JSON-tekenreeks.

Resultatentabel

| OrderDetails | JsonSet(JsonSet (OrderDetails,'/items/price',1100),'/total_ price',2200) |
|---|---|
| { "order_id": "12345", "customer": { "name": "John Doe", "email": "john.doe@example.com"}, "items": {"product": "Laptop", "quantity": 2, "price": 1200 }, "total_price": 2400 } | {"order_id":"12345","customer":{"name":"John Doe","email":"john.doe@example.com"},"items": {"product":"Laptop","quantity":2,"price":1100},"total_ price":2200} |

De uitvoer van de meting werkt de `price` en `total price` sleutels bij naar respectievelijk `1100` en `2200`.

## JsonSetEx

**JsonSetEx()** wijzigt tekenreeksen die JSON-gegevens (JavaScript Object Notation) bevat. Het kan JSON-waarden instellen of invoegen met de nieuwe locatie die is opgegeven door het pad. De JSON-gegevens moeten een geldige JSON zijn, maar kunnen extra spaties of nieuwe regels bevatten.

*U kunt een willekeurig aantal naam-waardeparen toevoegen met JsonSetEx.*

**Syntaxis:**

```
JsonSetEx(json, path, value, ...)
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| json | Tekenreeks met JSON-gegevens. |
| path | Het pad moet zijn gespecificeerd volgens ≤ RFC 6901. Dit maakt opbouwen van eigenschappen in json-gegevens mogelijk zonder gebruik van complexe subtekenreeks- of indexfuncties en aaneenschakeling. |
| value | De nieuwe tekenreekswaarde in JSON-indeling. |

**Voorbeeld:**

Met het volgende load-script worden gegevens geladen en geformatteerd in een JSON-indeling.

```
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
];
```

Dit resulteert in de volgende tabel met JSON-gegevens:

```
{"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()** retourneert een tekenreeks die bestaat uit tekens van de eerste rekenreeks die overeenkomen met de tekens in de tweede tekensreeks. Deze functie is hoofdlettergevoelig.

**Syntaxis:**

```
KeepChar(text, keep_chars)
```

**Retourgegevenstypen:** tekenreeks

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De oorspronkelijke tekenreeks. |
| **keep_chars** | Een tekenreeks die de tekens in **text** bevat die moeten worden bewaard. Dit argument is hoofdlettergevoelig. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `KeepChar ( 'a1b2c3','123' )` | Retourneert `123` |
| `KeepChar ( 'a1b2c3','1234' )` | Retourneert `123` |
| `KeepChar ( 'a1b22c3','1234' )` | Retourneert `1223` |
| `KeepChar( 'a1b2c3','312' )` | Retourneert `123` |

## Voorbeeld - Basisprincipes voor KeepChar

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `InputText` die de originele teksttekenreeks bevat die verwerkt moet worden.
  - `CharsToKeep` die de lijst met tekens bevat die na verwerking uit de oorspronkelijke teksttekenreeks moeten worden behouden.

### Load-script

```
Example:
Load * inline [
InputText, CharsToKeep
'a1b2c3', '123'
'a^b^c', 'abc'
'A^b^c', 'abc'
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `InputText`
- `CharsToKeep`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=KeepChar(InputText, CharsToKeep)`, om de tekens in `InputText` te berekenen die overeenkomen met de tekens in `CharsToKeep`.

Resultatentabel

| InputText | CharsToKeep | KeepChar(InputText, CharsToKeep) |
|---|---|---|
| a1b2c3 | 123 | 123 |
| a^b^c | abc | abc |
| A^b^c | abc | bc |

De kolom `KeepChar(InputText, CharsToKeep)` retourneert alleen de tekens in `InputText` die exact overeenkomen met de tekens in `CharsToKeep`. De derde rij retourneert bijvoorbeeld `bc`, omdat dat de enige tekens in `InputText` zijn die overeenkomen met de tekens in `CharsToKeep`. Het hoofdletterteken `A` wordt niet geretourneerd omdat de items in `CharsToKeep` alleen kleine letters zijn.

Load-script

De volgende code laat zien hoe u de functie in een load-script kunt gebruiken.

```
Example:
Load
*,
keepchar(String1, String2) as KeepChar;
Load * inline [
String1, String2
'a1b2c3', '123'
];
```

Resultatentabel

| String1 | String2 | KeepChar |
|---|---|---|
| a1b2c3 | 123 | 123 |

## Voorbeeld - Scenario met KeepChar om ongestructureerde tekst te parseren

Diagramuitdrukking

**Overzicht**

Dit voorbeeld gebruikt de functie om ongestructureerde tekst (letters, cijfers en symbooltekens) te ontleden en een gestructureerde uitvoer te leveren die alleen numerieke tekens behoudt.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `InputText` die de originele teksttekenreeks bevat die verwerkt moet worden.
  - `CharsToKeep` die de lijst met tekens bevat die na verwerking uit de oorspronkelijke teksttekenreeks moeten worden behouden.

### Load-script

```
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'
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `InputText`
- `CharsToKeep`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=KeepChar(InputText, CharsToKeep)`, om de tekens in `InputText` te berekenen die overeenkomen met de tekens in `CharsToKeep`.

Resultatentabel

| InputText | CharsToKeep | KeepChar(InputText, CharsToKeep) |
|---|---|---|
| Delivery of batch number: 333*456789 | 0123456789 | 333456789 |
| Find the goods received in bay 16a | 0123456789 | 16 |
| The numeric code for input processing is 123-456\|789 | 0123456789 | 123456789 |

Vergelijk de uitvoer van de functie `KeepChar` met de oorspronkelijke `InputText` tekenreekswaarden die in het script geladen zijn. De functie `KeepChar` heeft met succes alle niet-numerieke tekens uit de invoertekst verwijderd.

---

**Zie ook:**

p *PurgeChar (page 1835)*

# Left

**Left()** retourneert een tekenreeks die bestaat uit de eerste (meest linker) van de invoertekenreeks, waarbij het aantal tekens wordt bepaald door het tweede argument.

**Syntaxis:**

```
Left(text, count)
```

**Retourgegevenstypen:** tekenreeks

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De oorspronkelijke tekenreeks. |
| **count** | Definieert het aantal tekens dat moet worden opgenomen vanaf de linkerkant van de tekenreeks **text**. |

Voorbeeld: diagramuitdrukking

| Voorbeeld | Resultaat |
|---|---|
| `Left( 'abcdef', 3 )` | Retourneert `abc` |

## Voorbeeld - Basisprincipes voor Left

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `InputText`
  - `NoOfChars`

### Load-script

```
Example:
Load * inline [
InputText, NoOfChars
'abcdef', 3
'2021-07-14', 4
'2021-07-14', 2
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `InputText`
- `NoOfChars`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=Left(InputText,NoOfChars)`

Resultatentabel

| InputText | NoOfChars | Left(InputText,NoOfChars) |
|---|---|---|
| abcdef | 3 | abc |
| 2021-07-14 | 2 | 20 |
| 2021-07-14 | 4 | 2021 |

De eerste rij retourneert `abc`, omdat dit de eerste 3 tekens van links zijn van de tekenreeks `InputText`.

De tweede rij retourneert `20`, omdat dit de eerste 2 tekens van links zijn van de tekenreeks `InputText`.

De derde rij retourneert `2021`, omdat dit de eerste 4 tekens van links zijn van de tekenreeks `InputText`.

#### Load-script

De volgende code laat zien hoe u de functie in een load-script kunt gebruiken.

```
Example:
Load
*,
left(Text,NoOfChars) as Left;
Load * inline [
Text, NoOfChars
'abcdef', 3
'2021-07-14', 4
'2021-07-14', 2
];
```

Resultatentabel

| Tekst | NoOfChars | Links |
|---|---|---|
| abcdef | 3 | abc |
| 2021-07-14 | 4 | 2021 |
| 2021-07-14 | 2 | 20 |

## Voorbeeld - Scenario met Left

Diagramuitdrukking

### Overzicht

Dit voorbeeld parseert de bestelcodes in een gegevensverzameling, die een regiocode, jaar en bestelnummer bevatten, om een nieuw veld voor gegevensanalyse te maken.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `OrderID`
  - `CustomerName`
  - `SalesAmount`

### Load-script

```
Example:
Load * inline [
OrderID, CustomerName, SalesAmount
US2024001, John Doe, 1500
EU2024002, Jane Smith, 2300
AS2024003, Mike Davis, 1800
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `OrderID`
- `CustomerName`
- `SalesAmount`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=Left(OrderID, 2)` om de regio te berekenen.

Resultatentabel

| OrderID | CustomerName | SalesAmount | Left(OrderID, 2) |
|---|---|---|---|
| AS2024003 | Mike Davis | 1800 | AS |
| EU2024002 | Jane Smith | 2300 | EU |
| US2024001 | John Doe | 1500 | VS |

De uitvoer van de functie `Left` parseert de OrderID om de regio te extraheren, waarbij een nieuw veld wordt aangemaakt dat voor verdere gegevensanalyse kan worden gebruikt.

## Voorbeeld - Geavanceerd scenario met Left

Diagramuitdrukking

### Overzicht

Dit voorbeeld parseert de productcodes in een gegevensverzameling, die een afdelingscategoriecode en een unieke identificatie bevatten, om een nieuw veld voor gegevensanalyse te maken.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `ProductCode`.

### Load-script

```
Example:
Load * inline [
ProductCode
ELC12345
HOM67890
FAS09876
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `ProductCode`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=Pick(Match(left(ProductCode,3),'ELC','HOM','FAS'),'Electronics','Home Appliances','Fashion')` om de afdeling te berekenen.

Resultatentabel

| ProductCode | Pick(Match(left (ProductCode,3),'ELC','HOM','FAS'),'Electronics','Home Appliances','Fashion') |
|---|---|
| ELC12345 | Electronics |
| FAS09876 | Fashion |
| HOM67890 | Home Appliances |

De uitvoer gebruikt de functies `Pick`, `Match` en `Left` om de afdeling uit de productcode te extraheren, waardoor een nieuw veld voor gegevensanalyse wordt gemaakt.

p Zie *Index (page 1774)*, waardoor complexere tekenreeksanalyse mogelijk is.

# Len

**Len()** retourneert de lengte van de invoertekenreeks.

**Syntaxis:**

```
Len(text)
```

**Retourgegevenstypen:** geheel getal

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De tekenreeks die moet worden geëvalueerd. |

Voorbeeld: diagramuitdrukking

| Voorbeeld | Resultaat |
|---|---|
| `Len('Peter')` | Retourneert `5` |

## Voorbeeld - Basisprincipes voor Len

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `CustomerComment`. Het veld bevat de oorspronkelijke teksttekenreeks die geëvalueerd moet worden.

### Load-script

```
Example:
Load  * inline [
CustomerComment
Please deliver after 5 PM.
Thank you for the quick service!
Can you add a gift wrap?
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `CustomerComment`

Maak de volgende meting:

- `=Len(CustomerComment)`

Resultatentabel

| CustomerComment | Len(CustomerComment) |
|---|---|
| Can you add a gift wrap? | 24 |
| Please deliver after 5 PM. | 26 |
| Thank you for the quick service! | 32 |

De uitvoer van de meting `Len(CustomerComment)` retourneert de lengte van de `CustomerComment` invoertekenreeks.

## Voorbeeld - Scenario met Len

Diagramuitdrukking

### Overzicht

Dit voorbeeld analyseert feedback van klanten door te kijken naar de lengte van elke opmerking als indicator voor de diepte of het detailniveau ervan.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `TicketID`
  - `Description`

### Load-script

```
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."
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `TicketID`
- `Description`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=If(Len(Description) < 30, 'Short',If(Len(Description) <= 50, 'Medium', 'Long'))` om een categorie (Short, Medium, Long) te berekenen en toe te wijzen aan het ticket op basis van de lengte van de beschrijving.

Resultatentabel

| TicketID | Beschrijving | If(Len(Description) < 30, 'Short',If(Len (Description) <= 50, 'Medium', 'Long')) |
|---|---|---|
| 1001 | I received the wrong product. | Short |
| 1002 | The package arrived damaged, and I would like a replacement. | Long |
| 1003 | I've been trying to reset my password for two days, and I haven't received an email. | Long |
| 1004 | Is product XYZ available in size Large? | Gemiddeld |
| 1005 | My order status shows 'Delivered' but I have not received my package yet. | Long |
| 1006 | I need help with an exchange for a faulty product, and I've attached photos. | Long |

De uitvoer van de berekende dimensie laat zien hoe u de functie `Len` kunt gebruiken om gegevens te categoriseren door de lengte van teksttekenreeksen te interpreteren.

## Voorbeeld - Scenario met Len en het gebruik van tekenreeksmanipulatie

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `InputText`.

### Load-script

```
Example:
Load  * inline [
InputText
this is a sample text string
capitalize first letter only
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `InputText`

Maak de volgende berekende dimensies:

- `=Upper(Left(InputText,1))` , om de eerste letter van de teksttekenreeks om te zetten naar hoofdletters.
- `=Mid(InputText,Len(upper(Left(InputText,1)))+1)`, om de eerste letter uit de teksttekenreeks te verwijderen.
- `=Upper(left(InputText,1)) & Mid(InputText,len(upper(left(InputText,1)))+1)` om de uitvoer van de eerste berekende dimensie te combineren met de uitvoer van de tweede berekende dimensie.

Resultatentabel

| InputText | Upper(Left (InputText,1)) | Mid(InputText,Len (upper(Left (InputText,1)))+1) | Upper(left(InputText,1)) & mid (InputText,len(upper(left (InputText,1)))+1) |
|---|---|---|---|
| this is a sample text string | T | his is a sample text string | This is a sample text string |
| capitalize first letter only | C | apitalize first letter only | Capitalize first letter only |

In de eerste berekende dimensie worden de functies `Upper` en `Left` gecombineerd om de eerste letter van de `InputText` als hoofdletter te retourneren. In de tweede berekende dimensie gebruikt de functie `Mid` de functie `Len` om een teksttekenreeks te retourneren die het eerste teken uit `InputText` verwijdert. De derde berekende dimensie combineert de eerste en tweede berekende dimensie en retourneert de `InputText` tekenreeks met een hoofdletter als eerste teken.

#### Load-script

Dit voorbeeld gebruikt dezelfde functies (`Upper`, `Mid`, en `Len`) als het scenario met de diagramuitdrukking. Het load-script maakt een nieuw veld, `NewInputText`, dat de `InputText` met een hoofdletter als eerste teken retourneert.

```
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 ];
```

Resultatentabel

| InputText | NewInputText |
|---|---|
| this is a sample text string | This is a sample text string |
| capitalize first letter only | Capitalize first letter only |

## LevenshteinDist

**LevenshteinDist()** retourneert de afstand van Levenshtein tussen twee tekenreeksen. Het wordt gedefinieerd als het aantal bewerkingen van één teken (invoegingen, verwijderingen of vervangingen) dat is vereist om een tekenreeks te wijzigen. De functie kan worden gebruikt om fuzzy tekenreeksen te vergelijken.

**Syntaxis:**

```
LevenshteinDist(text1, text2)
```

**Retourgegevenstypen:** geheel getal

Argumenten

| Argument | Beschrijving |
|---|---|
| **text1** | De eerste tekenreeks. |
| **text2** | De tweede tekenreeks die vergeleken zal worden met de eerste tekenreeks om het minimumaantal bewerkingen van één teken te berekenen. |

Voorbeeld: diagramuitdrukking

| Voorbeeld | Resultaat |
|---|---|
| `LevenshteinDist( 'Kitten','Sitting' )` | Retourneert `3` |

### Voorbeeld - Basisprincipes voor LevenshteinDist

Diagramuitdrukking

**Overzicht**

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `InputText`.

**Load-script**

```
Example:
Load * inline [
InputText
```

```
Sliver
SSiver
SSiveer
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `InputText`

Maak de volgende meting:

- `=LevenshteinDist('Silver', InputText)` om het minimumaantal bewerkingen van één teken te berekenen dat nodig is om de tekenreekswaarde voor `InputText` te wijzigen in het woord `Silver`.

Resultatentabel

| InputText | LevenshteinDist('Silver', InputText) |
|---|---|
| Sliver | 2 |
| SSiveer | 3 |
| SSiver | 2 |

De uitvoer van de functie `LevenshteinDist` geeft het aantal vereiste wijzigingen om de `InputText` te wijzigen in de verwachte tekst, `Silver`. Voor de eerste rij zijn bijvoorbeeld twee wijzigingen nodig om het woord `Sliver` te wijzigen in `Silver`. De tweede rij vereist 3 wijzigingen: 1) Verwijderen van het extra teken `s`. 2) Verwijderen van het extra teken `e`. 3) Invoegen van een nieuw teken in `l`.

## Voorbeeld - Scenario met LevenshteinDist

Diagramuitdrukking

#### Overzicht

In dit voorbeeld worden productnamen uit verschillende systemen samengevoegd. De productnamen gebruiken niet altijd dezelfde schrijfwijze vanwege typefouten, afkortingen, spaties of andere variaties. Met behulp van de functie `LevenshteinDist` kunt u de gelijkenis tussen twee productnamen meten en vaststellen welke waarschijnlijk naar hetzelfde product verwijzen, zelfs als de namen niet identiek zijn.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `ProductA`
  - `ProductB`

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `ProductA`
- `ProductB`

Maak de volgende meting:

- `=LevenshteinDist(ProductA, ProductB)` om het minimum aantal bewerkingen van één teken te berekenen dat nodig is om de tekenreekswaarde voor `ProductB` te wijzigen zodat het overeenkomt met `ProductA`.

Resultatentabel

| 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 |

De Levenshtein-afstand is een type fuzzy matching dat veel wordt gebruikt als onderdeel van spellingscontroles, optische tekenherkenning en correctiesystemen in gebieden zoals klantgegevensbeheer, inventarisatiesystemen en documentverwerking, waar kleine variaties in tekst vaak voorkomen.

### Load-script

De volgende code laat zien hoe u de functie in een load-script kunt gebruiken.

### Load-script

```
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
ビビビ ];

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;
```

Resultatentabel

| 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 | Bold | 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()** converteert alle tekens in de invoertekenreeks naar kleine letters.

**Syntaxis:**

```
Lower(text)
```

**Retourgegevenstypen:** tekenreeks

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De tekenreeks die moet worden geëvalueerd. |

Voorbeeld: diagramuitdrukking

| Voorbeeld | Resultaat |
|---|---|
| `Lower( 'abcD' )` | Retourneert `abcd` |

### Voorbeeld - Basisprincipes voor Lower

Diagramuitdrukking

**Overzicht**

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `InputText`.

### Load-script

```
Example:
Load * inline [
InputText
rHode iSland
washingTon d.C.
new york
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `InputText`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=Lower(InputText)` om alle tekens in het veld `InputText` te converteren naar kleine letters.

Resultatentabel

| InputText | Lower(InputText) |
|---|---|
| new york | new york |
| rHode iSland | rhode island |
| washingTon d.C. | washington d.c. |

De uitvoer van de functie `Lower` converteert alle waarden naar kleine letters.

### Load-script

De volgende code laat zien hoe u de functie in een load-script kunt gebruiken.

```
Load
String,
Lower(String)
Inline
[String
rHode iSland
washingTon d.C.
new york];
```

Resultatentabel

| Tekenreeks | Lower(String) |
|---|---|
| rHode iSland | rhode island |
| washingTon d.C. | washington d.c. |
| new york | new york |

## Voorbeeld - Scenario met Lower

Diagramuitdrukking

### Overzicht

Een gegevensverzameling met klantgegevens bevat e-mailadressen met een combinatie van hoofdletters en kleine letters voor het domein. Het bedrijf wil deze adressen standaard voorzien van allemaal kleine letters.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `CustomerID`
  - `EmailAddress`
  - `Inquiry`

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `CustomerID`
- `EmailAddress`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=Lower(EmailAddress)` om het domeingedeelte van de e-mail als een nieuwe kolom te scheiden en de tekens om te zetten naar kleine letters.

Resultatentabel

| 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 |

De uitvoer van de `Lower` functie retourneert alle e-mailadressen in kleine letters.

# LTrim

**LTrim()** retourneert de invoertekenreeks, ontdaan van beginspaties.

**Syntaxis:**

```
LTrim(text)
```

**Retourgegevenstypen:** tekenreeks

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De tekenreeks die moet worden geëvalueerd. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `LTrim( ' abc' )` | Retourneert `abc` |
| `LTrim( 'abc ' )` | Retourneert `abc` |

## Voorbeeld - Basisprincipes voor LTrim

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `InputText`.

### Load-script

```
Set verbatim=1;
Example:
Load * inline [
InputText
'   abc    '
' def  '];
```

*De opdracht "`Set verbatim=1`" is opgenomen in het voorbeeld om ervoor te zorgen dat de ruimten niet automatisch worden afgekapt voordat de `LTrim` functie wordt getoond. Raadpleeg Verbatim (page 225) voor meer informatie.*

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `InputText`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=LTrim(InputText)` om voorloopspaties te verwijderen uit `InputText`.

Resultatentabel

| InputText | LTrim(InputText) |
|---|---|
| ' abc ' | 'abc ' |
| ' def ' | 'def ' |

De uitvoer van de functie `LTrim` verwijdert alle voorloopspaties links van de oorspronkelijke tekst, maar behoudt alle volgspaties.

## Voorbeeld - Scenario met LTrim

Diagramuitdrukking

### Overzicht

Een CRM-systeem (Customer Relationship Management) bevat records met inconsistente gegevensinvoer die extra voorloopspaties bevatten. Voor rapportagedoeleinden moeten de gegevens worden opgeschoond om deze spaties te verwijderen en om ervoor te zorgen dat de klantnamen goed worden gesorteerd en gegroepeerd.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `CustomerName`.

### Load-script

```
Set verbatim=1;
Example:
Load * inline [
CustomerName
'  John Doe'
'Jane Smith'
```

```
'  Michael Johnson'
'Emily Davis'
];
```

*De opdracht "`Set verbatim=1`" is opgenomen in het voorbeeld om ervoor te zorgen dat de ruimten niet automatisch worden afgekapt voordat de `LTrim` functie wordt getoond. Raadpleeg Verbatim (page 225) voor meer informatie.*

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `CustomerName`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=LTrim(CustomerName)` om alle voorloopspaties te verwijderen uit `CustomerName`.

Resultatentabel

| CustomerName | LTrim(CustomerName) |
|---|---|
| '  Michael Johnson' | 'Michael Johnson' |
| '  John Doe' | 'John Doe' |
| 'Emily Davis' | 'Emily Davis' |
| 'Jane Smith' | 'Jane Smith' |

De uitvoer laat zien dat de functie `LTrim` alle voorloopspaties heeft verwijderd uit de oorspronkelijke tekenreekswaarden in `CustomerName`.

## Voorbeeld - Geavanceerd scenario met LTrim

Diagramuitdrukking

#### Overzicht

Dit voorbeeld verwijdert alle voorloopspaties uit de oorspronkelijke teksttekensreeks. De diagramuitdrukking bevat metingen die de `Len` functie gebruiken om de tekens in de tekenreeks voor en na het gebruik van de `LTrim` functie te tellen.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `String`.

### Load-script

```
Set verbatim=1;
Example:
Load * inline [
String
'   abc    '
' def  '];
```

*De opdracht "Set verbatim=1" is opgenomen in het voorbeeld om ervoor te zorgen dat de ruimten niet automatisch worden afgekapt voordat de `LTrim` functie wordt getoond. Raadpleeg Verbatim (page 225) voor meer informatie.*

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `String`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=LTrim(String)` om alle voorloopspaties te verwijderen.

Maak de volgende metingen:

- `=Len(String)` om de lengte van de oorspronkelijke tekenreeks te tellen.
- `=Len(LTrim(String))` om de lengte van de tekenreeks te tellen nadat de voorloopspaties zijn verwijderd.

Resultatentabel

| Tekenreeks | LTrim(String) | Len(String) | Len(LTrim(String)) |
|---|---|---|---|
| ' abc ' | 'abc ' | 10 | 7 |
| ' def ' | 'def ' | 6 | 5 |

Wanneer u de uitvoer van de functie `LTrim` vergelijkt met de oorspronkelijke tekensreekswaarden in het script, kunt u zien hoe alle voorloopspaties zijn verwijderd.

### Load-script

De volgende code laat zien hoe u de functie in een load-script kunt gebruiken.

```
Set verbatim=1;
Example:

Load *,
len(LtrimString) as LtrimStringLength;
Load *,
ltrim(String) as LtrimString;
Load *,
len(String) as StringLength;
```

```
Load * Inline [
String
'   abc    '
' def  '];
```

*De opdracht "Set verbatim=1" is opgenomen in het voorbeeld om ervoor te zorgen dat de ruimten niet automatisch worden afgekapt voordat de ltrim-functie wordt getoond. Raadpleeg Verbatim (page 225) voor meer informatie.*

Resultatentabel

| Tekenreeks | StringLength | LtrimString | LtrimStringLength |
|---|---|---|---|
| abc | 10 | abc | 7 |
| def | 6 | def | 5 |

**Zie ook:**

p RTrim (page 1860)

## MatchRegEx

**MatchRegEx()** vergelijkt de invoertekenreeks met een of meer gespecificeerde reguliere expressiepatronen en retourneert de numerieke locatie van de overeenkomende reguliere expressiepatronen. Als er geen overeenkomsten worden gevonden, retourneert de functie 0. Deze functie zoekt alleen naar exacte overeenkomsten.

Deze functie voert regexbewerkingen uit die hoofdlettergevoelig zijn. U kunt eventueel ook de variant **MatchRegExI()** gebruiken om hoofdletteronafhankelijke regexbewerkingen uit te voeren.

**Syntaxis:**

```
MatchRegEx (text, regex1 [ , regex2,...regexN])
```

**Retourgegevenstypen:** geheel getal

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De tekst van de invoertekenreeks die u wilt afstemmen met één of meer reguliere expressies. |
| **regex** | De reguliere expressie om te proberen af te stemmen met de invoertekenreeks. U kunt meer dan één reguliere expressie in één expressie opgeven, gescheiden als opeenvolgende argumenten. |

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `MatchRegEx('abc123','[a-z]','[0-9]','[a-z]+[0-9]+','.+')` | Retourneert `3`.Het derde regexpatroon is het eerste dat overeenkomt met de gegevens. |
| `MatchRegEx('ABC','[abc]+','[123]')` | Retourneert `0`, omdat **MatchRegEx()** hoofdlettergevoelig is. |
| `MatchRegExI('ABC','[abc]+','[123]')` | Retourneert `1`. De hoofdletterongevoelige variant van de functie, **MatchRegExI()**, wordt gebruikt. |

## Wanneer gebruiken

Gebruiksscenario's voor deze functie zijn onder andere:

- Tekst valideren om te voldoen aan opmaak- en nalevingsnormen en om fouten in gegevens op te sporen.
- Verschillen in de opmaak van tekst identificeren. U kunt bijvoorbeeld vaststellen hoeveel telefoonnummers indeling A gebruiken, indeling B, enzovoort.

## Voorbeeld 1 - load-script om indelingen van telefoonnummers te analyseren

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een tabel `BusinessContactInfo` met telefoonnummers van contactpersonen voor verschillende bedrijven. De telefoonnummers zijn 10-cijferige NANP-telefoonnummers.
- Een berekend veld, `PhoneNumberFormat`, om te beoordelen of elk telefoonnummer overeenkomt met één van een specifieke set telefoonnummerindelingen.

Er zijn drie verschillende indelingen waarop we willen controleren: (###) ###-####, ###-###-#### en ##########.

### Load-script

```
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');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `CompanyName`
- `ContactInfo`
- `PhoneNumberFormat`

Resultatentabel

| 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 |

Van hieruit kunt u bijvoorbeeld KPI's maken om het totale aantal van elke telefoonnummerindeling te berekenen, evenals het aantal telefoonnummers dat geen erkende indeling heeft:

Voorbeelden van aggregatie-uitdrukkingen om exemplaren van elke indeling te tellen

| Opmaak | Aantal instanties | Diagramuitdrukking |
|---|---|---|
| (###) ###-#### | 3 | =Count({<PhoneNumberFormat={'1'}>} PhoneNumberFormat) |
| ###-###-#### | 3 | =Count({<PhoneNumberFormat={'2'}>} PhoneNumberFormat) |
| ########## | 3 | =Count({<PhoneNumberFormat={'3'}>} PhoneNumberFormat) |
| Unrecognized format | 1 | =Count({<PhoneNumberFormat={'0'}>} PhoneNumberFormat) |

## Voorbeeld 2 - diagramuitdrukking om ISBN-indeling te identificeren

Diagramuitdrukking en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een tabel `Books` met ISBN-codes voor verschillende boeken.

Op basis van deze gegevens willen we bepalen of elke ISBN-code een geldige 10-cijferige of 13-cijferige code is, op basis van de syntaxis die in een regexpatroon is gespecificeerd.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Book`

Voeg deze berekende dimensie toe en geef deze het label *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'))
```

Resultatentabel

| 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()** retourneert het gedeelte van de invoertekenreeks dat begint op de positie van het teken dat is gedefinieerd door het tweede argument, 'start', en retourneert het aantal tekens dat is gedefinieerd door het derde argument, 'count'. Als 'count' wordt weggelaten, wordt de rest van de invoertekenreeks geretourneerd. Het eerste teken in de invoertekenreeks heeft het nummer 1.

**Syntaxis:**

```
Mid(text, start[, count])
```

**Retourgegevenstypen:** tekenreeks

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De oorspronkelijke tekenreeks. |
| **start** | Geheel getal dat de positie definieert van het eerste teken in **text** dat moet worden opgenomen. |
| **count** | Definieert de tekenreekslengte van de uitvoertekenreeks. Bij weglating, worden alle tekens opgenomen vanaf de positie die is gedefinieerd door **start**. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `Mid( 'abcdef',3 )` | Retourneert `cdef` |
| `Mid( 'abcdef',3, 2 )` | Retourneert `cd` |

## Voorbeeld - Basisprincipes voor Mid

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `InputText` de oorspronkelijke teksttekenreeks
  - `StartPosition`, de beginpositie van het eerste teken dat na verwerking in de oorspronkelijke teksttekenreeks moet worden opgenomen
  - `CharacterCount`, het aantal tekens dat moet worden geretourneerd (optionele parameter)

### Load-script

```
Example:
Load * inline [
InputText, StartPosition, CharacterCount
abcdef, 3, 2
abcdef, 2, 3
210714, 3, 2
210714, 2, 3
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `InputText`
- `StartPosition`
- `CharacterCount`

Maak de volgende berekende dimensies:

- `=mid(InputText,StartPosition)` om de tekens te berekenen die uit `InputText` geëxtraheerd moeten worden, beginnend bij de waarde in het veld `StartPosition`. De functie retourneert alle tekens vanaf de waarde `StartPosition` tot het einde van `InputString` omdat er geen `Count` parameter is opgegeven.

- `=mid(InputText,StartPosition,CharacterCount)` om de tekens te berekenen die uit `InputText` geëxtraheerd moeten worden, beginnend bij de waarde in het veld `StartPosition`. De waarde in `CharacterCount` bepaalt het aantal tekens dat wordt geretourneerd, beginnend vanaf de waarde in het veld `StartPosition`.

Resultatentabel

| InputText | Startpositie | CharacterCount | Mid (Invoertekst,Startpositie) | Mid (Invoertekst,Startpositie,AantalTekens |
|---|---|---|---|---|
| abcdef | 2 | 3 | bcdef | bcd |
| abcdef | 3 | 2 | cdef | cd |
| 20210714 | 2 | 3 | 10714 | 107 |
| 20210714 | 3 | 2 | 0714 | 07 |

De eerste rij retourneert de waarde `bcdef` voor de eerste uitdrukking. De uitdrukking begint op positie 2 en retourneert alle tekens omdat het `Count` argument niet is ingesteld. De tweede uitdrukking begint ook op positie 2, maar retourneert alleen de tekens `bcd` omdat het `Count` argument 2 is.

De tweede rij retourneert de waarde `cdef` voor de eerste uitdrukking. De uitdrukking begint op positie 3 en retourneert alle tekens omdat het `Count` argument niet is ingesteld. De tweede uitdrukking begint ook op positie 3, maar retourneert alleen de tekens `cd` omdat het `Count` argument 2 is.

De derde rij retourneert de waarde `10714` voor de eerste uitdrukking. De uitdrukking begint op positie 2 en retourneert alle tekens omdat het `Count` argument niet is ingesteld. De tweede uitdrukking begint ook op positie 2, maar retourneert alleen de tekens `107` omdat het `Count` argument 3 is.

De vierde rij retourneert de waarde `0714` voor de eerste uitdrukking. De uitdrukking begint op positie 3 en retourneert alle tekens omdat het `Count` argument niet is ingesteld. De tweede uitdrukking begint ook op positie 2, maar retourneert alleen de tekens `07` omdat het `Count` argument 2 is.

### Load-script

De volgende code laat zien hoe u de functie in een load-script kunt gebruiken.

```
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
];
```

Resultatentabel

| Tekst | Startpositie | Mid1 | Aantal | Mid2 |
|---|---|---|---|---|
| abcdef | 3 | cdef | 2 | cd |
| abcdef | 2 | bcdef | 3 | bcd |
| 210714 | 3 | 0714 | 2 | 07 |
| 210714 | 2 | 10714 | 3 | 107 |

## Voorbeeld - Scenario met Mid

Diagramuitdrukking

### Overzicht

Een gegevensbron bevat een kolom met de naam `OrderID` die de volgende gestructureerde indeling gebruikt: `ORDYYYY-COUNTRYCODE-ORDERNUMBER`. Om aan een rapportagevereiste te voldoen, moet u de landcode als een aparte kolom extraheren en presenteren.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Orders`.
- Eén veld in de gegevenstabel met de naam `OrderID`.

### Load-script

```
Orders:
Load * inline [
OrderID
ORD2024-US-123456
ORD2024-UK-654321
ORD2024-CA-789123
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `OrderID`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=Mid(OrderID, 9, 2)` om de tweeletterige landcode te extraheren uit de `OrderID`.

Resultatentabel

| OrderID | Mid(OrderID, 9, 2) |
|---|---|
| ORD2024-CA-789123 | CA |
| ORD2024-UK-654321 | UK |
| ORD2024-US-123456 | VS |

De uitvoer van de `Mid` functie heeft met succes de tweeletterige landcodes uit de `OrderID` tekenreekswaarden gehaald.

**Zie ook:**

p *Index (page 1774)*

## Ord

**Ord()** returneert de numerieke (ASCII of Unicode) waarde van het eerste teken van een tekenreeks. Deze functie is handig om tekenreeksen te evalueren of te vergelijken op basis van hun onderliggende tekencodes, bijvoorbeeld bij het sorteren of filteren van tekenreeksen met niet-standaard tekens.

**Syntaxis:**

```
Ord(text)
```

**Retourgegevenstypen:** geheel getal

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De oorspronkelijke tekenreeks. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `Ord( 'A' )` | Retourneert het gehele getal `65` |
| `Ord( 'Ab' )` | Retourneert het gehele getal `65` |

### Voorbeeld - Basisprincipes voor Ord

Diagramuitdrukking

**Overzicht**

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
    - `Chinese`
    - `Western`

### Load-script

```
Example:
Load * inline [
Chinese, Western
古琴, Guqin
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `Chinese`
- `Western`

Maak de volgende berekende dimensies:

- `=ord(Chinese)` om de numerieke waarde van de waarde in het veld met de naam `Chinese` te retourneren.
- `=ord(Western)` om de numerieke waarde van de waarde in het veld met de naam `Western` te retourneren.

Resultatentabel

| Chinese | Western | ord(Chinese) | ord(Western) |
|---|---|---|---|
| 古琴 | Guqin | 21476 | 71 |

### Load-script

De volgende code laat zien hoe u de functie in een load-script kunt gebruiken.

```
Example:
Load *,
ord(Chinese) as OrdUnicode,
ord(Western) as OrdASCII;
Load * inline [
Chinese, Western
古琴, Guqin ];
```

Resultatentabel

| Chinese | Western | OrdUnicode | OrdASCII |
|---|---|---|---|
| 古琴 | Guqin | 21476 | 71 |

## Voorbeeld - Scenario met Ord

Diagramuitdrukking

### Overzicht

Een gegevensverzameling met productcodes bevat enkele codes die als ongeldig worden beschouwd omdat ze beginnen met een speciaal teken in plaats van een alfanumeriek teken. Met behulp van de functie `Ord` om de ongeldige productcodes te identificeren, wordt een nieuwe meting gemaakt om de records het label Geldig of Ongeldig te geven.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Products`.
- Eén veld in de gegevenstabel met de naam `ProductCode`.

### Load-script

```
Products:
Load * inline [
ProductCode
AB123
#EX45
12XY9
DEF456
*KLM78
GH789
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `ProductCode`

Maak de volgende berekende dimensies:

- `=Ord(ProductCode)` om de numerieke waarde van het eerste teken terug te geven.
- `=If(Ord(ProductCode) < 48 OR (Ord(ProductCode) > 57 AND Ord(ProductCode) < 65) OR Ord (ProductCode) > 122, 'Invalid Product Code', 'Valid Product Code')` om te controleren of de rangtelwoordwaarde van de code alfanumeriek is. Als dat niet het geval is, wordt dat record gemarkeerd als `Not Valid`.

Resultatentabel

| Productcode | 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 |

Door de uitvoer van de `Ord` functie te vergelijken met een bereik van geldige alfanumerieke waarden, kunt u bepalen welke productcodes niet geldig zijn.

## PurgeChar

**PurgeChar()** retourneert een tekenreeks die bestaat uit de tekens die zijn opgenomen in de invoertekenreeks ('text'), behalve de tekens die voorkomen in het tweede argument ('remove_chars').

**Syntaxis:**

```
PurgeChar(text, remove_chars)
```

**Retourgegevenstypen:** tekenreeks

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De oorspronkelijke tekenreeks. |
| **remove_chars** | Een tekenreeks die de tekens in **text** bevat die moeten worden verwijderd. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `PurgeChar( 'a1b2c3','123' )` | Retourneert `abc` |
| `PurgeChar( 'a1b2c3','1234' )` | Retourneert `abc` |
| `PurgeChar( 'a1b22c3','1234' )` | Retourneert `abc` |
| `PurgeChar( 'a1b2c3','312' )` | Retourneert `abc` |
| `PurgeChar ( 'a1b2c3','Abc' )` | Retourneert `a123` |

## Voorbeeld - Basisprincipes voor PurgeChar

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `InputText`
  - `CharsToPurge`

### Load-script

```
Example:
Load * inline [
InputText, CharsToPurge
'a1b2c3', '123'
'a^b^c', '^'
'A^b^c', 'A^'
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `InputText`
- `CharsToPurge`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=PurgeChar(InputText,CharsToPurge)`, om de tekens in `InputText` te berekenen die overeenkomen met de tekens in `CharsToPurge` en alleen de tekens te retourneren die niet overeenkomen.

Results table

| InputText | CharsToPurge | PurgeChar(InputText,CharsToPurge) |
|---|---|---|
| A^b^c | A^ | bc |
| a^b^c | ^ | abc |
| a1b2c3 | 123 | abc |

De uitvoer van de functie `PurgeChar` retourneert alleen de tekens van `InputText` die niet overeenkomen met de tekens in `CharsToPurge`. De eerste regel retourneert bijvoorbeeld `bc` en verwijdert de andere tekens.

### Load-script

De volgende code laat zien hoe u de functie in een load-script kunt gebruiken.

```
Example:
Load
*,
purgechar(String1, String2) as PurgeChar;
Load * inline [
String1, String2
'a1b2c3', '123'
];
```

Resultatentabel

| String1 | String2 | PurgeChar |
|---|---|---|
| a1b2c3 | 123 | abc |

## Voorbeeld - Scenario met PurgeChar

### Diagramuitdrukking

#### Overzicht

Een gegevensverzameling bevat klantnamen en telefoonnummers. Dit voorbeeld parseert de tekenreeksen met telefoonnummers om alleen numerieke tekens te behouden en alle andere tekens, zoals haakjes, streepjes, punten, plustekens en spaties te verwijderen.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `CustomerName`
  - `PhoneNumber`

#### Load-script

```
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
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `CustomerName`
- `PhoneNumber`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=PurgeChar(PhoneNumber, '()- .+ ')`, om de niet-numerieke tekens uit het veld `PhoneNumber` te extraheren en alleen numerieke tekens te retourneren.

Results table

| CustomerName | PhoneNumber | PurgeChar(PhoneNumber,'()-.+ ') |
|---|---|---|
| Alice Brown | +1 (123) 456 7890 | 11234567890 |
| Bob Johnson | 123.456.7890 | 1234567890 |
| Jane Smith | 123-456-7890 | 1234567890 |
| John Doe | (123) 456-7890 | 1234567890 |

De uitvoer van de functie `PurgeChar` heeft met succes alle niet-numerieke tekens uit de `PhoneNumber` tekenreeks verwijderd en alleen getallen geretourneerd.

**Zie ook:**

p *KeepChar (page 1801)*

## Repeat

**Repeat()** vormt een tekenreeks die bestaat uit de invoertekenreeks die het aantal keren wordt herhaald dat wordt gedefinieerd door het tweede argument.

**Syntaxis:**

```
Repeat(text[, repeat_count])
```

**Retourgegevenstypen:** tekenreeks

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De oorspronkelijke tekenreeks. |
| **repeat_ count** | Definieert het aantal keren dat de tekens in de tekenreeks **text** moet worden herhaald in de uitvoertekenreeks. |

Voorbeeld: diagramuitdrukking

| Voorbeeld | Resultaat |
|---|---|
| `Repeat( ' * ', 4 )` | Retourneert `****` |

## Voorbeeld - Basisprincipes voor Repeat

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `InputText`.

### Load-script

```
Example:
Load * inline [
InputText
hello world!
hOw aRe you?
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `InputText`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=Repeat(InputText,2)` om het aantal keren te berekenen dat de waarde in `InputText` moet worden herhaald. In dit voorbeeld wordt elke tekenreeks twee keer herhaald.

Resultatentabel

| InputText | Repeat(InputText,2) |
|---|---|
| hallo wereld! | `hello world!hello world!` |
| Hoe gaat het met je? | `hOw aRe you?hOw aRe you?` |

De uitvoer van de functie `Repeat` bevat twee exemplaren van de tekenreeks `InputText`, bijvoorbeeld, rij 1 retourneert `hello world!hello world!`.

Load-script

De volgende code laat zien hoe u de functie in een load-script kunt gebruiken.

```
Example:
Load *,
repeat(String,2) as Repeat;
Load * inline [
String
```

```
hello world!
hOw aRe you? ];
```

Resultatentabel

| Tekenreeks | Herhalen |
|---|---|
| hello world! | hello world!hello world! |
| hOw aRe you? | hOw aRe you?hOw aRe you? |

## Voorbeeld 5 – Scenario met Repeat

Diagramuitdrukking

### Overzicht

Dit voorbeeld gebruikt klanttevredenheidsgegevens om een numerieke tevredenheidscore om te zetten in een visuele weergave met behulp van smiley-emoji's ☺.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `Customer`
  - `SatisfactionScore`

### Load-script

```
Example:
Load * inline [
Customer, SatisfactionScore
John,4
Mary,2
Peter,5
Susan,3
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `Customer`
- `SatisfactionScore`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=Repeat('☺',(SatisfactionScore))` om de score te berekenen en visueel weer te geven met emoji's.

Resultatentabel

| Klant | SatisfactionScore | Repeat ('☺',SatisfactionScore) |
|---|---|---|
| John | 4 | ☺☺☺☺ |
| Mary | 2 | ☺☺ |
| Peter | 5 | ☺☺☺☺☺ |
| Susan | 3 | ☺☺☺ |

De uitvoer van de `Repeat` functie toont een visuele weergave van de klanttevredenheid met behulp van smiley-emoji's om de numerieke tevredenheidsscore weer te geven.

## Voorbeeld - Geavanceerd scenario met Repeat

Diagramuitdrukking

### Overzicht

Een gegevensverzameling met klantrekeningen bevat rekeningnummers met verschillende tekenlengtes. Dit voorbeeld laat zien hoe u de functie `Repeat` kunt gebruiken om de lengte van rekeningnummers te standaardiseren.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `AccountNo`.

### Load-script

```
Example:
Load * inline [
AccountNo
123
45125783
41523
489654
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `AccountNo`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=Repeat(0,10 - Len(AccountNo)) & AccountNo`, om het rekeningnummer op te vullen met nullen tot een totale lengte van 10 tekens.

Resultatentabel

| AccountNo | Repeat(0, 10 - Len(AccountNo)) & AccountNo |
|---|---|
| 123 | 0000000123 |
| 41523 | 0000041523 |
| 489654 | 0000489654 |
| 45125783 | 0045125783 |

De uitvoer van de `Repeat` functie toont een gestandaardiseerde lengte van 10 tekens voor alle rekeningnummers.

## Replace

**Replace()** retourneert een tekenreeks nadat alle treffers van een bepaalde subtekenreeks in de tekenreeks zijn vervangen door een andere subtekenreeks. De functie is niet-recursief en wordt van links naar rechts uitgevoerd.

**Syntaxis:**

```
Replace(text, from_str, to_str)
```

**Retourgegevenstypen:** tekenreeks

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De oorspronkelijke tekenreeks. |
| **from_str** | Een tekenreeks die een of meer keren in de invoerreeks **text** mag voorkomen. |
| **to_str** | De tekenreeks die alle treffers van **from_str** in de tekenreeks **text** vervangt. |

Voorbeeld: diagramuitdrukking

| Voorbeeld | Resultaat |
|---|---|
| `Replace( 'abccde','cc','xyz' )` | Retourneert `abxyzde` |

## Voorbeeld - Basisprincipes voor Replace

Diagramuitdrukking

### Overzicht

Een gegevensverzameling met klantnamen bevat enkele spelfouten. Dit voorbeeld gebruikt de `Replace` functie om elke onjuiste herhaling van 'Jhon' te verwerken en te vervangen door 'John'.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `CustomerID`
  - `CustomerName`

### Load-script

```
Example:
Load * inline [
CustomerID, CustomerName
1, Jhon Smith
2, Jhon Doe
3, John Williams
4, Jhonathan Harris
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `CustomerID`
- `CustomerName`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=Replace(CustomerName,'Jhon','John')` om alle gevallen van `Jhon` te vervangen door `John`.

Resultatentabel

| CustomerID | CustomerName | Replace (CustomerName,'Jhon','John') |
|---|---|---|
| 1 | Jhon Smith | John Smith |
| 2 | Jhon Doe | John Doe |
| 3 | John Williams | John Williams |
| 4 | Jhonathan Harris | Johnathan Harris |

De uitvoer van de functie `Repeat` vindt en corrigeert alle exemplaren van `Jhon` met `John`.

## Voorbeeld 5 – Scenario met Replace

Diagramuitdrukking

### Overzicht

Een gegevensverzameling van producten gebruikt inconsistente labels voor de categoriecodes. Elektrische producten hebben bijvoorbeeld meerdere codevarianten, zoals `ELEC`, `ELC` en `elc`. Om de codes te standaardiseren, wordt de functie `Replace` gebruikt om alle inconsistente categoriecodes voor zowel elektrische producten als meubelproducten te corrigeren.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `ProductID`
  - `CategoryCode`

### Load-script

```
Example:
Load * inline [
ProductID, CategoryCode
1, ELEC
2, ELC
3, FURN
4, FRN
5, ELEC
6, ELC
7, elc
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `ProductID`
- `CategoryCode`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=Replace(Replace(Upper(CategoryCode), 'ELC', 'ELEC'),'FRN', 'FURN')` om alle exemplaren van `ELC` te vervangen door `ELEC` en alle exemplaren van `FRN` door `FURN`.

Resultatentabel

| ProductID | CategoryCode | Replace(Replace(Upper (CategoryCode), 'ELC', 'ELEC'),'FRN', 'FURN') |
|---|---|---|
| 1 | ELEC | ELEC |
| 2 | ELC | ELEC |
| 3 | FURN | FURN |
| 4 | FRN | FURN |
| 5 | ELEC | ELEC |
| 6 | ELC | ELEC |
| 7 | elc | ELEC |

De uitvoer van de functie `Replace` heeft alle exemplaren van `ELC` vervangen door `ELEC`, en `FRN` door `FURN`. Let op dat de code `elc` ook werd vervangen. Hoewel de functie `Replace` hoofdlettergevoelig is, gebruikt de dimensieformule de functie `Upper` om de hoofdletters en kleine letters van alle `CategoryCodes` aan te passen voordat de tekenreeks wordt vervangen. Hierdoor wordt de vervanging van de tekenreeks hoofdletterongevoelig.

---

**Zie ook:**

## ReplaceRegEx

**ReplaceRegEx()** retourneert een tekenreeks na het vervangen van één of meer overeenkomsten tussen een invoertekenreeks en een gespecificeerd regulier expressiepatroon. De tekst die de overeenkomende tekst vervangt, wordt opgegeven in het argument **to_str**. De functie wordt van links naar rechts uitgevoerd, maar als u een negatieve **occurrence** waarde opgeeft, leest de functie van rechts naar links.

Deze functie voert regexbewerkingen uit die hoofdlettergevoelig zijn. U kunt eventueel ook de variant **ReplaceRegExl()** gebruiken om hoofdletteronafhankelijke regexbewerkingen uit te voeren.

**Syntaxis:**

```
ReplaceRegEx (text, regex, to_str [, occurrence])
```

**Retourgegevenstypen:** tekenreeks

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De tekst van de invoertekenreeks die u geheel of gedeeltelijk wilt vervangen door de nieuwe tekst van **to_str**. |

| Argument | Beschrijving |
|---|---|
| **regex** | De reguliere expressie die bepaalt wanneer tekst moet worden vervangen. Overeenkomsten tussen dit argument en het argument **text** worden vervangen. |
| **to_str** | De nieuwe tekst die u wilt gebruiken om bestaande inhoud van **text** te vervangen. |
| **occurrence** | Het nummer van de overeenkomst (tussen invoertekst en reguliere expressie) om te vervangen door nieuwe tekst.<br><br>Dit is een optioneel argument. De standaardwaarde is 0 als deze wordt weggelaten. Als een waarde van 0 wordt gebruikt, of als het argument wordt weggelaten, worden alle overeenkomsten tussen **text** en **regex** vervangen door **to_str**.<br><br>U kunt een negatieve waarde opgeven voor **occurrence** om overeenkomsten van rechts naar links te identificeren. |

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `ReplaceRegEx ('abc123','[0-9]','x')` | Retourneert `abcxxx`. In dit voorbeeld wordt alle tekst uit de invoertekenreeks die overeenkomt met het regexpatroon vervangen door de nieuwe tekst. |
| `ReplaceRegEx ('abc123','[0-9]','x',1)` | Retourneert `abcx23`. Alleen de eerste overeenkomsten tussen de invoertekenreeks en het regexpatroon worden vervangen. |
| `ReplaceRegEx ('abc123','[0-9]','x',4)` | Retourneert `abc123`. Er zijn slechts drie mogelijke overeenkomsten tussen de invoertekenreeks en het regexpatroon, dus er wordt geen wijziging uitgevoerd. |
| `ReplaceRegEx ('ABC123','[a-c]','5')` | Retourneert `ABC123`. Er wordt geen wijziging uitgevoerd omdat **ReplaceRegEx()** hoofdlettergevoelig is, en de hoofdletterongevoelige variant niet wordt gebruikt. |
| `ReplaceRegExI ('ABC123','[a-c]','5')` | Retourneert `555123`. De hoofdletterongevoelige variant van de functie, **ReplaceRegExI()**, wordt gebruikt. |

## Wanneer gebruiken

U kunt **ReplaceRegEx()** gebruiken voor het wijzigen van tekst om te voldoen aan opmaak- en nalevingsnormen, vooral wanneer er complexe patronen in de gegevens voorkomen. U kunt bijvoorbeeld telefoonnummertekenreeksen opnieuw formatteren zodat alleen numerieke waarden worden opgenomen.

Indien nodig kunt u deze functie ook gebruiken om gevoelige informatie te maskeren, zoals persoonlijk identificeerbare informatie (PII), zodat deze niet wordt getoond aan gebruikers die uw app analyseren.

## Voorbeeld 1 - load-script om e-mailadressen van werknemers bij te werken

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een tabel `Employees` met de e-mailadressen van werknemers van een organisatie.
- Aanmaken van een nieuw veld, `Email_Updated`, om een zoek- en vervangbewerking uit te voeren op waarden in het veld `Email_Original`.

Nadat het bedrijf van naam is veranderd, zijn alle e-mailadressen bijgewerkt om de nieuwe naam weer te geven.

Onze vereisten:

- Converteer alle domeinverwijzingen van `test.com` naar `newcompanyname.com`.
- We moeten hoofdletterongevoelige regexbewerkingen gebruiken.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `ID`
- `Email_Original`
- `Email_Updated`

Resultatentabel

| 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 |

## Voorbeeld 2 - load-script om ISBN-codes op te maken

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Variabelen `ISBN_13` en `ISBN_10`, waarin de reguliere expressies worden opgeslagen die moeten worden gebruikt voor respectievelijk 13-cijferige en 10-cijferige ISBN-tekenreekspatronen.
- Een tabel `Employees` met ruwe numerieke waarden voor ISBN-boekcodes. De tabel bevat zowel 13-cijferige als 10-cijferige ISBN's.
- Verschillende bewerkingen om elk deel van het ISBN te extraheren en vervolgens een tekenreeks met de volledige code in het standaard, verwachte formaat te retourneren. Wij gebruiken een combinatie van **ReplaceRegEx()**, **MatchRegEx()**, **ExtractRegExGroup()** en **Pick()**.

Onze vereisten:

- We willen de getallen opmaken in beter leesbare tekst, met behulp van de gestandaardiseerde opmaak:
  - ISBN ###-#-####-####-#
  - ISBN #-###-#####-#

### Load-script

```
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
2444444444
8880999900000
`;
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `ISBN_Formatted`

Resultatentabel

| ISBN_Formatted |
| --- |
| ISBN 000-1-0123-0123-2 |
| ISBN 0-111-23456-7 |
| ISBN 1-901-23456-8 |

| ISBN_Formatted |
| --- |
| 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 |

## Voorbeeld 3 - diagramuitdrukking om secundaire e-mailadressen bij te werken

Diagramuitdrukking en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een tabel `Employees` met de e-mailadressen van werknemers van een organisatie. Deze worden opgeslagen in een enkele tekenreeks voor elke werknemer, waarin meerdere e-mailadressen worden opgeslagen.

Onze vereisten:

- Secundaire e-mailadressen moeten worden bijgewerkt om het domein `newcompanyname.com` te gebruiken. We willen geen primaire e-mailadressen bijwerken.
- We moeten hoofdletterongevoelige regexbewerkingen gebruiken.

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `ID`
- `Emails`

Voeg de volgende berekende dimensie toe:

```
=ReplaceRegExI(Emails,'@[a-zA-Z0-9.-]*\.?[a-zA-Z0-9]*\.com','@newcompanyname.com',2)
```

Resultatentabel

| ID | Emails | =ReplaceRegExI(Emails,'@[a-zA-Z0-9.-]*\.?example\.com','@newcompanyname.com',2) |
|---|---|---|
| 1 | Primary: john_doe@test.com Secondary: john_michael_doe@example.com | Primary: john_doe@test.com Secondary: john_michael_doe@newcompanyname.com |
| 2 | Primary: jane_doe@test.com 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 Secondary: charlie_peter_kent@example.com | Primary: charlie_kent@test.com Secondary: charlie_peter_kent@newcompanyname.com.com |
| 5 | Primary: frances_johnson@test.com Secondary: frances_johnson@example.com | Primary: frances_johnson@test.com Secondary: frances_johnson@newcompanyname.com |
| 6 | Primary: rebecca_smith@test.com Secondary: rebecca_danielle_smith@example.com | Primary: rebecca_smith@test.com Secondary: rebecca_danielle_smith@newcompanyname.com |

Door een waarde van 2 op te geven voor het argument **occurrence**, kunnen we alleen de tweede instantie bijwerken die overeenkomt met het regexpatroon.

## ReplaceRegExGroup

**ReplaceRegExGroup()** retourneert een tekenreeks na het vervangen van één of meer overeenkomsten tussen een invoertekenreeks en het samengestelde reguliere

expressiepatroon dat is opgegeven. De tekst die de overeenkomende tekst vervangt, wordt opgegeven in het argument **to_str**. De functie wordt van links naar rechts uitgevoerd, maar als u een negatieve **occurrence** waarde opgeeft, leest de functie van rechts naar links.

Deze functie voert regexbewerkingen uit die hoofdlettergevoelig zijn. U kunt eventueel ook de variant **ReplaceRegExGroupI()** gebruiken om hoofdletteronafhankelijke regexbewerkingen uit te voeren.

**Syntaxis:**

```
ReplaceRegExGroup (text, regex, to_str, group [, occurrence])
```

**Retourgegevenstypen:** tekenreeks

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De tekst van de invoertekenreeks die u geheel of gedeeltelijk wilt vervangen door de nieuwe tekst van **to_str**. |
| **regex** | De reguliere expressie die bepaalt wanneer tekst moet worden vervangen. Overeenkomsten tussen dit argument en het argument **text** worden vervangen. |
| **to_str** | De nieuwe tekst die u wilt gebruiken om bestaande inhoud van **text** te vervangen. |
| **group** | Het nummer van de groep, in het geval van een samengestelde reguliere expressie.<br><br>Als de reguliere expressie slechts één groep bevat, gebruik dan in plaats daarvan de functie **ReplaceRegEx()**. Als alternatief kunt u **ReplaceRegExGroup()** gebruiken met een **group** waarde van `0`.<br><br>U kunt een negatieve **group** waarde opgeven om van rechts naar links naar groepen te zoeken. |
| **occurrence** | Het nummer van de overeenkomst (tussen invoertekst en reguliere expressie) om te vervangen door nieuwe tekst.<br><br>Dit is een optioneel argument. De standaardwaarde is 0 als deze wordt weggelaten. Als een waarde van 0 wordt gebruikt, of als het argument wordt weggelaten, worden alle overeenkomsten tussen **text** en **regex** vervangen door **to_str**.<br><br>U kunt een negatieve waarde opgeven voor **occurrence** om overeenkomsten van rechts naar links te identificeren. |

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `ReplaceRegExGroup ('abc123','([a-z])([0-9]+)','x',0)` | Retourneert `abx` (volledige overeenkomst vervangen). |

| Voorbeeld | Resultaat |
|---|---|
| `ReplaceRegExGroup ('abc123','([a-z])([0-9]+)','x',1)` | Retourneert `abx123` (vervang alleen de eerste groep in de overeenkomst). |
| `ReplaceRegExGroup ('abc123','([a-z])([0-9]+)','x',2)` | Retourneert `abcx` (vervang alleen de tweede groep in de overeenkomst). |
| `ReplaceRegExGroup ('abc123abc123','([a-z])([0-9]+)','x',1,2)` | Retourneert `abc123abx123`. Er zijn twee overeenkomsten tussen de eerste groep van de invoertekst en het regexpatroon. Alleen het tweede exemplaar wordt echter vervangen. |
| `ReplaceRegExGroup ('ABC123ABC123','([a-z])([0-9]+)','x',1)` | Retourneert `ABC123ABC123`. Met andere woorden, er wordt geen tekst vervangen. Dit komt doordat de functie hoofdlettergevoelig is en de hoofdletterongevoelige variant van de functie niet wordt gebruikt. |
| `ReplaceRegExGroupI ('ABC123ABC123','([a-z])([0-9]+)','x',1)` | Retourneert `ABx123ABx123`. In dit voorbeeld wordt tekst vervangen omdat de hoofdletterongevoelige variant van de functie, **ReplaceRegExGroupI()**, wordt gebruikt. |

## Wanneer gebruiken

U kunt **ReplaceRegExGroup()** gebruiken voor het wijzigen van tekst om te voldoen aan opmaak- en nalevingsnormen, vooral wanneer u complexe patronen in de gegevens wilt identificeren. Met het extra argument **group** voor deze functie kunt u alleen bepaalde delen van een groter tekstpatroon bijwerken. U kunt bijvoorbeeld specifieke delen van een URL bijwerken.

Indien nodig kunt u deze functie ook gebruiken om gevoelige informatie te maskeren, zoals persoonlijk identificeerbare informatie (PII), zodat deze niet wordt getoond aan gebruikers die uw app analyseren.

## Voorbeeld 1 - load-script om domein in URL's bij te werken

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een tabel `URLs` die in eerste instantie een lijst met invoertekenreeksen bevat waarin we specifieke URL's willen identificeren en bijwerken. We willen het domein in deze URL's bijwerken.
- Het maken van een veld, `ResourceURL_Updated`, dat gewijzigde invoertekst met de verwerkte URL's bevat.
- Een variabele `URL_RegEx` om de reguliere expressie op te slaan die wordt gebruikt om te verwerken URL's te identificeren.

Onze vereisten:

- Elke invoertekenreeks bevat twee URL's, waarbij de URL van de bron in de eerste URL staat. We willen geen andere URL's dan de eerste bijwerken.
- De bij te werken URL moet het formaat <https of https>://<domein>.com/<bronpad> hebben. Het bronpad is optioneel, maar wordt in dit voorbeeld niet gebruikt.

### Load-script

```
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');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `ResourceURL_Updated`

Resultatentabel

| ResourceURL_Updated |
| --- |
| The resource is located at https://replacement-server/datamodel. For more help, see: https://support.company.com. |
| The resource is located at https://replacement-server/files. For more help, see: https://support.company.com. |

| ResourceURL_Updated |
| --- |
| The resource is located at https://replacement-server/files/book. For more help, see: https://support.company.com. |
| The resource is located at https://replacement-server/files/form. For more help, see: https://support.company.com. |
| The resource is located at https://replacement-server/files/worksheet. For more help, see: https://support.company.com. |
| The resource is located at https://replacement-server/page. For more help, see: https://support.company.com. |
| 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. |

## Voorbeeld 2 - diagramuitdrukking om protocol in URL's bij te werken

Diagramuitdrukking en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een tabel `URLsList` die een lijst met invoertekenreeksen bevat waarin we specifieke URL's willen identificeren en bijwerken. Wij willen het protocol in de koppelingen naar de Support-website bijwerken van http naar https.

Onze vereisten:

- Elke invoer bevat twee URL's, waarbij de link naar de Support-website als tweede URL wordt weergegeven. We willen geen andere URL's dan de tweede URL bijwerken.
- De bij te werken URL moet het formaat <https of https>://<domein>.com/<bronpad> hebben. Het bronpad is optioneel, maar wordt in dit voorbeeld niet gebruikt.

### Load-script

```
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');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- URLsList

Voeg deze berekende dimensies toe:

```
=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)
```

Resultatentabel

| 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 https://testserver.com/files. For more help, see: http://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/files/worksheet. For more help, see: http://support.company.com. | 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/book. For more help, see: http://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: http://support.company.com. | The resource is located at https://testserver.com/files/form. For more help, see: https://support.company.com. |

| 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 https://testserver.com/datamodel. For more help, see: http://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/resourcenew. For more help, see: http://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. For more help, see: http://support.company.com. | The resource is located at https://testserver.com/page/overview. For more help, see: https://support.company.com. |
| The resource is located at https://testserver.com/page/overview. For more help, see: http://support.company.com. | The resource is located at https://testserver.com/resourcenew. For more help, see: https://support.company.com. |

## Right

**Right()** retourneert een tekenreeks die bestaat uit de laatste tekens (meest rechts) van de invoertekenreeks, waarbij het aantal tekens wordt bepaald door het tweede argument.

**Syntaxis:**

```
Right(text, count)
```

**Retourgegevenstypen:** tekenreeks

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De oorspronkelijke tekenreeks. |
| **count** | Definieert het aantal tekens dat moet worden opgenomen vanaf de rechterkant van de tekenreeks **text**. |

Voorbeeld: diagramuitdrukking

| Voorbeeld | Resultaat |
|---|---|
| `Right( 'abcdef', 3 )` | Retourneert `def` |

## Voorbeeld - Basisprincipes voor Right

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `InputText`
  - `NoOfChars`

### Load-script

```
Example:
Load * inline [
InputText, NoOfChars
'abcdef', 3
'2021-07-14', 5
'2021-07-14', 2
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `InputText`
- `NoOfChars`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=Right(InputText,NoOfChars)` om de tekens in `InputText` te berekenen die van de meest rechtse kant moeten worden geëxtraheerd.

Resultatentabel

| InputText | NoOfChars | Right(InputText,NoOfChars) |
|---|---|---|
| 2021-07-14 | 5 | 07-14 |
| abcdef | 3 | def |
| 2021-07-14 | 2 | 14 |

De eerste rij retourneert `07-14`, omdat dit de eerste 5 tekens van rechts zijn van de tekenreeks `InputText`.

De tweede rij retourneert `def`, omdat dit de eerste 3 tekens van rechts zijn van de tekenreeks `InputText`.

De derde rij retourneert `14`, omdat dit de eerste 2 tekens van rechts zijn van de tekenreeks `InputText`.

Load-script

De volgende code laat zien hoe u de functie in een load-script kunt gebruiken.

```
Example:
Load
*,
right(Text,NoOfChars) as Right;
Load * inline [
Text, NoOfChars
'abcdef', 3
'2021-07-14', 5
'2021-07-14', 2
];
```

Resultatentabel

| Tekst | NoOfChars | Rechts |
|---|---|---|
| abcdef | 3 | def |
| 2021-07-14 | 5 | 7-14 |
| 2021-07-14 | 2 | 14 |

## Voorbeeld 5 – Scenario met Right

Diagramuitdrukking

### Overzicht

Dit voorbeeld parseert de bestelcodes in een gegevensverzameling, die een regiocode en bestelnummer bevatten, om een nieuw veld voor gegevensanalyse te maken.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `OrderID`
  - `CustomerName`
  - `SalesAmount`

### Load-script

```
Example:
Load * inline [
OrderID, CustomerName, SalesAmount
US1000001, John Doe, 1500
EU3000002, Jane Smith, 2300
AS2000003, Mike Davis, 1800
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `OrderID`
- `CustomerName`
- `SalesAmount`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=Right(OrderID, 7)` om het bestelnummer uit te splitsen.

Resultatentabel

| OrderID | CustomerName | SalesAmount | Right (OrderID, 7) |
|---|---|---|---|
| AS2000003 | Mike Davis | 1800 | 2000003 |
| EU3000002 | Jane Smith | 2300 | 3000002 |
| US1000001 | John Doe | 1500 | 1000001 |

Vergelijk de uitvoer van de functie `Right` met de invoer `OrderID`. De berekende dimensie leidt een numerieke waarde af die nu kan worden gebruikt voor taken zoals het sorteren van de uitvoer in numerieke volgorde in plaats van alfabetische volgorde.

## RTrim

**RTrim()** retourneert de invoertekenreeks, ontdaan van eindspaties.

**Syntaxis:**

```
RTrim(text)
```

**Retourgegevenstypen:** tekenreeks

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De tekenreeks die moet worden geëvalueerd. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `RTrim( ' abc' )` | Retourneert `abc` |
| `RTrim( 'abc ' )` | Retourneert `abc` |

## Voorbeeld - Basisprincipes voor RTrim

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `InputText`.

### Load-script

```
Set verbatim=1;
Example:
Load * inline [
InputText
'   abc    '
' def  '];
```

> *De opdracht `Set verbatim=1` is opgenomen in het voorbeeld om ervoor te zorgen dat de ruimten niet automatisch worden afgekapt voordat de `RTrim` functie wordt getoond. Raadpleeg Verbatim (page 225) voor meer informatie.*

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `InputText`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=RTrim(InputText)` om volgspaties te verwijderen uit `InputText`.

Resultatentabel

| InputText | RTrim(InputText) |
|---|---|
| '  abc  ' | '  abc' |
| ' def ' | ' def' |

De uitvoer van de functie `RTrim` verwijdert alle volgspaties rechts van de oorspronkelijke tekst, maar behoudt alle voorloopspaties.

## Voorbeeld - Scenario met RTrim

Diagramuitdrukking

### Overzicht

Een CRM-systeem (Customer Relationship Management) bevat records met inconsistente gegevensinvoer die extra volgspaties bevatten. Voor rapportagedoeleinden moeten de gegevens worden opgeschoond om deze spaties te verwijderen en om ervoor te zorgen dat de klantnamen goed worden gesorteerd en gegroepeerd.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `CustomerName`.

### Load-script

```
Set verbatim=1;
Example:
Load * inline [
CustomerName
'John Doe  '
'Jane Smith'
'Michael Johnson  '
'Emily Davis'
];
```

*De opdracht "Set verbatim=1" is opgenomen in het voorbeeld om ervoor te zorgen dat de ruimten niet automatisch worden afgekapt voordat de trim-functie wordt getoond. Raadpleeg Verbatim (page 225) voor meer informatie.*

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `CustomerName`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=RTrim(CustomerName)` om alle volgspaties te verwijderen uit `CustomerName`.

Resultatentabel

| CustomerName | RTrim(CustomerName) |
|---|---|
| 'Emily Davis' | 'Emily Davis' |

| CustomerName | RTrim(CustomerName) |
|---|---|
| 'Jane Smith' | 'Jane Smith' |
| 'John Doe ' | 'John Doe' |
| 'Michael Johnson ' | 'Michael Johnson' |

De uitvoer laat zien dat de functie `RTrim` alle volgspaties heeft verwijderd uit de oorspronkelijke tekenreekswaarden in `CustomerName`.

## Voorbeeld - Geavanceerd scenario met RTrim

Diagramuitdrukking

### Overzicht

Dit voorbeeld verwijdert alle volgspaties uit de oorspronkelijke teksttekensreeks. De diagramuitdrukking bevat metingen die de `Len` functie gebruiken om de tekens in de tekenreeks voor en na het gebruik van de `RTrim` functie te tellen.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `String`.

### Load-script

```
Set verbatim=1;
Example:
Load * inline [
String
'   abc    '
' def  '];
```

*De opdracht "Set verbatim=1" is opgenomen in het voorbeeld om ervoor te zorgen dat de ruimten niet automatisch worden afgekapt voordat de `RTrim` functie wordt getoond. Raadpleeg Verbatim (page 225) voor meer informatie.*

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `String`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=RTrim(String)` om alle extra volgspaties te verwijderen.

Maak de volgende metingen:

- `=Len(String)` om de lengte van de oorspronkelijke tekenreeks te tellen.
- `=Len(RTrim(String))` om de lengte van de tekenreeks te tellen nadat de volgspaties zijn verwijderd.

Resultatentabel

| Tekenreeks | RTrim(String) | Len(String) | Len(RTrim(String)) |
|---|---|---|---|
| ' abc ' | ' abc' | 10 | 6 |
| 'def ' | 'def' | 6 | 4 |

Wanneer u de uitvoer van de functie `RTrim` vergelijkt met de oorspronkelijke tekensreekswaarden in het script, kunt u zien hoe alle volgspaties zijn verwijderd.

Load-script

De volgende code laat zien hoe u de functie in een load-script kunt gebruiken.

```
Set verbatim=1;
Example:
Load *, len(RtrimString) as RtrimStringLength;
Load *, rtrim(String) as RtrimString;
Load *, len(String) as StringLength;
Load * Inline [
String
'   abc    '
' def  '];
```

*De opdracht "`Set verbatim=1`" is opgenomen in het voorbeeld om ervoor te zorgen dat de ruimten niet automatisch worden afgekapt voordat de `rtrim` functie wordt getoond. Raadpleeg Verbatim (page 225) voor meer informatie.*

Resultatentabel

| Tekenreeks | StringLength | RtrimString | RtrimStringLength |
|---|---|---|---|
| ' abc ' | 10 | ' abc' | 6 |
| 'def ' | 6 | 'def' | 4 |

**Zie ook:**

p LTrim (page 1820)

## SubField

**Subfield()** wordt gebruikt voor het extraheren van onderdelen van subtekenreeksen van een bovenliggend tekenreeksveld, waarbij de oorspronkelijke recordvelden bestaat uit twee of meer onderdelen met een scheidingsteken ertussen.

De functie **Subfield()** kan bijvoorbeeld worden gebruikt om voornaam en achternaam te extraheren vanuit een lijst met records die bestaat uit volledige namen, de samenstellende delen van een padnaam of voor het extraheren van gegevens uit tabellen met door komma's gescheiden waarden.

Als u de functie **Subfield()** gebruikt in een opdracht **LOAD** terwijl de optionele parameter field_no wordt weggelaten, wordt één complete record gegenereerd voor elke subtekenreeks. Als meerdere velden worden geladen met **Subfield()**, worden de cartesische producten van alle combinaties gemaakt.

**Syntaxis:**

```
SubField(text, delimiter[, field_no ])
```

**Retourgegevenstypen:** tekenreeks

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De oorspronkelijke tekenreeks. Dit kan een hard gecodeerde tekst, een variabele, een uitbreiding met dollarteken of een andere uitdrukking zijn. |
| **delimiter** | Een teken binnen de invoerreeks **text** die de tekenreeks opsplitst in samenstellende delen. |
| **field_no** | Het optionele derde argument is een geheel getal waarmee wordt opgegeven welke van de subtekenreeksen van de bovenliggende tekenreeks **text** moeten worden geretourneerd. Gebruik de waarde 1 om terug te keren naar de eerste subtekenreeks, 2 om terug te keren naar de tweede subtekenreeks, enzovoorts.<br>• Als **field_no** een positieve waarde is, worden subtekenreeksen van links naar rechts geëxtraheerd.<br>• Als **field_no** een negatieve waarde is, worden subtekenreeksen van rechts naar links geëxtraheerd. |

*SubField() kan worden gebruikt in plaats van complexe combinaties van functies zoals Len(), Right(), Left(), Mid() en andere tekenreeksfuncties.*

Voorbeelden: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `SubField( 'abc;cde;efg', ';' ,2 )` | Retourneert `cde` |
| `SubField( Null, ';' ,1 )` | Retourneert een lege tekenreeks |
| `SubField( ';', ';' ,1 )` | Retourneert een lege tekenreeks |
| `SubField( '\Users\ext_jrb\Documents\Qlik\Sense\Apps;','\',-3 )` | Retourneert `Qlik` |

## Voorbeeld - Basisprincipes voor SubField

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `Name`.

### Load-script

```
Example:
Load * inline [
Name
Dave Owen
Joe Tem
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `Name`

Maak de volgende berekende dimensies:

- `=SubField(Name, ' ',1)`, om de eerste subtekenreeks te extraheren die vóór de spatie van het `' '` scheidingsteken verschijnt.
- `=SubField(Name, ' ',-1)` om de eerste subtekenreeks te extraheren die vóór de spatie van het `' '` scheidingsteken verschijnt, beginnend vanaf de rechterkant van de tekenreeks.

Results table

| Name | SubField(Name,' ',1) | SubField(Name,' ',-1) |
|---|---|---|
| Dave Owen | Dave | Owen |
| Joe Tem | Joe | Tem |

De functie `SubField()` haalt de eerste subtekenreeks op uit `Name` door het argument `field_no` in te stellen op 1. Aangezien de waarde van `field_no` positief is, wordt de volgorde links naar rechts gevolgd voor het ophalen van de subtekenreeks. Een tweede functieaanroep haalt de tweede subtekenreeks op door het argument `field_no` in te stellen op -1, waardoor de subtekenreeks wordt opgehaald volgens de volgorde rechts naar links.

Load-script

### Load-script

Open de Editor voor laden van gegevens en voeg het volgende load-script toe aan een nieuwe sectie.

```
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;
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `Name`
- `FirstName`
- `SurName`

Results table

| Name | FirstName | SurName |
|---|---|---|
| Dave Owen | Dave | Owen |
| Joe Tem | Joe | Tem |

### Uitleg

De functie `SubField` haalt de eerste subtekenreeks op, `Name`, door het argument `field_no` in te stellen op 1. Aangezien de waarde van `field_no` positief is, wordt de volgorde links naar rechts gevolgd voor het ophalen van de subtekenreeks. Een tweede functieaanroep haalt de tweede subtekenreeks op door het argument `field_no` in te stellen op -1, waardoor de subtekenreeks wordt opgehaald volgens de volgorde rechts naar links.

## Voorbeeld 5 – Scenario met SubField

Diagramuitdrukking

### Overzicht

Een gegevensverzameling van producten bevat een producttabel. Elk product in de tabel heeft een tagveld dat het productgebied identificeert. Producten kunnen meerdere waarden voor het tagveld hebben. Widget A heeft bijvoorbeeld de volgende tags: Electronics, Gadgets, Home. Tagwaarden worden gescheiden door het sluisteken (|) als scheidingsteken. Dit voorbeeld laat zien hoe u specifieke tagwaarden kunt extraheren met de functie `SubField`.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `ProductID`
  - `ProductName`
  - `Tags`
  - `Sales`

### Load-script

```
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
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `ProductID`
- `ProductName`
- `Tags`

Maak de volgende berekende dimensies:

- `=SubField(Tags, '|',1)`, om het eerste item uit het veld `Tags` te extraheren.
- `=SubField(Tags, '|',2)` om het tweede item uit het veld `Tags` te extraheren.
- `=SubField(Tags, '|',3)`, om het derde item uit het veld `Tags` te extraheren.

Results table

| ProductID | ProductName | Tags | SubField (Tags, '\|', 1) | SubField (Tags, '\|', 2) | SubField (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 | - |

De uitvoer van de `SubField` functies heeft met succes de tags van de relevante posities in de oorspronkelijke `Tags` tekenreeks geretourneerd.

Load-script

### Load-script

Open de Editor voor laden van gegevens en voeg het volgende load-script toe aan een nieuwe sectie.

```
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 '|');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `Instrument`
- `Player`
- `Project`

Results table

| Instrument | Player | Project |
|---|---|---|
| Guitar | Mike | Music |
| Guitar | Mike | Video |

| Instrument | Player | Project |
|---|---|---|
| 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 |

**Uitleg**

Dit voorbeeld laat zien dat als u meerdere exemplaren van de functie `Subfield()` gebruikt, elk met de parameter `field_no` weggelaten, binnen in dezelfde `LOAD`-opdracht, Cartesiaanse producten worden gemaakt van alle combinaties. De optie `DISTINCT` wordt gebruikt om te voorkomen dat dubbele records worden gemaakt.

## SubFieldRegEx

**SubFieldRegEx()** extraheert tekst uit een tekenreeksuitdrukking en gebruikt het opgegeven reguliere expressiepatroon als scheidingsteken. De functie retourneert een nulwaarde als er geen overeenkomsten worden gevonden.

Deze functie voert regexbewerkingen uit die hoofdlettergevoelig zijn. U kunt eventueel ook de variant **SubFieldRegExI()** gebruiken om hoofdletteronafhankelijke regexbewerkingen uit te voeren.

**Syntaxis:**

```
SubFieldRegEx (text, regex_delimiter [, field_no])
```

**Retourgegevenstypen:** tekenreeks

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De tekst van de invoertekenreeks waarin u wilt zoeken naar een regulier expressiepatroon. |
| **regex_ delimiter** | Tekrensreeksexpressie die de reguliere expressie bevat die als scheidingsteken wordt gebruikt. U kunt meer dan één scheidingsteken opgeven in hetzelfde reguliere expressiepatroon door ze te scheiden met een \| (sluisteken) teken. |
| **field_no** | Geheel getal dat het nummer aangeeft van de onderverdeling die u wilt extraheren. Geef bijvoorbeeld de waarde `4` op om de vierde onderverdeling te extraheren. Dit is een optioneel argument.<br><br>Het volgende is van toepassing ongeacht of de functie wordt gebruikt in een load-script of diagramuitdrukking:<br><br>• Als **field_no** positief is, zal de functie één waarde retourneren, die de overeenkomsten van links naar rechts identificeert.<br>• Als **field_no** negatief is, zal de functie één waarde retourneren, die de overeenkomsten van rechts naar links identificeert.<br><br>Wanneer u de functie in een load-script gebruikt:<br><br>• Als u de functie **SubFieldRegEx()** in een **LOAD**-instructie gebruikt en **field_no** is weggelaten, genereert de functie evenveel records als er overeenkomsten zijn.<br>• Als meerdere velden worden geladen met **SubFieldRegEx()** en bij geen darvan een **field_no** argument is opgegeven, worden de cartesische producten van alle combinaties gemaakt.<br><br>Wanneer u de functie in een diagramuitdrukking gebruikt:<br><br>• Als **field_no** is weggelaten, is de standaardwaarde `1`. |

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `SubFieldRegEx('a,b c;1 2,3',' \|,\|;', 1)` | Retourneert `a`. In dit voorbeeld worden drie scheidingstekens opgegeven (spatie, komma en puntkomma). |
| `SubFieldRegEx ('a;b;c;1;2;3',';', 2)` | Retourneert `b`. In dit geval wordt slechts één scheidingsteken opgegeven (puntkomma). |
| `SubFieldRegEx('a,b c;1 2,3',' \|,\|;', 4)` | Retourneert `1`. |

## Wanneer gebruiken

U kunt **SubFieldRegEx()** gebruiken om meerdere afzonderlijke waarden uit een lijst te halen. Als u bijvoorbeeld een door komma's gescheiden lijst in een groter blok natuurlijke taal hebt, kunt u deze functie gebruiken om elke individuele waarde uit de lijst in een nieuwe record in uw gegevensmodel te isoleren.

## Voorbeeld 1 - load-script om labels per transactie te scheiden

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een tabel `Transaction_By_Tag`, die in eerste instantie transactiegegevens bevat. Deze gegevens worden verwerkt om individuele labelwaarden in nieuwe records te extraheren.
- Het aanmaken van een veld, `TransactionTag`, om de geëxtraheerde labelwaarden op te slaan.

In het veld met de transactielabels staan meerdere labels in teksttekenreeksen, gescheiden door scheidingstekens. Het scheidingsteken kan een van de volgende tekens zijn: , | ;

### Load-script

```
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;
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `ID`
- `Product`
- `TransactionTag`

Resultatentabel

| 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 |

Dit voorbeeld laat zien dat bij gebruik in een load-script, **SubFieldRegEx()** een aparte rij retourneert voor elke gevonden overeenkomst in de invoertekenreeks.

## Voorbeeld 2 - diagramuitdrukkingen om labels per transactie te scheiden

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een tabel `Transaction_By_Tag`, die transactiegegevens bevat.
- Een veld, `Tags`, dat teksttekenreeksen bevat om alle labels op te slaan die aan een transactie verbonden zijn.

Onze vereisten:

- In het veld met de transactielabels staan meerdere labels in teksttekenreeksen, gescheiden door scheidingstekens. Het scheidingsteken kan een van de volgende tekens zijn: , | ;
- We willen diagramuitdrukkingen gebruiken om elk label in zijn eigen veld te scheiden, waarbij maximaal drie labels aan een transactie kunnen worden toegevoegd.

### Load-script

```
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');
```

**Resultaten**

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `ID`
- `Tags`

Voeg deze berekende dimensies toe:

- `=SubFieldRegEx(Tags, ',|;|\|', 1)`
- `=SubFieldRegEx(Tags, ',|;|\|', 2)`
- `=SubFieldRegEx(Tags, ',|;|\|', 3)`

Resultatentabel

| 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()** retourneert het aantal keren dat de opgegeven subtekenreeks voorkomt in de tekst van de invoertekenreeks. Als er geen overeenkomst wordt gevonden, wordt 0 geretourneerd.

**Syntaxis:**

```
SubStringCount(text, sub_string)
```

**Retourgegevenstypen:** geheel getal

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De oorspronkelijke tekenreeks. |
| **sub_string** | Een tekenreeks die een of meer keren in de invoerreeks **text** mag voorkomen. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `SubStringCount( 'abcdefgcdxyz', 'cd' )` | Retourneert `2` |
| `SubStringCount( 'abcdefgcdxyz', 'dc' )` | Retourneert `0` |

## Voorbeeld - Basisprincipes voor SubStringCount

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `FruitSentence`.

### Load-script

```
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!"
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `FruitSentence`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=SubStringCount(FruitSentence, 'apple')` om uit te rekenen hoe vaak het woord `apple` voorkomt in `FruitSentence`.

Results table

| FruitSentence | SubStringCount(FruitSentence, 'apple') |
|---|---|
| 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 |

De tabel wordt gesorteerd op de dimensie `FruitSentence` in oplopende volgorde. De functie `SubStringCount` retourneert het aantal keren dat het woord `apple` in elke zin voorkomt. De eerste zin retourneert bijvoorbeeld `2`, omdat het woord twee keer voorkomt in de zin. In de tweede en vierde zin negeert de functie het woord `Apple` omdat het niet overeenkomt met de hoofdletter van de zoekterm `apple`.

*Om hoofdletterongevoelig te zoeken, kunt u uw invoerreeks eerst omzetten naar hoofdletters of kleine letters en vervolgens de functie `SubStringCount` gebruiken voor het resultaat. Om bijvoorbeeld naar alle exemplaren van het woord apple te zoeken, ongeacht de hoofdletters, gebruikt u de volgende metinguitdrukking: `=SubStringCount(Upper(FruitSentence),'APPLE')`. De uitdrukking gebruikt de functie `SubStringCount` met de functie `Upper` om de hele tekenreeks naar hoofdletters te converteren, en zoekt vervolgens naar het woord `APPLE`.*

Load-script

De volgende code laat zien hoe u de functie in een load-script kunt gebruiken. Dit voorbeeld bevat een voorafgaande lading die de functie `SubStringCount` met de functie `Upper` gebruikt om de hele tekenreeks naar hoofdletters te converteren, en vervolgens naar het woord `APPLE` zoekt.

```
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!"
];
```

Resultatentabel

| FruitSentence | SubStringCount_APPLE |
|---|---|
| I love eating an apple every day. Apple juice is great too! | 2 |
| Apple pie is my favorite dessert. I also enjoy apple jam. | 2 |

| FruitSentence | SubStringCount_APPLE |
|---|---|
| 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 |

## Voorbeeld 5 – Scenario met SubStringCount

Diagramuitdrukking

### Overzicht

Een gegevensverzameling van fruitproducten bevat een veld met productbeschrijvingen. Dit voorbeeld berekent het aantal keren dat de volgende fruitnamen in elke beschrijving voorkomen: apple, banana of orange.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `ProductDescription`.

### Load-script

```
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."
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `ProductDescription`

Maak de volgende berekende dimensies:

- `=SubStringCount(ProductDescription, 'apple')`, om het aantal exemplaren van het woord `apple` in de tekenreeks `ProductDescription` te berekenen.
- `=SubStringCount(ProductDescription, 'banana')`, om het aantal exemplaren van het woord `banana` in de tekenreeks `ProductDescription` te berekenen.
- `=SubStringCount(ProductDescription, 'orange')`, om het aantal exemplaren van het woord `orange` in de tekenreeks `ProductDescription` te berekenen.

Results table

| ProductDescription | SubStringCount (ProductDescription, 'apple') | SubStringCount (ProductDescription, 'banana') | SubStringCount (ProductDescription, 'orange') |
|---|---|---|---|
| A refreshing orange soda made with real orange juice. | 0 | 0 | 2 |
| Banana chips with apple and cinnamon flavor. | 1 | 0 | 0 |
| Delicious apple pie with a hint of cinnamon and vanilla. | 1 | 0 | 0 |
| Fresh apple and banana smoothie. | 1 | 1 | 0 |
| Organic apple, banana, and orange juice. Great for breakfast. | 1 | 1 | 1 |
| Tropical banana and orange mix for smoothies. | 0 | 1 | 1 |

De uitvoer van de functie `SubStringCount` geeft met succes het aantal keren dat elke subtekenreeks (`apple, banana, orange`) voorkomt in de productomschrijving. In de eerste beschrijving retourneert de functie bijvoorbeeld 2 terug omdat er twee exemplaren van het woord `orange` zijn. Het argument `sub_string` is hoofdlettergevoelig, dus in de tweede beschrijving telt de functie het woord `Banana` niet omdat het niet overeenkomt met de subtekenreeks `banana` in de metinguitdrukking.

*Om hoofdletterongevoelig te zoeken, kunt u uw invoerreeks eerst omzetten naar hoofdletters of kleine letters en vervolgens de functie `SubStringCount` gebruiken voor het resultaat. Om bijvoorbeeld naar alle exemplaren van het woord apple te zoeken, ongeacht de hoofdletters, gebruikt u de volgende metinguitdrukking: `=SubStringCount(Upper (ProductDescription),'APPLE')`. De uitdrukking gebruikt de functie `SubStringCount` met de functie `Upper` om de hele tekenreeks naar hoofdletters te converteren, en zoekt vervolgens naar het woord `APPLE`.*

## Voorbeeld 5 – Scenario met SubStringCount voor het parseren van geneste gegevens

Diagramuitdrukking

### Overzicht

Een gegevensverzameling van IT-records bevat een lijst met id-nummers en categorieën van wijzigingen. In de gegevenstabel wordt het veld `Categories` afgeleid van een hiërarchie op meerdere niveaus die de hoogste bovenliggende categorie vermeldt, gevolgd door alle subcategorieën, bijvoorbeeld Software > Program > Patch. Dit scenario laat zien hoe u de laatste waarde, of bladwaarde, uit deze onregelmatige categoriehiërarchie kunt halen met behulp van een combinatie van de functies `SubStringCount` en `SubField`.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `ChangeID` bevat de oorspronkelijke wijzigings-ID die verwerkt moet worden.
  - `Categories` bevat een lijst met wijzigingscategorieën. De categorieën worden gescheiden door puntkomma's en in hiërarchische volgorde van links naar rechts vermeld, beginnend bij de bovenste bovenliggende categorie tot de meest gedetailleerde subcategorie.

### Load-script

```
Example:
Load * inline [
ChangeID,Categories
CHG00100,Software;Program;Patch
CHG00101,Hardware;Server
CHG00102,Hardware;Network;Router
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `ChangeID`
- `Categories`

Maak de volgende berekende dimensies:

- `=SubStringCount(Categories,';')+1`, om het aantal categorieniveaus in de wijzigingshiërarchie te berekenen.
- `=SubField(Categories, ';' ,SubStringCount(Categories,';')+1)`, om de meest rechtse categorie uit de wijzigingshiërarchie te extraheren.

Results table

| ChangeID | Categories | SubStringCount (Categories,';')+1 | SubField(Categories, ';' ,SubStringCount (Categories,';')+1) |
|---|---|---|---|
| CHG00100 | Software;Program;Patch | 3 | Patch |
| CHG00101 | Hardware;Server | 2 | Server |
| CHG00102 | Hardware;Network;Router | 3 | Router |

In dit voorbeeld is de `SubStringCount` functie in de `SubField` functie genest. De eerste berekende dimensie, `SubStringCount()`, berekent het aantal geneste waarden in het veld `Categories`. Dit wordt bepaald door het aantal puntkomma's `';'` te tellen en 1 bij het resultaat op te tellen. Deze uitvoer wordt dan geïnjecteerd als de derde parameter van de functie `SubField` om de meest rechtse categorie te extraheren die genest is in het veld `Categories`.

Bekijk bijvoorbeeld de resultaten voor de vermelding `ChangeID CHG00100` . In de eerste berekende dimensie is `SubStringCount` 2. De uitdrukking voegt vervolgens 1 toe aan dat resultaat om te bepalen dat het veld `Categories` drie geneste categoriewaarden heeft, `Software`, `Program`, en `Patch`. De functie `SubField` gebruikt dit resultaat vervolgens in de tweede berekende dimensie om de derde categorie, `Patch`, te extraheren.

## TextBetween

**TextBetween()** retourneert de tekst in de invoertekenreeks die tussen de tekens staat die zijn opgegeven als scheidingstekens.

**Syntaxis:**

```
TextBetween(text, delimiter1, delimiter2[, n])
```

**Retourgegevenstypen:** tekenreeks

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De oorspronkelijke tekenreeks. |

| Argument | Beschrijving |
|---|---|
| **delimiter1** | Geeft het eerste scheidingsteken (of de eerste scheidingsreeks) op waarnaar moet worden gezocht in **text**. |
| **delimiter2** | Geeft het tweede scheidingsteken (of de tweede scheidingsreeks) op waarnaar moet worden gezocht in **text**. |
| **n** | Definieert tussen welk exemplaar van het paar scheidingstekens moet worden gezocht. Een waarde van 2 retourneert bijvoorbeeld de tekens tussen het tweede exemplaar van **delimiter1** en het tweede exemplaar van **delimiter2**. |

Voorbeeld: diagramuitdrukkingen

| Voorbeeld | Resultaat |
|---|---|
| `TextBetween( '<abc>', '<', '>' )` | Retourneert `abc` |
| `TextBetween( '<abc><de>', '<', '>',2 )` | Retourneert `de` |
| `TextBetween( 'abc', '<', '>' )` | Retourneert `NULL`<br><br>Als één van de scheidingstekens niet wordt gevonden in de tekenreeks, wordt `NULL` geretourneerd. |
| `TextBetween( '<a<b', '<', '>' )` | Retourneert `NULL`<br><br>Als één van de scheidingstekens niet wordt gevonden in de tekenreeks, wordt `NULL` geretourneerd. |
| `TextBetween( '<>', '<', '>' )` | Retourneert een tekenreeks met lengte nul. |
| `TextBetween( '<abc>', '<', '>', 2 )` | Retourneert `NULL`, want n is groter dan het aantal scheidingstekens. |

## Voorbeeld - Basisprincipes voor TextBetween

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `InputText`.

### Load-script

```
Example:
Load * inline [
```

```
InputText,
<abc><de>
<def><ghi><jk>
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `InputText`

Maak de volgende berekende dimensies:

- `=TextBetween(InputText,'<','>')`, om het eerste exemplaar te berekenen van de tekens die tussen de scheidingstekens `'<'` en `'>'` in het veld `InputText` voorkomen.
- `=TextBetween(InputText,'<','>',2)` om het tweede exemplaar te berekenen van de tekens die tussen de scheidingstekens `'<'` en `'>'` in het veld `InputText` voorkomen.

Results table

| InputText | TextBetween(InputText,'<','>') | TextBetween(InputText,'<','>',2) |
|---|---|---|
| <abc><de> | abc | de |
| <def><ghi><jkl> | def | ghi |

In de eerste berekende uitvoering, retourneert de uitvoer van de `TextBetween` functie het eerste exemplaar van de tekens die tussen de scheidingstekens `'<'` en `'>'` voorkomen. Rij 1 retourneert bijvoorbeeld `abc`. De tweede berekende uitvoering heeft een `n` argument van `2`, zodat de uitvoer het tweede exemplaar retourneert van de tekens die tussen de scheidingstekens `'<'` en `'>'` voorkomen. Rij 1 retourneert bijvoorbeeld `de`.

### Load-script

De volgende code laat zien hoe u de functie in een load-script kunt gebruiken.

```
Load *,
textbetween(Text,'<','>') as TextBetween,
textbetween(Text,'<','>',2) as SecondTextBetween;
Load * inline [
Text
<abc><de>
<def><ghi><jkl>];
```

Resultatentabel

| Tekst | TextBetween | SecondTextBetween |
|---|---|---|
| <abc><de> | abc | de |
| <def><ghi><jkl> | def | ghi |

## Voorbeeld 5 – Scenario met TextBetween

Diagramuitdrukking

### Overzicht

Een gegevensverzameling bevat een gegevenstabel met e-mailadressen. Dit voorbeeld gebruikt de functie `TextBetween` om het domein van het tweede niveau uit het adres te extraheren.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- De volgende velden in de gegevenstabel:
  - `UserID`
  - `EmailAddress`

### Load-script

```
Example:
Load * inline [
UserID, EmailAddress
1, alice@example.com
2, bob@sample.org
3, charlie@domain.net
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `UserID`
- `EmailAddress`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=TextBetween(EmailAddress,'@', '.')` om de tekst tussen de tekens `'@'` en `'.'` binnen het veld `EmailAddress` te retourneren.

Results table

| UserID | EmailAddress | TextBetween(EmailAddress,'@', '.') |
|---|---|---|
| 1 | alice@example.com | example |
| 2 | bob@sample.org | sample |
| 3 | charlie@domain.net | domain |

De uitvoer van de functie `TextBetween` retourneert het domein van het tweede niveau voor elke gebruikers-id door te zoeken naar inhoud tussen de opgegeven scheidingstekens: `'@'` en `'.'`.

# Trim

**Trim()** retourneert de invoertekenreeks, ontdaan van alle begin- en eindspaties.

**Syntaxis:**

```
Trim(text)
```

**Retourgegevenstypen:** tekenreeks

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De tekenreeks die moet worden geëvalueerd. |

Voorbeeld: diagramuitdrukking

| Voorbeeld | Resultaat |
|---|---|
| `Trim( ' abc' )` | Retourneert `abc` |
| `Trim( 'abc ' )` | Retourneert `abc` |
| `Trim( ' abc ' )` | Retourneert `abc` |

## Voorbeeld - Basisprincipes voor Trim

Diagramuitdrukking

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `InputText`.

### Load-script

```
Set verbatim=1;
Example:
Load * inline [
InputText
'   abc    '
' def  '];
```

*De opdracht "Set verbatim=1" is opgenomen in het voorbeeld om ervoor te zorgen dat de ruimten niet automatisch worden afgekapt voordat de trim-functie wordt getoond. Raadpleeg Verbatim (page 225) voor meer informatie.*

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `InputText`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=Trim(InputText)`, om alle extra voorloop- en volgspaties te verwijderen uit `InputText`.

Resultatentabel

| InputText | Trim(InputText) |
|---|---|
| '  abc  ' | 'abc' |
| ' def ' | 'def' |

De uitvoer van de functie `Trim` verwijdert alle voorloop- en volgspaties.

## Voorbeeld - Scenario met Trim

Diagramuitdrukking

#### Overzicht

Een CRM-systeem (Customer Relationship Management) bevat records met inconsistente gegevensinvoer die extra voorloopspaties bevatten. Voor rapportagedoeleinden moeten de gegevens worden opgeschoond om deze spaties te verwijderen en om ervoor te zorgen dat de klantnamen goed worden gesorteerd en gegroepeerd.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `CustomerName`.

#### Load-script

```
Set verbatim=1;
Example:
Load * inline [
CustomerName
'  John Doe '
'Jane Smith '
'   Michael Johnson'
```

```
'Emily Davis'
];
```

*De opdracht "Set verbatim=1" is opgenomen in het voorbeeld om ervoor te zorgen dat de ruimten niet automatisch worden afgekapt voordat de trim-functie wordt getoond. Raadpleeg Verbatim (page 225) voor meer informatie.*

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `CustomerName`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=Trim(CustomerName)`, om alle extra voorloop- en volgspaties te verwijderen uit `CustomerName`.

Resultatentabel

| CustomerName | Trim(KlantNaam) |
|---|---|
| '  Michael Johnson' | 'Michael Johnson' |
| '  John Doe ' | 'John Doe' |
| 'Emily Davis' | 'Emily Davis' |
| 'Jane Smith ' | 'Jane Smith' |

De uitvoer laat zien dat de functie `Trim` alle voorloop- en volgspaties heeft verwijderd uit de oorspronkelijke tekenreekswaarden in `CustomerName`.

## Voorbeeld - Geavanceerd scenario met Trim

Diagramuitdrukking

#### Overzicht

Dit voorbeeld verwijdert alle voorloop- en volgspaties uit de oorspronkelijke teksttekensreeks. De diagramuitdrukking bevat metingen die de `Len` functie gebruiken om de tekens in de tekenreeks voor en na het gebruik van de `Trim` functie te tellen.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.
- Eén veld in de gegevenstabel met de naam `String`.

#### Load-script

```
Set verbatim=1;
Example:
```

```
Load * inline [
String
'   abc    '
' def  '];
```

*De opdracht "Set verbatim=1" is opgenomen in het voorbeeld om ervoor te zorgen dat de ruimten niet automatisch worden afgekapt voordat de trim-functie wordt getoond. Raadpleeg Verbatim (page 225) voor meer informatie.*

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `String`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=Trim(String)`, om alle extra voorloop- en volgspaties te verwijderen

Maak de volgende metingen:

- `=Len(String)`, om de lengte van de oorspronkelijke tekenreeks te tellen
- `=Len(Trim(String))` om de lengte van de tekenreeks te tellen nadat de spaties zijn verwijderd.

Resultatentabel

| Tekenreeks | Trim(String) | Len(String) | Len(Trim(String)) |
|---|---|---|---|
| '  abc  ' | 'abc' | 10 | 3 |
| ' def ' | 'def' | 6 | 3 |

Wanneer u de uitvoer van de functie `Trim` vergelijkt met de oorspronkelijke tekensreekswaarden in het script, kunt u zien hoe alle voorloop- en volgspaties zijn verwijderd.

#### Load-script

De volgende code laat zien hoe u de functie in een load-script kunt gebruiken.

```
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');
```

*De opdracht "Set verbatim=1" is opgenomen in het voorbeeld om ervoor te zorgen dat de ruimten niet automatisch worden afgekapt voordat de trim-functie wordt getoond. Raadpleeg Verbatim (page 225) voor meer informatie.*

Resultatentabel

| Tekenreeks | StringLength | TrimString | TrimStringLength |
|---|---|---|---|
| abc | 10 | abc | 3 |
| def | 6 | def | 3 |

## Upper

**Upper()** converteert alle tekens in de invoertekenreeks naar hoofdletters voor alle teksttekens in de uitdrukking. Cijfers en symbolen worden genegeerd.

**Syntaxis:**

```
Upper(text)
```

**Retourgegevenstypen:** tekenreeks

Argumenten

| Argument | Beschrijving |
|---|---|
| **text** | De tekenreeks die moet worden geëvalueerd. |

Voorbeeld: diagramuitdrukking

| Voorbeeld | Resultaat |
|---|---|
| `Upper( ' abcD' )` | Retourneert `ABCD` |

### Voorbeeld - Basisprincipes voor Upper

Diagramuitdrukking

**Overzicht**

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.

- Eén veld in de gegevenstabel met de naam `InputText`.

**Load-script**

```
Example:
Load * inline [
```

```
InputText
rHode iSland
washingTon d.C.
new york
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- `InputText`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=Upper(InputText)` om alle tekens in het veld `InputText` te converteren naar hoofdletters.

Resultatentabel

| InputText | Upper(InputText) |
|---|---|
| new york | NEW YORK |
| rHode iSland | RHODE ISLAND |
| washingTon d.C. | WASHINGTON D.C. |

De uitvoer van de functie `Upper` converteert alle waarden naar hoofdletters.

### Load-script

De volgende code laat zien hoe u de functie in een load-script kunt gebruiken.

```
Load
String,Upper(String)
Inline
[String
rHode iSland
washingTon d.C.
new york];
```

Resultatentabel

| Tekenreeks | Upper(String) |
|---|---|
| rHode iSland | RHODE ISLAND |
| washingTon d.C. | WASHINGTON D.C. |
| new york | NEW YORK |

## Voorbeeld – optimistisch scenario

Diagramuitdrukking

### Overzicht

Een gegevensverzameling bevat ISO-landencodes en landennamen in verschillende formaten, zoals alleen kleine letters, alleen hoofdletters of een combinatie van hoofdletters en kleine letters. Het doel is om de gegevens op te schonen zodat elke landcode en -naam wordt geformatteerd in hoofdletters.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuwe sectie.

Het load-script bevat:

Een gegevensverzameling die wordt geladen in een tabel met de naam `Example`.

- De volgende velden in de gegevenstabel:
  - `ISO CountryCode`
  - `CountryName`

### Load-script

```
Country:
Load  * inline [
ISO_CountryCode, CountryName
gbr, United kingdom
Ind, india
CAN, Canada
Are, United arab emirates
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `ISO_CountryCode`
- CountryName

Maak de volgende berekende dimensies:

- `=Upper(ISO_CountryCode)`, om een kolom met landcodes te maken waarbij alleen hoofdletters worden gebruikt.
- `=Upper(CountryName)`, om een kolom met landnamen te maken waarbij alleen hoofdletters worden gebruikt.

Resultatentabel

| 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 |

De functie `Upper` heeft alle landcodes en landnamen geconverteerd naar hoofdletters.

## Voorbeeld - Geavanceerd scenario met Upper

Diagramuitdrukking

### Overzicht

Gebruikmakend van dezelfde gegevensverzameling als het vorige scenario, combineert dit voorbeeld landnaam en landcode en formatteert deze tekenreeksen met de functies `Capitalize` en `Upper`.

### Resultaat

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `ISO_CountryCode`
- `CountryName`

Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:

- `=Capitalize(CountryName) & ' (' & Upper(ISO_CountryCode) & ')'` om een nieuwe dimensie te maken die `CountryName` en `ISO_CountryCode` aaneenschakelt en de landnamen opmaakt met alles beginhoofdletter en de landcodes met hoofdletters.

Resultatentabel

| 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 Systeemfuncties

Systeemfuncties zijn functies voor het benaderen van eigenschappen van het systeem, het apparaat en de Qlik Sense-app.

## Overzicht van systeemfuncties

Sommige functies worden na het overzicht nader beschreven. Bij deze functies kunt u ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

### Author()

Deze scriptfunctie retourneert een tekenreeks met de titel van de huidige app. Kan zowel worden gebruikt in het script voor het laden van gegevens als in een diagramuitdrukking.

> *De eigenschap Author kan niet worden ingesteld in de huidige versie van Qlik Sense. Als u een QlikView 12-document migreert, blijft de eigenschap Author behouden.*

### CalcDim()

Als ze worden gebruikt in een dimensieuitdrukking, overschrijft **CalcDim()** de automatische interpretatie van de engine voor de interne afhandeling van de uitdrukking. De functie voorkomt dat er een nieuw intern veld wordt gemaakt in het gegevensmodel. In plaats daarvan worden selecties die zijn gemaakt in de app behandeld als selecties in toepasselijke velden van het originele gegevensmodel.

```
CalcDim - diagramfunctie(expr)
```

### ClientPlatform()

Deze functie retourneert de tekenreeks voor de gebruikersagent van de clientbrowser. Kan zowel worden gebruikt in het script voor het laden van gegevens als in een diagramuitdrukking.

**Voorbeeld:**

```
Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/35.0.1916.114 Safari/537.36
```

### ComputerName

Deze functie retourneert een tekenreeks met de naam van de computer volgens het besturingssysteem. Kan zowel worden gebruikt in het script voor het laden van gegevens als in een diagramuitdrukking.

> *Als de naam van de computer meer dan 15 tekens bevat, bevat de reeks alleen de eerste 15 tekens.*

```
ComputerName( )
```

### DocumentName

Deze functie retourneert een tekenreeks met de naam van het huidige Qlik Sense-document, zonder pad maar met de extensie. Kan zowel worden gebruikt in het script voor het laden van gegevens als in een diagramuitdrukking.

```
DocumentName( )
```

### DocumentPath

Deze functie retourneert een tekenreeks met het volledige pad naar de huidige Qlik Sense-app. Kan zowel worden gebruikt in het script voor het laden van gegevens als in een diagramuitdrukking.

```
DocumentPath( )
```

*Deze functie wordt niet ondersteund in de standaardmodus. .*

### DocumentTitle

Deze functie retourneert een tekenreeks met de titel van de huidige Qlik Sense-app. Kan zowel worden gebruikt in het script voor het laden van gegevens als in een diagramuitdrukking.

```
DocumentTitle( )
```

### EngineVersion

Deze functie retourneert de volledige engineversie van Qlik Sense als tekenreeks.

```
EngineVersion ()
```

### GetCollationLocale

Deze scriptfunctie retourneert de cultuurnaam van de landinstellingen voor collatie die wordt gebruikt. Als de variabele CollationLocale niet is ingesteld, wordt de daadwerkelijke landinstellingen van de gebruikersmachine geretourneerd.

```
GetCollationLocale( )
```

### GetObjectField

**GetObjectField()** retourneert de velduitdrukking van de dimensie. **Index** is een optioneel geheel getal dat de dimensie aangeeft die moet worden geretourneerd.

```
GetObjectField - diagramfunctie([index])
```

### GetSysAttr

De functie **GetSysAttr()** wordt gebruikt om informatie te retourneren over de tenant en de ruimte waarin een app zich bevindt. Met de functie **GetSysAttr()** kunt u de ID en naam van de tenant, de ID en naam van de ruimte en andere details over de omgeving waar de app zich bevindt, evalueren. Deze functie vereenvoudigt de identificatie van ruimte en tenant van een Qlik Sense-app die mogelijk tussen verschillende omgevingen wordt verplaatst. De functie maakt het eenvoudig om de specifieke omgeving waarin u werkt te identificeren. Dit is voornamelijk praktisch tijdens de ontwikkel-, test-, acceptatie- en productiefasen van de levenscyclus van een app.

```
GetSysAttr (name)
```

*Als u deze functie gebruikt in Qlik Sense met clientbeheer, retourneert hij alleen lege gegevenswaarden.*

### GetUserAttr

De functie **GetUserAttr()** retourneert informatie voor de gebruiker die een app opent: Kan zowel worden gebruikt in het script voor het laden van gegevens als in een diagramuitdrukking.

```
GetUserAttr (name)
```

*Als u deze functie gebruikt in Qlik Sense met clientbeheer, retourneert hij alleen lege gegevenswaarden.*

### GroupDimensionIndex

Deze functie retourneert de index van het actieve veld voor de opgegeven cyclische dimensie. Het eerste veld heeft een indexwaarde van 1. Geef de naam van de cyclische dimensie op tussen enkele aanhalingstekens. Het kan alleen in een diagramuitdrukking worden gebruikt.

```
GroupDimensionIndex(dim_name Expression)
```

### GroupDimensionLabel

Deze functie retourneert het veldlabel van de huidige stap in de opgegeven cyclische dimensie. Geef de naam van de cyclische dimensie op tussen enkele aanhalingstekens. Het kan alleen in een diagramuitdrukking worden gebruikt.

```
GroupDimensionLabel(dim_name Expression)
```

### IsPartialReload

Deze functie retourneert -1 (True) als de huidige bewerking voor opnieuw laden gedeeltelijk is uitgevoerd, anders 0 (False).

```
IsPartialReload ()
```

### InObject

De **InObject()**-diagramfunctie evalueert of het huidige object zich al dan niet binnen een ander object bevindt waarvan de ID wordt gespecificeerd in de functie argument. Het object kan een werkblad zijn of een visualisatie.

```
InObject - diagramfunctie(id_str)
```

### ObjectId

De diagramfunctie **ObjectId()** retourneert de ID van het object waarin de uitdrukking wordt geëvalueerd. De functie gebruikt een optioneel argument om te specificeren op welk type object de functie betrekking heeft. Het object kan een werkblad zijn of een visualisatie. Deze functie is alleen beschikbaar in diagramuitdrukkingen.

```
ObjectId - diagramfunctie([object_type_str])
```

### OSUser

Deze functie retourneert een tekenreeks met de naam van de gebruiker die momenteel is verbonden. Kan zowel worden gebruikt in het script voor het laden van gegevens als in een diagramuitdrukking.

```
OSUser( )
```

> *In Qlik Sense Desktop en Qlik Sense Client-Managed Mobile retourneert deze functie altijd 'Personal\Me'.*

### ProductVersion

Deze functie retourneert het volledige versie- en buildnummer van Qlik Sense als tekenreeks.

Deze functie is verouderd en vervangen door **EngineVersion()**.

```
ProductVersion ()
```

### ReloadTime

Deze functie retourneert een tijdstempel voor het moment waarop het laden van de laatste gegevens werd voltooid. Kan zowel worden gebruikt in het script voor het laden van gegevens als in een diagramuitdrukking.

```
ReloadTime( )
```

### StateName

**StateName()** haalt de naam op van de alternatieve state van de visualisatie waarin deze wordt gebruikt. StateName kan bijvoorbeeld worden gebruikt om visualisaties te maken met dynamische tekst en kleuren om aan te geven wanneer de state van een visualisatie is gewijzigd. U kunt de functie gebruiken in een diagramuitdrukking maar niet om de state te bepalen waarnaar de uitdrukking verwijst.

```
StateName - diagramfunctie()
```

## CalcDim - diagramfunctie

Als ze worden gebruikt in een dimensieuitdrukking, overschrijft **CalcDim()** de automatische interpretatie van de engine voor de interne afhandeling van de uitdrukking. De functie voorkomt dat er een nieuw intern veld wordt gemaakt in het gegevensmodel. In plaats daarvan worden selecties die zijn gemaakt in de app behandeld als selecties in toepasselijke velden van het originele gegevensmodel.

Deze functie kan niet worden gebruikt in het load-script. Bovendien is het voor systeemprestaties een best practice om, indien mogelijk, de velden die in de berekende dimensie worden gebruikt, als logische velden via het load-script te laden. **CalcDim()** biedt een manier om het gedrag van de engine te regelen in scenario's waarin het laden van het berekende veld vanuit het script niet mogelijk is.

In Qlik Sense wordt de dimensie-uitdrukking standaard gerealiseerd als een veld-on-the-fly—dat wil zeggen, als een daadwerkelijk veld, zichtbaar in de huidige selectiestaaf, dat op aanvraag wordt gebouwd. Selecties worden toegepast en afgeleid op dit veld, vergelijkbaar met elk logisch veld uit de script.

Een expressie verpakt in **CalcDim()** zal zich op dezelfde manier gedragen als alle berekende dimensie-expressies werken in QlikView 12, wat een ad-hoc toewijzing is van de berekende waarden aan de onderliggende bronveldwaarden. Alle on-the-fly veldberekeningen kunnen worden overgezet naar berekende dimensies met behulp van de **CalcDim()** functie, aangezien de berekende dimensie aan minder vereisten hoeft te voldoen op basis van het gegevensmodel dan de overeenkomstige on-the-fly veldberekening. Deze transformatie kan wenselijk zijn om een vergelijkbaar gedrag te garanderen voor alle berekende dimensies, waar het gegevensmodel het gebruik van een 'field-on-the-fly'-berekening in sommige gevallen kan verhinderen, maar in andere niet.
≤ Berekende velden

**Syntaxis:**

```
CalcDim(expr)
```

**Retourgegevenstypen:** dubbele waarde

Argumenten

| Argument | Beschrijving |
|---|---|
| expr | Een expressie die wordt gebruikt als een dimensie, of als een segment binnen een dimensie-expressie. |

## Wanneer gebruiken

**CalcDim()** is nuttig wanneer u meer controle nodig heeft over hoe de Qlik engine omgaat met de velden die u aanroept in een dimensie-uitdrukking.

Standaard bepaalt Qlik Sense automatisch of interne velden in het gegevensmodel moeten worden aangemaakt en ernaar moet worden verwezen wanneer u een aangepaste dimensie-uitdrukking gebruikt. Qlik Sense maakt en gebruikt deze interne velden wanneer dit mogelijk is. Wanneer dit niet mogelijk is, wordt de expressie afgehandeld met verwijzing naar de bestaande gegevensmodelvelden waarop deze berust.

Afhankelijk van uw gebruiksscenario is deze automatische verwerking mogelijk niet altijd wenselijk. Bijvoorbeeld, als u een variabele gebruikt in een dimensie-uitdrukking, kan het automatisch aanmaken van nieuwe velden problematisch worden. In dit geval, gebruik **CalcDim()** om de expressie te dwingen verwerkt te worden zonder deze methode.

Hier volgen enkele van de meest voorkomende gebruiksscenario's:

- U hebt QlikView 12 inhoud gemigreerd naar Qlik Sense en u wilt de QlikView 12 verwerkingsmethode voor berekende dimensies behouden.
- U wilt het selectiegedrag in uw app vereenvoudigen. U wilt bijvoorbeeld veldselecties weergeven als selecties in velden uit het oorspronkelijke gegevensmodel, in plaats van selecties in nieuwe velden.

- Uw gebruiksscenario vereist een ad-hoc expressie voor consistentie en prestaties. Bijvoorbeeld, als uw dimensie-uitdrukking gebruikersgestuurde dynamische visualisatieparameters bevat, zoals configureerbare opmaak, afkapwaarden of perioden.

Voorbeelden van functies

| Voorbeeld | Resultaat |
|---|---|
| `=CalcDim(if(Year < 2023, 'Older', Year))` | Een selectie maken in een visualisatie die deze dimensie gebruikt, wordt verwerkt en weergegeven als een selectie in de bijbehorende *Year* veldwaarde uit het gegevensmodel.<br><br>Omgekeerd, stel dat u de expressie niet omhult in **CalcDim()**. In dit geval zou het er zo uitzien:<br><br>`=if(Year < 2023, 'Older', Year)`<br><br>Een selectie maken in de bovenstaande expressie zou worden verwerkt en weergegeven als een selectie in het interne veld *=if(Year < 2023, 'Older', Year)*, dat niet in het gegevensmodel zit. |

## Voorbeeld 1 – Eenvoudige vergelijking

Diagramuitdrukkingen en resultaten

### Overzicht

In dit voorbeeld zult u:

- Voeg een uitgelijnde tabel toe en laad gegevens in een app.
- Maak een tabel om de gegevens te visualiseren en berekeningen uit te voeren.
- Maak selecties in de tabel om het verschil in gedrag te observeren.

Open de Editor voor laden van gegevens en voeg het volgende load-script toe aan een nieuwe sectie.

De load-script laadt een uitgelijnde tabel. Deze tabel bevat twee velden: een veld voor de voornaam van de persoon, en een veld voor de achternaam van de persoon.

### Load-script

```
Names:

Load * inline [
FirstName,LastName
John,Doe
Amanda,Honda
Amalia,Craig
Cart,Lynch
];
```

### Tabel aanmaken

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- *FirstName*
- *LastName*

Vervolgens, voeg elk van de volgende uitdrukkingen toe als dimensies in de tabel:

- *=FirstName & ' ' & LastName*
- *=CalcDim(FirstName & ' ' & LastName)*

De tabel moet er als volgt uitzien.

Qlik Sense tabel met voorbeeldresultaten

| 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 |

### Vergelijking: Selecties maken

Selecteer in de *=FirstName & ' ' & LastName* kolom van de tabel de waarde *John Doe*. Merk op dat in de selectiewerkbalk de waarde is geselecteerd als een waarde uit een veld genaamd *=FirstName&' '&LastName*.

Wis de selectie.

Selecteer in de *=CalcDim(FirstName & ' ' & LastName)* kolom van de tabel de waarde *John Doe*. In de selectiewerkbalk wordt de waarde geselecteerd als waarden uit twee verschillende velden: *FirstName* en *LastName*.

Deze twee expressies presenteren dezelfde informatie: een concatenatie van de *FirstName* en *LastName* velden. Het verschil is dat zonder **CalcDim()** een synthetisch veld wordt aangemaakt om de berekening op te slaan. Met **CalcDim()** worden de oorspronkelijke velden uit het gegevensmodel gebruikt als basis voor de berekende dimensie.

## Voorbeeld 2 – Praktische use case om het standaardgedrag te wijzigen

Diagramuitdrukkingen en resultaten

### Overzicht

In dit voorbeeld creëren we dynamische dimensie-uitdrukkingen die verwijzen naar een variabele, in plaats van verwijzingen naar statische waarden.

In dit voorbeeld zult u:

- Voeg een uitgelijnde tabel toe en laad gegevens in de app.
- Maak een variabele om dynamisch een waarde op te slaan.
- Maak een tabel om de gegevens te visualiseren en berekeningen uit te voeren.
- Maak selecties in de tabel om het verschil in gedrag te observeren.

Open de Editor voor laden van gegevens en voeg het volgende load-script toe aan een nieuwe sectie.

De load-script laadt een uitgelijnde tabel. Deze tabel bevat een datumveld en een berekend veld, *Year*, dat is afgeleid van de ruwe datums.

### Load-script

```
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
];
```

### Variabelecreatie

Laad de gegevens en open een werkblad. Maak een variabele met de naam *vCutoffYear*, en geef deze de volgende **Definitie**: *2023*

### Tabel aanmaken

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- *date*
- *Year*
- *amount*

Vervolgens, voeg elk van de volgende uitdrukkingen toe als dimensies in de tabel:

- *=if(Year < $(vCutoffYear), 'Older', Year)*
- *=CalcDim(if(Year < $(vCutoffYear), 'Older', Year))*

De tabel moet er als volgt uitzien.

Qlik Sense tabel met voorbeeldresultaten

| 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 |

| date | Year | amount | if(Year < $(vCutoffYear), 'Older', Year) | =CalcDim(if(Year < $(vCutoffYear), 'Older', Year)) |
|---|---|---|---|---|
| 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 |
| 11/12/2023 | 2023 | 135.41 | 2023 | 2023 |
| 1/15/2024 | 2024 | 67.24 | 2024 | 2024 |

**Vergelijkingen, deel 1: Gedrag zonder CalcDim()**

In de bovenstaande tabel is er geen verschil tussen de twee berekende dimensiekolommen.

Selecteer in de *if(Year < $(vCutoffYear), 'Older', Year)* kolom van de tabel de waarde *2023*. Merk op dat in de selectiewerkbalk de waarde is geselecteerd als een waarde uit een veld met de naam *=if (Year<2023,'Older',Year)*.

Vervolgens wijzigt u de **Definitie** van de *vCutoffYear* variabele naar *2024*. Merk op dat hoewel het selectieveld *=if(Year<2023,'Older',Year)* hetzelfde is en de geselecteerde waarde *2023* behoudt, de gegevens in de tabel zijn gewijzigd. De waarde *2023* is gewijzigd naar de waarde *Older*. Als u *Older* selecteert in de *if(Year < $(vCutoffYear), 'Older', Year)* kolom, wordt de selectie *Older* gemaakt in een nieuw veld: *=if(Year<2024,'Older',Year)*.

In wezen wordt intern een nieuw veld aangemaakt voor elke waarde van de *vCutoffYear* variabele. Bij het werken met gelijktijdige en veranderende selecties is dit gedrag mogelijk niet wenselijk en kan het de interpretatie van gegevens complexer maken.

**Vergelijkingen, deel 2: Gedrag met CalcDim()**

Nu u het gedrag van de expressie hebt gezien wanneer deze niet is ingesloten in **CalcDim()**, wis alle selecties in de app. Reset de *vCutoffYear* variabele op de waarde *2023*.

Selecteer in de *=CalcDim(if(Year < $(vCutoffYear), 'Older', Year))* kolom van de tabel de waarde *2023*. Merk op dat in de selectiewerkbalk de waarde wordt geselecteerd als een waarde uit het onderliggende veld, *Year*.

Vervolgens wijzigt u de **Definitie** van de *vCutoffYear* variabele naar *2024*. Het selectieveld *Year* is hetzelfde, het behoudt de geselecteerde waarde *2023*, en de gegevens in de tabel zijn gewijzigd, zoals te zien is zonder **CalcDim()**. Als u echter nu *Older* selecteert in de *=CalcDim(if(Year < $(vCutoffYear), 'Older', Year))* kolom, wordt de waarde *2024* geselecteerd in *Year*.

Alle selecties die u maakt in een berekende dimensie omwikkeld met **CalcDim()** worden weergegeven als selecties in de onderliggende velden waarnaar wordt verwezen vanuit het oorspronkelijke gegevensmodel. Dit kan de gegevensinterpretatie vereenvoudigen en voorkomt dat rommelige, complexe uitdrukkingen verschijnen in de selectiestaaf wanneer gebruikers de analyse-inhoud consumeren.

## EngineVersion

Deze functie retourneert de volledige engineversie van Qlik Sense als tekenreeks.

**Syntaxis:**

```
EngineVersion()
```

## Systeemfuncties

Systeemfuncties zijn functies voor het benaderen van eigenschappen van het systeem, het apparaat en de Qlik Sense-app.

### Overzicht van systeemfuncties

Sommige functies worden na het overzicht nader beschreven. Bij deze functies kunt u ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

**Author()**
Deze scriptfunctie retourneert een tekenreeks met de titel van de huidige app. Kan zowel worden gebruikt in het script voor het laden van gegevens als in een diagramuitdrukking.

> *De eigenschap Author kan niet worden ingesteld in de huidige versie van Qlik Sense. Als u een QlikView 12-document migreert, blijft de eigenschap Author behouden.*

**CalcDim()**
Als ze worden gebruikt in een dimensieuitdrukking, overschrijft **CalcDim()** de automatische interpretatie van de engine voor de interne afhandeling van de uitdrukking. De functie voorkomt dat er een nieuw intern veld wordt gemaakt in het gegevensmodel. In plaats daarvan worden selecties die zijn gemaakt in de app behandeld als selecties in toepasselijke velden van het originele gegevensmodel.

```
CalcDim - diagramfunctie(expr)
```

**ClientPlatform()**
Deze functie retourneert de tekenreeks voor de gebruikersagent van de clientbrowser. Kan zowel worden gebruikt in het script voor het laden van gegevens als in een diagramuitdrukking.

**Voorbeeld:**

```
Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/35.0.1916.114 Safari/537.36
```

**ComputerName**

Deze functie retourneert een tekenreeks met de naam van de computer volgens het besturingssysteem. Kan zowel worden gebruikt in het script voor het laden van gegevens als in een diagramuitdrukking.

> *Als de naam van de computer meer dan 15 tekens bevat, bevat de reeks alleen de eerste 15 tekens.*

```
ComputerName( )
```

### DocumentName

Deze functie retourneert een tekenreeks met de naam van het huidige Qlik Sense-document, zonder pad maar met de extensie. Kan zowel worden gebruikt in het script voor het laden van gegevens als in een diagramuitdrukking.

```
DocumentName( )
```

### DocumentPath

Deze functie retourneert een tekenreeks met het volledige pad naar de huidige Qlik Sense-app. Kan zowel worden gebruikt in het script voor het laden van gegevens als in een diagramuitdrukking.

```
DocumentPath( )
```

> *Deze functie wordt niet ondersteund in de standaardmodus. .*

### DocumentTitle

Deze functie retourneert een tekenreeks met de titel van de huidige Qlik Sense-app. Kan zowel worden gebruikt in het script voor het laden van gegevens als in een diagramuitdrukking.

```
DocumentTitle( )
```

### EngineVersion

Deze functie retourneert de volledige engineversie van Qlik Sense als tekenreeks.

```
EngineVersion ()
```

### GetCollationLocale

Deze scriptfunctie retourneert de cultuurnaam van de landinstellingen voor collatie die wordt gebruikt. Als de variabele CollationLocale niet is ingesteld, wordt de daadwerkelijke landinstellingen van de gebruikersmachine geretourneerd.

```
GetCollationLocale( )
```

### GetObjectField

**GetObjectField()** retourneert de velduitdrukking van de dimensie. **Index** is een optioneel geheel getal dat de dimensie aangeeft die moet worden geretourneerd.

```
GetObjectField - diagramfunctie([index])
```

### GetSysAttr

De functie **GetSysAttr()** wordt gebruikt om informatie te retourneren over de tenant en de ruimte waarin een app zich bevindt. Met de functie **GetSysAttr()** kunt u de ID en naam van de tenant, de ID en naam van de ruimte en andere details over de omgeving waar de app zich bevindt, evalueren. Deze functie vereenvoudigt de identificatie van ruimte en tenant van een Qlik Sense-app die mogelijk tussen verschillende omgevingen wordt verplaatst. De functie maakt het eenvoudig om de specifieke omgeving waarin u werkt te identificeren. Dit is voornamelijk praktisch tijdens de ontwikkel-, test-, acceptatie- en productiefasen van de levenscyclus van een app.

```
GetSysAttr (name)
```

> *Als u deze functie gebruikt in Qlik Sense met clientbeheer, retourneert hij alleen lege gegevenswaarden.*

### GetUserAttr

De functie **GetUserAttr()** retourneert informatie voor de gebruiker die een app opent: Kan zowel worden gebruikt in het script voor het laden van gegevens als in een diagramuitdrukking.

```
GetUserAttr (name)
```

> *Als u deze functie gebruikt in Qlik Sense met clientbeheer, retourneert hij alleen lege gegevenswaarden.*

### GroupDimensionIndex

Deze functie retourneert de index van het actieve veld voor de opgegeven cyclische dimensie. Het eerste veld heeft een indexwaarde van 1. Geef de naam van de cyclische dimensie op tussen enkele aanhalingstekens. Het kan alleen in een diagramuitdrukking worden gebruikt.

```
GroupDimensionIndex(dim_name Expression)
```

### GroupDimensionLabel

Deze functie retourneert het veldlabel van de huidige stap in de opgegeven cyclische dimensie. Geef de naam van de cyclische dimensie op tussen enkele aanhalingstekens. Het kan alleen in een diagramuitdrukking worden gebruikt.

```
GroupDimensionLabel(dim_name Expression)
```

### IsPartialReload

Deze functie retourneert - 1 (True) als de huidige bewerking voor opnieuw laden gedeeltelijk is uitgevoerd, anders 0 (False).

```
IsPartialReload ()
```

### InObject

De **InObject()**-diagramfunctie evalueert of het huidige object zich al dan niet binnen een ander object bevindt waarvan de ID wordt gespecificeerd in de functie argument. Het object kan een werkblad zijn of een visualisatie.

```
InObject - diagramfunctie(id_str)
```

### ObjectId

De diagramfunctie **ObjectId()** retourneert de ID van het object waarin de uitdrukking wordt geëvalueerd. De functie gebruikt een optioneel argument om te specificeren op welk type object de functie betrekking heeft. Het object kan een werkblad zijn of een visualisatie. Deze functie is alleen beschikbaar in diagramuitdrukkingen.

```
ObjectId - diagramfunctie([object_type_str])
```

### OSUser

Deze functie retourneert een tekenreeks met de naam van de gebruiker die momenteel is verbonden. Kan zowel worden gebruikt in het script voor het laden van gegevens als in een diagramuitdrukking.

```
OSUser( )
```

> *In Qlik Sense Desktop en Qlik Sense Client-Managed Mobile retourneert deze functie altijd 'Personal\Me'.*

### ProductVersion

Deze functie retourneert het volledige versie- en buildnummer van Qlik Sense als tekenreeks.

Deze functie is verouderd en vervangen door **EngineVersion()**.

```
ProductVersion ()
```

### ReloadTime

Deze functie retourneert een tijdstempel voor het moment waarop het laden van de laatste gegevens werd voltooid. Kan zowel worden gebruikt in het script voor het laden van gegevens als in een diagramuitdrukking.

```
ReloadTime( )
```

### StateName

**StateName()** haalt de naam op van de alternatieve state van de visualisatie waarin deze wordt gebruikt. StateName kan bijvoorbeeld worden gebruikt om visualisaties te maken met dynamische tekst en kleuren om aan te geven wanneer de state van een visualisatie is gewijzigd. U kunt de functie gebruiken in een diagramuitdrukking maar niet om de state te bepalen waarnaar de uitdrukking verwijst.

```
StateName - diagramfunctie()
```

## GetSysAttr

De functie **GetSysAttr()** wordt gebruikt om informatie te retourneren over de tenant en de ruimte waarin een app zich bevindt. Met de functie **GetSysAttr()** kunt u de ID en naam van de tenant, de ID en naam van de ruimte en andere details over de omgeving waar de app zich

bevindt, evalueren. Deze functie vereenvoudigt de identificatie van ruimte en tenant van een Qlik Sense-app die mogelijk tussen verschillende omgevingen wordt verplaatst. De functie maakt het eenvoudig om de specifieke omgeving waarin u werkt te identificeren. Dit is voornamelijk praktisch tijdens de ontwikkel-, test-, acceptatie- en productiefasen van de levenscyclus van een app.

Deze functie kan worden gebruikt in het load-script of de diagramuitdrukking van de app, ofwel direct als een uitdrukking of in een uitdrukking die is opgeslagen als een variabele. Als u verwacht dat uw app gedurende zijn levencyclus tussen ruimten of tenants zal worden verplaatst, gebruikt u de functie in een diagramuitdrukking om dynamisch ruimte- en tenantinformatie te verkrijgen, zonder de app opnieuw te laden. Wanneer u dit doet, slaat u de functie-uitdrukking op in een variabele buiten het load-script, of gebruikt u de functie direct in de diagramuitdrukking.

Gebruik **GetSysAttr()** om de volgende informatie over uw app te bepalen:

- De id van de tenant waar de app zich bevindt.
- De naam van de tenant waar de app zich bevindt.
- De URL van de tenant waar de app zich bevindt.
- De alias-URL is gedefinieerd voor de tenant die de app bevat.
- De regio van de tenant die de app bevat.
- De id van de ruimte waar de app zich bevindt.
- De naam van de ruimte waar de app zich bevindt.
- Het type van de ruimte (gedeeld of beheerd) waar de app zich bevindt.

Als u deze functie gebruikt in Qlik Sense met clientbeheer, retourneert hij lege gegevenswaarden. U kunt de functie daarom gebruiken om load-scripts te ontwikkelen in Qlik Sense met clientbeheer, zonder dat er fouten optreden en om de apps later naar Qlik Cloud te uploaden.

Ga naar GetSysAttr - script- en diagramfunctie om toegang te krijgen tot de volledige documentatie van de Qlik Cloud-functie.

## GetUserAttr

Deze functie retourneert gebruikerskenmerken, inclusief e-mailadres, onderwerp en groepen, voor een geselecteerde app. Kan zowel worden gebruikt in het script voor het laden van gegevens als in een diagramuitdrukking.

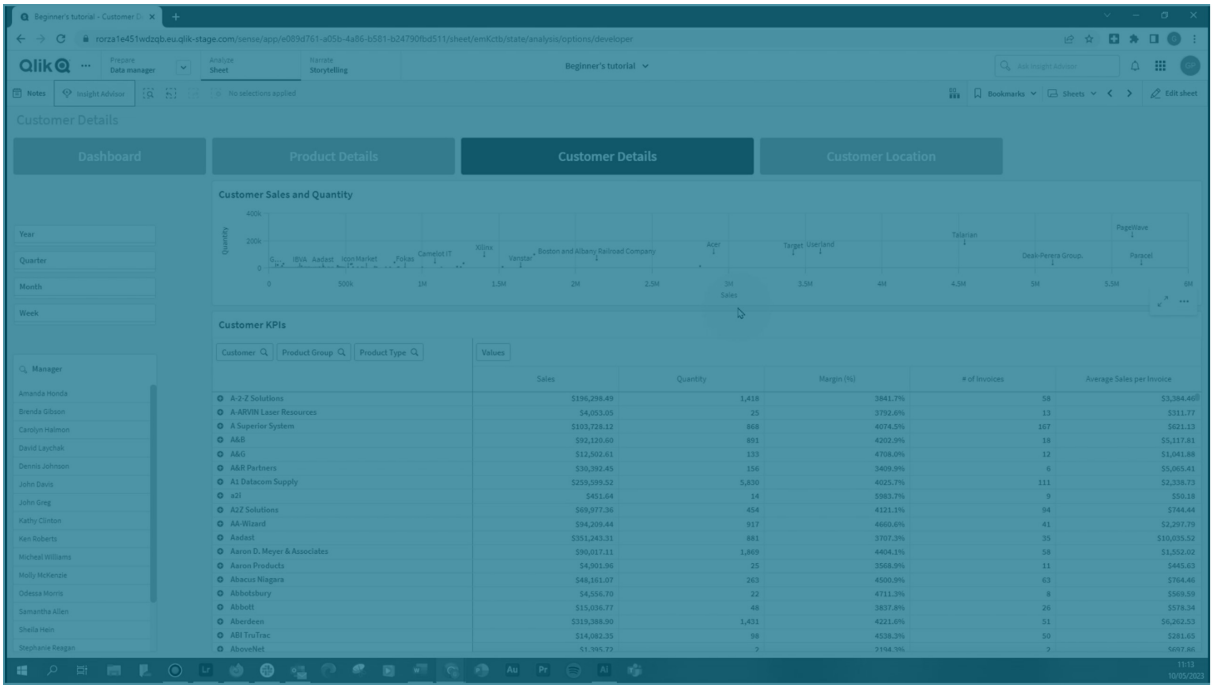
Als u deze functie gebruikt in Qlik Sense met clientbeheer, retourneert hij lege gegevenswaarden. U kunt de functie daarom gebruiken om load-scripts te ontwikkelen in Qlik Sense met clientbeheer, zonder dat er fouten optreden en om de apps later naar Qlik Cloud te uploaden.

Ga naar GetUserAttr - script- en diagramfunctie om toegang te krijgen tot de volledige documentatie van de Qlik Cloud-functie.

## InObject - diagramfunctie

De **InObject()**-diagramfunctie evalueert of het huidige object zich al dan niet binnen een ander object bevindt waarvan de ID wordt gespecificeerd in de functie argument. Het object kan een werkblad zijn of een visualisatie.

Deze functie kan worden gebruikt om de hiërarchie van objecten in een werkblad te tonen, van het toplevelwerkbladobject tot in andere visualisaties geneste visualisaties. Deze functie kan naast de functies **if** en **ObjectId** worden gebruikt om aangepast navigeren te creëren in uw apps.




**Syntaxis:**

```
InObject(id_str)
```

**Retourgegevenstypen:** Booleaanse waarde

In Qlik Sense wordt de booleaanse waarde Waar vertegenwoordigd door -1 en de waarde Onwaar door 0.

Argumenten

| Argument | Beschrijving |
|---|---|
| **id_str** | Een tekenreekswaarde die de ID vertegenwoordigt van het object dat wordt geëvalueerd. |

De werkblad-ID kan worden verkregen via de app-URL. Gebruik voor visualisaties de **Ontwikkelaar**-opties om de object-ID en de tekenreeks van het objecttype te identificeren.

**Doe het volgende:**

1. Voeg in analysemodus de volgende tekst toe aan uw URL: */opties/ontwikkelaar*
2. Klik met de rechtermuisknop op de visualisatie en klik op **Ontwikkelaar**.
3. Verkrijg onder **Eigenschappen** de object-ID uit de dialoogkop en het objecttype uit de eigenschap **"qType"**.

**Beperkingen:**

Deze functie kan tot onverwachte resultaten leiden wanneer ze wordt opgeroepen in een object (een knop bijvoorbeeld) binnen een container die een masteritem is. De beperking geldt ook voor het filtervak masteritems. Dit zijn containers voor een aantal keuzelijsten. Dit komt door de manier waarop masteritems gebruikmaken van de objecthiërarchie.

**InObject()** wordt vaak gebruikt in combinatie met de volgende functies:

Gerelateerde functies

| Functie | Interactie |
|---|---|
| *if (page 660)* | De functies **if** en **ObjectId** kunnen in combinatie worden gebruikt om voorwaardelijke uitdrukkingen te creëren. Zo kunnen visualisaties bijvoorbeeld voorwaardelijke kleuren krijgen dankzij uitdrukkingen die gebruikmaken van deze functies. |
| *ObjectId - diagramfunctie (page 1912)* | Net als **if** wordt **ObjectId** ook gebruikt met **InObject** om voorwaardelijke uitdrukkingen te creëren. |

## Voorbeeld 1 – Basisfunctionaliteit

### Diagramuitdrukking en resultaten

Onderstaand basisvoorbeeld laat zien hoe te bepalen of een object zich binnen een ander object bevindt. In dit geval controleren we of een **Tekst & afbeelding**-object zich binnen een werkbladobject bevindt en de ID van het werkblad als argument gebruikt.

**Doe het volgende:**

1. Open een nieuw werkblad en sleep een diagram met **tekst en afbeelding** op het werkblad.
2. Klik in het eigenschappenvenster op **Meting toevoegen**.
3. Klik op $f\times$ om de uitdrukkingseditor te openen.
4. Plak de volgende uitdrukking in de dialoog:
   `=InObject()`
5. Wijzig de uitdrukking om de ID van uw werkblad op te nemen als een tekenreeks tussen de haakjes. Zo zou u voor een werkblad met ID 1234-5678 bijvoorbeeld het onderstaande gebruiken:
   `=InObject('1234-5678')`
6. Klik op **Toepassen**.

De waarde `-1` wordt weergegeven in het diagram, waarmee wordt aangegeven dat de uitdrukking als waar werd beoordeeld.

## Voorbeeld 2 – Objecten met voorwaardelijke kleuren

### Diagramuitdrukking en resultaten

**Overzicht**

Het volgende voorbeeld laat zien hoe u aangepaste navigatieknoppen maakt met verschillende kleuren om aan te geven dat het werkblad momenteel geopend is.

Begin met het creëren van een nieuwe app en de editor voor het laden van gegevens te openen. Plak het volgende load-script in een nieuw tabblad. Merk op dat de gegevens zelf fungeren als plaatshouder en niet in de voorbeeldinhoud worden gebruikt.

**Load-script**

```
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
];
```

### De visualisaties maken

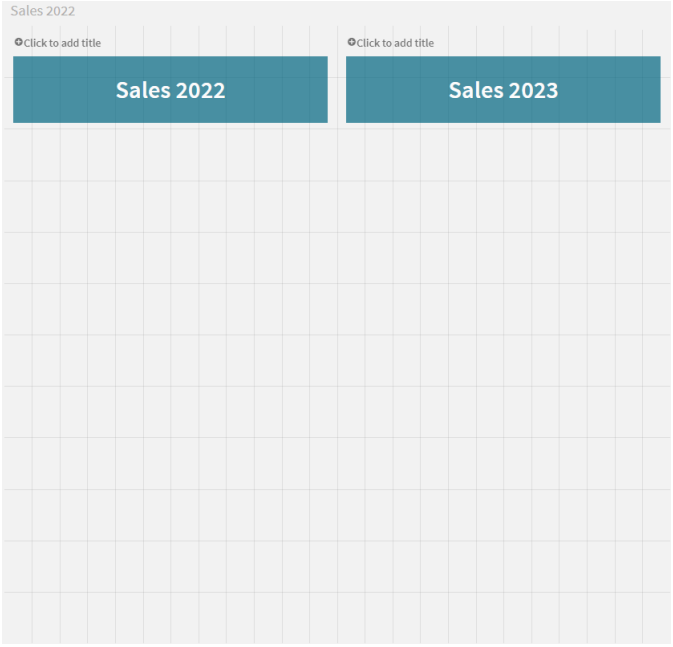
Laad de gegevens en maak twee nieuwe werkbladen. Geef ze de respectievelijke titels *Verkopen 2022* en *Verkopen 2023*.

Bouw vervolgens twee knopobjecten om tussen de twee werkbladen te kunnen navigeren.

### Doe het volgende:

1. Voeg twee **Knop**-objecten toe aan het werkblad.
2. Zet onder **Uiterlijk** > **Algemeen** het **Label** van elke knop op respectievelijk *Verkopen 2022* en *Verkopen 2023*.

3. Deel de knoppen zo in dat ze overeenkomen met de volgende afbeelding.
   *Verkopen 2022 werkbladindeling met twee navigatieknoppen*




4. Selecteer de *Verkopen 2022*-knop en open **Acties en navigatie** in het eigenschappenvenster.
5. Klik op **Actie toevoegen** en selecteer onder **Navigatie Ga naar een werkblad**.
6. Selecteer onder **Werkblad** *Verkopen 2022*.
7. Herhaal deze actie-instelling om de **Verkopen 2023**-knop aan het *Verkopen 2023*-werkblad te koppelen.
8. Converteer de knoppen naar masteritems door er met de rechtermuisknop op te klikken en **Toevoegen aan masteritems** te selecteren.

U kunt nu iedere knop kopiëren en in het werkblad *Verkopen 2023* plakken en daar gebruikmaken van dezelfde afmetingen en indeling.

**Voorwaardelijke kleuren creëren**

Configureer daarna de knoppen zodat ze blauw worden wanneer ze gekoppeld zijn aan het op dat moment geopende werkblad en lichtgrijs wanneer gekoppeld aan het werkblad dat niet geopend is.

**Doe het volgende:**

1. Open het *Verkopen 2022*-werkblad en verkrijg de werkblad-ID via de URL. Houd het *Verkopen 2022*-werkblad open.
2. Klik op het **Verkopen 2022**-knopmasteritem en selecteer **Bewerken** in het eigenschappenvenster.
3. Selecteer onder **Uiterlijk** > **Achtergrond Op uitdrukking** om de knop een kleur te geven.
4. Plak in **Uitdrukking** de volgende tekst:
   *=if(InObject(''), Blue(), LightGray())*

5. Plak in bovenstaande uitdrukking tussen de haakjes de werkblad-ID voor het *Verkopen 2022*-werkblad.

De knop is nu geconfigureerd om blauw te worden wanneer het *Verkopen 2022*-werkblad geopend is en lichtgrijs wanneer het niet geopend is.

Herhaal bovenstaande instructies voor het *Verkopen 2023*-werkblad, waarbij u het **Verkopen 2023**-knopmasteritem aan de werkblad-ID *Verkopen 2023* koppelt.

Als het goed is, heeft elk werkblad nu twee knoppen die het op dat moment geopende werkblad aangeven met de kleur blauw.

*Verkopen 2022-werkblad met blauwe kleur om aan te geven dat Verkopen 2022 op dit moment wordt weergegeven*

## IsPartialReload

Deze functie retourneert -1 (True) als de huidige bewerking voor opnieuw laden gedeeltelijk is uitgevoerd, anders 0 (False).

**Syntaxis:**

```
IsPartialReload()
```

## ObjectId - diagramfunctie

De diagramfunctie **ObjectId()** retourneert de ID van het object waarin de uitdrukking wordt geëvalueerd. De functie gebruikt een optioneel argument om te specificeren op welk type object de functie betrekking heeft. Het object kan een werkblad zijn of een visualisatie. Deze functie is alleen beschikbaar in diagramuitdrukkingen.

**Syntaxis:**

```
ObjectId([object_type_str])
```

**Retourgegevenstypen:** tekenreeks

Het enige argument van de functie, **object_type_str**, is optioneel en refereert aan een tekenreekswaarde die het objecttype vertegenwoordigt.

Argumenten

| Argument | Beschrijving |
|---|---|
| **object_type_str** | Een tekenreekswaarde die het type vertegenwoordigt van het object dat wordt geëvalueerd. |

Als er in de functie-uitdrukking geen argument wordt gespecificeerd, retourneert **ObjectId()** de ID van het object waarin de uitdrukking wordt gebruikt. Gebruik *ObjectId('sheet')* om de ID van het werkbladobject waarbinnen de visualisatie verschijnt te retourneren.

Specificeer in het geval van visualisatie-objecten die in andere visualisatie-objecten zijn genest het gewenste objecttype in het functie-argument voor verschillende resultaten. Gebruik bijvoorbeeld voor een diagram met **tekst en afbeelding** binnen een container 'text-image' om het *tekst & afbeelding*-object te retourneren en **'container'** om de ID van de *container* te retourneren.

**Doe het volgende:**

1. Voeg in analysemodus de volgende tekst toe aan uw URL:
   */opties/ontwikkelaar*
2. Klik met de rechtermuisknop op de visualisatie en klik op **Ontwikkelaar**.
3. Verkrijg onder **Eigenschappen** de object-ID uit de dialoogkop en het objecttype uit de eigenschap **"qType"**.

**Beperkingen:**

Deze functie kan tot onverwachte resultaten leiden wanneer ze wordt opgeroepen in een object (een knop bijvoorbeeld) binnen een container die een masteritem is. De beperking geldt ook voor het filtervak masteritems. Dit zijn containers voor een aantal keuzelijsten. Dit komt door de manier waarop masteritems gebruikmaken van de objecthiërarchie.

De diagramuitdrukking *ObjectId('werkblad')* zal in dergelijke gevallen een lege tekenreeks retourneren, terwijl *ObjectId('masterobject')* de identifier van het bezittende masteritem zal retourneren.

**ObjectId()** wordt vaak gebruikt in combinatie met de volgende functies:

Gerelateerde functies

| Functie | Interactie |
|---|---|
| *if (page 660)* | De functies **if** en **ObjectId** kunnen in combinatie worden gebruikt om voorwaardelijke uitdrukkingen te creëren. Zo kunnen visualisaties bijvoorbeeld voorwaardelijke kleuren krijgen dankzij uitdrukkingen die gebruikmaken van deze functies. |
| *InObject - diagramfunctie (page 1907)* | Net als **if** wordt **InObject** ook gebruikt met **ObjectId** om voorwaardelijke uitdrukkingen te creëren. |

## Voorbeeld 1 – Diagram object-ID retourneren

Diagramuitdrukking en resultaten

Het volgende basisvoorbeeld laat zien hoe u de ID van een visualisatie retourneert.

**Doe het volgende:**

1. Open een nieuw werkblad en sleep een **Tekst & afbeelding**-diagram op het werkblad.
2. Klik in het eigenschappenvenster op **Meting toevoegen**.
3. Klik op $f_x$ om de uitdrukkingseditor te openen.
4. Plak de volgende uitdrukking in de dialoog:
   `=ObjectId()`
5. Klik op **Toepassen**.

De ID van het **Tekst & afbeelding**-object wordt weergegeven in de visualisatie.

Hetzelfde resultaat kan worden bereikt met de volgende uitdrukking:

```
=ObjectId('text-image')
```

### Voorbeeld 2 – Werkblad-ID retourneren

**Diagramuitdrukking en resultaten**

Het volgende basisvoorbeeld laat zien hoe u de ID van het werkblad waarin een visualisatie optreedt retourneert.

**Doe het volgende:**

1. Open een nieuw werkblad en sleep een **Tekst & afbeelding**-diagram op het werkblad.
2. Klik in het eigenschappenvenster op **Meting toevoegen**.
3. Klik op $f_x$ om de uitdrukkingseditor te openen.
4. Plak de volgende uitdrukking in de dialoog:
   `=ObjectId('sheet')`
5. Klik op **Toepassen**.

De ID van het werkblad wordt weergegeven in de visualisatie.

### Voorbeeld 3 – Geneste uitdrukking

**Diagramuitdrukking en resultaten**

Het volgende voorbeeld laat zien hoe de **ObjectId()**-functie kan in andere uitdrukkingen worden genest.

**Doe het volgende:**

1. Open een nieuw werkblad en sleep een **Tekst & afbeelding**-diagram op het werkblad.
2. Klik in het eigenschappenvenster op **Meting toevoegen**.
3. Klik op $f_x$ om de uitdrukkingseditor te openen.
4. Plak de volgende uitdrukking in de dialoog:
   *=if(InObject(ObjectId('text-image')), 'In Tekst & afbeelding', 'Niet in Tekst & afbeelding')*
5. Klik op **Toepassen**.

De tekst *In Tekst & afbeelding* verschijnt in het diagram, waarmee wordt aangegeven dat het object waaraan wordt gerefereerd in de uitdrukking een **Tekst & afbeelding**-diagram is.

Raadpleeg voor een meer gedetailleerd voorbeeld dat gebruikmaakt van voorwaardelijke kleuren, het voorbeeld op *InObject - diagramfunctie (page 1907)*

## ProductVersion

Deze functie is verouderd en vervangen door EngineVersion().Deze functie retourneert het volledige versie- en buildnummer van Qlik Sense als tekenreeks.

**Syntaxis:**

```
ProductVersion()
```

## StateName - diagramfunctie

**StateName()** haalt de naam op van de alternatieve state van de visualisatie waarin deze wordt gebruikt. StateName kan bijvoorbeeld worden gebruikt om visualisaties te maken met dynamische tekst en kleuren om aan te geven wanneer de state van een visualisatie is gewijzigd. U kunt de functie gebruiken in een diagramuitdrukking maar niet om de state te bepalen waarnaar de uitdrukking verwijst.

**Syntaxis:**

```
StateName ()
```

**Example 1:**

Dynamische tekst

```
='Region - ' & if(StateName() = '$', 'Default', StateName())
```

**Example 2:**

Dynamische kleuren

```
if(StateName() = 'Group 1', rgb(152, 171, 206),
      if(StateName() = 'Group 2', rgb(187, 200, 179),
            rgb(210, 210, 210)
      )
)
```

# 8.26 Tabelfuncties

De tabelfuncties retourneren informatie over de gegevenstabel die op dat moment wordt gelezen. Als er geen tabelnaam is opgegeven en de functie wordt gebruikt in een **LOAD**-opdracht, wordt de huidige tabelnaam verondersteld.

Alle functies kunnen worden gebruikt in het script voor het laden van gegevens, terwijl alleen **NoOfRows** kan worden gebruikt in een diagramuitdrukking.

## Overzicht van tabelfuncties

Sommige functies worden na het overzicht nader beschreven. Bij deze functies kunt u ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

### FieldName

De scriptfunctie **FieldName** retourneert de naam van het veld met het opgegeven nummer in een eerder geladen tabel. Als de functie wordt gebruikt in een **LOAD**-opdracht, mag deze niet verwijzen naar de tabel die op dat moment wordt geladen.

```
FieldName (field_number ,table_name)
```

### FieldNumber

De scriptfunctie **FieldNumber** retourneert het nummer van een opgegeven veld in een eerder geladen tabel. Als de functie wordt gebruikt in een **LOAD**-opdracht, mag deze niet verwijzen naar de tabel die op dat moment wordt geladen.

```
FieldNumber (field_name ,table_name)
```

### NoOfFields

De scriptfunctie **NoOfFields** retourneert het aantal velden in een eerder geladen tabel. Als de functie wordt gebruikt in een **LOAD**-opdracht, mag deze niet verwijzen naar de tabel die op dat moment wordt geladen.

```
NoOfFields (table_name)
```

### NoOfRows

De functie **NoOfRows** retourneert het aantal rijen (records) in een eerder geladen tabel. Als de functie wordt gebruikt in een **LOAD**-opdracht, mag deze niet verwijzen naar de tabel die op dat moment wordt geladen.

```
NoOfRows (table_name)
```

### NoOfTables

Deze scriptfunctie retourneert het aantal tabellen dat eerder is geladen.

```
NoOfTables()
```

### TableName

Deze scriptfunctie retourneert de naam van de tabel met het opgegeven nummer.

```
TableName(table_number)
```

### TableNumber

Deze scriptfunctie retourneert het nummer van de opgegeven tabel. De eerste tabel heeft nummer 0.

Als table_name niet bestaat, wordt NULL geretourneerd.

```
TableNumber(table_name)
```

### Voorbeeld:

In dit voorbeeld willen we een tabel maken met informatie over de tabellen en velden die zijn geladen.

Eerst gaan we een aantal steekproefgegevens laden. Hiermee worden de twee tabellen gemaakt die worden gebruikt om de tabelfuncties die in deze sectie worden beschreven te illustreren.

```
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 ;
```

Vervolgens itereren we door de tabellen die zijn geladen, via de functie **NoOfTables** en daarna door de velden van elke tabel, via de functie **NoOfFields**, en laden wij informatie met behulp van de tabelfuncties.

```
//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;
```

De resulterende tabel Tables ziet er als volgt uit:

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

De scriptfunctie **FieldName** retourneert de naam van het veld met het opgegeven nummer in een eerder geladen tabel. Als de functie wordt gebruikt in een **LOAD**-opdracht, mag deze niet verwijzen naar de tabel die op dat moment wordt geladen.

**Syntaxis:**

```
FieldName(field_number ,table_name)
```

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| field_number | Het veldnummer van het veld waarnaar u wilt verwijzen. |
| table_name | De tabel die het veld bevat waarnaar u wilt verwijzen. |

**Voorbeeld:**

```
LET a = FieldName(4,'tab1');
```

## FieldNumber

De scriptfunctie **FieldNumber** retourneert het nummer van een opgegeven veld in een eerder geladen tabel. Als de functie wordt gebruikt in een **LOAD**-opdracht, mag deze niet verwijzen naar de tabel die op dat moment wordt geladen.

**Syntaxis:**

```
FieldNumber(field_name ,table_name)
```

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| field_name | De naam van het veld. |
| table_name | De naam van de tabel die het veld bevat. |

Als het veld field_name niet bestaat in table_name of als table_name niet bestaat, retourneert de functie de waarde 0.

**Voorbeeld:**

```
LET a = FieldNumber('Customer','tab1');
```

## NoOfFields

De scriptfunctie **NoOfFields** retourneert het aantal velden in een eerder geladen tabel. Als de functie wordt gebruikt in een **LOAD**-opdracht, mag deze niet verwijzen naar de tabel die op dat moment wordt geladen.

**Syntaxis:**

```
NoOfFields(table_name)
```

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| table_name | De naam van de tabel. |

**Voorbeeld:**

```
LET a = NoOfFields('tab1');
```

## NoOfRows

De functie **NoOfRows** retourneert het aantal rijen (records) in een eerder geladen tabel. Als de functie wordt gebruikt in een **LOAD**-opdracht, mag deze niet verwijzen naar de tabel die op dat moment wordt geladen.

**Syntaxis:**

```
NoOfRows(table_name)
```

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| table_name | De naam van de tabel. |

**Voorbeeld:**

```
LET a = NoOfRows('tab1');
```

# 8.27 Trigonometrische en hyperbolische functies

In dit hoofdstuk worden functies beschreven voor trigonometrische en hyperbolische bewerkingen. In alle functies zijn de argumenten uitdrukkingen waarmee hoeken worden herleid die zijn gemeten in radialen, waarbij **x** moet worden geïnterpreteerd als een reëel getal.

Alle hoeken zijn weergegeven in radialen.

Alle functies kunnen zowel worden gebruikt in het script voor het laden van gegevens als in diagramuitdrukkingen.

**cos**

Cosinus van **x**. Het resultaat is een getal tussen -1 en 1.

```
cos( x )
```

### acos

Inverse cosinus van **x**. De functie is alleen gedefinieerd als -1≤ **x** ≤1. Het resultaat is een getal tussen 0 en π.

```
acos( x )
```

### sin

Sinus van **x**. Het resultaat is een getal tussen -1 en 1.

```
sin( x )
```

### asin

Inverse sinus van **x**. De functie is alleen gedefinieerd als -1≤ **x** ≤1. Het resultaat is een getal tussen -π/2 en π/2.

```
asin( x )
```

### tan

Tangens van **x**. Het resultaat is een reëel getal.

```
tan( x )
```

### atan

Inverse tangens van **x**. Het resultaat is een getal tussen -π/2 en π/2.

```
atan( x )
```

### atan2

Tweedimensionale generalisatie van de functie inverse tangens. Het resultaat is de hoek tussen de oorsprong en het punt dat wordt weergegeven door de coördinaten **x** en **y**. Het resultaat is een getal tussen - π en + π.

```
atan2( y,x )
```

### cosh

Hyperbolische cosinus van **x**. Het resultaat is een positief reëel getal.

```
cosh( x )
```

### sinh

Hyperbolische sinus van **x**. Het resultaat is een reëel getal.

```
sinh( x )
```

### tanh

Hyperbolische tangens van **x**. Het resultaat is een reëel getal.

```
tanh( x )
```

### acosh

Omgekeerde hyperbolische cosinus van **x**. Het resultaat is een positief reëel getal.

```
acosh( x )
```

**asinh**
Omgekeerde hyperbolische cosinus van **x**. Het resultaat is een reëel getal.

```
asinh( x )
```

**atanh**
Omgekeerde hyperbolische tangens van **x**. Het resultaat is een reëel getal.

```
atanh( x )
```

**Voorbeelden:**

Met de volgende scriptcode wordt een steekproeftabel geladen en vervolgens wordt een tabel geladen met de berekende trigonometrische en hyperbolische bewerkingen op de waarden.

```
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 Vensterfuncties

Vensterfuncties voeren berekeningen uit met waarden uit meerdere rijen en produceren een waarde per rij. Vensterfuncties kunnen uitsluitend worden berekend zodra de hele tabel is gelezen.

U kunt de vensterfuncties gebruiken om bewerkingen uit te voeren zoals:

- Een individuele nummerwaarde in een rij vergelijken met het gemiddelde, maximum of minimum in de kolom vergelijken.
- De positie van een individuele waarde berekenen, hetzij binnen de kolom of in de gehele tabel.

De vensterfuncties veranderen het aantal records in de tabel niet, maar kunnen wel vergelijke taken uitvoeren zoals aggregatiefuncties, relationele functies en bereikfuncties.

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

**Window**
De **Window** voert berekeningen uit op meerdere rijen en produceert een waarde per rij.

```
Window (main_expr, [partition1, partition2, ...], [sort_type, [sort_expr],
[filter_expr], [start_expr, end_expr]])
```

**WRank**
De **WRank**-functie voert rangschikkingsberekeningen uit in **Window**.

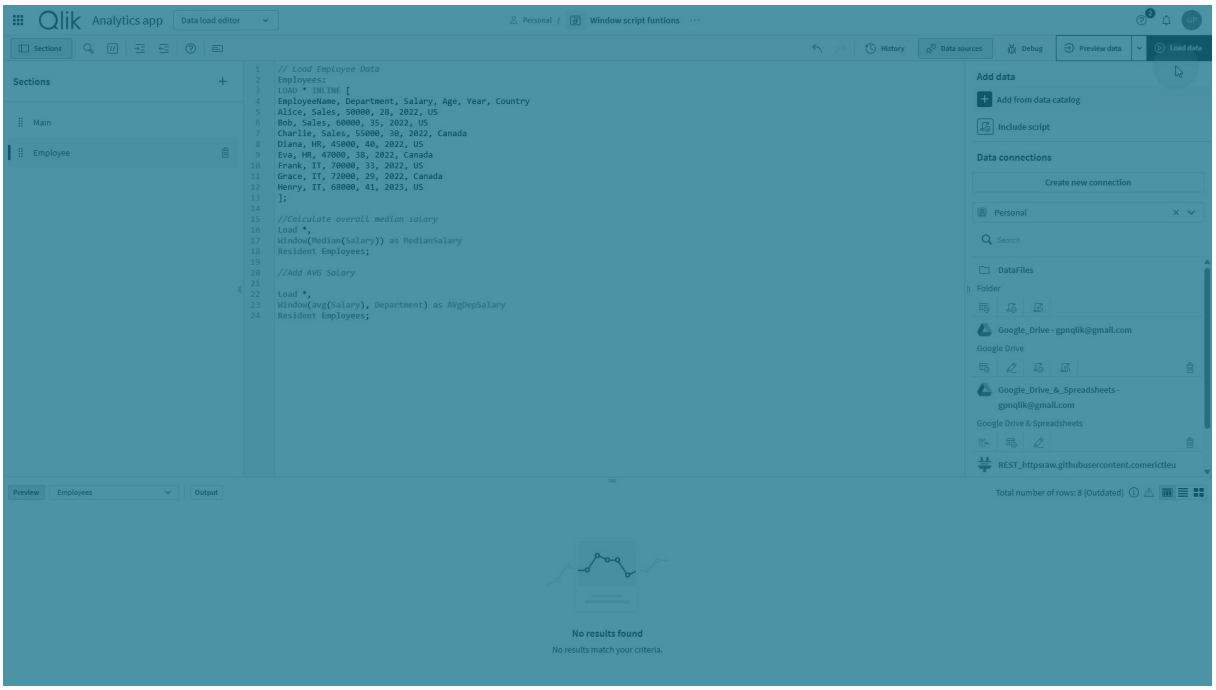
```
WRank ([mode[, fmt]])
```

## Window

**Window()** voert berekeningen uit op basis van meerdere rijen van een sectie (of "kader") van een volledige tabel, en produceert een enkele waarde voor elke rij.

U kunt de **Window**-functies gebruiken om bewerkingen uit te voeren zoals:

- Een individuele getalwaarde in een rij vergelijken met het gemiddelde, maximum of minimum binnen het kader van de tabel.
- De positie van een individuele waarde berekenen, hetzij binnen het kader of in de gehele tabel.



De **Window**-functie verandert het aantal records in de tabel niet, maar kan wel vergelijke taken uitvoeren zoals aggregatie-, relationele en bereikfuncties.

De functie **Window** moet worden opgenomen in de LOAD-opdracht van de tabel waarmee u werkt, of er moet naar de tabel worden verwezen, zoals met een Resident-LOAD. Zie bijvoorbeeld *Voorbeeld: een veld met een aggregatie toevoegen (page 1927)*.

De naam kan worden weggelaten, maar dit leidt ertoe dat de veldnaam wordt ingesteld op de volledige uitdrukking van de **Window** functie. Bijvoorbeeld *AvgDepSalary* in het onderstaande voorbeeld:

```
[AvgSalary]:
Load
       *,
       Window(avg(Salary), Department) as AvgDepSalary
Resident Employees;
```

*Het wordt aanbevolen om de functie **Window** een veldnaam te geven.*

**Window** ondersteunt algemene functies zoals afronding of basale numerieke bewerkingen. Bijvoorbeeld:

```
Load *, Round(Window(Sum(Salary),Department)) as SumDepSalary
Load *, Window(Sum(Salary),Department) + 5 as SumDepSalary
```

*De functie **Window** scant de volledige tabel. Met betrekking tot het geheugengebruik zijn de kosten van het scannen gelijk aan het aantal velden in de tabel. Een tabel met een paar velden, naast de velden waarop de **Window** functie werkt, verdient altijd de voorkeur boven een tabel met veel velden.*

*Over het algemeen verbruikt een **Window** functie 2-4 keer zoveel geheugen en duurt het 2-4 keer langer om opnieuw te laden in vergelijking met een vergelijkbare scriptbewerking die dezelfde aggregatie, een `Where` clausule (filter), een `Group by` clausule (partitie) en een `Order by` clausule (sorteren) bevat.*

**Syntaxis:**

```
Window (main_expr, [partition1, partition2, ...], [sort_type, [sort_expr],
[filter_expr], [start_expr, end_expr]])
```

Gegevenstype retourneren: één waarde per rij, die gecombineerd kan worden met of gebruikt kan worden in een andere uitdrukking, of direct toegevoegd kan worden aan de resulterende tabel die gemaakt is met de LOAD-instructie.

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| main_expr | De basis invoeruitdrukking die wordt berekend en geretourneerd door de functie. Het moet een uitdrukking zijn die is gebaseerd op een aggregatie, zoals `Median(Salary)`. Bijvoorbeeld:<br>`Window(Median(Salary)) as MedianSalary`<br>De uitdrukking kan ook een veldnaam zijn zonder dat aggregatie is toegepast. In dit geval behandelt **Window** het alsof de **Only()**-functie is toegepast op dat veld. Bijvoorbeeld:<br>`Window(Salary) as WSalary`<br>Uitdrukkingen die geneste aggregatiefuncties of geneste window-functies bevatten, zoals `Window(Sum(Avg(Salary)))` of `Window(Sum(Window(Salary)))`, zijn niet toegestaan. |
| partition 1, partition 2 | Na main_expr kunt u een willekeurig aantal partities definiëren. Partities zijn velden die definiëren met welke combinaties de aggregaties moeten worden toegepast. Bijvoorbeeld:<br>`Window(Avg(Salary), Department) as AvgDepSalary`<br>Dit berekent het gemiddelde salaris voor elke afdeling.<br>Als er meerdere partitieparameters worden opgegeven, wordt de aggregatie toegepast op elke rij met een unieke combinatie van partitievelden. Bijvoorbeeld:<br>`Window(Avg(Salary), Department, Country) as AvgDepSalaryPerCountry`<br>Dit berekent het gemiddelde salaris voor elke afdeling in elk land. Dubbele afdelingen in verschillende landen, zoals *Marketing* in *Zweden* en *Marketing* in *Canada*, worden als verschillende partities behandeld.<br>Partitionering is vergelijkbaar met de groepering die wordt bereikt met de **group by** clausule. Partitioneren vermindert het aantal records niet in vergelijking met de invoertabel. |

| Argument | Beschrijving |
|---|---|
| sort_type, [sort_expr] | U kunt optioneel het sorteertype en de sorteeruitdrukking specificeren. Dit bepaalt de volgorde waarin de aggregatie wordt uitgevoerd. sort_type is een constante tekenreeks en kan een van drie waarden hebben:<br>• ASC: oplopende sortering.<br>• DESC: aflopende sortering.<br>• NONE<br>Als u sort_type definieert als ASC of DESC, moet u een sorteeruitdrukking definiëren. Met sort_type NONE hoeft er geen sorteeruitdrukking worden opgegeven. De sorteeruitdrukking bepaalt de volgorde van de rijen in een partitie.<br>Bijvoorbeeld:<br>`Window(RecNo(), Department, 'ASC', Year)`<br>In het bovenstaande voorbeeld wordt de aggregratie in de partitie in oplopende volgorde uitgevoerd door het veld *Year*.<br>De sorteerparameters beïnvloeden ook de volgorde waarin een schuifvenster een tabel verwerkt. Zie bijvoorbeeld *Voorbeeld - een veld met een gesorteerd schuifvenster toevoegen (page 1930)*. |
| filter_expr | U kunt optioneel een filteruitdrukking toevoegen. Dit is een booleaanse uitdrukking die bepaalt of de record in de berekening moet worden opgenomen of niet.<br>Als filter_expr wordt opgegeven, moet er een sorteertype worden opgenomen. Als er geen sorteertype is gedefinieerd, dan kan het argument worden geïnterpreteerd als een partitieparameter in plaats van een filteruitdrukking. De parameter filter_expr kan volledig worden weggelaten of er kan een waarheidsgetrouwe waarde worden opgenomen. In beide gevallen worden er geen filters gebruikt, zodat er geen rijen worden uitgesloten.<br>Bijvoorbeeld:<br>`Window(avg(Salary), Department, 'ASC', Age, Country='US' Or Country='Canada') as AvgDepSalaryInUsOrCanada`<br>De bovenstaande **Window**-instructie zal elke rij wegfilteren waarvan het veld *Country* niet *US* of *Canada* is, waardoor die rijen worden uitgesloten van de aggregatie. Er zullen nog steeds uitvoerwaarden zijn voor de rijen die werden uitgefilterd, maar net als alle andere uitvoerrijen zullen deze alleen samengevoegde resultaten bevatten van rijen die niet door het filter zijn uitgezonderd. |

| Argument | Beschrijving |
|---|---|
| start_expr,end_expr | Stel de argumenten voor de schuifvensterfunctie in (optioneel). Dit beperkt het aantal rijen dat geaggregeerd moet worden nog verder. Elke uitvoerrij bevat alleen (geaggregeerde) resultaten van een aantal rijen vóór en na de huidige rij. Een schuifvenster vereist twee argumenten (start_expr,end_expr). Als deze argumenten worden gegeven, dan mag het sorteertype niet worden overgeslagen. Anders kunnen de argumenten worden geïnterpreteerd als een partitieparameter in plaats van een schuifvenster. De parameter start_expr,end_expr bestaat uit:<br>• Beginuitdrukking: het aantal rijen vóór de huidige rij om toe te voegen aan het venster.<br>• Einduitdrukking: het aantal rijen na de huidige rij om toe te voegen aan het venster.<br>Als u bijvoorbeeld 3 voorgaande rijen, de huidige rij en de volgende rij wilt toevoegen:<br>`Window(concat(Text(Salary),'-'), Department, 'NONE', 1, -3, 1) as WSlidingSalaryDepartment`<br>Om alle eerdere rijen of alle volgende rijen aan te geven kunt u de **Unbounded()**-functie gebruiken. Bijvoorbeeld: om alle voorgaande rijen, de huidige rij en de volgende rij toe te voegen:<br>`Window(concat(Text(Salary),'-'), Department, 'NONE', 1, UNBOUNDED(), 1) as WSlidingSalaryDepartment`<br>Bijvoorbeeld: om de derde rij uit de huidige rij en de daaropvolgende rijen toe te voegen:<br>`Window(concat(Text(Salary),'-'), Department, 'NONE', 1, 3, UNBOUNDED()) as WSlidingSalaryDepartment`<br>In de bovenstaande voorbeelden wordt het sorteertype NONE gegeven zodat het script onderscheid kan maken tussen partitie-argumenten en de parameters die na het sorteertype komen. |

## Voorbeeld: een veld met een aggregatie toevoegen

Voorbeeld: een veld met een aggregatie toevoegen

**Load-script**

Maak een nieuw tabblad in de editor voor laden van gegevens en laad de volgende gegevens als een inline load. Maak de onderstaande tabel in Qlik Sense om de resultaten te bekijken.

```
[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
];
```

#### Resultaten

Resultaten voor het toevoegen van een veld met een aggregatie

| 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 |

## Voorbeeld: een veld met een aggregatie toevoegen, gefilterd op specifieke waarden

Voorbeeld: een veld met een aggregatie toevoegen, gefilterd op specifieke waarden

#### Load-script

Maak een nieuw tabblad in de editor voor laden van gegevens en laad de volgende gegevens als een inline load. Maak de onderstaande tabel in Qlik Sense om de resultaten te bekijken.

```
[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
];
```

### Resultaten

Resultaten voor het toevoegen van een veld met een aggregatie, gefilterd op specifieke waarden

| 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 |

## Voorbeeld: een veld met een schuifvenster toevoegen

Voorbeeld: een veld met een schuifvenster toevoegen

### Load-script

Maak een nieuw tabblad in de editor voor laden van gegevens en laad de volgende gegevens als een inline load. Maak de onderstaande tabel in Qlik Sense om de resultaten te bekijken.

```
[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
];
```

### Resultaten

Resultaten voor het toevoegen van een veld met een aggregatie, gefilterd op specifieke waarden

| Name | Department | Salary | Age | Year | Country | SlidingWindow |
|---|---|---|---|---|---|---|
| David | Sales | 11 | 32 | 2019 | Canada | 11-22 |
| Sara | Sales | 22 | 25 | 2022 | Sweden | 11-22-55 |

| Name | Department | Salary | Age | Year | Country | SlidingWindow |
|---|---|---|---|---|---|---|
| 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 |

## Voorbeeld - een veld met een gesorteerd schuifvenster toevoegen

Voorbeeld: een veld met een schuifvenster toevoegen

### Load-script

Maak een nieuw tabblad in de editor voor laden van gegevens en laad de volgende gegevens als een inline load. Maak de onderstaande tabel in Qlik Sense om de resultaten te bekijken.

```
[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
];
```

### Resultaten

Resultaten voor het toevoegen van een veld met een aggregatie, gefilterd op specifieke waarden

| Name | Department | Salary | Age | Year | Country | SortedSlidingWindow |
|---|---|---|---|---|---|---|
| David | Sales | 11 | 32 | 2019 | Canada | 11-55-66 |
| 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 |

| Name | Department | Salary | Age | Year | Country | SortedSlidingWindow |
|---|---|---|---|---|---|---|
| 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 |

Om beter te begrijpen hoe de partities en sorteervolgorde de aggregaties in een schuifvenster beïnvloeden, sorteert u de tabel oplopend op naam.

Resultaten voor het toevoegen van een veld met een aggregatie, gefilterd op specifieke waarden

| 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 |

In de rij voor *Johan* heeft de aggregatie de salariswaarden voor *David* en *Sara* opgenomen. Dit zijn de voorgaande en volgende rijen binnen de partitie *Verkoopafdeling*, gesorteerd op leeftijd.

Voor een ander voorbeeld dat laat zien welke rijen zijn opgenomen, wijzigt u de uitdrukking **Windows** van `Window(concat(Text(Salary),'-'), Department, 'ASC', Age, -1, 1)` in `Window(concat(Text(Name),'-'), Department, 'ASC', Age, -1, 1)` en laadt u opnieuw. De waarden in SortedSlidingWindow in de onderstaande tabel geven de naam van de voorgaande en volgende rijen binnen de partitie *Verkoop*.

Resultaten voor het toevoegen van een veld met een aggregatie, gefilterd op specifieke waarden

| 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 |
| 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 |

## Beperkingen

**Window** heeft de volgende beperkingen:

- **Window** is een functie die veel resources gebruikt, vooral wat betreft geheugen.
- **Window** wordt niet ondersteund in Qlik Sense Mobile.
- Diagramuitdrukkingen bieden geen ondersteuning voor **Window**.
- U kunt **Window**-functies niet in andere **Window**-functies nesten.
- **Window** kan niet worden gebruikt in een aggregatiefunctie.
- **Window** kan niet worden gebruikt in een LOAD-instructie met een groeperen op-clausule.
- **Window** moet de hele tabel kunnen scannen.
- **WRank()**, **RecNo()** en **RowNo()** kunnen niet worden gebruikt met **Window** wanneer de schuifvensterfunctionaliteit wordt gebruikt.
- Wanneer de schuifvensterfunctionaliteit wordt gebruikt, neemt de berekeningstijd toe met de grootte van het venster. Grote tabellen in combinatie met grote vensters moeten over het algemeen worden vermeden.

# WRank

**WRank()** evalueert de rijen van een tabel in het load-script en geeft voor elke rij de relatieve positie van de waarde van het veld weer die in het load-script wordt geëvalueerd. Bij het evalueren van de tabel vergelijkt de functie het resultaat met het resultaat van de andere rijen die de huidige partitie bevatten en retourneert de rangschikking van de huidige rij binnen het segment.

*Partities in een tabel*

**WRank** kunnen alleen worden gebruikt in een **Window**-functie. De **Window**-functie moet een sorteertype en sorteeruitdrukking omvatten. De rangschikking wordt toegepast op de sorteeruitdrukking.

**Syntaxis:**

```
WRank ([mode[, fmt]])
```

**Retourgegevenstypen:** dubbele waarde

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| mode | Geeft de numerieke representatie aan van het resultaat van de functie (optioneel). |
| fmt | Geeft de tekstrepresentatie aan van het resultaat van de functie (optioneel). |

De rangorde wordt geretourneerd als een dubbele waarde. Als elke rij een unieke rangschikking heeft, is dit een geheel getal tussen 1 en het aantal rijen in de huidige partitie.

Als meerdere rijen dezelfde rangschikking hebben, kan de tekstuele en numerieke representatie worden bestuurd met de parameters **mode** en **fmt**.

**mode**

Het eerste argument, **mode**, kan de volgende waarden hebben:

**mode**-waarden

| Waarde | Beschrijving |
|---|---|
| 0 (standaard) | Als alle rangnummers in de delende groep lager zijn dan de middenwaarde van de volledige rangschikking, krijgen alle rijen het laagste positienummer in de delende groep.<br><br>Als alle rangnummers in de delende groep hoger zijn dan de middenwaarde van de volledige rangschikking, krijgen alle rijen het hoogste rangnummer in de delende groep.<br><br>Als rangnummers in de delende groep zowel lager als hoger zijn dan de middenwaarde van de volledige rangschikking, krijgen alle rijen de waarde die overeenkomt met het gemiddelde van het hoogste en laagste rangnummer in de volledige partitie. |
| 1 | Laagste rangnummer in alle rijen. |
| 2 | Gemiddelde rangnummer in alle rijen. |
| 3 | Hoogste rangnummer in alle rijen. |
| 4 | Laagste rangnummer in eerste rij, vervolgens opgehoogd met één voor elke rij. |

**fmt**

Het tweede argument, **fmt**, kan de volgende waarden hebben:

**fmt**-waarden

| Waarde | Beschrijving |
|---|---|
| 0 (standaard) | Lage waarde **-** hoge waarde in alle rijen (bijvoorbeeld 3 - 4). |
| 1 | Lage waarde in alle rijen. |
| 2 | Lage waarde in eerste rij, leeg in alle volgende rijen. |

De volgorde van rijen voor **mode** 4 en **fmt** 2 wordt bepaald door de laadvolgorde van de tabelvelden.

## Voorbeeld: een gerangschikt veld toevoegen

Voorbeeld: een gerangschikt veld toevoegen

### Load-script

Maak een nieuw tabblad in de editor voor laden van gegevens en laad de volgende gegevens als een inline load. Maak de onderstaande tabel in Qlik Sense om de resultaten te bekijken.

```
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
];
```

### Resultaten

Resultaten voor het toevoegen van een gerangschikt veld

| transaction_id | transaction_date | transaction_amount | transaction_quantity | customer_id | grootte | color_code | TransactionRanking |
|---|---|---|---|---|---|---|---|
| 3750 | 20180830 | 23.56 | 2 | 2038593 | L | Rood | 4-4 |
| 3751 | 20180907 | 556.31 | 6 | 203521 | M | Orange | 1-1 |
| 3752 | 20180916 | 5.75 | 1 | 5646471 | S | Blauw | 2-2 |
| 3754 | 20180922 | 484.21 | 13 | 049681 | XS | Rood | 1-1 |
| 3756 | 20180922 | 59.18 | 2 | 2038593 | M | Blauw | 3-3 |
| 3753 | 20180922 | 125.00 | 7 | 3036491 | L | Zwart | 1-1 |
| 3757 | 20180923 | 177.42 | 21 | 203521 | XL | Zwart | 3-3 |

| transactio n_id | transactio n_date | transactio n_amount | transactio n_quantity | custome r_id | grootte | color_ code | TransactionRan king |
|---|---|---|---|---|---|---|---|
| 3758 | 20180924 | 153.42 | 14 | 2038593 | L | Rood | 2-2 |
| 3759 | 20180925 | 7.42 | 5 | 203521 | M | Orang e | 4-4 |
| 3760 | 20180925 | 80.12 | 18 | 5646471 | M | Blauw | 1-1 |
| 3763 | 20180926 | 63.55 | 12 | 049681 | S | Rood | 2-2 |
| 3761 | 20180926 | 3.42 | 7 | 3036491 | XS | Zwart | 2-2 |
| 3764 | 20180927 | 325.95 | 8 | 203521 | XL | Zwart | 2-2 |
| 3763 | 20180927 | 177.56 | 10 | 2038593 | L | Blauw | 1-1 |

## Voorbeeld: een gerangschikt veld toevoegen met fmt voor een resultaat met één cijfer

Voorbeeld: een gerangschikt veld toevoegen met fmt voor een resultaat met één cijfer

### Load-script

Maak een nieuw tabblad in de editor voor laden van gegevens en laad de volgende gegevens als een inline load. Maak de onderstaande tabel in Qlik Sense om de resultaten te bekijken.

```
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
];
```

#### Resultaten

Resultaten voor het toevoegen van een gerangschikt veld met fmt voor een resultaat met één cijfer

| transaction_id | transaction_date | transaction_amount | transaction_quantity | customer_id | grootte | color_code | TransactionRanking |
|---|---|---|---|---|---|---|---|
| 3750 | 20180830 | 23.56 | 2 | 2038593 | L | Rood | 4 |
| 3751 | 20180907 | 556.31 | 6 | 203521 | M | Orange | 1 |
| 3752 | 20180916 | 5.75 | 1 | 5646471 | S | Blauw | 2 |
| 3754 | 20180922 | 484.21 | 13 | 049681 | XS | Rood | 1 |
| 3756 | 20180922 | 59.18 | 2 | 2038593 | M | Blauw | 3 |
| 3753 | 20180922 | 125.00 | 7 | 3036491 | L | Zwart | 1 |
| 3757 | 20180923 | 177.42 | 21 | 203521 | XL | Zwart | 3 |
| 3758 | 20180924 | 153.42 | 14 | 2038593 | L | Rood | 2 |
| 3759 | 20180925 | 7.42 | 5 | 203521 | M | Orange | 4 |
| 3760 | 20180925 | 80.12 | 18 | 5646471 | M | Blauw | 1 |
| 3763 | 20180926 | 63.55 | 12 | 049681 | S | Rood | 2 |
| 3761 | 20180926 | 3.42 | 7 | 3036491 | XS | Zwart | 2 |
| 3764 | 20180927 | 325.95 | 8 | 203521 | XL | Zwart | 2 |
| 3763 | 20180927 | 177.56 | 10 | 2038593 | L | Blauw | 1 |

## Voorbeeld: een gerangschikt veld met meerdere partities

Voorbeeld: een gerangschikt veld met meerdere partities toevoegen

#### Load-script

Maak een nieuw tabblad in de editor voor laden van gegevens en laad de volgende gegevens als een inline load. Maak de onderstaande tabel in Qlik Sense om de resultaten te bekijken.

```
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
];
```

### Resultaten

Resultaten voor het toevoegen van een gerangschikt veld met fmt voor een resultaat met één cijfer

| transaction_id | transaction_date | transaction_amount | transaction_quantity | customer_id | grootte | color_code | TransactionRanking |
|---|---|---|---|---|---|---|---|
| 3750 | 20180830 | 23.56 | 2 | 2038593 | L | Rood | 2 |
| 3751 | 20180907 | 556.31 | 6 | 203521 | M | Oranje | 1 |
| 3752 | 20180916 | 5.75 | 1 | 5646471 | S | Blauw | 1 |
| 3754 | 20180922 | 484.21 | 13 | 049681 | XS | Rood | 1 |
| 3756 | 20180922 | 59.18 | 2 | 2038593 | M | Blauw | 1 |
| 3753 | 20180922 | 125.00 | 7 | 3036491 | L | Zwart | 1 |
| 3757 | 20180923 | 177.42 | 21 | 203521 | XL | Zwart | 2 |
| 3758 | 20180924 | 153.42 | 14 | 2038593 | L | Rood | 1 |
| 3759 | 20180925 | 7.42 | 5 | 203521 | M | Oranje | 2 |
| 3760 | 20180925 | 80.12 | 18 | 5646471 | M | Blauw | 1 |
| 3763 | 20180926 | 63.55 | 12 | 049681 | S | Rood | 1 |
| 3761 | 20180926 | 3.42 | 7 | 3036491 | XS | Zwart | 1 |
| 3764 | 20180927 | 325.95 | 8 | 203521 | XL | Zwart | 1 |
| 3763 | 20180927 | 177.56 | 10 | 2038593 | L | Blauw | 1 |

## Beperkingen

WRank heeft de volgende beperkingen:

- Als uw fmt-waarde 0 is en u het tekstgedeelte van de dubbele resultaat wilt gebruiken voor **WRank**, moet u **Text()** met **Window(WRank)** gebruiken. Bijvoorbeeld: `Text(Window(WRank(0), Unit, 'DESC', Age)) as UnitWRankedByAgeText`.

# 9 Beperking van toegang tot bestandssysteem

Om beveiligingsredenen biedt Qlik Sense in de standaardmodus geen ondersteuning voor paden in het load-script voor gegevens of in functies en variabelen waarmee het bestandssysteem beschikbaar wordt gesteld.

Aangezien bestandssysteempaden echter werden ondersteund in QlikView 12, is het mogelijk om de standaardmodus uit te schakelen en de bestaande modus te gebruiken ten behoeve van hergebruik van QlikView 12 load-scripts.

*Bij het uitschakelen van de standaardmodus kan een beveiligingsrisico ontstaan doordat het bestandssysteem wordt opengesteld.*

*Standaardmodus uitschakelen (page 1945)*

## 9.1 Beveiligingsaspecten bij verbinding met op bestanden gebaseerde ODBC- en OLE DB-gegevensverbindingen

ODBC- en OLE DB-gegevensverbindingen die gebruikmaken van op bestanden gebaseerde stuurprogramma's geven het pad naar het verbonden gegevensbestand in de verbindingstekenreeks aan. Het pad kan worden weergegeven als de verbinding wordt bewerkt, in het dialoogvenster voor gegevensselectie of in bepaalde SQL-query's. Dit is het geval in zowel de standaardmodus als de bestaande modus.

*Als het weergeven van het pad in het gegevensbestand een punt van zorg is, wordt aanbevolen verbinding te maken met het gegevensbestand via een mapgegevensverbinding als dit mogelijk is.*

## 9.2 Beperkingen in standaardmodus

Verschillende opdrachten, variabelen en functies kunnen niet worden gebruikt of hebben beperkingen in de standaardmodus. Het gebruik van niet-ondersteunde opdrachten in het load-script voor gegevens levert een fout op bij het uitvoeren van het load-script. Foutberichten zijn te vinden in het logbestand van het script. Het gebruik van niet-ondersteunde variabelen en functies resulteert niet in foutberichten of vermeldingen in het logbestand. In plaats daarvan geeft de functie NULL als resultaat.

Er is geen indicatie dat een variabele, opdracht of functie niet wordt ondersteund bij het bewerken van het load-script voor gegevens.

## Systeemvariabelen

Systeemvariabelen

| Variabele | Standaardmodus | Bestaande modus | Definitie |
|---|---|---|---|
| Floppy | Niet ondersteund | Ondersteund | Retourneert de stationsletter van het eerst gevonden diskettestation. Dit is normaal gesproken *a:*. |
| CD | Niet ondersteund | Ondersteund | Retourneert de stationsletter van het eerst gevonden cd-rom-station. Als er geen cd-rom-station is gevonden, wordt *c:* geretourneerd. |
| QvPath | Niet ondersteund | Ondersteund | Retourneert het pad naar het Qlik Sense-programma. |
| QvRoot | Niet ondersteund | Ondersteund | Retourneert de hoofdmap van het Qlik Sense-programma. |
| QvWorkPath | Niet ondersteund | Ondersteund | Retourneert het pad naar de huidige Qlik Sense-app. |
| QvWorkRoot | Niet ondersteund | Ondersteund | Retourneert de hoofdmap van de huidige Qlik Sense-app. |
| WinPath | Niet ondersteund | Ondersteund | Retourneert het pad naar Windows. |
| WinRoot | Niet ondersteund | Ondersteund | Retourneert de hoofdmap van Windows. |

| Variabele | Standaardmodus | Bestaande modus | Definitie |
|---|---|---|---|
| $(include=...) | Ondersteunde invoer: Pad dat bibliotheekverbinding gebruikt | Ondersteunde invoer: Pad dat bibliotheekverbinding of bestandssysteem gebruikt | De variabele **Include/Must_Include** specificeert een bestand dat tekst bevat die in het script moet worden opgenomen en moet worden geëvalueerd als scriptcode. Deze wordt niet gebruikt om gegevens toe te voegen. U kunt delen van uw scriptcode opslaan in een apart tekstbestand en dit in meerdere apps hergebruiken. Dit is een door de gebruiker gedefinieerde variabele. |

## Reguliere scriptopdrachten

Reguliere scriptopdrachten

| Opdracht | Standaardmodus | Bestaande modus | Definitie |
|---|---|---|---|
| Binary | Ondersteunde invoer: Pad dat bibliotheekverbinding gebruikt | Ondersteunde invoer: Pad dat bibliotheekverbinding of bestandssysteem gebruikt | De opdracht **binary** wordt gebruikt voor het laden van gegevens vanuit een andere app. |
| Connect | Ondersteunde invoer: Pad dat bibliotheekverbinding gebruikt | Ondersteunde invoer: Pad dat bibliotheekverbinding of bestandssysteem gebruikt | Met de opdracht **CONNECT** wordt de toegang van Qlik Sense tot een algemene database via de OLE DB/ODBC-interface vastgelegd. Voor ODBC moet de gegevensbron eerst worden opgegeven met behulp van ODBC-beheer. |

| Opdracht | Standaardmodus | Bestaande modus | Definitie |
|---|---|---|---|
| Directory | Ondersteunde invoer: Pad dat bibliotheekverbinding gebruikt | Ondersteunde invoer: Pad dat bibliotheekverbinding of bestandssysteem gebruikt | De opdracht **Directory** legt vast in welke directory moet worden gezocht naar gegevensbestanden in volgende **LOAD**-opdrachten, totdat een nieuwe **Directory**opdracht wordt gegeven. |
| Execute | Niet ondersteund | Ondersteunde invoer: Pad dat bibliotheekverbinding of bestandssysteem gebruikt | De **Execute**-opdracht wordt gebruikt voor het uitvoeren van andere programma's terwijl gegevens worden geladen in Qlik Sense. Bijvoorbeeld voor het uitvoeren van noodzakelijke conversies. |
| LOAD from ... | Ondersteunde invoer: Pad dat bibliotheekverbinding gebruikt | Ondersteunde invoer: Pad dat bibliotheekverbinding of bestandssysteem gebruikt | Met de opdracht **LOAD** worden velden geladen uit een bestand, uit gegevens die in het script zijn gedefinieerd, uit een eerder geladen tabel, van een webpagina, uit het resultaat van een daaropvolgende **SELECT**-opdracht of door gegevens automatisch te genereren. |
| Store into ... | Ondersteunde invoer: Pad dat bibliotheekverbinding gebruikt | Ondersteunde invoer: Pad dat bibliotheekverbinding of bestandssysteem gebruikt | De **Store**-instructie genereert een QVD-, Parquet-, CSV- of TXT-bestand. |

## Scriptbesturingsopdrachten

Scriptbesturingsopdrachten

| Opdracht | Standaardmodus | Bestaande modus | Definitie |
|---|---|---|---|
| For each...<br><br>filelist mask/dirlist mask | Ondersteunde invoer: Pad dat bibliotheekverbinding gebruikt<br><br>Geretourneerde output: bibliotheekverbinding | Ondersteunde invoer: Pad dat bibliotheekverbinding of bestandssysteem gebruikt<br><br>Geretourneerde output: Bibliotheekverbinding of bestandssysteempad, afhankelijk van invoer | De syntaxis filelist mask produceert een lijst met door komma's gescheiden namen van alle bestanden in de huidige map die overeenkomen met het **filelist mask**. De syntaxis **dirlist mask** produceert een lijst met door komma's gescheiden namen van alle mappen in de huidige map die overeenkomen met het mapnaammasker. |

## Bestandsfuncties

Bestandsfuncties

| Functie | Standaardmodus | Bestaande modus | Definitie |
|---|---|---|---|
| Attribute() | Ondersteunde invoer: Pad dat bibliotheekverbinding gebruikt | Ondersteunde invoer: Pad dat bibliotheekverbinding of bestandssysteem gebruikt | Retourneert de waarde van de metatags van verschillende mediabestanden als tekst. |
| ConnectString() | Geretourneerde uitvoer: naam van bibliotheekverbinding | De naam van de bibliotheekverbinding of daadwerkelijke verbinding, afhankelijk van invoer | Retourneert de actieve verbindingstekenreeks voor ODBC- of OLE DB-verbindingen. |
| FileDir() | Geretourneerde output: bibliotheekverbinding | Geretourneerde output: Bibliotheekverbinding of bestandssysteempad, afhankelijk van invoer | De functie **FileDir** retourneert een tekenreeks met het pad naar de map van het tabelbestand dat op dat moment wordt gelezen. |

| Functie | Standaardmodus | Bestaande modus | Definitie |
|---|---|---|---|
| FilePath() | Geretourneerde output: bibliotheekverbinding | Geretourneerde output: Bibliotheekverbinding of bestandssysteempad, afhankelijk van invoer | De functie **FilePath** retourneert een tekenreeks met het volledige pad naar het tabelbestand dat op dat moment wordt gelezen. |
| FileSize() | Ondersteunde invoer: Pad dat bibliotheekverbinding gebruikt | Ondersteunde invoer: Pad dat bibliotheekverbinding of bestandssysteem gebruikt | De functie retourneert een geheel getal met de grootte in bytes van het bestand of, als geen is opgegeven, van het tabelbestand dat op dat moment wordt gelezen. filenamefilename |
| FileTime() | Ondersteunde invoer: Pad dat bibliotheekverbinding gebruikt | Ondersteunde invoer: Pad dat bibliotheekverbinding of bestandssysteem gebruikt | De functie **FileTime** geeft een tijdstempel in UTC-indeling van de laatste wijziging van een opgegeven bestand. Als er geen bestand is opgegeven, geeft de functie een tijdstempel in UTC-indeling van de laatste wijziging van het momenteel gelezen tabelbestand. |
| GetFolderPath() | Niet ondersteund | Geretourneerde output: absoluut pad | De functie **GetFolderPath** retourneert de waarde van de functie Microsoft Windows *SHGetFolderPath*. Deze functie neemt de naam van een map in Microsoft Windows als invoer en retourneert het volledige pad van de map. |

| Functie | Standaardmodus | Bestaande modus | Definitie |
|---|---|---|---|
| QvdCreateTime() | Ondersteunde invoer: Pad dat bibliotheekverbinding gebruikt | Ondersteunde invoer: Pad dat bibliotheekverbinding of bestandssysteem gebruikt | Deze scriptfunctie retourneert de tijdstempel voor de XML-koptekst uit een QVD-bestand, indien aanwezig. Anders wordt NULL geretourneerd. In de tijdstempel wordt tijd weergegeven in UTC. |
| QvdFieldName() | Ondersteunde invoer: Pad dat bibliotheekverbinding gebruikt | Ondersteunde invoer: Pad dat bibliotheekverbinding of bestandssysteem gebruikt | Deze script-functie retourneert de naam van veldnummer **fieldno** in een QVD-bestand. Als het veld niet bestaat, wordt NULL geretourneerd. |
| QvdNoOfFields() | Ondersteunde invoer: Pad dat bibliotheekverbinding gebruikt | Ondersteunde invoer: Pad dat bibliotheekverbinding of bestandssysteem gebruikt | Deze scriptfunctie retourneert het aantal velden in een QVD-bestand. |
| QvdNoOfRecords() | Ondersteunde invoer: Pad dat bibliotheekverbinding gebruikt | Ondersteunde invoer: Pad dat bibliotheekverbinding of bestandssysteem gebruikt | Deze scriptfunctie retourneert het huidige aantal records in een QVD-bestand. |
| QvdTableName() | Ondersteunde invoer: Pad dat bibliotheekverbinding gebruikt | Ondersteunde invoer: Pad dat bibliotheekverbinding of bestandssysteem gebruikt | Deze scriptfunctie retourneert de naam van de tabel die is opgeslagen in een QVD-bestand. |

## Systeemfuncties

Systeemfuncties

| Functie | Standaardmodus | Bestaande modus | Definitie |
|---|---|---|---|
| DocumentPath() | Niet ondersteund | Geretourneerde output: absoluut pad | Deze functie retourneert een tekenreeks met het volledige pad naar de huidige Qlik Sense-app. |

## 9.3 Standaardmodus uitschakelen

U kunt de standaardmodus uitschakelen, met andere woorden de bestaande modus instellen, om QlikView 12 load-scripts die verwijzen naar absolute of relatieve bestandspaden en bibliotheekverbindingen opnieuw te gebruiken.

*Bij het uitschakelen van de standaardmodus kan een beveiligingsrisico ontstaan doordat het bestandssysteem wordt opengesteld.*

### Qlik Sense

Voor Qlik Sense, kan de standaardmodus worden uitgeschakeld in QMC met de eigenschap **Standaardmodus**.

### Qlik Sense Desktop

In Qlik Sense Desktop kunt u de standaard-/bestaande modus instellen in *Settings.ini*.

Als u Qlik Sense Desktop in de standaardinstallatielocatie hebt geïnstalleerd, bevindt *Settings.ini* zich in *C:\Users\{user}\Documents\Qlik\Sense\Settings.ini*. Als u Qlik Sense Desktop hebt geïnstalleerd in een door u geselecteerde map, bevindt *Settings.ini* zich in de map *Engine* in het installatiepad.

**Doe het volgende:**

1. Open *Settings.ini* in een teksteditor.
2. Wijzig *StandardReload=1* in *StandardReload=0*.
3. Sla het bestand op en start Qlik Sense Desktop.

Qlik Sense Desktop wordt nu in de bestaande modus uitgevoerd.

#### Instellingen

De beschikbare instellingen voor StandardReload zijn:

- 1 (standaardmodus)
- 0 (bestaande modus)

# 10 Scripts op diagramniveau

Wanneer u diagramgegevens aanpast, gebruikt u een subset van het Qlik Sense script dat uit verschillende opdrachten bestaat. Een opdracht kan een normale scriptopdracht of een scriptbesturingsopdracht zijn. Bepaalde opdrachten kunnen worden voorafgegaan door prefixen.

Normale opdrachten worden over het algemeen gebruikt om gegevens op een bepaalde manier te manipuleren. Deze opdrachten kunnen worden geschreven op een willekeurig aantal regels in het script en moeten altijd worden afgesloten met een puntkomma (;).

Besturingsopdrachten worden gebruikt om het verloop van de scriptuitvoering te besturen. Elke zin van een besturingsopdracht moet op één scriptregel blijven en kan worden afgesloten met een puntkomma of een regeleinde.

Prefixen kunnen worden toegevoegd aan normale opdrachten, maar nooit aan besturingsopdrachten.

Bij het typen van sleutelwoorden in het script kunt u elke combinatie van hoofdletters en kleine letters gebruiken. Gebruikte namen van velden en variabelen in de opdrachten zijn echter hoofdlettergevoelig.

In deze sectie vindt u een alfabetische lijst met alle scriptopdrachten, besturingsopdrachten en prefixen die beschikbaar zijn in de subset van het script dat wordt gebruikt tijdens het aanpassen van diagramgegevens.

## 10.1 Scriptbesturingsopdrachten

Wanneer u diagramgegevens aanpast, gebruikt u een subset van het Qlik Sense script dat uit verschillende opdrachten bestaat. Een opdracht kan een normale scriptopdracht of een scriptbesturingsopdracht zijn.

Besturingsopdrachten worden gebruikt om het verloop van de scriptuitvoering te besturen. Elke zin van een besturingsopdracht moet op één scriptregel blijven en kan worden afgesloten met een puntkomma of een regeleinde.

Prefixen worden nooit toegepast op besturingsopdrachten.

Bij het typen van sleutelwoorden in het script kunt u elke combinatie van hoofdletters en kleine letters gebruiken.

### Overzicht besturingsopdracht diagramaanpassing

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

#### Call

De besturingsopdracht **call** roept een subroutine aan die door een vorige **sub**-opdracht is gedefinieerd.

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

### Do..loop

De besturingsopdracht **do..loop** is een herhalingsconstructie voor gebruik in scripts waarmee één of meer opdrachten worden uitgevoerd totdat aan een logische voorwaarde wordt voldaan.

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

### End

Het sleutelwoord **End** voor scripts wordt gebruikt om **If**-, **Sub**- en **Switch**-clausules af te sluiten.

### Exit

Het sleutelwoord **Exit** voor scripts maakt onderdeel uit van de **Exit Script**-opdracht, maar kan ook worden gebruikt voor het afsluiten van **Do**-, **For**- of **Sub**-clausules.

### Exit script

Met deze besturingsopdracht wordt de uitvoering van het script stopgezet. Deze opdracht kan overal in het script worden ingevoegd.

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

### For..next

De besturingsopdracht **for..next** is een herhalingsconstructie voor gebruik in scripts met een teller. De opdrachten in de lus tussen **for** en **next** worden uitgevoerd voor elke waarde van de tellervariabele tussen de opgegeven onder- en bovengrens.

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

[statements]

[exit for [ ( when | unless ) condition ]

[statements]

Next [counter]
```

### For each ..next

De besturingsopdracht **for each..next** is een herhalingsconstructie voor gebruik in scripts waarmee een of meer opdrachten worden uitgevoerd voor elke waarde in een door komma's gescheiden lijst. De opdrachten in de lus tussen **for** en **next** worden uitgevoerd voor elke waarde van de lijst.

```
For each..next var in list

[statements]

[exit for [ ( when | unless ) condition ]

[statements]

next [var]
```

### If..then

De besturingsopdracht **if..then** is een constructie voor scriptselectie waarmee de uitvoering van het script, afhankelijk van één of meer logische voorwaarden, op verschillende manieren kan verlopen.

> *Omdat de instructie **if..then** een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag geen van de vier mogelijke clausules (**if..then**, **elseif..then**, **else** en **end if**) de regelgrens overschrijden.*

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

  [ statements ]

{ elseif condition then

  [ statements ] }

[ else

  [ statements ] ]

end if
```

### Next

Het sleutelwoord **Next** voor scripts wordt gebruikt om**For**-lussen af te sluiten.

### Sub

De besturingsopdracht **sub..end sub** geeft een subroutine aan die vanuit een **call**-opdracht kan worden aangeroepen.

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

### Switch

De besturingsopdracht **switch** is een constructie voor scriptselectie waarmee de uitvoering van het script, afhankelijk van de waarde van een uitdrukking, op verschillende manieren kan verlopen.

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

### To

Het sleutelwoord **To** voor scripts wordt gebruikt in verschillende scriptopdrachten.

## Call

De besturingsopdracht **call** roept een subroutine aan die door een vorige **sub**-opdracht is gedefinieerd.

**Syntaxis:**

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

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| name | De naam van de subroutine. |
| paramlist | Een door komma's gescheiden lijst met de werkelijke parameters die worden gezonden aan de subroutine. Elk item in de lijst kan een veldnaam, een variabele of een willekeurige uitdrukking zijn. |

De subroutine die wordt opgeroepen door een **call**-opdracht moet op een eerdere positie in het script door een **sub** zijn gedefinieerd.

Parameters worden gekopieerd naar de subroutine en worden, als de parameter in de **call**-instructie een variabele is en geen uitdrukking, bij het beëindigen van de subroutine weer teruggekopieerd.

**Beperkingen:**

- Aangezien de **call**-opdracht een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag de opdracht niet op meerdere regels staan.
- Als u een subroutine definieert met `Sub..end sub` binnen een besturingsopdracht, bijvoorbeeld `if..then`, kunt u de subroutine alleen oproepen binnen dezelfde besturingsopdracht.

## Do..lus

De besturingsopdracht **do..loop** is een herhalingsconstructie voor gebruik in scripts waarmee één of meer opdrachten worden uitgevoerd totdat aan een logische voorwaarde wordt voldaan.

**Syntaxis:**

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

*Omdat de instructie **do..loop** een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag geen van de drie mogelijke clausules ervan (**do**, **exit do** en **loop**) de regelgrens overschrijden.*

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| condition | Een logische uitdrukking die resulteert in True of False. |
| statements | Een willekeurige groep van een of meer Qlik Sense-scriptopdrachten. |
| while / until | De voorwaardelijke clausule **while** of **until** mag slechts eenmaal voorkomen in elke **do..loop**-opdracht, namelijk na **do** of na **loop**. Elke voorwaarde wordt alleen de eerste keer geïnterpreteerd wanneer deze wordt aangetroffen, maar wordt telkens in de lus geëvalueerd. |
| exit do | Als een clausule **exit do** binnen de lus wordt aangetroffen, gaat de uitvoering van het script over op de eerste opdracht na de clausule **loop** die het einde van de lus aangeeft. De clausule **exit do** kan voorwaardelijk worden gemaakt met het optionele suffix **when** of **unless**. |

## End

Het sleutelwoord **End** voor scripts wordt gebruikt om **If**-, **Sub**- en **Switch**-clausules af te sluiten.

## Exit

Het sleutelwoord **Exit** voor scripts maakt onderdeel uit van de **Exit Script**-opdracht, maar kan ook worden gebruikt voor het afsluiten van **Do**-, **For**- of **Sub**-clausules.

## Exit script

Met deze besturingsopdracht wordt de uitvoering van het script stopgezet. Deze opdracht kan overal in het script worden ingevoegd.

**Syntaxis:**

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

Aangezien de **exit script**-opdracht een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag de opdracht niet op meerdere regels staan.

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| condition | Een logische uitdrukking die resulteert in True of False. |

| Argument | Beschrijving |
|---|---|
| when / unless | De opdracht **exit script** kan voorwaardelijk worden gemaakt met de optionele clausule **when** of **unless**. |

**Voorbeelden:**

```
//Exit script
Exit Script;

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

## For..next

De besturingsopdracht **for..next** is een herhalingsconstructie voor gebruik in scripts met een teller. De opdrachten in de lus tussen **for** en **next** worden uitgevoerd voor elke waarde van de tellervariabele tussen de opgegeven onder- en bovengrens.

**Syntaxis:**

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

[statements]

[exit for [ ( when | unless ) condition ]

[statements]

Next [counter]
```

De uitdrukkingen *expr1*, *expr2* en *expr3* worden alleen de eerste keer dat de lus wordt ingevoerd geëvalueerd. De waarde van de tellervariabele kan worden gewijzigd door opdrachten binnen de lus. Dit is echter met het oog op de gebruikelijke manier van programmeren niet aan te raden.

Als een clausule **exit for** binnen de lus wordt aangetroffen, gaat de uitvoering van het script over op de eerste opdracht na de clausule **next** die het einde van de lus aangeeft. De clausule **exit for** kan voorwaardelijk worden gemaakt met het optionele suffix **when** of **unless**.

*Omdat de instructie **for..next** een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag geen van de drie mogelijke clausules ervan (**for..to..step**, **exit for** en **next**) de regelgrens overschrijden.*

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| counter | De naam van een variabele. Als *counter* wordt opgegeven na **next** moet de variabele dezelfde naam hebben als de variabele die wordt gevonden na de overeenkomende **for**. |
| expr1 | Een uitdrukking die de eerste waarde van de variabele *counter* bepaalt waarvoor de lus wordt uitgevoerd. |
| expr2 | Een uitdrukking die de laatste waarde van de variabele *counter* bepaalt waarvoor de lus wordt uitgevoerd. |
| expr3 | Een uitdrukking die de waarde bepaalt van de stap waarmee de variabele *counter* toeneemt bij elke uitvoering van de lus. |
| condition | Een logische uitdrukking die resulteert in True of False. |
| statements | Een willekeurige groep van een of meer Qlik Sense-scriptopdrachten. |

## For each..next

De besturingsopdracht **for each..next** is een herhalingsconstructie voor gebruik in scripts waarmee een of meer opdrachten worden uitgevoerd voor elke waarde in een door komma's gescheiden lijst. De opdrachten in de lus tussen **for** en **next** worden uitgevoerd voor elke waarde van de lijst.

**Syntaxis:**

Door een speciale syntaxis kunnen lijsten worden gegenereerd met namen van bestanden en mappen in de huidige map.

```
for each var in list

[statements]

[exit for [ ( when | unless ) condition ]

[statements]

next [var]
```

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| var | De naam van een scriptvariabele die een nieuwe waarde uit de lijst ophaalt, telkens wanneer de lus wordt uitgevoerd. Als **var** wordt opgegeven na **next** moet de variabele dezelfde naam hebben als de variabele die wordt gevonden na de overeenkomende **for each**. |

De waarde van de **var** kan worden gewijzigd door opdrachten binnen de lus. Dit is echter met het oog op de gebruikelijke manier van programmeren niet aan te raden.

Als een clausule **exit for** binnen de lus wordt aangetroffen, gaat de uitvoering van het script over op de eerste opdracht na de clausule **next** die het einde van de lus aangeeft. De clausule **exit for** kan voorwaardelijk worden gemaakt met het optionele suffix **when** of **unless**.

*Omdat de instructie **for each..next** een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag geen van de drie mogelijke clausules ervan (**for each**, **exit for** en **next**) de regelgrens overschrijden.*

**Syntaxis:**

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

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

Argumenten

| Argument | Beschrijving |
|---|---|
| constant | Elk getal of elke tekenreeks. Een tekenreeks die rechtstreeks in het script wordt opgenomen moet tussen enkele aanhalingstekens worden geplaatst. Een tekenreeks zonder enkele aanhalingstekens wordt geïnterpreteerd als een variabele en vervolgens wordt de waarde van de variabele gebruikt. Getallen hoeven niet tussen enkele aanhalingstekens te worden geplaatst. |
| expression | Een willekeurige uitdrukking. |
| mask | Een masker voor de naam van een bestand of map die geldige tekens voor bestandsnamen plus de standaardjokertekens, * en ?, kan bevatten.<br><br>U kunt absolute bestandspaden of paden lib:// gebruiken. |
| condition | Een logische uitdrukking die resulteert in True of False. |
| statements | Een willekeurige groep van een of meer Qlik Sense-scriptopdrachten. |

| Argument | Beschrijving |
|---|---|
| filelist mask | Deze syntaxis produceert een lijst met door komma's gescheiden namen van alle bestanden in de huidige map die overeenkomen met het bestandsnaammasker.<br><br>*Dit argument ondersteunt alleen bibliotheekverbindingen in de standaardmodus.* |
| dirlist mask | Deze syntaxis produceert een lijst met door komma's gescheiden namen van alle mappen in de huidige map die overeenkomen met het mapnaammasker.<br><br>*Dit argument ondersteunt alleen bibliotheekverbindingen in de standaardmodus.* |
| fieldvaluelist mask | Deze syntaxis itereert door de waarden van een veld die al in Qlik Sense zijn geladen. |

*De Qlik Connectoren van internetopslagprovider en andere DataFiles-verbindingen bieden geen ondersteuning voor filtermaskeringen die jokertekens (***** en **?**) gebruiken.*

### Example 1: Een lijst van bestanden laden

```
// 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: Een lijst van bestanden maken op schijf

Via dit voorbeeld wordt een lijst met alle aan Qlik Sense gerelateerde bestanden in een map geladen.

```
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: Itereren door de waarden van een veld**

Dit voorbeeld itereert door de waarden van geladen waarden van FIELD en genereert een nieuw veld, NEWFIELD. Voor elke waarde van FIELD, worden twee NEWFIELD-records gemaakt.

```
load * inline [
FIELD
one
two
three
];

FOR Each a in FieldValueList('FIELD')
LOAD '$(a)' &'-'&RecNo() as NEWFIELD AutoGenerate 2;
NEXT a
```

De resulterende tabel ziet er als volgt uit:

Example table

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

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

De besturingsopdracht **if..then** is een constructie voor scriptselectie waarmee de uitvoering van het script, afhankelijk van één of meer logische voorwaarden, op verschillende manieren kan verlopen.

Besturingsopdrachten worden gebruikt om het verloop van de scriptuitvoering te besturen. Gebruik in een diagramuitdrukking in plaats daarvan de **if** voorwaardelijke functie.

**Syntaxis:**

```
If  condition then

  [ statements ]

{ elseif condition then
```

```
  [ statements ] }

[ else

  [ statements ] ]

end if
```

Omdat de instructie **if..then** een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag geen van de vier mogelijke clausules (**if..then**, **elseif..then**, **else** en **end if**) de regelgrens overschrijden.

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| condition | Een logische uitdrukking die resulteert in True of False. |
| statements | Een willekeurige groep van een of meer Qlik Sense-scriptopdrachten. |

**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

Het sleutelwoord **Next** voor scripts wordt gebruikt om**For**-lussen af te sluiten.

## Sub..end sub

De besturingsopdracht **sub..end sub** geeft een subroutine aan die vanuit een **call**-opdracht kan worden aangeroepen.

**Syntaxis:**

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

De argumenten worden in de subroutine gekopieerd en, als de corresponderende parameter in de **call**-opdracht een variabelenaam is, weer teruggekopieerd bij het verlaten van de subroutine.

Als een subroutine meer formele parameters heeft dan er aan daadwerkelijke parameters door een **call**-opdracht worden doorgegeven, worden de overtollige parameters op NULL ingesteld en kunnen ze als lokale variabelen binnen de subroutine worden gebruikt.

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| name | De naam van de subroutine. |
| paramlist | Een door komma's gescheiden lijst variabelen voor de formele parameters van de subroutine. Deze kunnen worden gebruikt als willekeurige variabelen binnen de subroutine. |
| statements | Een willekeurige groep van een of meer Qlik Sense-scriptopdrachten. |

**Beperkingen:**

- Omdat de instructie **sub** een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag geen van de twee mogelijke clausules (**sub** en **end sub**) de regelgrens overschrijden.
- Als u een subroutine definieert met `Sub..end sub` binnen een besturingsopdracht, bijvoorbeeld `if..then`, kunt u de subroutine alleen oproepen binnen dezelfde besturingsopdracht.

**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: - parameteroverdracht**

```
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)
```

Het resultaat van het bovenstaande is dat lokaal, binnen de subroutine, A wordt geïnitialiseerd op 1, B op 4 en C op NULL.

Bij het verlaten van de subroutine krijgt de globale variabele A de waarde 2 (teruggekopieerd van de subroutine). De tweede werkelijke parameter "(X+1)*2" wordt niet teruggekopieerd omdat dit geen variabele is. Tot slot verandert de globale variabele C niet door de aanroep van de subroutine.

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

De besturingsopdracht **switch** is een constructie voor scriptselectie waarmee de uitvoering van het script, afhankelijk van de waarde van een uitdrukking, op verschillende manieren kan verlopen.

**Syntaxis:**

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

> *Omdat de instructie **switch** een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag geen van de vier mogelijke clausules (**switch**, **case**, **default** en **end switch**) de regelgrens overschrijden.*

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| expression | Een willekeurige uitdrukking. |

| Argument | Beschrijving |
|---|---|
| valuelist | Een door komma's gescheiden lijst waarden waarmee de waarde van de uitdrukking zal worden vergeleken. Uitvoering van het script wordt vervolgd met de opdrachten in de eerste groep waarvoor een waarde in valuelist is gevonden die gelijk is aan de waarde in de uitdrukking. Elke waarde in valuelist kan een willekeurige uitdrukking zijn. Als in geen van de **case**-opdrachten een overeenkomst is gevonden, worden de opdrachten achter **default**, indien gespecificeerd, uitgevoerd. |
| statements | Een willekeurige groep van een of meer Qlik Sense-scriptopdrachten. |

**Voorbeeld:**

```
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

Het sleutelwoord **To** voor scripts wordt gebruikt in verschillende scriptopdrachten.

## 10.2 Prefixen

Prefixen kunnen worden toegevoegd aan normale opdrachten, maar nooit aan besturingsopdrachten.

Bij het typen van sleutelwoorden in het script kunt u elke combinatie van hoofdletters en kleine letters gebruiken. Gebruikte namen van velden en variabelen in de opdrachten zijn echter hoofdlettergevoelig.

### Overzicht prefixen diagramaanpassing

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

#### Add

Het voorvoegsel **Add** kan aan elke **LOAD**- of **SELECT**-opdracht in het script worden toegevoegd om op te geven dat er records aan een andere tabel moeten worden toegevoegd. Dit geeft ook aan dat deze opdracht met behulp van gedeeltelijk laden moet worden uitgevoerd. Het voorvoegsel **Add** kan ook worden

gebruikt in een **Map**-opdracht.

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

**Replace**

het voorvoegsel **Replace** kan aan elke **LOAD**- of **SELECT**-opdracht in het script worden toegevoegd om op te geven dat de geladen tabel een andere tabel moet vervangen. Dit geeft ook aan dat deze opdracht met behulp van gedeeltelijk laden moet worden uitgevoerd. Het voorvoegsel **Replace** kan ook worden gebruikt in een **Map**-opdracht.

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

## Add

In de context van de aanpassing van diagrammen, wordt het prefix **Add** gebruikt in combinatie met **LOAD** om waarde toe te voegen aan de *HC1*-tabel. Dit vertegenwoordigt de hyperkubus die is berekend door de Qlik Associative Engine. U kunt één of meerdere kolommen specificeren. Ontbrekende waarden worden automatisch ingevuld door de Qlik Associative Engine.

**Syntaxis:**

```
Add loadstatement
```

**Voorbeeld:**

In dit voorbeeld worden twee rijen aan de kolommen *Datums* en *Verkoop* toegevoegd vanuit the inline-opdracht

```
Add Load
x as Dates,
y as Sales
Inline
[
Dates,Sales
2001/09/1,1000
2001/09/10,-300
]
```

## Replace

In de context van aanpassing van het diagram, verandert het prefix **Replace** alle waarden van de *HC1*-tabel met een berekende waarde die is gedefinieerd door het script.

**Syntaxis:**

```
Replace loadstatement
```

**Voorbeeld:**

Dit voorbeeld overschrijft alle waarden in kolom `z` met de som van `x` en `y`.

```
Replace Load
x+y as z
Resident HC1;
```

## 10.3 Reguliere scriptopdrachten

Normale opdrachten worden over het algemeen gebruikt om gegevens op een bepaalde manier te manipuleren. Deze opdrachten kunnen worden geschreven op een willekeurig aantal regels in het script en moeten altijd worden afgesloten met een puntkomma (;).

Bij het typen van sleutelwoorden in het script kunt u elke combinatie van hoofdletters en kleine letters gebruiken. Gebruikte namen van velden en variabelen in de opdrachten zijn echter hoofdlettergevoelig.

### Overzicht besturingsopdracht diagramaanpassing

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

#### LOAD

In de context van aanpassing van een diagram, laadt de opdracht **LOAD** meer gegevens naar de hyperkubus uit de gegevens die in het script zijn gedefinieerd, of vanuit een eerder geladen tabel. U kunt tevens gegevens van analytische verbindingen inladen.

> *De opdracht **LOAD** moet ofwel het prefix **Replace** ofwel het prefix **Add** bevatten. Zo niet, dan wordt hij afgewezen.*

```
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

De opdracht **let** is een aanvulling op de opdracht **set** voor het definiëren van scriptvariabelen. In tegenstelling tot de opdracht **set** evalueert de opdracht **let** de uitdrukking rechts van '=' tijdens runtime van het script voordat deze wordt toegewezen aan de variabele.

```
Let variablename=expression
```

### Set

Met de opdracht **set** legt u scriptvariabelen vast. Deze kunnen worden gebruikt om tekenreeksen, paden, stations en dergelijke te vervangen.

```
Set variablename=string
```

### Put

De opdracht **Put** wordt gebruikt om een numerieke waarde in de hyperkubus te stellen.

### HCValue

De opdracht **HCValue** wordt gebruikt om waarden in een rij van een bepaalde kolom op te halen.

## Load

In de context van aanpassing van een diagram, laadt de opdracht **LOAD** meer gegevens naar de hyperkubus uit de gegevens die in het script zijn gedefinieerd, of vanuit een eerder geladen tabel. U kunt tevens gegevens van analytische verbindingen inladen.

*De opdracht **LOAD** moet ofwel het prefix **Replace** ofwel het prefix **Add** bevatten. Zo niet, dan wordt hij afgewezen.*

**Syntaxis:**

```
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 ]
```

**Argumenten:**

Argumenten

| Argument | Beschrijving |
|---|---|
| fieldlist | *fieldlist ::= ( * \| field{, * \| field } )*<br>Een lijst van de velden die geladen moeten worden. Gebruik een * als veldlijst om alle velden in de tabel aan te duiden.<br>*field ::= ( fieldref \| expression ) [***as** *aliasname ]*<br>De velddefinitie moet altijd een literal, een referentie naar een bestaand veld of een uitdrukking bevatten.<br>*fieldref ::= ( fieldname \|@fieldnumber \|@startpos:endpos [ **I** \| **U** \| **R** \| **B** \| **T**] )*<br>*fieldname* is een tekst die identiek is aan een veldnaam in de tabel. Houd er rekening mee dat de veldnaam tussen rechte dubbele aanhalingstekens of vierkante haken moet staan als deze bijvoorbeeld spaties bevat. Soms zijn veldnamen niet expliciet beschikbaar. In dat geval wordt een andere notatie gebruikt:<br><br>*@fieldnumber* geeft het veldnummer in een tabelbestand met scheidingstekens aan. Het moet een positief geheel getal zijn, voorafgegaan door "@". De nummering is altijd vanaf 1 optellend tot het aantal velden.<br><br>*@startpos*:*endpos* geeft de begin- en eindposities van een veld aan in een bestand met een vaste recordlengte. De posities moeten beide positieve gehele getallen zijn. De twee getallen moeten worden voorafgegaan door "@" en gescheiden door een dubbele punt. De nummering is altijd vanaf 1 optellend tot het aantal posities. In het laatste veld wordt **n** gebruikt als eindpositie.<br>• Als *@startpos*:*endpos* direct wordt gevolgd door de tekens **I** of**U**, worden de gelezen bytes geïnterpreteerd als een binair ondertekend (**I**) of niet-ondertekend (**U**) geheel getal (Intel-bytevolgorde). Het aantal gelezen posities moet 1, 2 of 4 zijn.<br>• Als *@startpos*:*endpos* direct wordt gevolgd door het teken **R**, worden de bytes geïnterpreteerd als een binair reëel getal (IEEE 32-bits of 64-bits drijvende komma). Het aantal gelezen posities moet 4 of 8 zijn.<br>• Als *@startpos*:*endpos* direct wordt gevolgd door het teken **B**, worden de gelezen bytes geïnterpreteerd als een BCD (Binary Coded Decimal) volgens de COMP-3-standaard. U kunt een willekeurig aantal bytes opgeven.<br><br>*expression* kan een numerieke functie of een tekenreeksfunctie zijn, gebaseerd op een of meer andere velden in dezelfde tabel. Zie de syntaxis van uitdrukkingen voor meer informatie.<br><br>**as** wordt gebruikt om een nieuwe naam aan een veld toe te wijzen. |

| Argument | Beschrijving |
|---|---|
| inline | **inline** wordt gebruikt als gegevens moeten worden getypt in het script en niet uit een bestand moeten worden geladen.<br>*data ::= [ text ]*<br><br>Gegevens die u invoert met een **inline**-clausule moeten tussen dubbele aanhalingstekens of tussen vierkante haakjes worden geplaatst. De tekst ertussen wordt net zo geïnterpreteerd als de inhoud van een bestand. Waar u bijvoorbeeld een nieuwe regel in een tekstbestand zou invoegen, moet u dat ook doen in de tekst van een **inline**-clausule. U drukt dus op Enter bij het typen van het script. Het aantal kolommen wordt bepaald door de eerste regel.<br>*format-spec ::= ( fspec-item {, fspec-item } )*<br>De opmaakspecificatie bestaat uit een lijst met verschillende opmaakspecificaties, tussen haakjes. Ga voor meer informatie naar *Opmaakspecificaties (page 176)*. |
| resident | **resident** wordt gebruikt als gegevens moeten worden geladen uit een eerder geladen tabel.<br>*table label* is een label dat vooraf gaat aan de opdracht **LOAD** of waarmee de oorspronkelijke tabel is gemaakt. Het label moet worden opgegeven met een puntkomma aan het eind. |

| Argument | Beschrijving |
|---|---|
| extension | U kunt gegevens via analytische verbindingen inladen. Als u een functie wilt aanroepen die in de SSE-plugin is gedefinieerd, moet u de clausule **extension** gebruiken of een script evalueren.<br><br>U kunt één tabel naar de SSE-plugin verzenden, waarop één gegevenstabel wordt teruggegeven. Als door de plugin niet de namen worden aangegeven van de velden die worden teruggegeven, krijgen de velden de namen Field1, Field2 enzovoort.<br><br>`Extension pluginname.functionname( tabledescription );`<br><br>• Gegevens inladen met een functie in een SSE-plugin<br>*tabledescription ::= (table { ,tablefield} )*<br>Als u geen tabelvelden aangeeft, worden de velden gebruikt in de volgorde waarin deze zijn ingeladen.<br>• Gegevens inladen door een script in een SSE-plugin te evalueren<br>*tabledescription ::= ( script, table { ,tablefield} )*<br><br>**Afhandeling van gegevenssoorten in de tabelvelddefinitie**<br><br>Gegevenssoorten in analytische verbindingen worden automatisch gedetecteerd. Als de gegevens geen getalswaarden bevatten en ten minste één teksttekenreeks die niet leeg is, wordt het veld als tekstveld beschouwd. In alle andere gevallen wordt het veld als getalsveld beschouwd.<br><br>U kunt de gegevenssoort afdwingen door een veldnaam op te nemen in **String()** voor een tekenreeks of **Mixed()** voor gemengd.<br><br>• Met **String()** maakt u het veld een tekstveld. Als het veld numeriek is, wordt het tekstgedeelte van de dubbele waarde geëxtraheerd. Er wordt niet geconverteerd.<br>• Met **Mixed()** maakt u het veld een veld met dubbele waarden.<br><br>**String()** en **Mixed()** kunnen niet buiten tabelvelddefinities van **extension** worden gebruikt, en u kunt geen andere Qlik Sense -functies in een tabelvelddefinitie gebruiken. |
| where | **where** is een clausule die wordt gebruikt om aan te geven of een record wel of niet in de selectie moet worden opgenomen. De selectie wordt opgenomen als *criterion* de waarde True heeft.<br>*criterion* is een logische uitdrukking. |
| while | **while** is een clausule die aangeeft of een record herhaaldelijk moet worden gelezen. Hetzelfde record wordt gelezen zolang *criterion* de waarde True heeft. Een **while**-clausule is vaak pas nuttig wanneer u deze samen met de functie **IterNo( )** gebruikt.<br><br>*criterion* is een logische uitdrukking. |

| Argument | Beschrijving |
|---|---|
| group by | **group by** is een clausule die wordt gebruikt om te definiëren over welke velden de gegevens moeten worden geaggregeerd (gegroepeerd). De aggregatievelden moeten op de een of andere manier in de geladen uitdrukkingen worden opgenomen. U mag geen andere velden dan de aggregatievelden gebruiken buiten aggregatiefuncties in de geladen uitdrukkingen.<br><br>*groupbyfieldlist ::= (fieldname { ,fieldname } )* |
| order by | **order by** is een clausule die wordt gebruikt om de records van een residente tabel te sorteren voordat ze worden verwerkt door de **load**-opdracht. De residente tabel kan worden gesorteerd op één of meer velden in oplopende of aflopende volgorde. Het sorteren gebeurt primair op numerieke waarde en secundair op nationale sorteervolgorde. U kunt deze clausule alleen gebruiken wanneer de gegevensbron een residente tabel is.<br>De volgordevelden geven aan op welk veld de residente tabel wordt gesorteerd. U kunt het veld opgeven met z'n naam of met z'n nummer in de residente tabel (het eerste veld is nummer 1).<br><br>orderbyfieldlist ::= fieldname [ sortorder ] { , fieldname [ sortorder ] }<br><br>*sortorder* is *asc* voor oplopend of *desc* voor aflopend. Als u geen *sortorder* opgeeft, wordt *asc* verondersteld.<br><br>*fieldname*, *path*, *filename* en *aliasname* zijn tekenreeksen die aangeven waar de respectievelijke namen voor staan. Elk veld in de brontabel kan worden gebruikt als *fieldname*. Velden die zijn gemaakt via de as-clausule (*aliasname*) vallen echter buiten het bereik en kunnen niet worden gebruikt binnen dezelfde **load**-opdracht. |

## Let

De opdracht **let** is een aanvulling op de opdracht **set** voor het definiëren van scriptvariabelen. In tegenstelling tot de opdracht **set** evalueert de opdracht **let** de uitdrukking rechts van '=' tijdens runtime van het script voordat deze wordt toegewezen aan de variabele.

**Syntaxis:**

```
Let variablename=expression
```

Voorbeelden en resultaten:

| Voorbeeld | Resultaat |
|---|---|
| `Set x=3+4;`<br><br>`Let y=3+4;`<br><br>`z=$(y)+1;` | `$(x)` wordt geëvalueerd als ' 3+4 '<br><br>`$(y)` wordt geëvalueerd als ' 7 '<br><br>`$(z)` wordt geëvalueerd als ' 8 '<br><br>Let op het verschil tussen de **Set**- en **Let**-instructies. De **Set**-instructie wijst de reeks '3+4' toe aan de variabele, terwijl de **Let**-instructie de reeks evalueert en 7 toewijst aan de variabele. |
| Let T=now( ); | `$(T)` krijgt de waarde van de huidige tijd. |

## Set

Met de opdracht **set** legt u scriptvariabelen vast. Deze kunnen worden gebruikt om tekenreeksen, paden, stations en dergelijke te vervangen.

**Syntaxis:**

```
Set variablename=string
```

**Example 1:**

```
Set FileToUse=Data1.csv;
```

**Example 2:**

```
Set Constant="My string";
```

**Example 3:**

```
Set BudgetYear=2012;
```

## Put

De opdracht **put** wordt gebruikt om een numerieke waarde in de hyperkubus te stellen.

De kolommen zijn toegankelijk via labels. U kunt kolommen en rijen ook openen op volgorde van declaratie. Zie de onderstaande voorbeelden voor meer details.

**Syntaxis:**

```
put column(position)=value
```

**Example 1:**

De kolommen zijn toegankelijk via labels.

In dit voorbeeld wordt een waarde van `1` gesteld in de positie van de kolom met het label *Verkoop*.

```
Put Sales(1) = 1;
```

**Example 2:**

U hebt toegang tot metingkolommen op volgorde van declaratie met gebruik van de `#hc1.measure`-indeling voor metingen.

In dit voorbeeld wordt de waarde 1000 in de tiende positie geplaatst van de laatst gesorteerde hyperkubus.

```
Put #hc1.measure.2(10) = 1000;
```

**Example 3:**

U kunt de dimensierijen openen op volgorde van declaratie met de `#hc1.dimension`-indeling voor dimensies.

In dit voorbeeld wordt de waarde van de constante Pi in de vijfde rij van de derde gedeclareerde dimensie.

```
Put #hc1.dimension.3(5) = Pi();
```

*Als dat soort dimensies of uitdrukkingen niet aanwezig zijn, in waarden of labels, dan wordt er een foutmelding gegeneerd dat de kolom niet is gevonden. Als de indexwaarde voor de kolom buiten bereik ligt, wordt er NULL geretourneerd.*

## HCValue

De functie **HCValue** wordt gebruikt om waarden in een rij van een bepaalde kolom op te halen.

**Syntaxis:**

```
HCValue(column,position)
```

**Example 1:**

In dit voorbeeld wordt de waarde uit de eerste positie van de kolom met het label Verkoop geretourneerd.

```
HCValue(Sales,1)
```

**Example 2:**

In dit voorbeeld wordt de waarde uit de tiende positie van de gesorteerde hyperkubus geretourneerd.

```
HCValue(#hc1.measure2,10)
```

**Example 3:**

In dit voorbeeld wordt de waarde uit de vijfde rij in de derde dimensie geretourneerd.

```
HCValue(#hc1.dimension.3,5)
```

*Als dat soort dimensies of uitdrukkingen niet aanwezig zijn, in waarde of labels, dan wordt er een foutmelding gegeneerd dat de kolom niet is gevonden. Als de indexwaarde voor de kolom buiten bereik ligt, wordt er NULL geretourneerd.*

# 11 QlikView 12-functies en -opdrachten die niet worden ondersteund in Qlik Sense

De meeste functies en opdrachten die kunnen worden gebruikt in load-scripts en diagramuitdrukkingen in QlikView 12 worden tevens ondersteund in Qlik Sense. Maar er zijn enkele uitzonderingen, zoals hier beschreven.

## 11.1 Scriptopdrachten die niet worden ondersteund in Qlik Sense

QlikView 12 Scriptopdrachten die niet worden ondersteund in Qlik Sense

| Opdracht | Commentaar |
|---|---|
| **Command** | Gebruik in plaats hiervan **SQL**. |
| **InputField** | |

## 11.2 Functies die niet worden ondersteund in Qlik Sense

Deze lijst bevat script- en diagramfuncties in QlikView 12 die niet worden ondersteund in Qlik Sense.

- **GetCurrentField**
- **GetExtendedProperty**
- **Input**
- **InputAvg**
- **InputSum**
- **MsgBox**
- **NoOfReports**
- **ReportComment**
- **ReportId**
- **ReportName**
- **ReportNumber**

## 11.3 Prefixes die niet worden ondersteund in Qlik Sense

Deze lijst bevat pefixes in QlikView 12 die niet worden ondersteund in Qlik Sense.

- **Bundle**
- **Image_Size**
- **Info**

# 12 Functies en opdrachten die niet worden aanbevolen in Qlik Sense

De meeste functies en opdrachten die kunnen worden gebruikt in load-scripts en diagramuitdrukkingen in QlikView 12 worden tevens ondersteund in Qlik Sense, maar voor sommige hiervan wordt het gebruik in Qlik Sense niet aanbevolen. Er zijn ook functies en opdrachten beschikbaar in eerdere versies van Qlik Sense die zijn verouderd.

Om compatibiliteitsredenen werken zij nog steeds zoals bedoeld, maar het is raadzaam om de code bij te werken op basis van de aanbevelingen in deze sectie, aangezien zij mogelijk worden verwijderd in komende versies.

## 12.1 Scriptopdrachten die niet worden aanbevolen in Qlik Sense

Deze tabel bevat scriptopdrachten die niet worden aanbevolen voor gebruik in Qlik Sense.

Scriptopdrachten die niet worden aanbevolen

| Opdracht | Aanbeveling |
|---|---|
| **Command** | Gebruik in plaats hiervan **SQL**. |
| **CustomConnect** | Gebruik in plaats hiervan **Custom Connect**. |

## 12.2 Parameters voor scriptopdrachten die niet worden aanbevolen in Qlik Sense

Deze tabel bevat parameters voor scriptopdrachten die niet worden aanbevolen voor gebruik in Qlik Sense.

Parameters voor scriptopdrachten die niet worden aanbevolen

| Opdracht | Parameters |
|---|---|
| **Buffer** | Gebruik **Incremental** in plaats van:<br>• **Inc** (niet aanbevolen)<br>• **Incr** (niet aanbevolen) |

# 12 Functies en opdrachten die niet worden aanbevolen in Qlik Sense

| Opdracht | Parameters |
|---|---|
| LOAD | De volgende trefwoorden voor parameters worden gegenereerd door QlikView 12-wizards voor bestandstransformatie. De functionaliteit blijft behouden als gegevens opnieuw worden geladen, maar Qlik Sense biedt geen gerichte ondersteuning/wizards voor het genereren van de opdracht met deze parameters:<br>• Bottom<br>• Cellvalue<br>• Col<br>• Colmatch<br>• Colsplit<br>• Colxtr<br>• Compound<br>• Contain<br>• Equal<br>• Every<br>• Expand<br>• Filters<br>• Intarray<br>• Interpret<br>• Length<br>• Longer<br>• Numerical<br>• Pos<br>• Remove<br>• Rotate<br>• Row<br>• Rowcnd<br>• Shorter |

## 12.3 Functies die niet worden aanbevolen in Qlik Sense

Deze tabel bevat script- en diagramfuncties die niet worden aanbevolen voor gebruik in Qlik Sense.

Functies die niet worden aanbevolen

| Functie | Aanbeveling |
|---|---|
| **NumAvg**<br>**NumCount**<br>**NumMax**<br>**NumMin**<br>**NumSum** | Gebruik in plaats hiervan Range-functies.<br>*Bereikfuncties (page 1600)* |
| **Color()**<br>**QliktechBlue**<br>**QliktechGray** | Gebruik in plaats hiervan kleurfuncties. **QliktechBlue()** kan worden vervangen door **RGB(8, 18, 90)** en **QliktechGray** kan worden vervangen door **RGB(158, 148, 137)** om dezelfde kleuren te krijgen.<br>*Kleurfuncties (page 622)* |
| **QlikViewVersion** | Gebruik in plaats hiervan **EngineVersion**.<br>*EngineVersion (page 1902)* |
| **ProductVersion** | Gebruik in plaats hiervan **EngineVersion**.<br>*EngineVersion (page 1902)* |
| **QVUser** | |
| **Year2Date** | Gebruik in plaats hiervan **YearToDate**. |
| **Vrank** | Gebruik in plaats hiervan **Rank**. |
| **WildMatch5** | Gebruik in plaats hiervan **WildMatch**. |

### Kwalificatie ALL

In QlikView 12 kan de kwalificatie **ALL** voorkomen vóór een uitdrukking. Dit is gelijk aan het gebruik van **{1} TOTAL**. In een dergelijk geval wordt de berekening gemaakt op basis van alle waarden van het veld in het document, en worden grafiekdimensies en huidige selecties genegeerd. Dezelfde waarde wordt altijd geretourneerd, ongeacht de logische status in het document. Als de kwalificatie **ALL** wordt gebruikt, kan de uitdrukking set niet worden gebruikt omdat de kwalificatie **ALL** zelf een set definieert. Om de functionaliteit uit eerdere versies te behouden werkt de kwalificatie **ALL** nog in deze versie van Qlik Sense. Het is echter mogelijk dat deze kwalificatie in toekomstige versies wordt verwijderd.