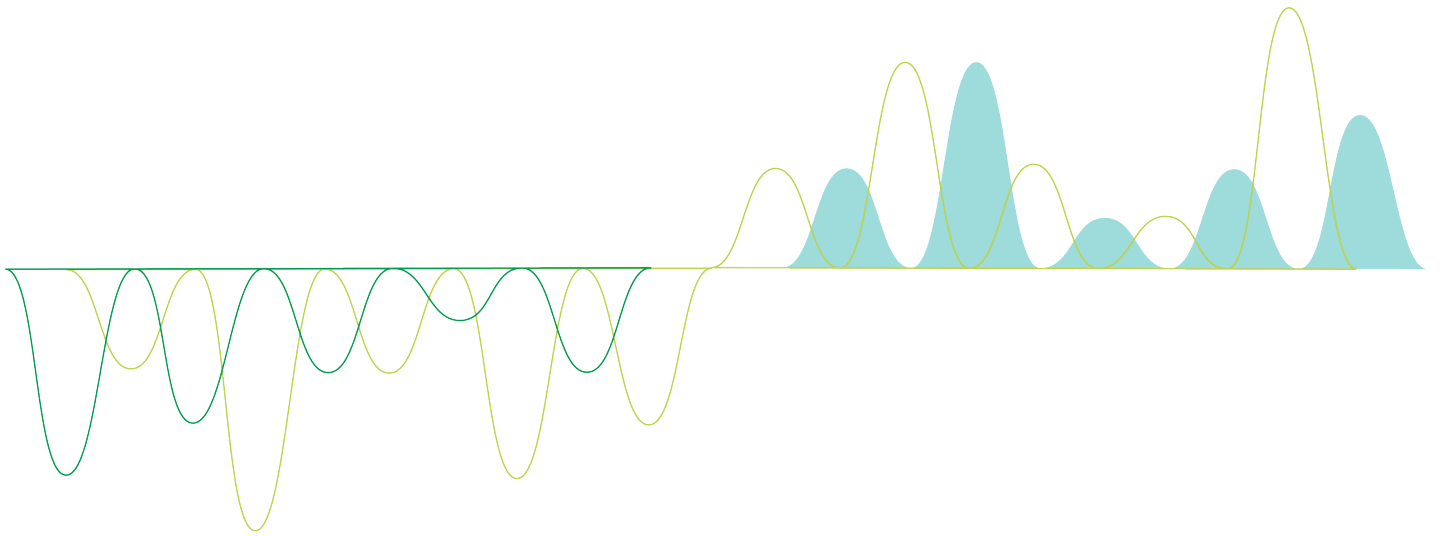


データの管理

Qlik Sense®

May 2022

Copyright © 1993-2022 QlikTech International AB. All rights reserved.



1 本書について	10
2 データの管理	11
3 データマネージャーを使用したデータのロードと管理	12
3.1 データテーブルのプレビュー	12
3.2 新規データテーブルの追加	12
3.3 データテーブルの編集	13
3.4 データテーブルの削除	13
3.5 データテーブル関連付けの管理	13
3.6 変更の適用とデータのリロード	14
3.7 [データマネージャー] で、操作を [元に戻す]/[やり直す]	15
3.8 データマネージャーでのテーブル変換の詳細の表示	15
3.9 [データマネージャー] およびデータロードスクリプト間の操作	15
3.10 データマネージャーでのテーブルの連結	16
3.11 アプリへのデータの追加	16
アプリ内	17
ファイルの場所	17
データ接続	17
[データコンテンツ]	18
ファイルをこのアプリに添付	18
新しいデータソースへの接続	18
データの追加	18
使用可能なデータソースとは	18
既存のデータソースからデータを追加する	18
新しいデータソースからデータを追加する	20
データファイルの添付とデータのアプリへの追加	22
でのデータの手動追加 Qlik Sense	25
データ項目の選択	27
ファイルのデータのフィルタリング	32
データコネクタからのデータのフィルタリング	34
3.12 データテーブルの編集	35
テーブル名の変更	35
項目名の変更	35
他のテーブルとの関連付けの管理	36
項目の種類と表示形式の変更	36
全体的な分析から項目を非表示にする	36
データロード前のテーブル項目データの評価	37
テーブル内での項目値の置換	37
テーブル内の項目値を NULL に設定する	37
項目値にカスタム順序を設定	37
テーブル内での項目の分割	37
メジャーの範囲へのグループ化	38
項目変換の詳細の表示	38
クロスタブのデータのピボットを無効にする	38
データソースからのテーブルの更新	38
計算項目の追加	39
テーブルのソート	39
元に戻す/操作のやり直し	39

テーブル エディタでデータを関連付ける	39
計算された項目の使用	41
項目の種類を変更する	53
全体的な分析から項目を非表示にする	55
データロード前のテーブル項目データの評価	56
テーブル内での項目値の置換	58
テーブル内の項目値を NULL に設定する	59
軸の値の順序のカスタマイズ	60
テーブル内での項目の分割	61
メジャーの範囲へのグループ化	64
データマネージャーでクロスタブのデータのピボットを無効にする	67
3.13 データマネージャーでのテーブルの連結	69
テーブルの自動連結	69
テーブル間の強制連結	70
連結されたテーブルの分割	72
3.14 データマネージャーでのテーブルの結合	73
Join 演算子	73
テーブルの結合	76
結合テーブルの分割	78
3.15 データマネージャーでのテーブルおよび項目の変換の詳細の表示	78
テーブルの詳細の表示	79
項目の詳細の表示	79
3.16 強制連結を使用した段階的なテーブルの結合	79
連結の概要	79
手順 - 強制連結	80
その先の手順 - 新しいテーブルの追加とデータ項目の連結	86
3.17 [データマネージャー] でのスクリプト化されたテーブルの同期	88
スクリプト化されたテーブルの同期	89
管理対象のスクリプト化されたテーブルの削除	89
3.18 データ関連付けの管理	90
[Recommended associations (推奨される関連付け)] パネルの使用によるテーブルの関連付	
け	91
手動によるテーブルの関連付け	92
関連付けの解除	93
関連付けの編集	93
データのプレビュー	94
合成キー	94
制限事項	95
変更の適用とデータのリロード	95
4 スクリプトを使用したデータのロードと変換	97
4.1 [データマネージャー] およびデータロードスクリプト間の操作	97
4.2 データロードエディタの使用	98
ツールバー	98
データ接続	98
テキストエディタ	98
セクション	99
アウトプット	99

データロードエディターでのデータソースへの接続	99
データロードエディターでデータを選択	102
データロードスクリプトの編集	108
スクリプトコードの整理	112
データロードスクリプトのデバッグ	112
ロードスクリプトの保存	115
スクリプト実行によるデータのロード	115
[データロードエディタ]でのキーボードショートカット	116
4.3 スクリプト構文とデータ構造を理解する	117
抽出、変換、ロード	117
データロードのステートメント	118
スクリプトの実行	119
項目	119
論理テーブル	124
Qlik Sense のデータ型	134
ドル記号展開	137
スクリプトでの引用符の使用	148
データ内のワイルドカード	151
NULL 値の処理	152
4.4 データおよび項目のガイドライン	155
ロードされるデータ量のガイドライン	155
データテーブルと項目の上限	155
ロードスクリプトセクションの推奨制限	155
数値や日付書式の説明	155
4.5 QVD ファイルでの作業	159
QVD ファイルの使用目的	160
QVD ファイルの作成	160
QVD ファイルからのデータのロード	160
QVD 形式	161
4.6 における分析接続の構成 Qlik Sense Desktop	161
Qlik オープンソース SSE リポジトリ	162
要素の説明	162
4 Section Access によるデータセキュリティの管理	163
4.7 ロードスクリプトのセクション	163
セクションアクセスのシステム項目	163
4.8 アプリへのユーザーアクセスの管理	165
4.9 アプリ内の特定のデータへのユーザーアクセスの管理	166
行レベルのデータへのアクセスの管理	166
列レベルのデータへのアクセスの管理	167
ユーザーグループへのアクセスの管理	168
4.10 なりすましを使用したデータのリロード	169
4.11 マルチクラウド環境におけるユーザーアクセスの管理	169
4.12 セクションアクセスと Insight Advisor Chat の使用	170
4.13 Section Access を使用するためのガイドラインとヒント	171
5 On-demand アプリによるビッグデータの管理	172
5.1 On-demand アプリの構成要素	172
5.2 On-demand アプリの作成	173

5.3 On-demand アプリの公開	174
5.4 On-Demand アプリの利点	174
5.5 制限事項	175
5.6 On-demand 選択 アプリの作成	175
5.7 On-demand テンプレート アプリの作成	176
テンプレート アプリの構造	176
シングル サインオン (SSO)	177
テンプレート アプリのノードのリロード	178
On-demand テンプレート アプリのバインディング数式	178
5.8 On-demand アプリの構築	184
5 ダイナミック ビューによるデータの管理	188
5.9 ダイナミック ビューの概要	188
ダイナミック ビュー	189
ダイナミック ビューのテンプレート アプリ	190
ダイナミック チャート	190
5.10 ダイナミック ビューの制限	190
5.11 ストリームとダイナミック ビュー	191
5.12 ダイナミック ビューとダイナミック チャートの作成	191
ダイナミック ビューを作成	192
シートにダイナミック チャートを追加	192
ダイナミック ビューの編集	192
5.13 ダイナミック ビューとダイナミック チャートの使用	192
ダイナミック ビューでの選択	193
ダイナミック ビューの詳細表示	194
ダイナミック ビューの更新	197
6 データ ソースへの接続	198
6.1 接続の作成	198
6.2 データ接続の種類	198
添付ファイル	198
データベース コネクタ	198
Essbase	199
ローカル ファイルまたはネットワーク ファイル	199
DSN 経由の ODBC 接続	199
Qlik Web コネクタ	199
REST	200
Salesforce	201
SAP	201
Web ファイル	201
Web ストレージ プロバイダ コネクタ	201
サードパーティ コネクタ	201
6.3 データ接続の保存場所	201
6.4 ファイルからのデータのロード	202
ファイル形式	202
接続の種類	202
ファイルからデータをロードする方法	202
ローカルおよびネットワークのファイル フォルダーからのファイルのロード	203

Web リソースからのファイルのロード	203
Microsoft Excel スプレッドシートからのデータのロード	204
6.5 データベースからのデータのロード	207
ODBC データベースからのデータのロード	207
ODBC	208
ODBC データソースからのデータのロード	208
OLE DB	211
データベースのロジック	213
6.6 を利用した膨大なデータセットへのアクセスDirect Discovery	213
Direct Discovery とインメモリデータとの違い	214
Direct Discovery 項目の種類	220
でサポートされているデータソースDirect Discovery	221
を使用する際の制限事項Direct Discovery	222
のマルチテーブル サポートDirect Discovery	224
でのサブクエリの使用Direct Discovery	226
Direct Discovery アクセスの ログ	228
7 データモデルの表示および変換	229
7.1 テーブルの移動	230
7.2 テーブルのサイズ変更	230
7.3 データモデルのパフォーマンス	230
7.4 データモデル ビューアでのテーブルおよび項目のプレビュー	233
テーブルのプレビューを表示する	234
項目のプレビューを表示する	234
7.5 データモデル ビューアからのマスター軸の作成	235
7.6 データモデル ビューアからのマスター メジャーの作成	235
8 データモデリングのベストプラクティス	236
8.1 データ列を行に変える	236
8.2 データ行を項目に変える	236
8.3 階層レベルに整理されたデータ(組織図など)をロードする	237
8.4 大きなデータベースから新しいレコードや更新されたレコードのみをロードする	238
8.5 共通項目を持つ2つのテーブルのデータを結合する	238
8.6 不連続値を間隔に一致させる	238
8.7 不一致項目値を処理する	239
8.8 不一致の項目値の大文字の処理	240
8.9 地理空間データをロードしてマップのデータを可視化する	241
8.10 増分ロードにより新規および更新された記録をロード	241
追加のみ	242
挿入のみ(更新または削除なし)	242
挿入および更新(削除なし)	243
挿入、更新、および削除	243
8.11 Join とKeep を使用したテーブルの結合	244
SQL SELECT ステートメント内で結合	245
Join	245
Keep	246
Inner	246
Left	247

Right	248
8.12 結合の代替としてのマッピング	250
8.13 データロードスクリプトでのクロス集計の使用	251
1つの行見出し列により、クロスタブのピボットを無効にする	251
2つの行見出し列により、クロスタブのピボットを無効にする	252
8.14 汎用データベース	253
8.15 不連続データと間隔の一致	256
Intervalmatch の例	256
拡張された intervalmatch 構文を使用した、穏やかに変化する軸 (SCD) 問題の解決	257
8.16 単一日付からの日付間隔の作成	259
8.17 階層データのロード	262
8.18 ユーザー独自のマップデータのロード	263
マップ ビジュアライゼーションの項目で対応している名前データ	263
KML ファイルからのポイントデータおよびエリアデータのロード	264
データプロファイリングを使用したマップデータのロード	264
ポイントデータのロードと書式設定	265
8.19 データクレンジング	267
マッピング テーブル	267
マッピング テーブルの使用	267
8 Insight Advisor の論理モデルのカスタマイズ	269
8.20 ビジネス ロジックを使用した Insight Advisor の論理モデルの構築	269
論理モデルの理解	269
論理モデルのカスタマイズ	270
カスタム ビジネス ロジックの有効化	271
ビジネス ロジックのリセット	271
ビジネス ロジックの無効化	271
項目とグループの定義	271
パッケージを使用した論理モデル範囲の設定	275
階層を使用したドリルダウン分析の作成	276
論理モデルへの動作の適用	276
カレンダー期間を使用した分析期間の定義	279
8.21 Insight Advisor の語彙の作成	285
制限事項	286
Insight Advisor への同義語の追加	286
Insight Advisor へのカスタム分析の追加	287
9 トラブルシューティング - データのロード	292
9.1 [データの追加] にドロップしてもファイルが添付されない	292
9.2 ANSI 以外でエンコードされたデータファイルでの文字セットの問題	292
9.3 データロード時の循環参照についての警告	292
9.4 固定長レコードファイルからデータを選択するときに列が予想通りに並ばない	293
9.5 コネクタが動作しない	293
コネクタが正常にインストールされない	293
コネクタが Qlik Sense 対応しない	293
9.6 SQL Server の再起動後にデータ接続が機能しなくなる	294
9.7 データロードエディターがスクリプトを表示しない	294
9.8 データロードスクリプトはエラーなしで実行されますが、データがロードされません	295

ステートメントがセミコロンで終了していない	295
文字列内の単一引用符	295
9.9 データマネージャーがデータを含むアプリでテーブルを表示しない	295
9.10 サーバーでアプリを作成しているすべてのユーザーのデータマネージャーワークフローが壊れている	296
9.11 OLE DB データソースでのデータ選択の問題	296
9.12 シート表示でデータ項目が日付項目として認識されない	296
テーブルを追加したときに、データプロファイルが解除された	296
日付形式が認識されない	297
9.13 ファイル添付時に [無効なパス] というエラーメッセージが表示される	297
9.14 QlikView ドキュメントから変換したアプリをロードする際にエラーが発生する	297
絶対ファイルパス参照がスクリプト内で使用されている	297
サポートされていない関数/ステートメントがスクリプトで使用されている	298
9.15 Microsoft Excel: データマネージャーまたはデータロードエディタでのファイルからのデータのロードが失敗する	298
9.16 Microsoft Excel: ファイルへの接続時、およびファイルからのデータのロード時に問題が発生する (ODBC 経由)	298
9.17 ディスクスペースが不足している	299
9.18 データロード時の合成キーについての警告	299
9.19 共通項目を含むテーブルが項目名によって自動的に関連付けられない	299

1 本書について

このドキュメントでは、データを追加して管理する方法、より高度なデータモデルのデータロードスクリプトを構築する方法、データモデルビューアーでデータモデルを表示する方法について説明し、**Qlik Sense** のデータモデリングのベストプラクティスを示します。



スクリプト関数とチャート関数の詳細については、スクリプト構文およびチャート関数を参照してください。

本書は **Qlik Sense** のオンラインヘルプをもとに作成されています。オフラインでの情報入手や印刷の便宜を図った簡易版のため、すべての情報が掲載されているわけではありません。

help.qlik.com/sense では、オンラインヘルプ、付加的なガイド、その他多くのリソースをご提供しております。

2 データの管理

Qlik Sense アプリを作成したら、まず探索と分析を行うデータを追加します。このセクションでは、データを追加して管理する方法、より高度なデータモデルのデータロードスクリプトを構築する方法、データモデルビューアーでデータモデルを表示する方法について説明し、**Qlik Sense** のデータモデリングのベストプラクティスを示します。

アプリにデータを追加する方法は、2種類あります。

- **データマネージャー**

スクリプト言語を学習しなくても、独自のデータソースや **Qlik DataMarket** などの他のソースからデータを追加できます。データ選択を編集でき、データモデルでのデータの関連付けの作成がサポートされます。

- **データロードエディター**

Qlik Sense データロードスクリプト言語を使用して、**ETL (Extract, Transform & Load)** プロセスでデータモデルを構築できます。このスクリプト言語は強力であるため、複雑な変換を実行し、拡張性に優れたデータモデルを作成できます。





[データマネージャー] で構築したデータモデルをデータロードスクリプトに変換して**[データロードエディター]** で手を加えることもできますが、データロードスクリプトを**[データマネージャー]** データモデルに変換することはできません。**[データマネージャー]** データモデルとデータロードスクリプトで定義したデータテーブルは、引き続き共存できますが、データモデルの問題のトラブルシューティングを行うのが難しくなります。

3 データ マネージャーを使用したデータのロードと管理

アプリでデータを使用 することができるよう、データ マネージャーからデータを追加および管理 します。

データ マネージャーには、次の 2 つのビューがあります。

-  **関連付け**
テーブル間で関連付けを作成して編集できます。
-  **テーブル**
[データの追加] を使用してデータを追加した、またはデータ ロードスクリプトを使用してロードしたかによらず、アプリのすべてのデータ テーブルの概要が表示 されます。各 テーブルには、テーブル名、データ項目の数、データソースの数が表示 されます。

3.1 データ テーブルのプレビュー

テーブルをプレビューして、含まれている列およびデータのサンプル セットを確認 することができます。

次の手順を実行 します。

- プレビューするデータ テーブルを選択 します。

テーブル データセットのプレビューが表示 されます。

3.2 新規データ テーブルの追加

データ テーブルは、アプリに簡単に追加 できます。データ マネージャーを開いて、**+** をクリック します。**☰** で [データの追加] をクリック することもできます。新しいアプリを作成 する場合には、データを追加 するよう要求 するプロンプトも表示 されます。

以下のデータソースからデータを追加 できます。


データソース

データソース	説明
アプリ内	アプリ内に存在 するデータソースから選択 します。アプリに添付 したファイルなどが該当 します。 [手動入力] を使用 して、データソースを作成 し、そのソースにデータを手動 で追加 することもできます。
ファイルの場所	例えば、管理者 が定義 したドライブなどのネットワーク ドライブ上のファイルから選択 します。
データ接続	ユーザーや管理者 によって定義 されている、既存のデータ接続から選択 します。

3.3 データ テーブルの編集


[データの追加] で追加したすべてのデータテーブルを編集できます。データテーブルでテーブルと項目の名前を変更し、データソースからの項目を更新することができます。また、計算項目を追加し、日付と時刻の形式を調整することもできます。

次の手順を実行します。

1. 編集したいデータテーブルで  をクリックします。
データテーブル エディタが開き、編集や変換を実行できます。
2. [閉じる] をクリックして戻ります。

このテーブルのステータスは [更新保留] となり、変更は、次のデータリロードの際に適用されます。




[データの追加] で追加したデータテーブルのみを編集できます。ロードスクリプトを使用してロードされたテーブルで  をクリックすると、データロードエディタが開きます。詳細については、「データロードエディタの使用 (page 98)」を参照してください。



3.4 データ テーブルの削除

[データの追加] で追加したデータテーブルのみを削除できます。ロードスクリプトを使用してロードされたデータテーブルは、データロードエディタでスクリプトを編集することでのみ削除できます。

次の手順を実行します。

- 削除したいデータテーブルで  をクリックします。

これでテーブルが [削除対象] になり、次回データをリロードする際に削除されます。

以前に実行した削除を元に戻す/やり直すには、 または  をクリックします。



データテーブルからの項目をビジュアライゼーションで使用している場合は、データテーブルを削除すると、アプリでエラーが表示されます。

3.5 データ テーブル関連付けの管理

関連付ける必要があるテーブルを追加するのに最適な状況は、テーブルを、異なるテーブルにおいて名前が同じキー項目と関連付ける場合です。そのような場合には、[データの追加] のデータプロファイリング無効化オプションによって Qlik Sense にそれらのテーブルを追加することができ、その結果、テーブルが正しく関連付けられたデータ構造になります。

データソースがそのような理想的な状態になっていない場合には、関連付けの問題がいくつか発生する可能性があります。


3 データマネージャーを使用したデータのロードと管理

- 同じデータを含むが項目名の異なる2つの項目を2つの異なるテーブルからロードした場合、テーブルを関連付けるために、それらの項目を同じ名前にすることができます。
- 異なるデータを含むが項目名が同じ2つの項目を2つの異なるテーブルからロードした場合、それらの項目を別々の項目としてロードするために少なくとも一方の項目の名前を変更する必要があります。
- 共通項目が複数含まれるテーブルを2つロードした場合。

データを関連付ける場合は、データプロファイリングを有効にして **[dデータの追加]** オプションを使用するようお勧めします。これはデフォルトのオプションです。この設定を検証するには、[データの追加] ページ右下隅の **[データの追加]** ボタンの横にある ******* をクリックします。

Qlik Senseは、テーブルの関連付けの修正をサポートするために、ロードするデータのデータプロファイリングを実行します。既存の不適切な関連付けと適切な関連付けになる可能性がある箇所が強調表示され、データの分析にもとづいて、関連付け項目の選択がサポートされます。

テーブルの関連付けは、次の2種類の方法で管理できます。


- データマネージャーの  **[関連付け]** ビューを使用する。
Insight Advisor 推奨事項に基づいて関連付けを作成するか、または1つまたは複数の項目に基づいてカスタム関連付けを作成できます。
- テーブルエディタの **[関連付け]** オプションを使用する。
いくつかの項目に基づいてカスタム関連付けと複合キーの関連付けを作成できます。




データを追加するときにデータプロファイリングを無効化した場合、Qlik Senseは、共通項目名に基づいてテーブルに自動的に関連づけられます。

3.6 変更の適用とデータのリロード


[データマネージャー] で行った変更は、データをリロードするまでアプリでは使用できません。データをリロードすると、変更が適用され、追加したすべての新規データが外部データソースからロードされます。以前ロードしたデータはリロードされません。

[データマネージャー] フッターの  ボタンを使用して、外部データソースからすべてのデータをリロードすることができます。



 ボタンを使用すると、選択したテーブルのすべてのデータがリロードされます。アプリ内のすべてのテーブルのすべてのデータはリロードされません。

[データマネージャー] 内のデータがアプリのデータと同期していない場合は、[データのロード] ボタンが緑色で表示されます。**[関連付け]** 表示では、新規作成または更新されたテーブルにはすべて*が付き、削除されたテーブルは薄いグレーで表示されます。**[テーブル]** 表示では、新規作成および更新、削除されたテーブルはすべて青でハイライトされ、テーブルのステータスを示すアイコンが表示されます。

- [削除保留 


次の手順を実行します。

- [データのロード] をクリックして、アプリで変更をロードします。


これで [データ マネージャー] での変更がアプリのデータに反映されます。

変更を適用し、外部データソースから、選択したテーブルのすべてのデータをリロードするには、次の手順に従います。

次の手順を実行します。

- [データ マネージャー] フッターで 

3.7 [データ マネージャー] で、操作を [元に戻す]/[やり直す]

[データ マネージャー] で編集しているときは、

アクション ログは次の場合にクリアされます。

- 表示を切り替えた場合 (テーブルの概要から [関連付け] に変えた場合など)。
- データをロードした場合。
- [データ マネージャー] を開いた場合。

3.8 データ マネージャーでのテーブル変換の詳細の表示

[詳細] ダイアログを使用して、データ マネージャー内のテーブルで実行された操作と変換を表示できます。[詳細] ダイアログは、[関連付け] ビューと [テーブル] ビューで表示できます。

[詳細] には、選択したテーブルに対して実行されている現在の操作と変換が表示されます。このダイアログには、テーブルのソース、行われた現在の変更、および変更が適用された順序が表示されます。[詳細] により、テーブルが現在の状態になった経過を容易に理解できます。例えば、[詳細] を使用して、テーブルが連結された順序を簡単に確認できます。

3.9 [データ マネージャー] およびデータ ロード スクリプト間の操作

[データ マネージャー] でデータ テーブルを追加すると、データ ロード スクリプト コードが生成されます。データ ロード エディタの [自動生成 セクション] でスクリプト コードを表示できます。生成されたスクリプト コードをロック解除して編集することもできますが、その場合、データ テーブルは [データ マネージャー] で管理されなくなります。

3 データマネージャーを使用したデータのロードと管理

デフォルトでは、ロードスクリプトで定義されたデータテーブルは、[データマネージャー]で管理されません。これは、データの概要でテーブルを閲覧することはできても、[データマネージャー]内でテーブルを削除したり編集したりはできないことを意味します。また、スクリプトでロードしたテーブルに対しては、関連付けは提案されません。ただし、スクリプトテーブルを[データマネージャー]と同期する場合、そのテーブルはマネージドスクリプトテーブルとして[データマネージャー]に追加されます。



テーブルを同期した場合、[データマネージャー]を別のタブで開いた状態で、データロードエディターで変更を行わないでください。

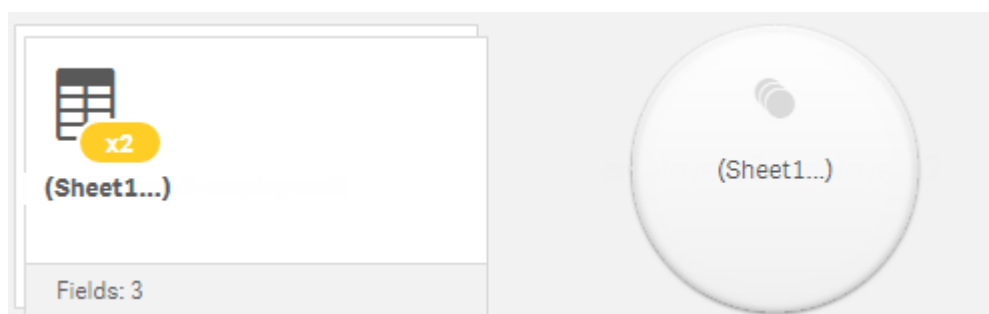
スクリプトセクションの追加や、[データマネージャー]で作成したデータモデルを強化したり、通信やコードの開発は可能ですが、注意を必要とするエリアがあります。ユーザー記述のスクリプトコードは、例えば、次のような場合に[データマネージャー]のデータモデルを妨害し、問題を生じさせる可能性があります。

- [データマネージャー]を使って追加したテーブルの名前をスクリプトで変更したりテーブルをドロップした場合。
- [データマネージャー]を使って追加したテーブルから項目をドロップした場合。
- [データマネージャー]を使って追加したテーブルとスクリプトにロードしたテーブルを連結した場合。
- [データマネージャー]を使って追加したテーブル内にある項目と**Qualify**ステートメントを併用した場合。
- スクリプトで**Resident**を使用している[データマネージャー]を使って追加したテーブルをロードした場合。
- 生成されたコードセクションの後にスクリプトコードを追加した場合。データモデルの変更結果は、[データマネージャー]に反映されません。

3.10 データマネージャーでのテーブルの連結

連結により、2つのテーブルを項目が組み合わされた単一のテーブルに結合します。これによりコンテンツを統合して、コンテンツを共有する個別のテーブルおよび項目の数を減らします。データマネージャーのテーブルは、自動的または強制的に連結することが可能です。

[テーブル]表示と[関連付け]表示で連結されたテーブル。



3.11 アプリへのデータの追加

アプリにデータをすばやく追加できます。データマネージャーを開いて、**+**をクリックします。**☰**で[データの追加]をクリックすることもできます。新しいアプリを作成する場合には、データを追加するよう要求するプロンプトも表示されます。シートを編集する場合には、[項目]パネルで[追加]をク

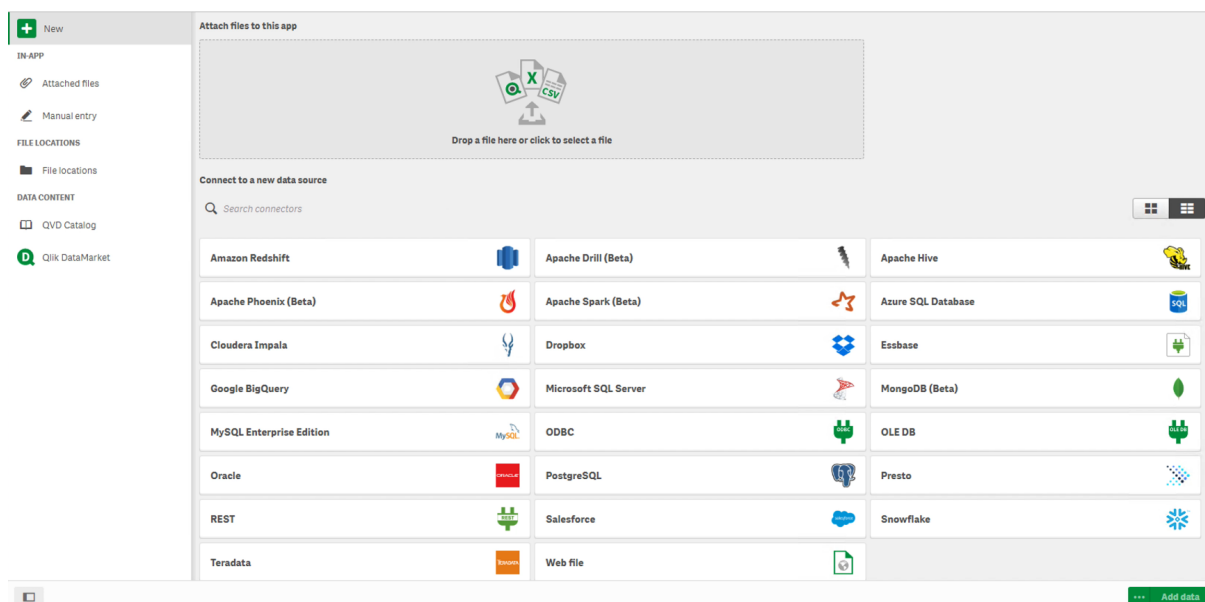
3 データ マネージャーを使用したデータのロードと管理

リックしてデータを追加することもできます。



使用できるデータの追加オプションとデータソースは、使用する **Qlik Sense** プラットフォームと構成によって決まります。

データビューの追加



アプリ内

添付ファイル。 展開: Qlik Sense Enterprise on Windows. アプリにファイルを添付するか、アプリに添付済みのファイルを表示する場合にクリックします。これらのファイルからデータをロードできます。

手動入力。 アプリ内 テーブルを作成し、アプリにデータを追加する場合にクリックします。

ファイルの場所

ファイルの場所。 展開: Qlik Sense Enterprise on Windows. 管理者が定義したフォルダの場所へのアクセス権を付与します。

マイ コンピューター。 展開: Qlik Sense Desktop.

データファイルをアップロードする場合、またはすでにアップロードされているファイルのデータを追加する場合にクリックします。

データ接続

展開: すべて。

外部のデータソースに対して作成された接続を表示します。この接続は、作成された後に **[新しいデータソースへの接続]** に表示されます。

接続をクリックして、アプリにデータを追加します。

[データ コンテンツ]

Qlik Catalog Service. プラットフォーム: Qlik Sense Enterprise on Windows.

Qlik Catalog Service QVD ソースへのアクセスを提供します。管理者が Qlik Catalog への接続を作成した場合にのみ使用できます。

アプリにデータを追加する場合にクリックします。

ファイルをこのアプリに添付

クリックしてアプリにファイルを添付します。

新しいデータソースへの接続

クリックして新しいデータソースへの接続を作成します。

データの追加

アプリにデータを追加する場合にクリックし、このボタンは、接続を作成し、ロードするデータを選択した後に有効になります。クリックして、アプリにデータを追加します。プロファイリングは有効でも無効でもデータを追加できます。データプロファイリングは推奨されており、デフォルトで有効です。●●● をクリックしてデータプロファイリングを無効にします。

使用可能なデータソースとは

使用可能なデータソースの種類は、多くの要因によって異なります。

- アクセス設定
管理者設定によって、接続できるデータソースの種類が決まります。
- インストールされているコネクタ
Qlik Sense には、多くのデータソース用のビルトインサポートが含まれています。組み込みコネクタは、Qlik Sense によって自動的にインストールされます。追加のデータソースに接続するには、それらのデータソースを特に対象としたコネクタを別々にインストールしなければならない場合があります。それらの別々にインストールされるコネクタは、Qlik またはサードパーティによって提供されます。
- ローカル ファイルの可用性
Qlik Sense Desktop では、デスクトップ コンピュータ上のローカル ファイルのみを使用できます。Qlik Sense のサーバー環境では使用できません。



Qlik Sense のサーバー インストール環境にロードするローカル ファイルがある場合は、そのファイルをアプリに添付するか、または Qlik Sense サーバーで使用可能なフォルダ(既にフォルダデータ接続として定義されているフォルダを推奨) にファイルを転送する必要があります。

既存のデータソースからデータを追加する

ユーザーや管理者によって既に定義されている接続から、データをアプリに追加できます。例えば、データベース、データファイルを含むフォルダー、Salesforce.com などの外部データソースへのコネクタなどです。

3 データマネージャーを使用したデータのロードと管理



[データロードエディター]で、同じ名前および同じ列でスクリプト化されたテーブルとして既に追加済みのテーブルは、[データマネージャー]に追加しないでください。

接続を右クリックしてを選択すると、[データの追加]から接続を削除できます



接続を削除する場合、データをロードする前に、その接続を使用していた[データマネージャー]からテーブルをすべて削除する必要があります。

次の手順を実行します。

1. アプリを開きます。
2. データマネージャーを開いて、**+** をクリックします。**☰** で [データの追加] をクリックすることもできます。
3. [データ接続] の下で、既存の接続を選択します。
一部の接続では、データソースに直接移動して、ロードするテーブルと項目を選択します。例えば、Salesforce.com やデータベースへの接続では、ソースに直接移動してデータを選択します。
4. 接続が選択内容を指定している場合は、データの追加元の特定のデータソースを選択します。
これは、データソースの種類によって異なります。
 - ファイルベースのデータソース: ファイルを選択します。
 - データベース: 使用するデータベースを設定します。
 - Web ファイル: Web ファイルの URL を入力します。
 - その他のデータソース: コネクターごとに指定されます。
5. ロードするテーブルと項目を選択します。
6. オプションで、選択した項目に含まれるデータのサブセットを選択する場合は、データフィルターの適用を選択します。
データソースがファイルである場合、[フィルター] を選択します。フィルターを追加するテーブルのそばにある、[Add filter] (フィルターを追加) をクリックして項目、条件を選択し、フィルターに使用する値を入力します。



Qlik Sense は、QVD ファイルからのデータ項目に対するフィルターには対応していません。

次の点に注意してください。

- 複数のフィルターを同じ項目に適用できます。
- [データマネージャー] の [関連付け] ビューのフィルターは、[ソースからデータを選択] から削除できます。変更を反映させるには、[データのロード] ボタンをクリックしてデータをリロードします。

データベースおよびコネクターに対して [データのフィルタリング] を選択すると、フィルター条件用のテキストボックスが開きます。

次の点に注意してください。

- データベースの項目名にフィルターを適用します。データマネージャーで項目の名前を変更する場合は、データベースの元の項目名にフィルターを適用する必要があります。例えば、データベースで項目に EMP という名前を付け、[データマネージャー] で EMPLOYEE という名前に変更した場合、フィルター EMP = 「filter_value」を適用する必要があります。

3 データ マネージャーを使用したデータのロードと管理

- [データ マネージャー] の [関連付け] ビューのデータフィルターをクリアできます。変更を反映させるには、[データのロード] ボタンをクリックしてデータをリロードします。フィルターをクリアする前に、連結されたテーブルを分割する必要があります。
 - すべてのコネクター、または添付ファイルでは現在、データのフィルタリングは使用できません。
7. [データの追加] をクリックして、データ マネージャーの [関連付け] ビューでデータを開きます。これにより、データ マネージャーでのデータソースの追加、データの変換、テーブルの関連付けを続行できます。データプロファイリングは、[データの追加] をクリックするとデフォルトで有効になります。データプロファイリングでは次の処理が行われます。
- データ関連付けを推奨する。
 - テーブル間の共通項目を自動記述する。これにより、テーブル名に基づいて一意のプレフィックスが追加されます。
 - 日付項目と時刻項目を autoCalendar にマッピングする。


テーブルは共通の項目名で自動的に関連付けられません。テーブルは [関連付け] ビューで関連付けることができます。



データをアプリに直接ロードする場合は、*** をクリックしてからデータプロファイリングを無効にします。無効にすることで、データを追加するときに、新しく選択したデータが外部データソースからロードされるようになります。テーブルは共通の項目名で自動的に関連付けられます。日付項目と時刻項目は作成されません。

詳細については、「データ関連付けの管理 (page 90)」を参照してください。

8. データの準備が完了したら、[データのロード] をクリックします。重大な問題が検出された場合は、[データ マネージャー] で解決してからアプリにデータをロードする必要があります。
- 詳細については、「トラブルシューティング - データのロード (page 292)」を参照してください。


選択したすべてのデータを外部ソースからリロードするには、データ マネージャーのフッターの  ボタンを使用します。このボタンを使用すると、行った選択について、ソースから現在のデータすべてが取得されます。すべてのデータのリロードは、新しいデータのみロードよりも時間がかかる場合があります。以前ロードしたデータがデータソースで変更されていない場合、すべてのデータをリロードする必要はありません。

新しいデータソースからデータを追加する

データを新しいデータソースからアプリに追加することができます。新しいデータソースからデータを追加すると、同じデータソースからのデータの追加を容易にするため、[データ接続] にそのデータソースへの接続が作成されます。



[データロードエディター] で、同じ名前および同じ列でスクリプト化されたテーブルとして既に追加済みのテーブルは、[データ マネージャー] に追加しないでください。

接続を右クリックしてを選択すると、[データの追加] から接続を削除できます .

3 データマネージャーを使用したデータのロードと管理



接続を削除する場合、データをロードする前に、その接続を使用していた[データマネージャー]からテーブルをすべて削除する必要があります。

次の手順を実行します。

1. アプリを開きます。
2. データマネージャーを開いて、**+** をクリックします。**☰** で **[データの追加]** をクリックすることもできます。
3. **[新しいデータソースへの接続]** の下でソースを選択します。
4. データソースで必要になる接続パラメーターを入力します。
例:
 - ファイルベースのデータソースでは、ファイルへのパスを指定して、ファイルの種類を選択する必要があります。
 - Oracle や IBM DB2 などのデータベースでは、データベースプロパティと資格情報が必要になります。
 - Web ファイルでは、Web ファイルの URL が必要になります。
 - ODBC 接続では、DSN 資格情報が必要になります。
5. ロードするテーブルと項目を選択します。
6. オプションで、選択した項目に含まれるデータのサブセットを選択する場合は、データフィルターの適用を選択します。
データソースがファイルである場合、**[フィルター]** を選択します。フィルターを追加するテーブルのそばにある、**[Add filter]** (フィルターを追加) をクリックして項目、条件を選択し、フィルターに使用する値を入力します。



Qlik Sense は、QVD ファイルからのデータ項目に対するフィルターには対応していません。

次の点に注意してください。

- 複数のフィルターを同じ項目に適用できます。
- **[データマネージャー]** の **[関連付け]** ビューのフィルターは、**[ソースからデータを選択]** から削除できます。変更を反映させるには、**[データのロード]** ボタンをクリックしてデータをリロードします。

データベースおよびコネクタに対して **[データのフィルタリング]** を選択すると、フィルター条件用のテキストボックスが開きます。

次の点に注意してください。

- データベースの項目名にフィルターを適用します。**データマネージャー**で項目の名前を変更する場合は、データベースの元の項目名にフィルターを適用する必要があります。例えば、データベースで項目に EMP という名前を付け、**[データマネージャー]** で EMPLOYEE という名前に変更した場合、フィルター EMP = 「filter_value」を適用する必要があります。
- **[データマネージャー]** の **[関連付け]** ビューのデータフィルターをクリアできます。変更を反映させるには、**[データのロード]** ボタンをクリックしてデータをリロードします。フィルターをクリアする前に、連結されたテーブルを分割する必要があります。
- すべてのコネクタ、または添付ファイルでは現在、データのフィルタリングは使用できません。

3 データ マネージャーを使用したデータのロードと管理

7. **[データの追加]** をクリックして、データ マネージャーの **[関連付け]** ビューでデータを開きます。これにより、**データ マネージャー** でのデータソースの追加、データの変換、テーブルの関連付けを続行できます。データプロファイリングは、**[データの追加]** をクリックするとデフォルトで有効になります。データプロファイリングでは次の処理が行われます。

- データ関連付けを推奨する。
- テーブル間の共通項目を自動記述する。これにより、テーブル名に基づいて一意のプレフィックスが追加されます。
- 日付項目と時刻項目を **autoCalendar** にマッピングする。

テーブルは共通の項目名で自動的に関連付けられません。テーブルは **[関連付け]** ビューで関連付けることができます。



データをアプリに直接ロードする場合は、******* をクリックしてからデータプロファイリングを無効にします。これによっても、データを追加するときに、すべての既存データがデータソースから再ロードされます。テーブルは共通の項目名で自動的に関連付けられます。日付項目と時刻項目は作成されません。

詳細については、「**データ関連付けの管理 (page 90)**」を参照してください。

8. データの準備が完了したら、**[データのロード]** をクリックします。重大な問題が検出された場合は、**[データ マネージャー]** で解決してからアプリにデータをロードする必要があります。詳細については、「**トラブルシューティング - データのロード (page 292)**」を参照してください。

データファイルの添付とデータのアプリへの追加

データファイルをアプリに添付してから、アプリでデータを使用できます。

添付されたファイルは、ファイルが添付されたアプリでのみ使用できます。元のデータファイルには接続がないため、元のデータファイルを更新した場合は、添付ファイルの更新も必要です。



制限されているデータを公開しないようにするには、アプリを公開する前に、**セクション アクセス設定** が含まれるすべての添付ファイルを削除します。添付ファイルはアプリの公開時に含まれます。公開済みアプリがコピーされると、添付ファイルがそのコピーに含まれます。ただし、添付したデータファイルに**セクション アクセス制限**が適用されている場合、ファイルがコピーされるときに**セクション アクセス設定**が保持されません。そのため、コピーしたアプリのユーザーは、添付ファイルのすべてのデータを表示することができます。

制限事項

- アプリに添付されるファイルの最大サイズは **50 MB** です。
- アプリに添付されるファイルの最大合計サイズは、メディア ライブラリにアップロードされる画像ファイルも含めて **200 MB** です。
- **Qlik Sense Desktop** にファイルを添付することはできません。

複数のデータファイルの迅速な添付

ほとんどの場合、1 セットのデータファイルをアプリに添付して追加する最も速くて簡単な方法は、ファイルをアプリ内にドロップすることです。

3 データマネージャーを使用したデータのロードと管理

次の手順を実行します。

- アプリ内に1つまたは複数のデータファイルをドロップします。
ファイルは、アプリにアップロードされて添付され、データモデルに追加されます。

この方法でファイルを添付すると、Qlik Sense はデータをロードするために最適な設定の選択を試行します。例えば、埋め込まれた項目名、項目区切り文字または文字セットなどの認識です。設定が最適ではないテーブルを追加した場合に、設定を修正するには、テーブル エディターでテーブルを開き、[ソースからデータを選択] をクリックします。



データロードエディターまたはデータモデルビューアーには、ファイルをドロップすることはできません。

単一データファイルの添付

データファイルは、1つずつ添付することができます。このようにして、ファイルのインポート設定全体をより詳細に制御します。例えば、埋め込まれた項目名、使用されている項目区切り文字または文字セットなどの設定です。



[データロードエディター] で、同じ名前および同じ列でスクリプト化されたテーブルとして既に追加済みのテーブルは、[データマネージャー] に追加しないでください。

次の手順を実行します。

1. アプリを開きます。
2. データマネージャーを開いて、**+** をクリックします。**☰** で [データの追加] をクリックすることもできます。
3. データファイルをドロップするか、コンピュータからロードするファイルをクリックして選択します。
すでに添付されているファイルと同じ名前のファイルを添付しようとする、新しいファイルと置き換えるかどうか確認するメッセージが表示されます。



添付ファイルには、固有のファイル名が付けられていなければなりません。

4. ロードするテーブルと項目を選択します。
5. オプションで、選択した項目に含まれるデータのサブセットを選択する場合は、データフィルターの適用を選択します。
データソースがファイルである場合、[フィルター] を選択します。フィルターを追加するテーブルのそばにある、[Add filter] (フィルターを追加) をクリックして項目、条件を選択し、フィルターに使用する値を入力します。



Qlik Sense は、QVD ファイルからのデータ項目に対するフィルターには対応していません。

次の点に注意してください。

- 複数のフィルターを同じ項目に適用できます。
- [データマネージャー] の [関連付け] ビューのフィルターは、[ソースからデータを選択] から削除できます。変更を反映させるには、[データのロード] ボタンをクリックしてデータをリロードします。

6. **[データの追加]** をクリックして、データ マネージャーの **[関連付け]** ビューでデータを開きます。これにより、**データ マネージャー** でのデータソースの追加、データの変換、テーブルの関連付けを続行できます。データプロファイリングは、**[データの追加]** をクリックするとデフォルトで有効になります。データプロファイリングでは次の処理が行われます。

- データ関連付けを推奨する。
- テーブル間の共通項目を自動記述する。これにより、テーブル名に基づいて一意のプレフィックスが追加されます。
- 日付項目と時刻項目を **autoCalendar** にマッピングする。

テーブルは共通の項目名で自動的に関連付けられません。テーブルは **[関連付け]** ビューで関連付けることができます。




データをアプリに直接ロードする場合は、******* をクリックしてからデータプロファイリングを無効にします。これによっても、データを追加するときに、すべての既存データがデータソースから再ロードされます。テーブルは共通の項目名で自動的に関連付けられます。日付項目と時刻項目は作成されません。

7. データの準備が完了したら、**[データのロード]** をクリックします。重大な問題が検出された場合は、**[データ マネージャー]** で解決してからアプリにデータをロードする必要があります。

添付ファイルの削除

データ マネージャー内の添付ファイルに基づくテーブルを削除すると、テーブルはデータモデルから削除されますが、添付データファイルはアプリ内に残されたままになります。データファイルはアプリから永続的に削除することができます。

次の手順を実行します。

1. アプリを開きます。
2. **データ マネージャー** を開いて、**+** をクリックします。
3.  **[添付ファイル]** をクリックします。
4. 該当するファイルを削除します。

これで、データファイルは、アプリから永続的に削除されます。




アプリで使用している添付ファイルを削除すると、**[データ マネージャー]** のロードスクリプトでファイルへの参照を削除しない限り、アプリをリロードできなくなります。ロードスクリプトの編集は、**[データロードエディター]** で行います。

添付ファイルからのデータのリロード

アプリ用にアップロードするファイルは、そのアプリに添付されます。これはそのアプリにのみ使用できます。





3 データ マネージャーを使用したデータのロードと管理

元のデータファイルには接続されていません。元のファイルを更新する場合は、アプリに添付されているファイルも更新する必要があります。その後、更新後のデータをアプリにロードできます。データ マネージャーでデータをロードしたら、 ([Refresh data from source] (ソースのデータを更新)) をクリックして、更新されたデータをテーブル ビューで表示できます。



[データ ロード エディター] で、同じ名前および同じ列でスクリプト化されたテーブルとして既に追加済みのテーブルは、[データ マネージャー] に追加しないでください。

次の手順を実行します。

1. アプリを開きます。
2. データ マネージャーを開いて、 をクリックします。
3.  [添付ファイル] をクリックします。
4. 既存のファイルを置き換えます。更新したファイルは、元のファイルと同じ名前である必要があります。データファイルの内容が更新されます。
5. [データの追加] をクリックします。 をクリックして、データプロファイルが有効になっていることを確認します。
6. [関連付け] ビューまたは [テーブル] ビューで、テーブルをクリックします。
7.  をクリックしてデータを更新します。
8. [データのロード] をクリックしてデータをアプリにリロードします。



データファイルの項目構造に変更を加えた場合 (例えば、項目の削除、名前の変更など) は、アプリ内のデータモデルに影響を与えることがあります。特に、テーブルを関連付けるのに使用される項目が関係している場合です。

でのデータの手動追加 Qlik Sense

[データの追加] の [手動入力] を使用すると、データをエディタに入力し、データ マネージャーのテーブルとして追加することができます。

データの手動入力は、別のソースからの限られたデータのセットを使用する場合に有益です。例えば、Excel のスプレッドシートまたは Web ページのテーブルから選択した行をテーブルとしてデータ マネージャーにロードするだけの場合、選択したデータを [手動入力] にコピーして貼り付け、データ マネージャーのテーブルとして追加することができます。データの量が少なく、別のデータソースからインポートするよりも手動で追加するほうが速い場合にも有益です。

データを手動で追加するには、[データの追加] を開いて [手動入力] を選択し、テーブルにデータを入力し、テーブルをデータ マネージャーに追加します。テーブル エディタは 1 つの行と 2 つの列から始まりますが、テーブルにデータが追加されるにつれて、空の行と列が自動的にテーブルに追加されます。

3 データマネージャーを使用したデータのロードと管理



[手動入力]ではデータは入力時に自動的に保存されません。画面が更新された場合、セッションがタイムアウトした場合、またはデータがデータマネージャーに追加される前に接続が切れた場合、入力されたデータが失われることがあります。

データは入力するほか、他のソースからコピーして貼り付けることができます。**[手動入力]**では、Excel テーブルからコピーしたデータの列と行が保持されます。

[手動入力]で効果的かつ簡単に作業を行うために使用できるキーボードのショートカットが数多く用意されています。ショートカットの動作は、セル、行、列を選択しているか、またはテーブルでセルを編集しているかによって異なります。次の表に、選択のショートカットを示します。

選択に使用するキーボードショートカット

ショートカット	説明
矢印キー	セルの各選択を切り替えます。
Tab	セルの選択を右に移動します。右にセルがない場合、次の行の最初のセルに移動します。
Shift+Tab	セルの選択を左に移動します。左にセルがない場合、前の行の最初のセルに移動します。
Enter	編集モードに切り替えます。
Delete	現在の選択条件を削除します。

次の表に、編集のショートカットを示します。

編集に使用するキーボードショートカット

ショートカット	説明
矢印キー	セル内でカーソルを移動します。
Tab	編集を確定して、右の次のセルに移動します。
Shift+Tab	編集を確定して、左の前のセルに移動します。
Enter	編集を確定して、下の次のセルに移動します。
Shift+Enter	編集を確定して、上の前のセルに移動します。
Esc	編集をキャンセルして、選択モードに切り替えます。

[手動入力]で作成したテーブルは後から編集して、コンテンツの追加または削除ができます。詳細については、「データソースからのテーブルの更新 (page 38)」を参照してください。

手動でのデータの追加

次の手順を実行します。

1. アプリを開きます。
2. データマネージャーを開いて、**+** をクリックします。

3 データ マネージャーを使用したデータのロードと管理

☰ メニューで **[データの追加]** をクリックすることもできます。

3. **[アプリ内]** で **[手動入力]** をクリックします。
4. テーブルの名前を入力します。
5. テーブル エディタでデータを入力します。
セルをダブルクリックして、セルへのデータ入力を開始します。
セルの編集集中にテーブル内の別のセルをクリックすると、編集が確定され、他のセルが選択されます。
6. データが完成したら、**[データの追加]** をクリックします。

データ マネージャーにテーブルが追加されます。

データ項目の選択

データの追加やテーブルの編集で使用するテーブルと項目を選択できます。

CSV ファイルなど一部のデータソースには単一テーブルが含まれますが、Microsoft Excel スプレッドシートやデータベースなどその他のデータソースには、複数のテーブルが含まれる場合があります。

テーブルにヘッダー行が含まれる場合、通常、項目名は自動で検出されますが、**[項目名]** の設定の変更が必要な場合もあります。また、データを正しく変換するために、**[ヘッダー サイズ]** や **[文字セット]** などの他のテーブル オプションの変更が必要な場合もあります。テーブル オプションは、データソースの種類によって異なります。

データベースからのデータの選択

データベースからデータを選択する手順は、データベースへの接続方法に応じて異なります。ODBC ドライバーに DSN ソースとして接続するか、Qlik Sense と合わせてインストールされている Qlik ODBC Connector Package の一部である Qlik Database コネクタを介して、直接接続できます。

詳細については、「*ODBC (page 208)*」を参照してください。

データベースからデータを追加する場合は、データソースに複数のテーブルを含めることができます。

次の手順を実行します。

1. ドロップダウン リストから **[データベース]** を選択します。
一部の選択ダイアログでは、接続の構成時にデータベース名が入力されているため、**[データベース]** ドロップダウン リストはありません。
2. データベースの **[所有者]** を選択します。
[テーブル] のリストには、選択したデータベースで使用可能なビューとテーブルが自動的に反映されます。
一部のデータベースでは、データ選択プロセス時に、所有者を指定する必要はありません。
3. テーブルを選択します。
4. 各項目名の横にあるボックスにチェックを付けて、ロードする項目を選択します。
テーブル名の横にあるボックスにチェックを付けるとテーブルのすべての項目を選択できます。



既存の項目名をクリックして新しい名前を入力すると、項目名を編集できます。規定値では共通項目に加わるため、その他のテーブルにテーブルがリンクする方法に影響が及ぶ場合があります。

3 データ マネージャーを使用したデータのロードと管理

5. 追加のテーブルからデータを追加する場合は、そのテーブルを選択します。
6. データを選択し終えたら、**[データの追加]** をクリックしてデータプロファイリングを続行し、テーブル関係の推奨事項を確認します。

データをアプリに直接ロードする場合は、**[データの追加]** の横にある ******* をクリックしてからデータプロファイリングを無効にします。これにより、選択したデータがそのままロードされ、データプロファイリングの手順がスキップされて、ビジュアライゼーションの作成を開始できます。テーブルは、未処理状態の関連付けを使用して(つまり、共通した項目名により)リンクされます。

Microsoft Excel スプレッドシートからのデータの選択

Microsoft Excel スプレッドシートからデータを追加する場合は、複数のシートを含めることができます。各シートは個別のテーブルとしてロードされます。例外となるのは、シートに別のシートやロード済みのテーブルと同じ項目や列構造がある場合であり、この場合はテーブルが連結されます。

次の手順を実行します。

1. シートが正しく設定されていることを確認します。

テーブルデータの正しい解釈に役立つ設定

UI 項目	説明
項目名	テーブルに [埋め込まれた項目名] が含まれているか、 [項目名が存在しない] が指定します。通常、Excel スプレッドシートでは、最初の行には埋め込まれた項目名が含まれます。 [項目名なし] を選択すると、項目名は A,B,C... となります。
ヘッダーサイズ	テーブルヘッダーとして無視する行数を設定します。通常は、列形式になっていない一般情報が含まれる行です。

例

スプレッドシートは次のようになっています。

スプレッドシートの例

マシン:	AEJ12B	-	-
【日付】:	2015-10-05 09	-	-
タイムスタンプ	順序	演算子	生産額
2015-10-05 09:22	00122344	A	52
2015-10-05 10:31	00153534	A	67
2015-10-05 13:46	00747899	B	86

この場合、最初の2行を無視して、Timestamp、Order、Operator、およびYieldの項目を含むテーブルをロードしたいと考えるでしょう。これを行うには、次の設定を使用します。

3 データ マネージャーを使用したデータのロードと管理

最初の 2 行を無視して項目をロードするための設定

UI 項目	説明
ヘッダー サイズ	2 これは、最初の 2 行がヘッダーデータとみなされて、ファイルをロードするときに無視されることを意味します。この場合、 Machine: および Date: で開始する 2 行はテーブルデータの一部分ではないため無視されます。
項目 名	[埋め込まれた項目名] これは、読み込まれる最初の行が、それぞれの列の項目名として使用されることを意味します。この場合は、最初の 2 行がヘッダーデータになっているため、読み込まれる最初の行は 3 行目となります。

2. データを選択する最初のシートを選択します。シート名の横にあるボックスにチェックを付けるとシートのすべての項目を選択できます。
3. 各項目名の横にあるボックスにチェックを付けて、ロードする項目を選択します。



既存の項目名をクリックして新しい名前を入力すると、項目名を編集できます。テーブルは既定では共通項目により結合されているため、他のテーブルにリンクされる方法に影響することがあります。

4. データを選択し終わったら、**[データの追加]** をクリックしてデータプロファイリングを続行し、テーブル関係の推奨事項を確認します。
データをアプリに直接ロードする場合は、**[データの追加]** の横にある ******* をクリックしてからデータプロファイリングを無効にします。これにより、選択したデータがそのままロードされ、データプロファイリングの手順がスキップされて、ビジュアライゼーションの作成を開始できます。テーブルは、未処理状態の関連付けを使用して(つまり、共通した項目名により)リンクされます。

テーブル ファイルからのデータの選択

大量のデータファイルからデータを追加できます。

次の手順を実行します。

1. **[ファイル形式]** で、ファイルの種類が正しく選択されていることを確認します。
2. ファイルが正しく設定されていることを確認します。ファイル設定は、ファイルの種類によって異なります。
3. 各項目名の横にあるボックスにチェックを付けて、ロードする項目を選択します。シート名の横にあるボックスにチェックを付けるとファイルのすべての項目を選択できます。



既存の項目名をクリックして新しい名前を入力すると、項目名を編集できます。テーブルは既定では共通項目により結合されているため、他のテーブルにリンクされる方法に影響することがあります。

4. データを選択し終わったら、**[データの追加]** をクリックしてデータプロファイリングを続行し、テーブル関係の推奨事項を確認します。



データをアプリに直接ロードする場合は、[データの追加]の横にある***をクリックしてからデータプロファイリングを無効にします。これにより、選択したデータがそのままロードされ、データプロファイリングの手順がスキップされて、ビジュアライゼーションの作成を開始できます。テーブルは、未処理状態の関連付けを使用して(つまり、共通した項目名により)リンクされません。

ファイルの種類の設定の選択

区切り記号付きテーブル ファイル

これらの設定は、各レコードが改行で区切られた単一テーブルや、各項目が区切り文字で区切られた(CSVファイルなど)の単一テーブルを含む、区切り記号付きテーブルファイルで検証されます。

ファイル形式の設定

区切り記号付きテーブルファイルのファイル形式の設定

UI 項目	説明
区切り記号付きテーブルファイルのファイル形式	[区切り記号付き] または [固定長レコード] に設定します。 選択すると、データ選択ダイアログが選択したファイル形式に応じて変化します。
項目名	テーブルに [埋め込まれた項目名] が含まれているか、[項目名が存在しない] が指定します。
区切り記号	テーブルファイルで使用する [区切り記号] 文字を設定します。
引用符	引用符の取り扱い方法を指定します。 [なし]= 引用符の使用不可 [標準]= 標準的な引用符 (項目値の最初および最後の文字として使用できます) [MSQ]= 新しいスタイルの引用符 (項目に複数行の内容を含めることができます)
ヘッダー サイズ	テーブルヘッダーとして省略される行数を設定します。
文字セット	テーブルファイルで使用される文字セットを設定します。
コメント	データファイルでは、レコード間にコメントを含めることができます。1つ以上の特殊文字 (// など) で始まる行はコメントを意味します。 ここでは、コメント行を示す文字を1つ以上指定します。Qlik Sense は、ここで指定された文字で始まる行をロードしません。
EOF を無視	項目値の一部として、データにファイルの終端を意味する文字が含まれている場合、[EOF を無視する] を選択します。

3 データ マネージャーを使用したデータのロードと管理

固定長レコードのデータファイル

固定長レコードのデータファイルには単一テーブルが含まれています。このテーブルでは通常、各レコード(データの行)に、空白やタブ記号が埋め込まれた固定項目サイズの列が多数含まれています。

項目の区切り位置の設定

項目の区切り位置の設定には、2通りの方法があります。

- **[項目の区切り位置]** に手動でコンマを入力して区切ります。各位置が項目の開始点となります。

1,12,24

- **[項目の区切り]** を有効にして、項目データのプレビューで項目の区切り位置を編集します。**[項目の区切り位置]** が、選択した位置により更新されます。次の操作が可能です。
 - 項目データのプレビュー内をクリックして、項目を区切ります。
 - 項目の区切りをクリックして、その区切りを削除します。
 - 項目の区切りをドラッグして、その区切りを移動させます。

ファイル形式の設定

固定長レコードのデータファイルのファイル形式の設定

UI 項目	説明
項目名	テーブルに [埋め込まれた項目名] が含まれているか、 [項目名が存在しない] か指定します。
ヘッダーサイズ	テーブルヘッダーとして省略される行数に、 [ヘッダーサイズ] を設定します。
文字セット	テーブルファイルで使用される文字セットに設定します。
タブサイズ	1つのタブ文字がテーブルファイルで示すスペースの数に設定します。
レコード行のサイズ	1つのレコードがテーブルファイルで占めるライン数に設定します。デフォルトは1です。

HTML ファイル

HTML ファイルには複数のテーブルを含めることができます。Qlik Sense は、<TABLE> タグにより、1つのテーブルとしてすべての要素を解釈します。

ファイル形式の設定

HTML ファイルのファイル形式の設定

UI 項目	説明
項目名	テーブルに [埋め込まれた項目名] が含まれているか、 [項目名が存在しない] か指定します。
文字セット	テーブルファイルで使用される文字セットを設定します。

XML ファイル

XML 形式で保存されているデータをロードできます。

XML ファイルには特定のファイル形式の設定はありません。

3 データ マネージャーを使用したデータのロードと管理

QVD ファイル

QVD 形式で保存されているデータをロードできます。QVD はネイティブの Qlik 形式で、Qlik Sense または QlikView でしか書き込みと読み出しを行えません。ファイル形式は、Qlik Sense スクリプトからデータを読み取る際の速度について最適化されていますが、それでも非常にコンパクトです。

QVD ファイルには特定のファイル形式の設定はありません。

QVX ファイル

Qlik data eXchange (QVX) 形式で保存されたデータをロードできます。QVX ファイルは、Qlik QVX SDK で開発されたカスタム コネクタにより作成されます。

QVX ファイルには特定のファイル形式の設定はありません。

KML ファイル

KML 形式で保存されているマップ ファイルをロードし、マップ ビジュアライゼーションで使用できます。

KML ファイルには特定のファイル形式の設定はありません。

前の手順に戻る ([データの追加])

データを追加するときに、前の手順に戻ることができます。

次の手順を実行します。

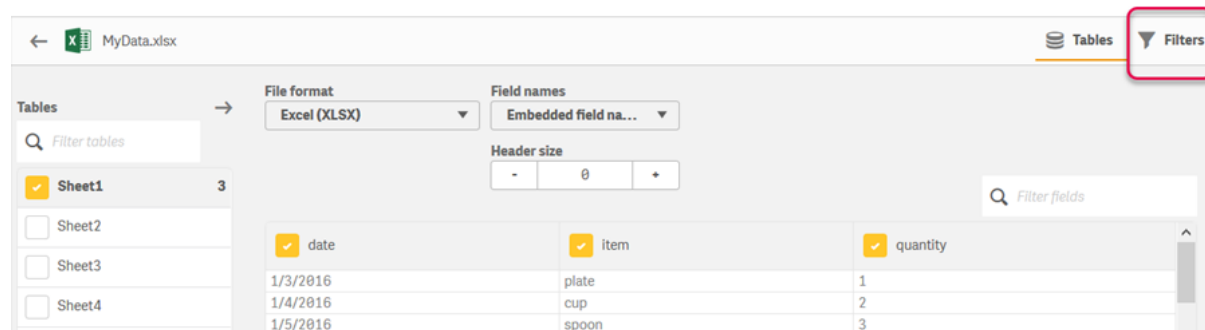
- 戻る矢印をクリックして、[データの追加] の前の手順に戻ります。

ファイルのデータのフィルタリング

ファイルからデータを追加する際のフィルター条件を作成することができます。これにより、ロードするデータのサブセットを選択できます。データのロード量を軽減する場合や、特定のデータのみ (4 万ドル以上の売上のみなど) を使用する場合に、この機能が役立ちます。

[データの追加] の手順で初めてファイルからデータを追加する場合、[フィルター] をクリックしてフィルター条件を適用できます。

[フィルター] ボタン



その後は、データ マネージャーでテーブルをクリックし、[このテーブルを編集] をクリックして、条件を変更できます。[ソースからデータを選択] をクリックし、[フィルター] をクリックします。

3 データ マネージャーを使用したデータのロードと管理

データをアプリに追加してフィルター条件を適用すると、データをアプリ内にロードすることができます



より高度なデータ操作の場合には、[データロードエディター]を使用します。詳細については、スクリプト作成チュートリアルで初心者用スクリプトおよびその次のステップを参照してください。

利用可能なフィルターの条件は以下のとおりです。

- =
- >
- <
- >=
- <=

データをフィルタリングする際には以下を考慮します。下記に具体例を示します。

- フィルター条件は、数値、日付、またはテキストに対して適用できます。
- ワイルドカード文字はサポートしていません。
- 条件は、複数適用できます。ただし、同じ項目について条件が競合する場合、データが返されない場合があります。
- 条件はアルファベット順にテキストデータに適用されます。条件では、大文字と小文字が区別されます。
- テキストデータに対して2文字以上の文字を使用できます。たとえば、>ctは>=cuと同様にcupという語を返します。>cもcupという語を返します。
- =条件を複数使用すると、すべての条件がtrueと評価されなければ値は返されません。ただし、同じ項目について複数の=条件を使用すると、trueと評価されるすべての値が返されます。
- <条件と>条件を組み合わせると、すべてがtrueと評価されなければ値は返されません。これらの条件と=を組み合わせると、すべての条件がtrueと評価されなければなりません。
- <=条件と>=条件を組み合わせると、すべてがtrueと評価されなければ値は返されませこれらの条件と=を組み合わせると、すべての条件がtrueと評価されなければなりません。
- QVD ファイルの日付項目に対するフィルターはサポートされていません。

例

次の例では、1つの項目(テーブル内の1列)に含まれるcup、fork、およびknifeという値を使用します。

- 条件:
 - =cup
 - =fork
 - =knife
 - 戻り値: cup、fork、knife
 - 等号条件は、trueであるすべての値を返します。
- 条件:
 - >b
 - <d

- 戻り値: *cup*
- 文字 *c* は、*b* より大きく、かつ *d* より小さくなります。
- 条件:
 - *<b*
 - *>d*
 - 戻り値: 値なし
 - *b* より小さく、かつ *d* より大きい値は存在しません。
- 条件:
 - *=fork*
 - *>g*
 - 戻り値: 値なし
 - *fork* と等しく、かつ *g* より大きい値はありません。

データ コネクタからのデータのフィルタリング



データのフィルタリングは、すべてのコネクタで使用できません。

データフィルター数式を入力するには、[ロードするデータの選択] ステップの [データのフィルタリング] を選択します。[データのフィルタリング] を選択すると、フィルター数式を入力するテキストボックスが開きます。例:

Sales >= 40000

[データのフィルタリング] によって、[売上] など個別の項目が選択されます。このオプションは SQL WHERE 節として機能します。WHERE 節で使用されるほとんどの演算子とキーワードは、[データのフィルタリング] と共に使用できます。有効な演算子は次のとおりです。

- =
- >
- >=
- <
- <=
- IN
- BETWEEN
- LIKE
- IS NULL
- IS NOT NULL

Qlik Sense により、[データのフィルタリング] で入力した数式から、データロードスクリプトに WHERE 節が作成されます。

複数の項目に対してフィルタリングしたい場合など、AND 演算子を使用して演算子を組み合わせることができます。例:

Sales <= 30000 AND RegionID = 45

OR 演算子は、いずれかの条件に一致するデータをフィルタリングするために使用できます。例:



`Name = 'Smith' OR Name = 'Jones'`

IN 演算子を使用して同じ結果を得ることができます。IN 演算子は、複数の OR 条件を使用する簡単な方法です。例:

`Name IN ('Smith', 'Jones')`

3.12 データ テーブルの編集

アプリに追加したテーブルは、**[データ マネージャー]** の概要で編集し、テーブル名の変更、テーブルの他のテーブルへの関連付けや項目の変換を行うことができます。

テーブルを編集するには、**[データ マネージャー]** でテーブルを選択して、 をクリックします。テーブル エディタが開き、テーブルのデータのプレビューが表示されます。各項目には、変換オプションを含む項目メニューがあります。 をクリックして項目メニューを開きます。項目を選択するとデータプロファイリングカードパネルが表示されます。パネルには、項目データの概要やその他の変換オプションが含まれます。



データの中に、ロードされるすべての項目で等しいデータがレコードが含まれている場合、それらのレコードは単一のレコードとしてプレビューテーブルに表示されます。

テーブル名の変更

[データ マネージャー] で追加したテーブルは、データベース テーブルやデータファイル、Excel ワークシートなどの名前に基づき、既定名が割り当てられます。この名前がわかりづらい場合や不適切な場合は変更しても構いません。

次の手順を実行します。

1. テーブル名をクリックします。
2. テーブル名を編集します。
3. エンターキーを押すか、テーブル名の外側をクリックします。

これで、テーブル名が変更されます。

項目名の変更

テーブル内の項目の名前を変更して、簡単に判別できる適切な名前を作成することができます。

次の手順を実行します。

1. 名前を変更する項目をクリックするか、項目メニューで**[名前変更]**を選択します。
2. 新しい名前を入力します。



項目名は、固有の名前にする必要があります。複数のテーブルで同じ名前の項目が存在する場合は、データの追加時に **Qlik Sense** が項目名を修飾します。つまり、プレフィックスとしてテーブル名を追加します。

3. エンターキーを押すか、項目の外側をクリックしてください。

これで項目名が変更されました。

他のテーブルとの関連付けの管理

項目メニューの **[関連付け]** により、他のテーブルの項目とのカスタム関連付けを作成することができます。

通常これらは、提案に従わずにカスタム関連付けを作成することが必要となる場合に最も一般的なケースです。

- テーブルを関連付ける項目はわかっても、このテーブルペアのスコアが低すぎるため、提案の一覧に表示されません。
各テーブルの単一項目に基づいて関連付けを作成します。
- テーブルには共通項目が複数含まれており、それらの項目を使用して関連付けを作成する必要があります。
複合キーを作成します。

多くの場合では、**[関連付け]** ビューで関連付けを管理するのが簡単な方法です。

項目の種類と表示形式の変更

データが追加されると、**Qlik Sense** は各項目の種類を解釈します。現在サポートされている項目の種類は以下のとおりです。

- 一般
- 日付
- タイムスタンプ
- 地理データ

データが正しく解釈されなかった場合は、項目の種類を修正できます。日付やタイムスタンプ項目の入力および表示形式を変更することも可能です。

郵便地域などの、名前またはコードの形式の地理情報を含む項目は、**[地理データ]** 項目として指定されない限り、マッピングでは使用できません。

全体的な分析から項目を非表示にする

項目をシート表示または **Insights** から非表示にして、**データマネージャー** および **データロードエディター** でのみ使用可能にすることができます。たとえば、別の項目値の計算にのみ使用される項目値がある場合があります。これらの項目を非表示にして、シートのアセットパネルや **Insights** では使用できないようにし、**データマネージャー** では引き続き使用可能にすることができます。

詳細については、「**全体的な分析から項目を非表示にする (page 55)**」を参照してください。

データロード前のテーブル項目データの評価

テーブルのデータに NULL 値や外れ値などの品質問題の恐れがないか、データをロードする前に、**[サマリー]** データプロファイリング カードを使用して検査できます。**[サマリー]** カードでは、項目は軸、メジャー、一時項目に分類され、それぞれに異なるデータ概要が用意され、他のデータプロファイリング カードでさまざまな変換オプションが利用できるようになります。**[サマリー]** カードでメジャーとして設定された項目は、**[バケット]** カードを使用してグループ化できます。**[サマリー]** カードで軸に設定された項目は、**[順序]** カードでカスタム順序を適用することができます。複数のカテゴリーに分類できる項目の場合、項目が取り得る各カテゴリーサマリーの間で変換が可能です。

詳細については、「[データロード前のテーブル項目データの評価 \(page 56\)](#)」を参照してください。

テーブル内での項目値の置換

[置換] データプロファイリング カードを使用して、項目内の値を置換できます。**[置換]** カードを使用すると、項目の1つ以上の値を選択して別の値に置換できます。例えば、国名が完全な国名と省略形で含まれているデータセットでは、これらの国名を1つの共有値に置き換えることができます。

詳細については、「[テーブル内での項目値の置換 \(page 58\)](#)」を参照してください。

テーブル内の項目値を NULL に設定する

[NULL の設定] データプロファイリング カードを使用して、軸項目のユニーク値を NULL 値として処理するように設定することができます。**[NULL の設定]** カードを使用すると、テーブル項目から値を選択して、NULL に手動で設定できます。例えば、データセットが、X などの値を使って NULL を表す場合に **[NULL の設定]** カードを使用すると、Qlik Sense によってその値が NULL として処理されるように設定できます。また、**[NULL の設定]** カードを使って、不要な値を NULL に設定することでデータセットをクリーンアップすることもできます。

詳細については、「[テーブル内の項目値を NULL に設定する \(page 59\)](#)」を参照してください。

項目値にカスタム順序を設定

データによっては、軸の値をアルファベットや数字以外の順番で表示した方が効果的な場合があります。**[サマリー]** データプロファイリング カードで軸として設定された項目は、**[順序]** データプロファイリング カードを使用してデータにカスタム順序を適用し、ビジュアルイゼーションで項目データの既定組織を設定できるようになります。

詳細については、「[軸の値の順序のカスタマイズ \(page 60\)](#)」を参照してください。

テーブル内での項目の分割

[分割] データプロファイリング カードを使用して、既存の項目からの情報を新しい項目に抽出できます。**[分割]** カードを使用すると、1つの項目の内容を複数の項目に分割できます。例えば、住所を含む項目を分割して、郵便番号を抽出できます。この機能により、既存のデータの関連セクションを含む新しい項目を素早く作成できます。

詳細については、「[テーブル内での項目の分割 \(page 61\)](#)」を参照してください。

メジャーの範囲へのグループ化

[バケット] データプロファイリング カードを使用して、テーブルのメジャー項目の値を範囲にグループ化できます。
[バケット] カードを使用して、ユーザーが定義したバケットの項目値をグループ化し、新しい項目を作成してテーブルに追加できます。例えば、年齢を年齢範囲にグループ化してビジュアライゼーションで軸として使用することもできます。

項目変換の詳細の表示

項目で実行されている現在の操作と変換、およびそれらの順序を[詳細] ダイアログで表示できます。[詳細] では、項目の取得元、項目に加えられた変更、および変換が適用された順序について理解することができます。

詳細については、「データマネージャーでのテーブルおよび項目の変換の詳細の表示 (page 78)」を参照してください。

クロスタブのデータのピボットを無効にする

クロスタブ形式でデータをロードした場合、テーブルのピボットを無効にするのが一番良い方法です。つまり、テーブルの一部を行と入れ替えます。これにより、データの使用や、他のデータテーブルとの関連付けがしやすくなります。

Year	Europe	RoW
2007	234	567
2008	345	534

Year	Region	Sales
2007	Europe	234
2007	RoW	567
2008	Europe	345
2008	RoW	534

詳細については、「データマネージャーでクロスタブのデータのピボットを無効にする (page 67)」を参照してください。

データソースからのテーブルの更新

項目の選択をデータソースから変更しなければならない場合があります。たとえば、除外された項目を追加したり、追加した項目でデータソースが更新された場合がこれに該当します。このような場合、データソースからテーブルを更新できます。そのテーブルが[手動入力]で作成された場合は、テーブルデータを追加、編集、または削除するとともに、新しい行と列を追加することができます。詳しくは、「でのデータの手動追加 Qlik Sense (page 25)」を参照してください。

次の手順を実行します。

1. [ソースからデータを選択] をクリックします。
データ選択ウィザードが開き、現在の選択が表示されます。
2. 必要な変更を行います。
3. データプロファイリングを有効にした状態で[データの追加] をクリックします。

選択内容に応じて、テーブルが更新されます。

計算項目の追加

ロードされている項目データを修正または変換しなければならないことがよくあります。例えば、姓と名を連結して氏名(フルネーム)にする場合、製品番号の一部を抽出する場合、データ形式を変換したり2つの数字を乗算する場合などです。

こうしたケースに対処するには、計算項目を追加します。計算項目は、数式を使用して項目の結果を定義します。数式内では関数、項目および演算子を使用できます。参照できる項目は、編集集中のテーブル内にあるものに限られます。

テーブルのソート

テーブルの編集集中は、特定の項目を基準にテーブルをソートして、データの概要を把握しやすくすることができます。ソートは、一度に1つの項目でのみ実行できます。

次の手順を実行します。

- 項目メニューで、[ソート]を選択します。

この項目を基準にして、テーブルのデータが昇順でソートされます。降順でソートする場合は、もう一度[ソート]を選択します。



ソート順はロードされたアプリデータに保持されません。

元に戻す/操作のやり直し

テーブルへの編集を元に戻す/やり直すには、 および をクリックします。

元に戻した/やり直した操作の履歴は、[テーブル エディタ]を閉じるとクリアされます。

テーブル エディタでデータを関連付ける

[データマネージャー] テーブル エディタの項目メニューの[関連付け]により、他のテーブルの項目とのカスタム関連付けを作成することができます。

多くの場合では、[関連付け]ビューで関連付けを管理するのが簡単な方法です。




通常これらは、提案に従わずにカスタム関連付けを作成することが必要となる場合に最も一般的なケースです。



- テーブルを関連付ける項目はわかっても、このテーブルペアのスコアが低すぎるため、提案の一覧に表示されません。
各テーブルの単一項目に基づいて関連付けを作成します。
- テーブルには共通項目が複数含まれており、それらの項目を使用して関連付けを作成する必要があります。
複合キーを作成します。

単一項目を使用した関連付けの作成

2つのテーブルに関連データが含まれていても、その関連付けが提案として表示されない場合は、テーブルエディタでカスタム関連付けを定義することができます。これにより、テーブルを関連付けるキー項目が作成されます。

次の手順を実行します。




1. データマネージャーの概要から、関連付けるテーブルのいずれかで  をクリックします。
テーブルエディタが開きます。
2. キー項目で使用する項目の項目メニューで、**[関連付け]** を選択します。
[テーブルの関連付け] エディタが開き、左側のテーブルで選択した項目のプレビューが表示されます。ここで、このテーブルと関連付ける項目を右側のテーブルで選択する必要があります。
3. **[テーブルの選択]** をクリックし、関連付けるテーブルを選択します。
4.  をクリックし、**+** 関連付ける項目を選択します。
右側のテーブルに、選択した項目のプレビューデータが表示されます。ここで、左側と右側のテーブルを比較し、含まれるデータが一致していることを確認できます。を使用してテーブルを検索すると、 より簡単に比較できます。
5. 作成されるキー項目の名前を **[名前]** に入力します。
どちらかのテーブルに含まれている既存の項目と同じ名前を使用することはできません。
6. **[関連付け]** をクリックします。

これで、キー項目を使用して、選択した2つの項目によってそれらのテーブルが関連付けられます。これはで示されます 。をクリックすると、 関連付けを編集または解除するためのオプションが表示されます。

複合キーの作成


関連付けを作成できる複数の共通項目が2つのテーブルに含まれている場合は、Qlik Senseにより、関連付けを処理するための合成キーが作成されます。これを修正するためにお勧めするのは、複合キーを作成する方法です。これは、関連付ける必要があるすべての項目を含んだカスタム関連付けを作成することによって行うことができます。

次の手順を実行します。

1. データマネージャーの概要から、関連付けるテーブルのいずれかで  をクリックします。
テーブルエディタが開きます。
2. 複合キー項目に含めるいずれかの項目の項目メニューで、**[関連付け]** を選択します。
[テーブルの関連付け] エディタが開き、左側のテーブルで選択した項目のプレビューが表示されます。
3.  をクリックして、**+** 複合キー項目に含める他の項目を追加します。
プレビューが更新され、複合キーのデータが表示されます。
ここで、右側のテーブルで、このテーブルで関連付ける項目を選択する必要があります。
4. **[テーブルの選択]** をクリックし、複合キー項目に含める項目を追加します。
5.  をクリックし、**+** 関連付ける項目を選択します。項目は、左側のテーブルの場合と同じ順序で選択する必要があります。
キーのデータを解釈しやすくするため、区切り文字を追加することもできます。

3 データマネージャーを使用したデータのロードと管理

右側のテーブルに、選択した項目のプレビューデータが表示されます。

ここで、左側と右側のテーブルを比較し、含まれるデータが一致していることを確認できます。を使用してテーブルを検索すると、より簡単に比較できます。

6. 作成されるキー項目の名前を**[名前]**に入力します。
7. **[関連付け]**をクリックします。

これで、複合キー項目を使用して、選択した項によってそれらのテーブルが関連付けられます。

制限事項



複合キーの使用に関しては、制限事項がいくつかあります。

- 連結テーブルで複合キーを作成することはできません。
- 複合キーで計算項目を使用すると、その計算項目の数式は複合キーの数式で展開されます。計算項目への参照は存在しないため、計算項目を編集しても複合キーは更新されません。

関連付けの編集

関連付けを編集し、名前を変更したり、関連付けられた項目を変更したりすることができます。

次の手順を実行します。



1.  をクリックして関連付けメニューを表示します。
2.  をクリックして関連付けを編集します。

[テーブルの関連付け] エディタが開き、関連付けの名前を変更したり、関連付けられた項目を変更したりすることができます。

関連付けの解除

2つのテーブル間で不要な関連付けを作成した場合は、関連付けを解除できます。

次の手順を実行します。

1.  をクリックして関連付けメニューを表示します。
2.  をクリックして関連付けを解除します。

これで、テーブルの関連付けが解除されます。

計算された項目の使用

ロードされている項目データを修正または変換しなければならないことがよくあります。例えば、姓と名を連結して氏名(フルネーム)にする場合、製品番号の一部を抽出する場合、データ形式を変換したり2つの数字を乗算する場合などです。

こうしたケースに対処するには、計算項目を追加します。計算項目は、数式を使用して項目の結果を定義します。数式内では関数、項目および演算子を使用できます。参照できる項目は、編集集中のテーブル内にあるものに限られます。計算項目内で別の計算項目を参照することができます。

計算項目は、データマネージャーのテーブルエディタで追加および編集します。

計算項目を追加する

次の手順を実行します。

1. **[項目を追加]** をクリックし、**[計算項目]** を選択します。
[計算された項目を追加] するためのエディタが開きます。
2. **[名前]** に計算項目の名前を入力します。
3. **[数式]** で計算項目の数式を定義します。これには 2 種類の方法があります。
 - ***fx*** (**[関数]**)、**[項目]** および **[演算子]** リストを使用し、アイテムを選択して数式に挿入します。
選択したアイテムは、**[数式]** 内のカーソル位置に挿入されます。
 - **[数式]** に計算項目の数式を入力します。
入力中には推奨される関数および項目に関するアシスタンスと関数構文が提供されます。計算された項目の結果の例は、**[プレビュー]** に表示されます。
4. **[作成]** をクリックして計算項目を作成し、計算項目エディタを閉じます。

計算項目を編集する

計算項目の名前は変更可能で、数式を編集することもできます。

次の手順を実行します。

1. 項目名の横にあるドロップダウンメニューから**[編集]** を選択します。
[計算された項目を更新] するためのエディタが開きます。
2. 必要に応じて**[名前]** にある計算項目の名前を編集します。
3. 計算項目の数式を編集します。
4. **[更新]** をクリックして計算項目を更新し、計算項目エディタを閉じます。

計算項目数式で利用可能な関数

計算項目数式を作成する際、下記の関数を利用できます。これは、データロードスクリプト内にある数式のサブセットです。数式が複数のレコードからのデータの集計になることはなく、レコード間関数を使用してその他のレコードにあるデータを参照することもできません。

計算項目数式で利用可能な文字列関数

これらの関数は、テキスト文字列形式におけるデータの修正や抽出に使用できます。

文字列関数

関数	説明
Capitalize	Capitalize() は、すべての単語の頭文字が大文字の文字列を返します。
Chr	Chr() は、指定された整数に対応する Unicode 文字を返します。
FindOneOf	FindOneOf() は、文字列を検索して、指定された文字のセットのいずれかの文字が出現する位置を取得します。3 番目の引数に 1 よりも大きい値が指定されていない限り、指定された文字のセットのいずれかの文字が最初に出現した位置が返されます。出現しない場合は、 0 が返されます。

3 データマネージャーを使用したデータのロードと管理

関数	説明
Index	Index() は、文字列を検索して、指定されたサブストリングが n 回目に出現する開始位置を取得します。 n の値は、オプションの 3 番目の引数で指定されます。省略されている場合は、 1 になります。負の値が指定された場合は、文字列の末尾から検索を行います。文字列内での位置は、 1 から順に番号が付けられます。
KeepChar	KeepChar() は、最初の文字列である 'text' で構成される文字列から、2 番目の文字列である "keep_chars" に含まれない文字を除いて返します。
[左]	Left() は、入力文字列の最初の (一番左にある) 文字で構成される文字列を返します。ここで、文字数は 2 番目の引数により決定されます。
Len	Len() は、指定された文字列の長さを返します。
Lower	Lower() は、指定された文字列のすべての文字を小文字に変換します。
LTrim	LTrim() は、指定された文字列を先頭のスペースを削除して返します。
Mid	Mid() は、2 番目の引数 'start' で定義された文字の位置で始まり、3 番目の引数 'count' で定義された文字数を返す入力文字列の一部を返します。'count' が省略されている場合、入力文字列の残りが返されます。入力文字列の最初の文字には、 1 が付けられます。
Ord	Ord() は、指定された文字列の最初の文字の Unicode コードポイント番号を返します。
PurgeChar	PurgeChar() は、2 番目の引数 ('remove_chars') に表示されるものを除き、入力文字列 ('text') に含まれる文字で構成される文字列を返します。
Repeat	Repeat() は、指定された文字列を、2 番目の引数で指定された回数分繰り返した文字列を返します。
Replace	Replace() は、指定された文字列内に含まれる指定されたサブストリングすべてを別のサブストリングで置き換えた文字列を返します。この関数は非再帰関数で、左から右へ処理されます。
[右]	Right() は、指定された文字列の末尾 (右端) から、2 番目の引数で指定された文字数の文字列を返します。
RTrim	RTrim() は、指定された文字列を末尾のスペースを削除して返します。
SubStringCount	SubStringCount() は、指定された文字列テキストに、指定されたサブストリングが出現する回数を返します。出現しない場合は、 0 を返します。
TextBetween	TextBetween() は、区切り文字として指定された文字間で行われる入力文字列でのテキストを返します。
Trim	Trim() は、指定された文字列を先頭と末尾のスペースを削除して返します。
Upper	Upper() は、数式のすべてのテキスト文字について、入力文字列のすべての文字を大文字に変換します。数字と記号は無視されます。

計算項目数式で利用可能な日付関数

Qlik Sense の日付および時刻関数は、日付と時間の値を転送、変換するために使用されます。

3 データマネージャーを使用したデータのロードと管理

関数は、1899年12月30日からの経過日数と等しい日時のシリアル値に基づいています。整数値は日付を表し、小数値はその日付の時刻を表します。

Qlik Sense は引数の数値を使用するため、日付や時刻として形式設定されていない場合でも、数値は引数として有効です。引数が文字列の場合など、数値ではない場合、Qlik Sense は、日付と時刻の環境変数に従って、その文字列の解釈を試みます。

引数で使用されている日付形式が、DateFormat システム変数で設定されている形式に対応していない場合、Qlik Sense は正しく解釈することができません。この問題を解決するには、設定を変更するか、変換関数を使用します。

日付関数

関数	説明
addmonths	この関数は、 startdate 後の n か月後の日付、または n が負の場合には startdate の n か月前の日付を返します。
addyears	この関数は、 startdate の n 年の日付、または n が負の場合には startdate の n 年前の日付を返します。
age	age 関数は、 date_of_birth に生まれた人の timestamp 時点での年齢 (満年齢) を返します。
convertlocaltime	UTC または GMT のタイムスタンプをデュアル値として現地時間に変換します。世界中の任意の都市、場所、タイムゾーンを指定できます。
day	この関数は、 expression の小数部が標準的な数値の解釈に従って日付と判断される場合に、日付を表す整数を返します。
dayend	この関数は、 time を含む日の最後のミリ秒のタイムスタンプに相当する値を返します。デフォルトの出力形式は、スクリプトに設定されている TimestampFormat です。
daylightsaving	UTC または GMT のタイムスタンプをデュアル値として現地時間に変換します。世界中の任意の都市、場所、タイムゾーンを指定できます。
dayname	この関数は、 time を含む日の最初のミリ秒のタイムスタンプに対応する数値を基底として、日付を表示する値を返します。
daynumberofquarter	UTC または GMT のタイムスタンプをデュアル値として現地時間に変換します。世界中の任意の都市、場所、タイムゾーンを指定できます。
daynumberofyear	この関数は、タイムスタンプの年の日番号を計算します。計算は、年の初日の最初のミリ秒から行われますが、最初の月を補正することもできます。
daystart	この関数は、 time 引数に含まれる日の最初のミリ秒で、タイムスタンプに対応する値を返します。デフォルトの出力形式は、スクリプトに設定されている TimestampFormat です。
firstworkdate	firstworkdate 関数は、 end_date までに no_of_workdays (月～金曜日) の日数に達するように、オプションで指定された休日 を考慮した最遅開始日を返します。 end_date および holiday は有効な日付またはタイムスタンプでなければなりません。

3 データマネージャーを使用したデータのロードと管理

関数	説明
GMT	この関数は、現在の Greenwich Mean Time を返します。これはシステム クロックと Windows の時刻設定から導かれます。
hour	この関数は、 expression の小数部が標準的な数値の解釈に従って時間と判断される場合に、時間を表す整数を返します。
inday	この関数は、 base_timestamp を含む日に timestamp が含まれている場合、 True を返します。
indaytotime	この関数は、 timestamp が base_timestamp のミリ秒単位まで正確に base_timestamp を含む日の範囲内にある場合、 True を返します。
inlunarweek	この関数は、 timestamp が base_date を含む週周期の範囲内か確認します。 Qlik Sense の週周期は、1月1日を週の最初の日と見なして定義されています。
inlunarweektodate	この関数は、 timestamp が base_date の最後のミリ秒までの週周期の範囲内か確認します。 Qlik Sense の週周期は、1月1日を週の最初の日と見なして定義されています。
inmonth	この関数は、 timestamp が base_date を含む月にある場合、 True を返します。
inmonths	この関数は、タイムスタンプが、ベース日付と同じ月、2か月、四半期、4か月、半年のいずれかの期間の範囲内か確認します。タイムスタンプがその前後の期間に該当するか確認することもできます。
inmonthstodate	この関数は、タイムスタンプが、 base_date の最後のミリ秒までの月、2か月、四半期、4か月、半年のいずれかの期間の範囲内か確認します。タイムスタンプがその前後の期間に該当するか確認することもできます。
inmonthtodate	basedate の最後のミリ秒まで basedate を含む月に date がある場合に True を返します。
inquarter	この関数は、 timestamp が base_date を含む四半期に含まれる場合、 True を返します。
inquartertodate	この関数は、 timestamp が base_date のミリ秒単位まで正確に base_date を含む四半期の範囲内にある場合、 True を返します。
inweek	この関数は、 timestamp が base_date を含む週にある場合、 True を返します。
inweektodate	この関数は、 timestamp が base_date のミリ秒単位まで正確に base_date を含む週の範囲内にある場合、 True を返します。
inyear	この関数は、 timestamp が base_date を含む年の範囲内にある場合、 True を返します。
inyeartodate	この関数は、 timestamp が base_date のミリ秒単位まで正確に base_date を含む年の範囲内にある場合、 True を返します。

3 データ マネージャーを使用したデータのロードと管理

関数	説明
lastworkdate	lastworkdate 関数は、オプションで指定された holiday を考慮した上で、 start_date に開始した場合に no_of_workdays (月 ~ 金曜日) の日数に達する最早終了日を返します。 start_date と holiday は、有効な日付またはタイムスタンプでなければなりません。
localtime	この関数は、指定されたタイムゾーンのシステムクロックから取得した現在の時刻のタイムスタンプを返します。
lunarweekend	この関数は、 date を含む週周期の最後のミリ秒のタイムスタンプに相当する値を返します。Qlik Sense の週周期は、1月1日を週の最初の日と見なして定義されています。
lunarweekname	この関数は、 date を含む週周期の初日の最初のミリ秒のタイムスタンプに対応する年と週周期番号を表示する表示値を返します。Qlik Sense の週周期は、1月1日を週の最初の日と見なして定義されています。
lunarweekstart	この関数は、 date を含む週周期の最初のミリ秒のタイムスタンプに相当する値を返します。Qlik Sense の週周期は、1月1日を週の最初の日と見なして定義されています。
makedate	この関数は、年 YYYY 、月 MM 、日 DD から算出された日付を返します。
maketime	この関数は、時間 hh 、分 mm 、秒 ss から算出された時間を返します。
makeweekdate	この関数は、年 YYYY 、週 WW 、曜日 D から算出された日付を返します。
minute	この関数は、 expression の小数部が標準的な数値の解釈に従って時間と判断される場合に、分を表す整数を返します。
month	この関数は、環境変数 MonthNames および 1 から 12 までの整数で定義されている月名を持つデュアル値を返します。月は標準的な数値の解釈に従って、数式の日付の解釈により計算されます。
monthend	この関数は、 date を含む月の最終日の最後のミリ秒のタイムスタンプに相当する値を返します。デフォルトの出力形式は、スクリプトに設定されている DateFormat です。
monthname	この関数は、月の初日の最初のミリ秒のタイムスタンプに対応する基底の数値を持つ、月 (MonthNames スクリプト変数に従った書式) および年の表示値を返します。
monthsend	この関数は、ベース日付を含む月、2 か月、四半期、4 か月、半年のいずれかの期間の最後のミリ秒のタイムスタンプに相当する値を返します。その前後の期間のタイムスタンプを取得することもできます。
monthsname	この関数は、期間の月の範囲 (MonthNames スクリプト変数に従った書式で表示) および年を表す表示値を返します。基底値は、ベース日付を含む月、2 か月、四半期、4 か月、半年のいずれかの期間の最初のミリ秒のタイムスタンプに相当する値です。

3 データマネージャーを使用したデータのロードと管理

関数	説明
monthsstart	この関数は、ベース日付を含む、月、2か月、四半期、4か月、半年のいずれかの期間の最初のミリ秒のタイムスタンプに相当する値を返します。その前後の期間のタイムスタンプを取得することもできます。
monthstart	この関数は、 date を含む月の初日の最初のミリ秒のタイムスタンプに対応する値を返します。デフォルトの出力形式は、スクリプトに設定されている DateFormat です。
networkdays	networkdays 関数は、オプションで指定された holiday を考慮した上で、 start_date と end_date の間の当日を含む作業日数 (月～金曜日) を返します。
now	この関数は、システムクロックから取得した現在の時刻のタイムスタンプを返します。デフォルト値は 1 です。
quarterend	この関数は、 date を含む四半期の最後のミリ秒のタイムスタンプに相当する値を返します。デフォルトの出力形式は、スクリプトに設定されている DateFormat です。
quartername	この関数は、四半期の初日の最初のミリ秒のタイムスタンプに対応する値を基底として、四半期の月数 (MonthNames スクリプト変数に従った書式) および年の表示値を返します。
quarterstart	この関数は、 date を含む四半期の最初のミリ秒のタイムスタンプに相当する値を返します。デフォルトの出力形式は、スクリプトに設定されている DateFormat です。
second	この関数は、 expression の小数部が標準的な数値の解釈に従って時間と判断される場合に、秒を表す整数を返します。
timezone	この関数は、Windows で定義されている現在のタイムゾーンの名前を返します。
today	この関数は、システムクロックから取得した現在の日付を返します。
UTC	現在の Coordinated Universal Time を返します。
week	この関数は、ISO 8601 に従って、週番号を表す整数を返します。週番号は標準的な数値の解釈に従って、数式の日付の解釈により計算されます。
weekday	この関数は、以下を持つデュアル値を返します。環境変数 DayNames で定義される日の名前。曜日に相当する 0 から 6 までの整数。
weekend	この関数は、 date を含む暦週の最終日 (日曜日) の最後のミリ秒のタイムスタンプに対応する値を返します。デフォルトの出力形式は、スクリプトに設定されている DateFormat です。
weekname	この関数は、 date を含む週の初日の最初のミリ秒のタイムスタンプに対応する数値を基底として、年と週番号を表示する値を返します。
weekstart	この関数は、 date を含む暦週の初日 (月曜日) の最初のミリ秒のタイムスタンプに対応する値を返します。デフォルトの出力形式は、スクリプトに設定されている DateFormat です。

3 データマネージャーを使用したデータのロードと管理

関数	説明
weekyear	この関数は、ISO 8601 に基づいた週番号が含まれる年を返します。週番号の範囲は、1 からおよそ 52 となります。
year	この関数は、 expression が標準的な数値の解釈に従って日付と判断される場合に、年を表す整数を返します。
yearend	この関数は、 date を含む年の最終日の最後のミリ秒のタイムスタンプに相当する値を返します。デフォルトの出力形式は、スクリプトに設定されている DateFormat です。
yearname	この関数は、 date を含む年の初日の最初のミリ秒のタイムスタンプに対応する数値を基底として、4 桁の年の表示値を返します。
yearstart	この関数は、 date を含む年の最初の日開始に対応するタイムスタンプを返します。デフォルトの出力形式は、スクリプトに設定されている DateFormat です。
yeartodate	この関数は、入力したタイムスタンプがスクリプトが最後にロードされた日付の年に該当するかどうかを算出し、該当する場合は True を返し、該当しない場合は False を返します。

計算項目数式で利用可能な形式設定関数および変換関数

書式設定関数は、入力式の数値を使用し、テキスト値に変換します。それに対して、変換関数は上記と正反対のことを行います。つまり、文字列式を取得し、それを数値として評価し、結果として返される数値の書式を指定します。どちらの場合においても、出力値はデュアル値 (テキスト値と数値) となります。

たとえば、**Date** 関数および **Date#** 関数における出力の差異を考慮します。

Date 関数および Date# 関数

関数	出力 (テキスト値)	出力 (数値)
Date# ('20140831', 'YYYYMMDD')	20140831	41882
Date (41882, 'YYYY.MM.DD')	2014.08.31	41882

これらの関数は、書式が Qlik Sense の日付書式設定に対応していないため、日付として解釈されない日付項目がデータに含まれている場合に有用です。このケースでは、関数をネストするのに役立ちます。

Date (Date# (DateInput, 'YYYYMMDD'), 'YYYY.MM.DD')

この場合、**DateInput** 項目は入力形式 **YYYYMMDD** に従って解釈され、指定の書式 (**YYYY.MM.DD**) で値を返します。

フォーマット関数および変換関数

関数	説明
Date	Date() は、データロードスクリプトのシステム変数またはオペレーティングシステムの書式設定、または提供されている場合は書式文字列を使用して、数式を日付として書式設定します。
Date#	Date# は、2 番目の引数で指定されている書式で日付として、数式を評価します。

3 データマネージャーを使用したデータのロードと管理

関数	説明
Dual	Dual() は、数値と文字列を組み合わせて1つのレコードにし、そのレコードの数値表現をソートや計算に使用できるようにする一方で、文字列値を表示に使用できるようにします。
Interval	Interval() は、データロードスクリプトのシステム変数またはオペレーティングシステムの書式、または提供されている場合は書式文字列を使用して、数値を時間間隔として書式設定します。
Interval#	Interval#() は、既定では、テキスト表現をオペレーティングシステムの書式設定の時間間隔として評価します。ただし、2番目の引数で書式が指定されている場合は、その書式の時間間隔として評価します。
Money	Money() は、データロードスクリプトのシステム変数またはオペレーティングシステムの書式設定 (書式文字列が提供されている場合を除く)、およびオプションの小数点記号と千の桁区切りを使用して数式を数字で金額値として書式設定します。
Money#	Money#() は、形式の文字列がある場合を除き、ロードスクリプトまたはオペレーティングシステムで設定された形式で、テキスト文字列を金額値に変換します。カスタムの小数点および桁区切り記号は、オプションのパラメータです。
Num	Num() は数値をフォーマットします。つまり、2番目のパラメーターで指定された形式を使用して、入力の数値をテキストを表示するように変換します。2番目のパラメーターを省略すると、データロードスクリプトで設定された10進数と1000進数の区切り文字が使用されます。カスタムの小数点および桁区切り記号は、オプションのパラメータです。
Num#	Num#() は、テキスト文字列を数値として解釈します。つまり、2番目のパラメーターで指定された形式を使用して、入力文字列を数値に変換します。2番目のパラメーターを省略すると、データロードスクリプトで設定された10進数と1000進数の区切り文字が使用されます。カスタムの小数点および桁区切り記号は、オプションのパラメータです。
Text	Text() は、数値として解釈できる場合でも、数式をテキストとして処理します。
Time	Time() は、書式文字列が提供されている場合を除き、データロードスクリプトのシステム変数またはオペレーティングシステムの時刻書式設定を使用して、数式を時刻値として書式設定します。
Time#	Time#() は、書式文字列が提供されている場合を除き、データロードスクリプトまたはオペレーティングシステムの時刻書式設定で、数式を時刻値として評価します。
Timestamp	TimeStamp() は、書式文字列が提供されている場合を除き、データロードスクリプトのシステム変数またはオペレーティングシステムのタイムスタンプ書式設定を使用して、数式を日付と時刻の値として書式設定します。
Timestamp#	Timestamp#() は、書式文字列が提供されている場合を除き、データロードスクリプトまたはオペレーティングシステムのタイムスタンプ書式設定で、数式を日付と時刻の値として評価します。

計算項目数式で利用可能な数値関数

数値関数を使用して数値を四捨五入します。

3 データマネージャーを使用したデータのロードと管理

数値関数

関数	説明
ceil	Ceil() は、 step (offset によりシフト)に最も近い倍数に数値を切り上げます。
div	Div() は、1番目の引数を2番目の引数で割り算して得られる整数部分を返します。パラメータは両方とも実数として解釈されるため、整数である必要はありません。
evens	Even() はTrue (-1)を返します。これは、 integer_number が偶数の整数またはゼロの場合です。これがFalse (0)を返すのは integer_number が奇数の整数である場合で、NULLを返すのは integer_number が整数ではない場合です。
fabs	Fabs() は、 x の絶対値を返します。結果は正の数値です。
fact	Fact() は、正の整数 x の階乗を返します。
floor	Floor() は、 step (offset によりシフト)に最も近い倍数に数値を切下げます。
fmod	fmod() は、1番目の引数(被除数)を2番目の引数(除数)で割り算して得られる整数部分を返す、一般化モジュール関数です。結果は実数です。引数は両方とも実数として解釈されるため、整数である必要はありません。
frac	Frac() は、 x の小数部を返します。
mod	Mod() は、整数除算による負でない余りを返す、数学的モジュール関数です。1番目の引数は被除数で、2番目の引数は除数で、両方の引数とも整数でなければなりません。
odd	Odd() はTrue (-1)を返します。これは、 integer_number が奇数の整数またはゼロの場合です。これがFalse (0)を返すのは integer_number が偶数の整数である場合で、NULLを返すのは integer_number が整数ではない場合です。
round	Round() は、 offset によりシフトされた step の最も近い倍数で切り上げた/切り下げた結果を返します。
sign	Sign() は、 x が正の数か0、あるいは負の数かによって1、0、-1を返します。

計算項目数式で利用可能な条件分岐関数

条件分岐関数を使用して条件を評価し、条件値に応じて異なる答えを返します。

条件分岐関数

関数	説明
alt	alt 関数は、有効な数値表現を持つ最初のパラメータを返します。一致が見つからない場合は、最後のパラメータを返します。任意の数のパラメータを使用できます。
class	class 関数は、 class の間隔に最初のパラメータを割り当てます。結果は dual 値であり、 a<=x<b がテキスト値として含まれています(a と b はピンの上限と下限で、下限は数値で示されます)。
if	if 関数は、与えられた条件が True または False のどちらかに評価されるかによって異なる値を返します。
match	match 関数は、最初のパラメータを後続のすべてのパラメータと比較し、一致する数式の数値の位置を返します。比較では大文字と小文字が区別されます。

3 データマネージャーを使用したデータのロードと管理

関数	説明
mixmatch	mixmatch 関数は、最初のパラメータを後続のすべてのパラメータと比較し、一致する数式の数値の位置を返します。比較では大文字と小文字は区別されません。
pick	pick 関数は、リストの n 番目の数式を返します。
wildmatch	wildmatch 関数は、最初のパラメータとそれに続くすべてのパラメータと比較し、一致した数式の数値を返します。比較文字列では、ワイルドカード文字 (* と ?) を使用できます。 * は、任意の順序の文字と一致する必要があります。 ? は任意の 1 文字と一致します。比較では大文字と小文字は区別されません。

計算された項目の数式で利用可能な NULL 関数

NULL 値を返すか、または削除するには、以下の関数を使用できます。

NULL 関数

関数	説明
Null	Null 関数は、NULL 値を返します。
IsNull	IsNull 関数は、数式の値が NULL かどうかを検定します。NULL の場合は -1 (True)、NULL でない場合は 0 (False) を返します。

計算された項目の数式で利用可能な数学関数

数学計算には以下の関数を使用できます。

数学関数

関数	説明
e	この関数は、自然対数の底 e (2.71828...) を返します。
rand	この関数は、0 ~ 1 間の乱数を返します。この関数は、サンプルデータの作成に使用できます。

計算された項目の数式で利用可能な指数関数と対数関数

指数計算と対数計算には以下の関数を使用できます。

指数関数と対数関数

関数	説明
exp	自然対数の底 e を底として使用する自然指数関数 e^x 。結果は正の数値です。
log	x の自然対数。関数は、 $x > 0$ の場合にのみ定義されます。結果は数値で返されます。
log10	x の常用対数 (10 を底とする対数) です。関数は、 $x > 0$ の場合にのみ定義されます。結果は数値で返されます。
pow	x の y 乗を返します。結果は数値で返されます。
sqr	x の 2 乗 (x の 2 のべき乗)。結果は数値で返されます。
sqrt	x の平方根です。関数は、 $x \geq 0$ の場合にのみ定義されます。結果は正の数値です。

3 データマネージャーを使用したデータのロードと管理

計算された項目の数式で利用可能な分布関数

統計分布の計算には以下の関数を使用できます。

分布関数

関数	説明
CHIDIST	CHIDIST() は、 chi^2 分布の片側確率の値を返します。 chi^2 分布は、 ch^2 検定に関連付けられています。
CHIINV	CHIINV() は、 chi^2 分布の片側確率の逆関数の値を返します。
FDIST	FDIST() は、F 確率分布を返します。
FINV	FINV() は、F 確率分布の逆関数を返します。
NORMDIST	NORMDIST() は、指定された平均と標準偏差について、累積正規分布を返します。例えば、 <code>mean = 0</code> 、 <code>standard_dev = 1</code> の場合は、標準正規分布の値が返されます。
NORMINV	NORMINV() は、指定された平均と標準偏差について、累積正規分布の逆関数分布を返します。
TDIST	TDIST() は、スチューデント t 分布における確率を返します。ここでの数値は t の計算値であり、この値に対して確率が計算されます。
TINV	TINV() は、スチューデント t 分布の t 値を確率と自由度の関数として返します。

計算された項目の数式で利用可能な地理空間関数

地理空間データの処理にはこの関数を使用できます。

地理空間関数

関数	説明
GeoMakePoint	GeoMakePoint() は、スクリプトとチャートの数式で緯度と経度を使ってポイントを作成してタグ付けします。

計算された項目の数式で利用可能な色関数

色プロパティの設定と評価には以下の関数を使用できます。

カラー関数

関数	説明
ARGB	ARGB() は、チャートオブジェクトのカラープロパティを設定または評価する数式で使用されます。色は alpha のアルファ係数 (不透明度) を使用した、赤の要素 r 、緑の要素 g 、青の要素 b によって定義されます。
HSL	HSL() は、チャートオブジェクトのカラープロパティを設定、または評価する数式で使用されます。色は、 hue 、 saturation 、 luminosity の 0 ~ 1 の値で定義されます。
RGB	RGB() は、赤の成分 r 、緑の成分 g 、青の成分 b の 3 つのパラメータで定義された色のカラーコードに対応する整数を返します。これらの成分は、0 ~ 255 の整数値である必要があります。この関数を数式で使用して、チャートオブジェクトのカラープロパティを設定または評価できます。

3 データマネージャーを使用したデータのロードと管理

計算された項目の数式で利用可能な論理関数

論理演算の処理には以下の関数を使用できます。

論理関数

関数	説明
IsNum	数式を数値として解釈できる場合は -1 (True) 、それ以外の場合は 0 (False) を返します。
IsText	数式にテキスト表現がある場合は -1 (True) 、それ以外の場合は 0 (False) を返します。

計算された項目の数式で利用可能なシステム関数



システム、デバイスおよび Qlik Sense アプリのプロパティにアクセスするには、以下の関数を使用できます。

システム関数


関数	説明
OSuser	この関数は、現在接続しているユーザーの名前を含む文字列を返します。これは、データロードスクリプトおよびチャート式の両方で使用できます。
ReloadTime	この関数は、最後にデータロードを実行したときのタイムスタンプを返します。これは、データロードスクリプトおよびチャート式の両方で使用できます。

項目の種類を変更する

データが追加されると、Qlik Sense は各項目の種類を解釈します。現在サポートされている項目の種類は以下のとおりです。

-  基本設定
-  日付
-  タイムスタンプ
-  地理データ


データが正しく解釈されなかった場合は、項目の種類を修正できます。データやタイムスタンプ項目の入力および表示形式を変更することも可能です。

テーブル エディタを開くには、編集するデータテーブルで  をクリックします。

ただし、項目の種類や表示形式を変更できないケースもあります。


- 2 つ以上のテーブルを連結している場合がその一例です。
- すでに日付やタイムスタンプとして認識されている項目もこれに該当します。

日付およびタイムスタンプ項目が正しく認識されていることの確認

日付またはタイムスタンプ項目が日付やタイムスタンプとして認識されていない場合  ([基本設定] とマークされている)、解釈が正しいかどうか確認することができます。

次の手順を実行します。

3 データマネージャーを使用したデータのロードと管理

1. 項目の見出しの上にある  をクリックします。
データ形式のダイアログが開きます。
2. **[項目の種類]** を **[日付]** または **[タイムスタンプ]** に変更します。
3. **[入力形式]** で形式文字列を変更し、日付の解釈が正しく行われるようにします。ドロップダウン リストから既存の形式を選択することも、オリジナルの形式文字列を記述することもできます。





形式文字列では、単一引用符を使用できません。

4. アプリにあるデフォルト以外の表示形式を使用するには、**[表示形式]** で形式文字列を記述するか、既存の文字列を選択してください。
空白のままにしておくと、アプリに設定されているデフォルトの表示形式が採用されます。

日付およびタイムスタンプ項目の表示形式の変更

各アプリには、日付およびタイムスタンプ項目のデフォルトの表示形式が用意されています。日付およびタイムスタンプ項目の表示形式は、個別に変更することができます。


次の手順を実行します。

1. 項目の見出しの上にある  または  をクリックします。
データ形式のダイアログが開きます。
2. **[表示形式]** で形式文字列を変更します。ドロップダウン リストから既存の形式を選択することも、オリジナルの形式文字列を記述することもできます。

項目の種類を地理データに変更する

都市や国名、ISO 記号などの値を含む項目が地理データとして認識されない場合、その項目の種類を **[地理データ]** に変更することができます。

次の手順を実行します。

1. 項目の見出しの上にある  をクリックします。
データ形式のダイアログが開きます。
2. **[項目の種類]** ドロップダウン メニューから **[地理データ]** を選択します。
3. **[地理データ]** ドロップダウンメニューから地理データの種類を選択します。
[市区町村]、**[国]**、**[国コードISO2]** および **[国コードISO3]** のオプションがあります。ISO2 と ISO3 は、ISO 3166 (国の International Standards Organization コード) が提供するコードです。ISO2 には 2 文字のコード、ISO3 には 3 文字のコードが含まれます。たとえば、スウェーデンのコードは SE と SWE です。ISO コードを割り当てる場合、割り当てるコードと項目の値を一致させる必要があります。2 文字のコード値を含む項目に ISO3 を割り当てると、所在地座標が正しく割り当てられません。
4. **[市区町村]** データの場合、国の地理データを含むテーブルで関連項目を選択します。
関連する国項目が 1 つしかない場合もありますが、国の地理データを使用して、複数の項目を含めることができます。例えば、1 つの項目で、フランスなどの長い名前と、ISO2 などの国コードで国を指定する別の項目を持つことができます。ただし、項目は、**[地理データ]** として指定されている場合にのみ **[関連国]** リストに表示されます。
[地理データ] 国項目として指定されている値がない場合、項目を **[市区町村]** と指定した場合に **[関連国]** リストが表示されません。

3 データマネージャーを使用したデータのロードと管理

ユーザーによって、または Qlik Sense によって自動的に、項目が **[地理データ]** の項目の種類に割り当てられている場合、地理座標 (ポイントまたはポリゴンのいずれか) を含む項目はその項目に関連付けられます。座標を含む関連付け項目は、**[データモデルビュー]** で確認できます。このような座標は、**[マップ]** オブジェクトを使用するアプリで必要です。

郵便地域などの、名前またはコードの形式の地理情報を含む項目は、**[地理データ]** 項目として指定されない限り、マッピングでは使用できません。

[地理データ] の種類に割り当てられた項目は、メキシコや MX などの文字列値を保持しますが、これらが **[マップ]** オブジェクトで使用される場合、マッピング座標はポイントまたはポリゴンデータを含む項目から取得されません。

全体的な分析から項目を非表示にする

シート表示または **Insight Advisor** でビジュアライゼーションを構築するときに、使用しないデータ項目を非表示にすることができます。

たとえば、別の項目値の計算にのみ使用される項目値がある場合があります。このような項目を **データマネージャー** で非表示にして、シートのアセットパネルや **Insight Advisor** では使用できないようにし、**データマネージャー** および **データロードエディター** では引き続き使用可能にすることができます。これを使用して、関連情報のみを含め、現在の分析や洞察には冗長または不要な情報を削除し、読みやすく分析しやすくすることができます。


項目を非表示にするときに、計算での関連付けや使用法などの、項目に含まれる既存のすべての関係は維持されます。マスターアイテムや既存のチャートで項目を現在使用している場合は、引き続きその場所で使用できますが、新しいマスターアイテムやビジュアライゼーションでは、再度表示しない限り使用できなくなります。




非表示になっているすべての項目を **データマネージャー** で表示するには、**データロードエディター** に移動して自動生成セクションを開きます。非表示になっているすべての項目が `TAG FIELD <field name>` でリストされます。 `WITH '$hidden';`

分析から項目を非表示にする



次の手順を実行します。


1. 項目の見出しの上にある  をクリックします。
2. **[Analysis で非表示にする]** をクリックします。

シート表示および **Insight Advisor** で、この項目が非表示になります。非表示の項目では、項目の見出しの上に  が追加されます。

非表示の項目を表示する

次の手順を実行します。

1. 項目の見出しの上にある  または  をクリックします。
2. **[Analysis で表示]** をクリックしてください。

シート表示および Insight Advisor で、この項目が使用可能になります。項目の見出しの上にある  は削除されます。

データロード前のテーブル項目データの評価

Qlik Sense にロードする前に NULL 値または外れ値などの潜在的な品質問題がないかどうかデータを調べるには、**[サマリー]** データプロファイリング カードを使用してデータの概要を表示します。

また **[サマリー]** カードでは、軸やメジャーとして項目のデータを表示するなど、考えられるさまざまなデータの解釈を確認することができます。

[サマリー] カードには、**[データマネージャー]** でテーブルを編集し、テーブル項目を選択することによってアクセスできます。テーブルエディターで項目を選択すると、Qlik Sense でデータ型、メタデータ、および存在する値が調べられます。次にその項目は、軸、メジャー、または一時的項目のいずれかとしてカテゴリーに分類され、分析結果が **[サマリー]** カードに表示されます。入力されているデータを軸またはメジャーのどちらかとしてカテゴリーに分類可能な項目では、プレビューを切り換えて軸またはメジャーとしてその項目を表示できます。**[サマリー]** カードでのデータ項目のカテゴリーへの分類方法は、Qlik Sense ビジュアライゼーションでのその項目の使用方法には影響しませんが、他のデータプロファイリングカードにおいてそのデータ項目に対して選択できる変換オプションが決まります。

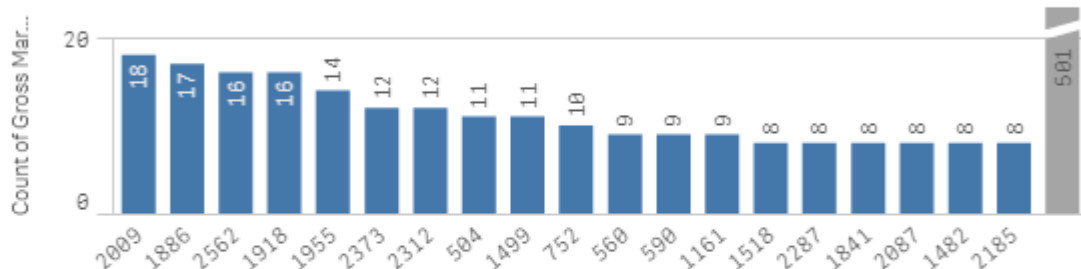
[サマリー] データプロファイリングカードを使用するデータの概要。

Gross Margin Summary (Measure)

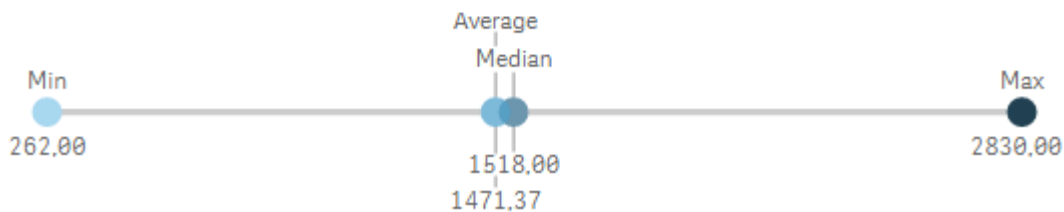


Distinct values: **225** Total values: **713**

Preview of distinct value distribution



Value Range



[サマリー] カードには次の情報が表示されます。

- **[ユニーク値]**: 項目内のユニーク値の数。
- **[値の総数]**: 項目内の値の数。
- **[ユニーク値分布のプレビュー]**: ユニーク値が 20 個より多い項目では、上位 20 個のユニーク値のみが表示されます。他のすべての値は、チャート上で 1 つの値にグループ化されます。すべての値がユニークである場合、棒グラフは表示されません。
- **[値範囲]**: (メジャーおよび一時的のみ) メジャー項目の場合、**[値範囲]** は、項目の最小値、中央値、平均値、および最大値を表示したチャートです。一時的項目の場合、**[値範囲]** は項目のデータによってカバーされる時間間隔です。
- **[NULL 値]**: 項目内の NULL 値の数。このビジュアライゼーションは、項目に NULL 値が含まれる場合にのみ表示されます。
- **[混合値]**: テキスト値と数値の両方を含む項目のテキスト値の数。このビジュアライゼーションは、項目に混合値が含まれる場合にのみ表示されます。

項目は、[サマリー] カードでカテゴリーに分類される方法に応じて、他のデータプロファイリングカードで変更可能です。メジャーとして設定されている項目には、[バケット] カードを使用して項目から作成されたグループ化値を含めることができます。詳細については、「メジャーの範囲へのグループ化 (page 64)」を参照してください。


軸として設定した項目には、以下を含めることができます。

- **[置換]** カードを使って他の値に置換されたユニーク値。
詳細については、「テーブル内での項目値の置換 (page 58)」を参照してください。
- **[NULL の設定]** カードを使って NULL 値に設定されたユニーク値。
詳細については、「テーブル内の項目値を NULL に設定する (page 59)」を参照してください。

- **[順序]** カードを使って値に適用されたカスタム順序。
詳細については、「**軸の値の順序のカスタマイズ (page 60)**」を参照してください。
- **[分割]** カードを使って、新しいテーブルの項目に分割された項目データ。
詳細については、「**テーブル内での項目の分割 (page 61)**」を参照してください。

[サマリー] カードへのアクセス

次の手順を実行します。

1. **[データマネージャー]** で、テーブルを選択し、をクリックします .
2. テーブルの項目を選択します。

[サマリー] カードが表示されます。

項目のデータカテゴリーの変更

Qlik Sense では、項目内のデータを複数のカテゴリーに分類することが可能であれば、項目のデータカテゴリーを変更できます。

次の手順を実行します。

- **[サマリー]** カードで、▼をクリックし、異なるデータカテゴリーを選択します。

テーブル内での項目値の置換

[置換] カードを使用すると、テーブルのさまざまな項目値を別の値に置き換えることができます。

データセット内の同じオブジェクトまたはコンセプトに対して異なる用語を使用できる場合があります。例えば、国名の完全名および省略形の両方が同じデータセット内にある場合があります。**[置換]** カードを使用すると、これらのインスタンスを異なるユニーク値ではなく、1つのユニーク値として処理されるようにすることができます。例えば、国データが入った項目で、**U.S.**、**US.**、**U.S.A** を **USA** に置き換えることができます。また、データセット内の名前を変更する必要がある場合などに、**[置換]** カードを使って個々の値を変更することもできます。


最大 5,000 個のユニーク値を含む項目で、値の置換を設定できます。

また、計算された項目では、置換値が無視され、代わりに元の値が使用されます。

[置換] カードは 2 つのセクションで構成されています。**[ユニーク値]** と **[置換値]** です。**[ユニーク値]** には、すべてのユニーク値と置換値が表示されます。**[置換値]** には、置換語を入力する項目、置換対象として選択した値のリストが含まれます。項目値を置換するには、ユニーク値を選択して、置換値を入力し、置換を適用します。また、**[ユニーク値]** で置換値を選択して編集することで、置換対象の値または置換に使われる値を変更できます。置換を適用する前に、複数の置換値を追加または編集することができます。

値の置換

次の手順を実行します。

1. **[データマネージャー]** で、テーブルを選択し、 をクリックします。
2. 軸項目を選択します。

3. データプロファイリング カードの **[置換]** タブをクリックします。
4. **[ユニーク値]** で、置換するユニーク値を選択します。
5. **[置換値]** で、これらのユニーク値の新しい値を入力します。
6. **[置換]** をクリックします。

置換値の編集

置換値にさらに多くのユニーク値を追加したり、置換値からユニーク値を削除したり、置換値を変更したりすることができます。

置換値へのユニーク値の追加

次の手順を実行します。

1. **[置換]** カードの **[ユニーク値]** で、置換値を選択します。
2. **[ユニーク値]** で、追加するユニーク値を選択します。
3. **[置換]** をクリックします。

置換値からのユニーク値の削除

次の手順を実行します。

1. **[置換]** カードの **[ユニーク値]** で、置換値を選択します。
2. 削除する各ユニーク値の横にある **✕** をクリックします。
3. **[置換]** をクリックします。

置換値の変更

次の手順を実行します。

1. **[置換]** カードの **[ユニーク値]** で、置換値を選択します。
2. **[置換]** で、新しい値を入力します。
3. **[置換]** をクリックします。

置換値の削除

置換値を削除する場合は、関連するすべてのユニーク値を削除します。

次の手順を実行します。

1. **[置換]** カードの **[ユニーク値]** で、置換値を選択します。
2. **[すべて削除]** をクリックします。
3. **[置換]** をクリックします。

テーブル内の項目値を NULL に設定する

Qlik Sense で NULL 値として処理するユニーク値を **[NULL の設定]** カードの軸項目から選択します。

3 データマネージャーを使用したデータのロードと管理

例えば、NULL として X などの表現がソースデータに含まれている場合、[NULL の設定] カードを使って、その値が Qlik Sense で NULL 値として処理されるように設定できます。空きスペースのある項目がテーブルに含まれている場合は、[NULL の設定] カードを使用してこれらの値を NULL 値として設定できます。また、不要な値のデータを NULL に設定してクリアする場合も、[NULL の設定] カードを使用できます。


NULL 値として特定の値を使用する場合は、[置換] カードを使用して、既定の NULL 値、-(Null) を置換できます。詳細については、「テーブル内での項目値の置換 (page 58)」を参照してください。

最大 5,000 個のユニーク値を含む項目で、項目値を NULL に設定できます。

[NULL の設定] カードは、[ユニーク値] と [手動での NULL 値] の 2 つのセクションで構成されています。[ユニーク値] から値を選択すると、その値は [手動での NULL 値] に追加されます。NULL 値を適用すると、選択した値のすべてのインスタンスは、項目のデータ内で NULL に設定されます。NULL として設定された値は、個別に復元することも、またはすべて復元することもできます。



項目値を NULL として設定する

次の手順を実行します。

1. [データマネージャー] で、テーブルを選択し、 をクリックします。
2. 項目を選択します。
3. データプロファイリング カードで、[NULL の設定] カードをクリックします。
4. [ユニーク値] で、NULL として設定する値を選択します。
5. [NULL 値の設定] をクリックします。

手動で NULL として設定された値を復元する

次の手順を実行します。

1. [データマネージャー] で、テーブルを選択し、 をクリックします。
2. 項目を選択します。
3. データプロファイリング カードで、[NULL の設定] カードをクリックします。
4. [NULL の設定] カードの [手動での NULL 値] で、次のいずれかを実行します。
 - NULL の設定を復元する値の後ろにある  をクリックします。
 - [すべて削除] をクリックして、NULL として設定したすべての値を復元します。
5. [NULL 値の設定] をクリックします。

軸の値の順序のカスタマイズ

カスタム順序により、ビジュアライゼーションでの軸の値の順序を設定できます。

多くのビジュアライゼーションで英字または数値の順序を使用できますが、場合によっては代替順序を利用できます。たとえば、市区町村を含む項目でカスタム順序を設定し、市区町村が東から西の順にチャートに表示されるようにすることができます。以下の要件を満たす項目のカスタム順序を定義できます。

3 データマネージャーを使用したデータのロードと管理


- 各項目は [サマリー] カードで軸として設定されていなければならない。
- 項目には最大 5,000 個のユニーク値が含まれていなければならない。
項目に含める値は最大で 25 個にすることを推奨しますが、[順序] カードで最大 5,000 個のユニーク値を含む項目を使用するよう選択できます。

[順序] カードは 2 つのセクションで構成されています。[現在の順序] と [順序のプレビュー] です。[現在の順序] には、軸のすべてのユニーク値が表示されます。ユニーク値は、デフォルトでロード順序に従って整理されます。カスタム順序は、[現在の順序] の値を必要な順序までドラッグすることによって設定します。[順序のプレビュー] は、各ユニーク値の値の数が現在の順序に従って整理されて表示される棒グラフです。

カスタム順序は、ロード順序によるソートを除いて、Qlik Sense のビジュアライゼーションで使用可能な他のすべてのソートオプションより優先されます。この項目で英字または数値順序が必要な場合は、順序をリセットすることによってカスタム順序を削除する必要があります。

軸項目の値の順序の変更

次の手順を実行します。

1. [データマネージャー] で、テーブルを選択し、 をクリックします。
2. 軸項目を選択します。
3. データプロファイリング カードの [順序] タブをクリックします。
4. 値をクリックして新しい順序にドラッグします。
5. 現在の順序をキャンセルするには、[キャンセル] をクリックします。




[キャンセル] は、新しいカスタム順序に対してのみ有効です。既存のカスタム順序の値の順序を変更する場合には、その変更をキャンセルするには、テーブルで別の項目を選択してからこの項目をもう一度選択します。

6. [順序変更] をクリックします。

軸項目の値の順序のリセット

次の手順を実行します。

1. [データマネージャー] で、テーブルを選択し、 をクリックします。
2. 順序変更された軸項目を選択します。
3. データプロファイリング カードの [順序] タブをクリックします。
4. [リセット] をクリックします。
5. [OK] をクリックします。

テーブル内での項目の分割

[分割] カードで、既存の項目のデータを利用して新しい項目を作成できます。

例えば、住所を含む項目を分割して、郵便番号のみを含む新しい項目を作成できます。この機能により、既存のデータのセグメントを含む新しい項目を素早く作成できます。[分割] カードでは、以下の要件を満たすテーブルの項目から新しい項目を作成できます。

3 データマネージャーを使用したデータのロードと管理

- 各項目は [サマリー] カードで軸として設定されていなければならない。



日付と時刻の情報を含むテーブル項目は、そのテーブルが [データマネージャー] で準備され、[分割] カードを必要としない場合は、自動的に日付項目に分割されます。

[分割] カードは、テンプレート値を含む入力項目、および新しい項目とその値のプレビューで構成されています。デフォルトでは、テンプレート値は項目の数値順での最初の値ですが、ソース項目から他の値を選択して、テンプレートとして使用することができます。テーブル内のすべての値の典型となる値を選択する必要があります。外れ値をテンプレートとして使って分割すると、新しい項目の質に影響する可能性があります。

項目を分割するには、テンプレート値 (項目を分割したい位置) に分割 マーカーを挿入します。分割 マーカーを追加するには、分割 マーカーを追加するサンプル項目内のポイントを選択し、選択内容を調整して、インスタンスで分割するか位置で分割するかを選択します。[分割] カードにより、推奨分割 マーカーがテンプレート値に自動的に追加できます。

インスタンスとは、文字 @ や文字間のスペースなどの、選択された区切り文字の出現箇所です。インスタンス分割 マーカーの位置は、以下のいずれかに対して相対的です。

- 値内の @ の最初のインスタンスなど、値の先頭。
- @ の後に続く、最初のインスタンスなど、別のインスタンスまたは位置分割 マーカーの右側の位置。

もう1つのインスタンスに相対するインスタンスを削除した場合、もう一方のインスタンスの位置は、分割 マーカーまたは値の先頭として設定された異なる区切り文字の次のインスタンスに相対する同じ位置に調整されます。項目は、最高 9 つの区切り記号を用いて分割できます。



[分割] カードは、分割 マーカーに指定された文字を使って値を分割します。アクセント記号付き文字など、データ内のこれらの文字の構成に相違がある場合、これらの相違は分割に含まれません。

位置とは、最初の 4 文字の後などの項目値内の場所です。位置は、以下のいずれかに対して相対的です。

- 値の先頭。
- インスタンス分割 マーカーの後の 2 番目の文字など、インスタンス分割 マーカーの右側の位置。

右側に位置するインスタンスを削除すると、その位置は値の先頭、または値の左か先頭にある最初の次のインスタンス分割 マーカーに対して相対的な同じ位置に移動します。

分割 マーカーを追加すると、項目のプレビューが更新され、新しい項目とそのデータが表示されます。項目のプレビューで、新しい項目の名前を変更できます。また、分割を適用する前に、分割項目をテーブルに追加したり、テーブルから削除したりすることができます。分割を適用すると、項目のプレビューで選択した項目がテーブルに追加されます。

項目は複数回分割できます。


項目を新しい項目に分割するには、以下の手順を実行します。

1. [分割] カードにアクセスします。
2. 入力項目に分割 マーカーを設定します。

3. また、次のタスクを任意で実行することもできます。
 - 分割 マーカーを削除する。
 - テーブルに追加する新しい項目を選択する。
 - 新しい項目の名前を変更する。
4. 新しい項目を作成します。

[分割] カードへのアクセス

次の手順を実行します。

1. [データマネージャー] で、テーブルを選択し、 をクリックします。
2. 軸項目を選択します。
3. データプロファイリング カードで、[分割] タブをクリックします。

分割 マーカーの挿入

分割 マーカーを挿入するには、サンプル値内の位置をクリックして、適用する分割の種類を選択します。



入力項目に表示されたサンプル値を変更する場合は、▼ をクリックして別の値を選択します。


次の手順を実行します。

1. [分割] カードで、サンプル値内で分割 マーカーを追加する位置をクリックします。
シングルクリックすると、その他の分割 マーカーにいたるまでのテンプレート値がすべて選択されます。
ダブルクリックすると、テンプレート値内でのカーソルの挿入ポイントが選択されます。
2. 選択内容を調整するには、選択 タブをクリックしてドラッグするか、または選択するセクションをハイライトします。
3. 適用する分割の種類に対応するボタンをクリックします。
 - **このインスタンス:** 区切り文字の選択されたインスタンスで項目を分割します。
 - **すべてのインスタンス:** 区切り文字のすべてのインスタンスで項目を分割します。
 - **これらの位置:** 選択のいずれかの側で項目を分割します。
 - **この位置:** この位置で項目を分割します。

テンプレート値に分割 マーカーが挿入されます。

分割 マーカーの削除

次の手順を実行します。

- [分割] カードで、次のいずれかを実行します。
 - 分割 マーカーを1つ削除する場合は、削除する分割 マーカーの上にある をクリックします。
 - 分割 マーカーをすべて削除する場合は、[リセット] をクリックします。
推奨分割 マーカーは削除されず、削除された推奨分割 マーカーが再びテンプレートに追加されます。これらを個別に削除する必要があります。

テーブルに追加する項目の選択

[分割] カードで作成した項目について、テーブルに含める項目またはテーブルから除外する項目を選択できます。既定ではすべての分割項目が含まれています。

次の手順を実行します。

- **[分割]** カードの項目のプレビューで、次のいずれかを実行します。
 - 項目を含める場合は、その項目の列のチェックボックスを選択します。
 - 項目を除外する場合は、その項目の列のチェックボックスのチェックを外します。

新しい項目の名前の変更

次の手順を実行します。

- **[分割]** カードの項目のプレビューにある項目ヘッダーに、新しい項目名を入力します。

新しい項目の作成

次の手順を実行します。

- **[分割]** カードで**[分割]** をクリックします。

メジャーの範囲へのグループ化

メジャーを範囲にグループ化し、ビジュアライゼーションの軸として使用することにより、メジャーから分析のためのインサイトをさらに得ることができます。

たとえば、項目に顧客の年齢が入力されている場合は、値を年齢の範囲にグループ化することができます。**[バケット]** データプロファイリング カードでは、項目データを範囲にグループ化し、指定したグループで新しい項目を作成することができます。**[バケット]** カードには、**[データ マネージャー]** でテーブルを編集し、**[サマリー]** カードでメジャーとして設定されているテーブル項目を選択することによってアクセスできます。

要件

メジャーは、以下の要件を満たす項目を使用して範囲へとグループ化することができます。

- 各項目が**[サマリー]** カードでメジャーとして分類されている
- 項目には 10 個以上のユニーク値が含まれている。
[バケット] カードは、低い範囲の値を含む項目には推奨されませんが、非推奨項目では**[バケット]** カードの使用を選択できます。
- 項目内の値は、20% を超える混合タイプを含むことはできません。
[バケット] カードは、混合タイプの値を含む項目には推奨されませんが、非推奨項目では**[バケット]** カードの使用を選択できます。



[バケット] カードによって作成されるバケット項目は、[サマリー] カードで軸としてカテゴリーに分類されます。ただしバケット項目には、[順序] カードを使用してカスタム順序を適用することはできません。バケット項目を計算された項目で使用することはできません。

概要

[バケット] カードには、提案された数のグループ化、グループ化のプレビュー、およびグループ化を含んだスライダーバーが表示され、スライダーバーにより各バケットの名前と値の範囲を変更することができます。提案されるグループ化の数は、新しい数を [バケット] 項目に入力することによって変更できます。Qlik Sense では、最大で 20 バケット、最小で 2 バケットがサポートされます。新しいバケットは、バケットが右から左に向かって削除される間に、バケット範囲の右側に追加されます。個々のバケットを何も変更していない場合、各バケットの範囲は、項目の値の範囲内に均等に適合するように調整されます。個々のバケットを変更した後にバケットの数を変更すると、新しいバケットがバケット範囲の右に追加され、右端から 2 番目のバケットのサイズと等しい値の範囲が指定されます。

[データバケットのプレビュー] 棒グラフには、各バケット内のユニーク値のカウント数と一緒にバケット内のデータの概要が表示されます。このチャートは、バケットに変更を加えると更新されます。バケットに値がない場合、棒チャートの棒は表示されません。

[バケット] スライダーバーを使用してバケットを変更することができます。バケットセグメントをクリックすることにより、バケットの範囲を設定するか、バケットの名前を変更するか、またはバケットを全体的に削除することができます。マウスカーソルをバケットに合わせると、バケットの名前と値の範囲が表示されます。バケットはデフォルトで、時間間隔の表記規則に従い、バケットの値範囲を付けて命名されます。バケットの範囲には、開始値と終了値の間の値 (終了値自体は除く) が含まれます。

1 つのバケットの値範囲を調整すると、Qlik Sense はすべてのバケットの値をずらし、他のバケットの既存の数量範囲を順守しながらギャップや重なりがないようにします。左端バケットには常に下側の境界がなく、右端バケットには上側の境界がありません。これにより、すべてのバケットの設定範囲に入らない可能性があるすべての値をキャプチャすることができます。バケットの下限範囲を変更するとバケットの範囲が右側に変更され、バケットの上限範囲を変更するとバケットが左側に変更されます。

1 つの項目からバケットを作成すると、新しい項目が生成され、ソース項目の対応するメジャー値が入力されている行に割り当てられているすべてのバケットが取り込まれます。デフォルトでは、<項目> (バケット化済み) と命名されます。この項目は、他のテーブルの項目と同様に、名前変更、関連付け、ソート、および削除することができます。テーブルで項目を選択し、バケットオプションを変更することにより、生成されるバケット項目でバケットを編集できます。同じソースメジャー項目から複数のバケット項目を作成できます。


メジャーのグループ化

メジャー値を範囲にグループ化するには、以下の手順を実行します。

1. [バケット] カードにアクセスします。
2. 任意で、バケットの数を変更します。
3. 任意で、個々のバケットを変更します。
4. 任意で、バケットに対する手動調整をデフォルト設定にリセットします。
5. バケット済み項目を作成します。

[バケット] カードへのアクセス

次の手順を実行します。

1. [データマネージャー] で、テーブルを選択し、 をクリックします。
2. 新しい値範囲のグループを作成する場合は、項目を1つ選択します。
3. 既存のバケット項目を編集する場合は、バケット項目を1つ選択します。

バケットの数の変更

バケットの数を変更すると、範囲の終わりにバケットが追加されるか、範囲の終わりからバケットが削除されます。

次の手順を実行します。

- [バケット] カードで、[バケット] の前に新しい数を入力します。

バケットの変更

バケットの名前変更、バケットの値範囲の調整、またはバケットの削除を行うことができます。



バケットを変更する場合は、左端のバケットセグメントから右端のバケットセグメントにかけて順番にバケットを変更するようお勧めします。

バケットの名前変更

次の手順を実行します。

1. [バケット] カードで、バケットセグメントをクリックします。
2. [名前] 項目に新しい名前を入力します。
3. 変更を適用するには、バケットセグメントの外側の任意の部分をクリックします。

バケットの値範囲の調整




10進数値を使用する場合は、**[開始値]** フィールドと**[最大値]** フィールドに手動で値を入力する必要があります。

次の手順を実行します。

1. [バケット] カードで、バケットセグメントをクリックします。
2. 以下のいずれかを行います。
 - **[開始値]** と**[最大値]** の後に、新しい値を入力してバケットの範囲を設定します。
バケットには、**[開始値]** と最大値の間の値が含まれますが、**[最大値]** として入力した値自体は除外されます。
 - セグメントのスライダーを調整してバケットの範囲を設定します。
3. 変更を適用するには、バケットセグメントの外側の任意の部分をクリックします。

バケットの削除

次の手順を実行します。

1. [バケット] カードで、バケットのセグメントをクリックします。
2.  をクリックします。

項目のバケットのリセット

バケットのリセットすることにより、バケットカードの設定を既定の状態に復元することができます。これがメジャー項目の場合、既定の状態は Qlik Sense で推奨されるバケット状態です。これがバケット項目の場合、既定の状態は、最後に [バケットの作成] がクリックされた後の項目の状態です。

次の手順を実行します。

- [バケット] カードで、[既定にリセット] をクリックします。

バケット項目の作成

次の手順を実行します。

1. [バケット] カードで、[バケットの作成] をクリックします。
2. [OK] をクリックします。


データのグループ化が入力された新しい項目がテーブルに追加されます。

データマネージャーでクロスタブのデータのピボットを無効にする

クロスタブは、ヘッダーデータが直交する2つのリストに値のマトリクスを持つ一般的なテーブルの種類です。

他のデータテーブルにデータを関連付けたい場合は、必ずしも最適なデータ形式ではありません。このトピックでは、クロスタブ形式でロードされたデータのピボットを無効にする方法、つまりデータマネージャーを使用して、データの一部を行と入れ替える方法を説明します。

クロスタブ形式でロードされたデータのピボットを無効にします。つまり、テーブルの一部を行と入れ替えます。



Year	Europe	RoW
2007	234	567
2008	345	534

Year	Region	Sales
2007	Europe	234
2007	RoW	567
2008	Europe	345
2008	RoW	534

クロスタブとは？

クロスタブには直接的な方法で読み取るいくつかの行見出しと、値のマトリクスが含まれます。この場合、行見出しとなる列 Year が1つあり、1か月当たりの販売データのマトリクスがあります。

3 データマネージャーを使用したデータのロードと管理

クロスタブ

Year	Jan	Feb	Mar	Apr	May	Jun
2008	45	65	78	12	78	22
2009	11	23	22	22	45	85
2010	65	56	22	79	12	56
2011	45	24	32	78	55	15
2012	45	56	35	78	68	82

このテーブルを単純に Qlik Sense にロードすると、**Year**(年) に 1 つ、そして各月に 1 つずつ項目が生成されます。ただし、通常はこのようなテーブルではデータを適切に管理できません。3 つの項目を生成したいはずです。

- 行見出しとなる項目、この場合では **Year** には、上のテーブルで緑色の印が付いています。
- この場合では月の名前 **Jan - Jun** で表される属性項目には、黄色の印が付いています。この項目には、**Month** と名前を付けることができます。
- データ項目には、青色の印が付いています。この場合、値は販売データを表すので、**Sales** と名前を付けることができます。


これをするには、データマネージャー テーブル エディタで **Unpivot** オプションを使用し、**Jan - Jun** 項目を選択します。これにより、次のテーブルが作成されます。

ピボットが無効化されたテーブル

Year	Month	Sales
2008	Jan	45
2008	Feb	65
2008	Mar	78
2008	Apr	12
2008	May	78
2008	Jun	22
2009	Jan	11
2009	Feb	23
...

クロスタブ テーブルのピボットを無効にしてフラットなテーブルにする

次の手順を実行します。

1. クロスタブ形式でデータファイルをアプリに追加します。
2. データマネージャーでテーブルの  をクリックしてテーブル エディタを開きます。
3. **[ピボットを無効]** をクリックします。

3 データ マネージャーを使用したデータのロードと管理

4. 行に入れ替えをする項目を選択します。ピボットが無効になっていない見出し項目が少なくとも1つは必要です。選択には、2つの方法があります。
 - 入れ替える項目の項目ヘッダーをクリックします。見出し項目として残す項目は選択しないでください。
 - 見出し項目として残す項目の項目ヘッダーをクリックし、フィールドメニューから**[選択を反転]**を選択します。入れ替える項目が多い場合はこれが最も簡単な方法です。
5. **[ピボットの無効を適用]** をクリックします。
選択されたデータが2つの項目 **TableName. [Attribute field]** と **TableName. [Data field]** の行に入れ替えられます。
6. 上述の例のように **Month**、**[Attribute field]** を意味を持つ名前に変更します。
7. 上述の例のように **Sales**、**[Data field]** を意味を持つ名前に変更します。

クロス集計はピボットが無効になりフラットな形式になりました。これでアプリの他のデータとの関連付けがしやすくなります。

元のクロスタブ テーブルに戻す

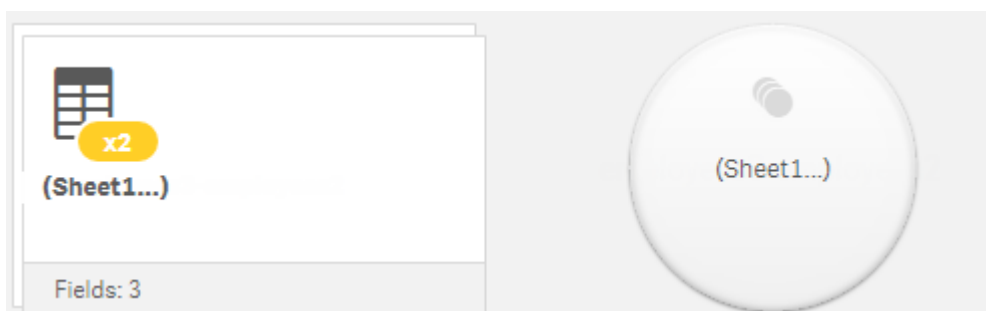
テーブル エディタの**[ピボットを無効を取り消す]** をクリックして、データソースの元のクロスタブ形式に戻すことができます。アプリの他のデータと関連付けをしていた場合は、その関連付けは削除されます。

3.13 データ マネージャーでのテーブルの連結

[連結] により、2つのテーブルを項目が組み合わせられた単一のテーブルに結合します。これによりコンテンツを統合して、コンテンツを共有する個別のテーブルおよび項目の数を減らします。データ マネージャーのテーブルは、自動的または強制的に連結することが可能です。

結合されたテーブルに含めるデータをもっと細かく制御する必要がある場合は、データ マネージャーでのテーブルの結合 ([page 73](#)) を参照して**結合演算子**について学習してください。

[テーブル] 表示と**[関連付け]** 表示で連結されたテーブル。



テーブルの自動連結

追加される1つ以上のテーブルについて、含まれる項目の数と項目名の両方が別のテーブルと同じであることが Qlik Sense によって検出されると、それらのテーブルはデータ マネージャーで自動的に連結されます。この場合 Qlik Sense は、複数のテーブルを単一のテーブルに自動連結します。自動連結されたテーブルは、誤って連結されていたり、連結を避けたい場合には、分割することができます。自動連結されたテーブルは、その他のテーブルに強制的に連結できます。

3 データ マネージャーを使用したデータのロードと管理

自動連結されたテーブルには次の制約があります。

- 項目のカテゴリは変更できません。
- 自動連結されたテーブルはピボット解除できません。
- **[ソースからデータを選択]** を使用してデータを追加または削除することはできません。

テーブル間の強制連結



[データ マネージャー]にあるテーブルは、同じ項目またはデータを全体的に共有していない場合でも、**[連結または結合]** パネルを使用して強制的に連結できます。強制連結により、マッピングを制御し、最終的な連結テーブルから項目を除外することができます。強制連結では、追加の変換がサポートされています。強制連結を使用することにより、以下の操作を実行できます。

- 連結されたテーブルを別のテーブルと連結する。
- ピボット解除されたテーブルを別のテーブルと連結する。強制的に連結されたテーブルはピボット解除できません。
- 計算された項目を含むテーブルを連結する。計算された項目は、強制連結で他の項目と連結できません。計算された項目は、強制連結されたテーブルに追加できます。



強制連結されたテーブルには次の制約があります。

- 強制連結では、連結されたテーブルに各テーブルの少なくとも1つの項目を含める必要がありますが、共にマッピングする必要はありません。
- 連結後に日付項目の書式設定を行うことはできません。日付項目には連結前と同じ書式を適用する必要があります。連結された日付項目では、**[データ ロード エディター]**の**[DateFormat]**で設定した既定の時間書式を使用します。
- 連結後に項目のカテゴリを変更することはできません。
- 連結されたテーブル内の別の項目にマッピングされている項目を参照する計算された項目には、連結された項目の結合済みのデータではなく、元の項目のデータのみが入力されます。連結されたテーブル内の項目を参照する、2つのテーブルの連結後に作成される計算された項目では、その項目内のすべてのデータが使用されます。
- **[ソースからデータを選択]** を使用して、連結されたテーブルにデータを追加または削除することはできません。ただし、**[データの追加]** をクリックし、ソース テーブルを選択して項目を除外することにより、項目を削除できます。削除された項目のデータとしてNULL値が追加されます。

[連結または結合] パネルにアクセスするには、*** で**[データ マネージャー]**、**[連結または結合]**の順にクリックし、2つのテーブルを選択します。**[連結または結合]** でテーブルを選択すると、Qlik Sense で項目が分析され、一致するすべての項目が自動的にマッピングされます。明確に一致する項目がない場合、項目は未マッピングのまま残されます。連結が適用されると、マッピングされた項目は連結されたテーブルで結合され、一方で未マッピング項目は、対応する値がない行にNULL値が入力された個別の項目として含められます。

[連結または結合] で選択された最初のテーブルは、プライマリテーブル、つまり他のテーブルが連結されるテーブルとして設定されます。連結されたテーブルでは、手動で名前を変更しない限り、プライマリ テーブルのテーブル名と項目名が使用されます。プライマリ テーブルは、 ボタンを使用して変更できます。**[連結または結合]** では、プライマリテーブルの項目を上段行に、セカンダリテーブルの項目を下段行にして、項目を2行に並べます。プライマリ テーブルとセカンダリ テーブルは  ボタンを使用して入れ替えることができます。

3 データマネージャーを使用したデータのロードと管理

[マッピングの編集]を使用して既定のマッピングを変更し、マッピングする項目を選択するか、未マッピングのままにするか、または連結されたテーブルから除外することができます。[マッピングの編集]には、マッピングを編集するためのドラッグアンドドロップインターフェイスと、テーブルのすべての項目が一覧表示される[項目]パネルが表示されます。項目は、プライマリーテーブルの項目の下までドラッグすることによってマッピングできます。項目は、[項目]パネルの項目の横にあるにより、または項目の上段行にドラッグすることにより、新しい未マッピング項目として追加できます。[項目]パネルでは、未マッピング項目にのマークが付きます。連結されたテーブルから削除された項目はそのテーブルに含められず、テーブルに連結が適用された後にQlik Senseで使用することはできません。




マッピングが適用され、テーブルが連結されると、それらのマッピングを編集することはできませんが、連結されたテーブルを分割してテーブルを元の状態に復元することによってテーブルから削除することができます。

[データマネージャー]で強制的にテーブルを連結するには、以下の手順を実行します。

1. 連結するテーブルを選択します。
2. 任意で、連結されたテーブルの名前と項目名を変更します。
3. 任意で、連結マッピングを編集します。
4. 連結演算子を選択します。
5. テーブルを連結します。

連結するテーブルの選択

次の手順を実行します。

1. [データマネージャー]の下段行で、をクリックします。
2. [連結または結合]をクリックします。
[連結または結合]パネルが開きます。
3. 2つのテーブルを選択します。
両方のテーブルの項目は、[連結または結合]パネルでマッピングされるか、未マッピングのままになります。
4. 各項目の一意の値のサンプルをプレビューするには、をクリックします。
5. プライマリーテーブルとセカンダリーテーブルを切り換えるには、をクリックします。



テーブル名と項目名の変更

次の手順を実行します。

1. [連結または結合]パネルで、テーブル名項目に新しいテーブル名を入力します。
2. [項目名]フィールドに、新しい項目名を入力します。

連結 マッピングの編集

次の手順を実行します。

1. **[連結または結合]** パネルで、**[マッピングの編集]** をクリックします。
2. 2つの項目をマッピングするには、プライマリー テーブル項目の下でテーブル項目をクリックしてドラッグします。
3. 新しい未マッピング項目を追加するには、テーブル項目をクリックして項目の上段行にドラッグします。
4. 連結されたテーブルから項目を削除するには、項目で  をクリックします。
5. 削除された項目をテーブルに戻すには、**[項目]** パネルで項目の横の  をクリックします。
6. **[マッピングの編集]** をクリックして **[マッピングの編集]** を閉じます。

連結演算子の選択

次の手順を実行します。

1. **[連結または結合]** パネルで、**[アクションを選択]** をクリックします。
2. リストから**[連結]**を選択します。

テーブルの連結

次の手順を実行します。



- **[連結または結合]** パネルで、**[適用]** をクリックします。

これでテーブルが連結されます。

連結されたテーブルの分割


Qlik Sense で不必要な自動連結が実行されたなど、連結が不要になった場合は、連結されたテーブルをそのソース テーブルへと分割することができます。



連結されたテーブルを分割すると、連結されたテーブルのすべての関連付けとともに、プライマリー テーブルとセカンダリー テーブルの相互の関連付けすべてが削除されます。連結されたテーブルを分割する際に関連付けを維持したい場合は、テーブルを分割する代わりに  をクリックして連結を元に戻します。データを **[データ マネージャー]** にロードした後に、 を使用して連結を元に戻すことはできません。

自動連結されたテーブルの分割

次の手順を実行します。

1. 連結されたテーブルを選択します。
2.  をクリックします。

3. 連結されたテーブルから分割するテーブルを選択します。
4. **[分割]** をクリックします。

これでテーブルがソーステーブルに分割され、ソーステーブルのすべての項目が修飾されます。修飾された項目の名前は、テーブル名の後にピリオド(文字「.」)で区切って項目名を続けた形式の名前に変更されます。


Table1 および Table2 には、項目 Field1 と Field2 が含まれています。**[データマネージャー]** でこれらの項目を追加すると、Table1-Table2 という名前のテーブルに連結されます(このテーブルには、項目 Field1 と Field2 が含まれます)。

Table1-Table2 テーブルを分割すると、次の2つのテーブルが生成されます。

- Table1 (項目 Table1.Field1 と Table1.Field2 を含む)
- Table2 (項目 Table2.Field1 と Table2.Field2 を含む)

強制連結されたテーブルの分割

次の手順を実行します。

1. 連結されたテーブルを選択します。
2.  をクリックします。

これでソーステーブルに分割されます。ソーステーブルのすべての項目とその項目は、連結前の名前となります。連結されたテーブルを分割すると、連結の1つのレベルのみが分割されるため、分割された連結テーブルの一部であったすべての連結されたテーブルでは独自の連結が保持されます。

3.14 データマネージャーでのテーブルの結合

Join は、2つのテーブルデータを手動で結合するために使用できる演算であり、選択した設定に応じてさまざまな結果を生成します。

これにより、連結よりも結合されたテーブルをより細かく制御できます。

Join 演算は、2つのテーブルを取得して1つに結合し、1つまたは複数の共通項目の共通の値の重複に基づいて、元のテーブル両方の項目を結合します。次の複数の演算子を、**Join** 演算に適用できます。**Outer**、**Inner**、**Left**、および **Right**。



結合されたテーブルは、その中で結合されているテーブルと同じぐらいのメモリ量を占有し続けます。結合されたテーブルを過度に使用すると、**Qlik Sense** の速度が低下する場合があります。**join** 演算によって除外される情報には、テーブルが分割されるまで **Qlik Sense** からアクセスできません。

Join 演算子

Join 演算子には次の4つがあります。**Outer join**、**Inner join**、**Left join**、および **Right join**。選択した演算子によって、重複する項目または値を含めるか除外するかを指定します。



Join 演算子が **Left** テーブルと **Right** テーブルを参照する場合、選択順にそれぞれ 1 番目と 2 番目のテーブルを参照します。

Outer join

一方または両方のテーブルに重複する項目値が表されている場合、**Outer join** 演算子は 2 つのテーブルの値について可能な組み合わせをすべて含めます。

First table

A	B
1	aa
2	cc
3	ee

Second table

A	C
1	xx
4	yy

結合されたテーブル

A	B	C
1	aa	xx
2	cc	-
3	ee	-
4	-	yy

Inner join

両方のテーブルに重複する項目値が表されている場合、**Inner join** 演算子は 2 つのテーブルの値の組み合わせのみを含めます。

First table

A	B
1	aa
2	cc
3	ee

3 データマネージャーを使用したデータのロードと管理

Second table

A	C
1	xx
4	yy

Joined table

A	B	C
1	aa	xx

Left join

最初のテーブルに重複する項目値が表されている場合、**Left join** 演算子は 2 つのテーブル値の組み合わせを含めます。

First table

A	B
1	aa
2	cc
3	ee

Second table

A	C
1	xx
4	yy

Joined table

A	B	C
1	aa	xx
2	cc	-
3	ee	-

Right join

2 番目のテーブルに重複する項目値が表されている場合、**Right join** 演算子は 2 つのテーブルの値の組み合わせを含めます。

First table

A	B
1	aa
2	cc
3	ee

Second table


A	C
1	xx
4	yy


Joined table


A	B	C
1	aa	xx
4	-	yy

テーブルの結合

[連結または結合] パネルにアクセスするには、*** で **[データマネージャー]**、**[連結または結合]** の順にクリックし、2 つのテーブルを選択します。**[テーブルの連結または結合]** でテーブルを選択すると、Qlik Sense で項目が分析され、一致するすべての項目が自動的にマッピングされます。明確に一致する項目がない場合、項目は未マッピングのまま残されます。結合が適用されると、マッピングされた項目が結合テーブルで組み合わせられます。マッピングされていない項目は、対応する値が存在しない行に null 値を持つ個別の項目として含まれるか、値のインスタンスが重複していない場合は完全に除外されます。

[テーブルの連結または結合] で選択された最初のテーブルは、プライマリーテーブルとして、つまり他のテーブルの結合先のテーブルとして設定されます。結合されたテーブルでは、手動で名前を変更しない限り、プライマリーテーブルのテーブル名と項目名が使用されます。プライマリーテーブルは、 ボタンを使用して変更できます。

[テーブルの連結または結合] では、プライマリーテーブルの項目を上段行に、セカンダリーテーブルの項目を下段行にして、項目を2行に並べます。プライマリーテーブルとセカンダリーテーブルは  ボタンを使用して入れ替えることができます。

[マッピングの編集] を使用して既定のマッピングを変更し、マッピングする項目を選択するか、未マッピングのままにするか、または結合されたテーブルから除外することができます。**[マッピングの編集]** には、マッピングを編集するためのドラッグアンドドロップインターフェイスと、テーブルのすべての項目が一覧表示される**[項目]**パネルが表示されます。項目は、プライマリーテーブルの項目の下までドラッグすることによってマッピングできます。項目は、**[項目]**パネルの項目の横にある  により、または項目の上段行にドラッグすることにより、新しい未マッピング

3 データ マネージャーを使用したデータのロードと管理

項目として追加できます。**[項目]** パネルでは、未マッピング項目に❗のマークが付きます。結合されたテーブルから削除された項目はそのテーブルに含まれていないので、テーブルに結合が適用された後に Qlik Sense で使用することはできません。

マッピングが適用され、テーブルが結合されると、マッピングされた項目を編集することはできませんが、結合されたテーブルを分割してテーブルを元の状態に復元することによってテーブルから削除することができます。

データ マネージャーでテーブルを結合するには、次の手順を実行します。

1. 結合するテーブルを選択します。
2. 任意で、結合されたテーブルの名前と項目名を変更します。
3. 任意で、マッピングを編集します。
4. Join 演算子を選択します。
5. テーブルを結合します。

結合するテーブルの選択

次の手順を実行します。

1. [データ マネージャー] の下段行で、*** をクリックします。
2. [連結または結合] をクリックします。
[連結または結合] パネルが開きます。
3. 2つのテーブルを選択します。
両方のテーブルの項目は、[テーブルの連結または結合] パネルでマッピングされるか、未マッピングのままになります。
4. 各項目の一意の値のサンプルをプレビューするには、👁️ をクリックします。
5. プライマリー テーブルとセカンダリー テーブルを切り換えるには、↔️ をクリックします。

テーブル名と項目名の変更


次の手順を実行します。

1. [連結または結合] パネルで、テーブル名項目に新しいテーブル名を入力します。
2. [項目名] フィールドに、新しい項目名を入力します。

マッピングの編集

次の手順を実行します。

1. [連結または結合] パネルで、[マッピングの編集] をクリックします。
2. 2つの項目をマッピングするには、プライマリー テーブル項目の下でテーブル項目をクリックしてドラッグします。
3. 新しい未マッピング項目を追加するには、テーブル項目をクリックして項目の上段行にドラッグします。
4. 結合されたテーブルから項目を削除するには、項目で✕をクリックします。

- 削除された項目をテーブルに戻すには、[項目] パネルで項目の横の  をクリックします。
- [マッピングの編集] をクリックして [マッピングの編集] を閉じます。

join 演算子の選択

次の手順を実行します。

- [連結または結合] パネルで、[アクションを選択] をクリックします。
- 次のリストから演算子を選択します。Outer join、Inner join、Left join、または Right join。

テーブルの結合

次の手順を実行します。



- [連結または結合] パネルで、[適用] をクリックします。

これでテーブルが結合されます。

結合 テーブルの分割


結合が不要になった場合、結合されたテーブルをソース テーブルに分割できます。



結合されたテーブルを分割すると、結合されたテーブルのすべての関連付けとともに、プライマリー テーブルとセカンダリー テーブルの相互の関連付けすべてが削除されます。結合されたテーブルを分割する際に関連付けを維持したい場合は、テーブルを分割する代わりに  をクリックして結合を元に戻します。データをデータ マネージャーにロードした後に、 を使用して結合を元に戻すことはできません。

結合 テーブルの分割

次の手順を実行します。

- 結合されたテーブルを選択します。
-  をクリックします。

これでソース テーブルに分割されます。ソース テーブルのすべての項目とその項目は、結合前の名前となります。結合されたテーブルを分割すると、結合の1つのレベルのみが分割されるため、分割された結合 テーブルの一部であったすべての結合されたテーブルでは独自の結合が保持されます。

3.15 データ マネージャーでのテーブルおよび項目の変換の詳細の表示

[詳細] ダイアログを使用して、データ マネージャーのテーブルと項目で行われる操作と変換を表示することができますようになりました。[詳細] ダイアログはテーブルの [関連付け] および [テーブル] ビューと、項目のデータ テーブル エディタで利用できます。

[詳細] に、選択したテーブルまたは項目に対して行った現在の操作と変換が、生成されたデータロードスクリプト内で適用される順序で表示されます。ここで、テーブルまたは項目のソース、行った現在の変更、変更が適用された順序を容易に確認できます。例えば、**[詳細]** を使用して、どのテーブルが連結されているか、項目の順序が調整されているかを容易に確認することができます。

[詳細] に表示される情報は、表示しているのがテーブルか項目かによって異なります。テーブルの **[詳細]** には以下が表示されます。

- 選択したテーブルのためのソース テーブル
- ピボット解除、連結など、テーブルで使用される変換

項目の **[詳細]** には以下が表示されます。

- 選択した項目のためのソース テーブルと項目
- 項目の型の変化
- データプロファイリング カードから、連結からなど、項目で使用される変換

テーブルの詳細の表示



次の手順を実行します。

- **[データマネージャー]** で、テーブルを選択し、をクリックして、*****[詳細を表示]** をクリックします。

[詳細] ダイアログが開きます。

項目の詳細の表示

次の手順を実行します。

1. **[データマネージャー]** で、テーブルを選択し、をクリックします .
2. 項目のヘッダーの上で  をクリックし、**[詳細の表示]** をクリックします。

[詳細] ダイアログが開きます。

3.16 強制連結を使用した段階的なテーブルの結合

ここでは、強制連結を使用して2つの類似したデータテーブルを結合する方法を段階的に説明します。

強制連結は、シートで分析のために使用する前にデータをクリーンアップする目的で使用できます。2つのテーブルを1つのテーブルに連結することができます。また、後で別のテーブルを追加することもでき、例えば、最初に **June** からテーブルを追加し、後で **July** から2つめのテーブルを追加することが可能です。

連結の概要

- 追加される1つ以上のテーブルについて、含まれる項目の数と項目名の両方が別のテーブルと同じであることが **Qlik Sense** によって検出されると、それらのテーブルはデータマネージャーで自動的に連結されま

3 データ マネージャーを使用したデータのロードと管理

す。この場合、必要に応じてテーブルを分割することができます。

- 2つのテーブルは、完全に同じ項目またはデータを共有していない場合、強制的に連結することができます。強制的に連結できるのは2つのテーブルのみです。例えば、3つのテーブルを連結する場合、最初に2つのテーブルを1つに連結してから、作成したテーブルに3つめのテーブルを連結します。
- 類似性の低いテーブルは自動的に連結されません。このようなテーブルは強制的に連結することもできません。この場合、テーブル内の項目を代わりにデータマネージャーに関連付ける必要があります。

手順 - 強制連結

手順の完了には、以下の作業が必要です。

1. データテーブルの準備
2. アプリにデータテーブルを追加
3. テーブルの連結とアプリへのデータテーブルのロード
4. その先の手順 - 新しいテーブルの追加とデータ項目の連結

前提条件

Qlik Sense でアプリを作成する方法を知っている必要があります。

データ テーブルの準備

強制連結のデモ用のサンプルデータを用意しています。2つのテーブルの項目とデータがほとんど同じである場合、独自のデータを使用することもできます。

例えば、以下のようなヘッダーとデータの最初の行があるとします。それらは、2つの Excel テーブルに貼り付けられています。項目の違いに注目してください。

	A	B	C	D	E	F	G	H
1	SalesOrderID	SalesOrderDetailID	TrackingNumber	OrderQty	PID	SpecialOfferID	UnitPrice	Modified Date
2	43659	1	4911-403C-98	1	776	1	1822.4946	7/1/2013 0:00

	A	B	C	D	E	F	G	H
1	SalesOrderID	SalesOrderDetailID	TrackingNumber	OrderQty	ProductID	UnitPrice	Modified Date	
2	43662	20	2E53-4802-85	3	764	377.51301	7/12/2013 0:00	

If you want to use the sample data, copy the entire table, including the column headings, into an empty Excel file on your computer. For this walkthrough, we named the Excel tabs *Data Table 1* and *Data Table 2*. We named the Excel file *Concatenate_Data.xlsx*.

Data Table 1 (テーブル データ 1)

Data Table 1 (テーブル データ 1)

SalesOrderID	SalesOrderDetailID	TrackingNumber	OrderQty	PID	SpecialOfferID	UnitPrice	Modified Date
43659	1	4911-403C-98	1	776	1	1822.4946	7/1/2013 0:00

3 データマネージャーを使用したデータのロードと管理

SalesOrderID	SalesOrderDetailID	TrackingNumber	OrderQty	PID	SpecialOfferID	UnitPrice	ModifiedDate
43659	2	4911-403C-98	3	777	1	1822.4946	7/2/2013 0:00
43659	3	4911-403C-98	1	778	1	1098.9	7/2/2013 0:00
43659	4	4911-403C-98	1	771	1	1835.9946	7/2/2013 0:00
43659	5	4911-403C-98	1	772	1	1835.9946	7/3/2013 0:00
43659	6	4911-403C-98	2	773	1	1835.9946	7/3/2013 0:00
43659	7	4911-403C-98	1	774	1	1835.9946	7/4/2013 0:00
43659	8	4911-403C-98	3	714	1	25.95636	7/4/2013 0:00
43659	9	4911-403C-98	1	716	1	25.95636	7/5/2013 0:00
43659	10	4911-403C-98	6	709	1	5.13	7/5/2013 0:00
43659	11	4911-403C-98	2	712	1	4.66785	7/6/2013 0:00
43659	12	4911-403C-98	4	711	1	18.16785	7/6/2013 0:00
43660	13	6431-4D57-83	1	762	1	377.51301	7/7/2013 0:00
43660	14	6431-4D57-83	1	758	1	787.3146	7/7/2013 0:00
43661	15	4E0A-4F89-AE	1	745	1	728.784	7/7/2013 0:00
43661	16	4E0A-4F89-AE	1	743	1	643.23387	7/8/2013 0:00
43661	17	4E0A-4F89-AE	2	747	1	643.23387	7/8/2013 0:00
43661	18	4E0A-4F89-AE	4	712	1	4.66785	7/8/2013 0:00
43661	19	4E0A-4F89-AE	4	715	1	25.95636	7/9/2013 0:00

3 データマネージャーを使用したデータのロードと管理

DataTable 2

データテーブル 2

SalesOrderID	SalesOrderDetailID	TrackingNumber	OrderQty	ProductID	UnitPrice	ModifiedDate
43662	20	2E53-4802-85	3	764	377.51301	7/12/2013 0:00
43662	21	2E53-4802-85	5	770	377.51301	7/12/2013 0:00
43662	22	2E53-4802-85	2	730	165.54438	7/13/2013 0:00
43662	23	2E53-4802-85	4	754	787.3146	7/14/2013 0:00
43662	24	2E53-4802-85	3	725	165.54438	7/14/2013 0:00
43662	25	2E53-4802-85	5	762	377.51301	7/14/2013 0:00
43662	26	2E53-4802-85	3	765	377.51301	7/14/2013 0:00
43662	27	2E53-4802-85	2	768	377.51301	7/15/2013 0:00
43662	28	2E53-4802-85	1	753	1932.2658	7/15/2013 0:00
43663	29	2E53-4802-85	1	756	787.3146	7/16/2013 0:00
43663	30	2E53-4802-85	3	763	377.51301	7/17/2013 0:00
43664	31	2E53-4802-85	1	732	321.2082	7/18/2013 0:00
43664	32	2E53-4802-85	6	758	787.3146	7/19/2013 0:00
43665	33	2E53-4802-85	1	729	165.54438	7/19/2013 0:00

アプリにデータテーブルを追加

次の手順を実行します。

1. Qlik Sense を起動します。
2. [作業] エリアで [アプリの新規作成] をクリックします。[アプリの新規作成] ウィンドウが開きます。

3 データ マネージャーを使用したデータのロードと管理

3. アプリに名前を付け、**[作成]** をクリックします。これでアプリが作成されました。ここでは、アプリに *ConcatenateExample* という名前を付けました。
4. **[アプリを開く]** をクリックします。アプリが開かれ、データを追加するためのダイアログが表示されます。
5. **Excel** ファイルを **[ファイルおよびその他のソースからデータを追加]** ダイアログにドラッグ アンド ドロップします。データ マネージャーの **[関連付け]** ビューにテーブルが表示されます。バブルをクリックして、そのテーブルのデータを表示します。



代わりにデータ マネージャーからデータを追加する場合、データ マネージャーの **[関連付け]** ビューに取り込む前にテーブルの項目を選択するよう求められます。この場合、両方のテーブルのすべての項目を選択します。

The screenshot shows the Qlik Sense Data Manager interface. At the top, there are two data tables represented by bubbles: 'Data Table 2*' and 'Data Table 1*'. Below them is a data table view for 'Data Table 1' with the following data:

Data Table 1.SalesOrderID	Data Table 1.SalesOrder...	Data Table 1.TrackingNum...	Data Table 1.OrderQty	PID	SpecialOfferID	Data Table 1.UnitPrice	Data Table 1.Modified Date
43659		1 4911-403C-98	1	776	1	1822.4946	7/1/2013 12:00:00 AM
43659		2 4911-403C-98	3	777	1	1822.4946	7/2/2013 12:00:00 AM
43659		3 4911-403C-98	1	778	1	1098.9	7/2/2013 12:00:00 AM
43659		4 4911-403C-98	1	771	1	1835.9946	7/2/2013 12:00:00 AM
43659		5 4911-403C-98	1	772	1	1835.9946	7/3/2013 12:00:00 AM

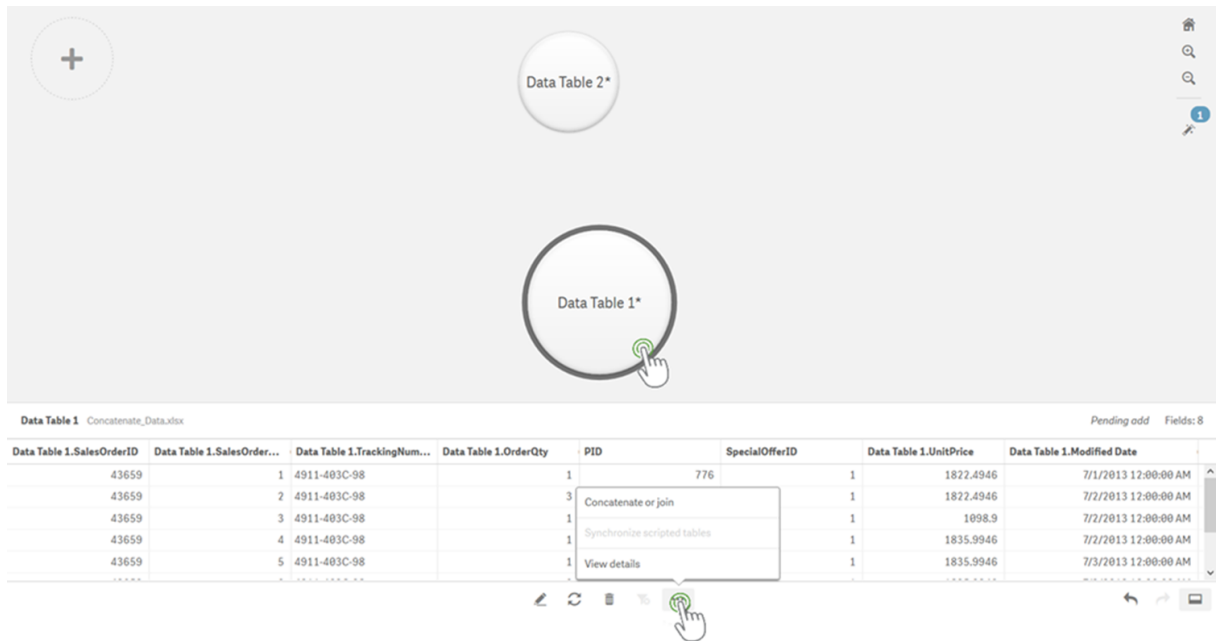
テーブルの連結とアプリへのデータ テーブルのロード

アプリへのデータ テーブルの追加が完了したら、テーブルを連結することができます。

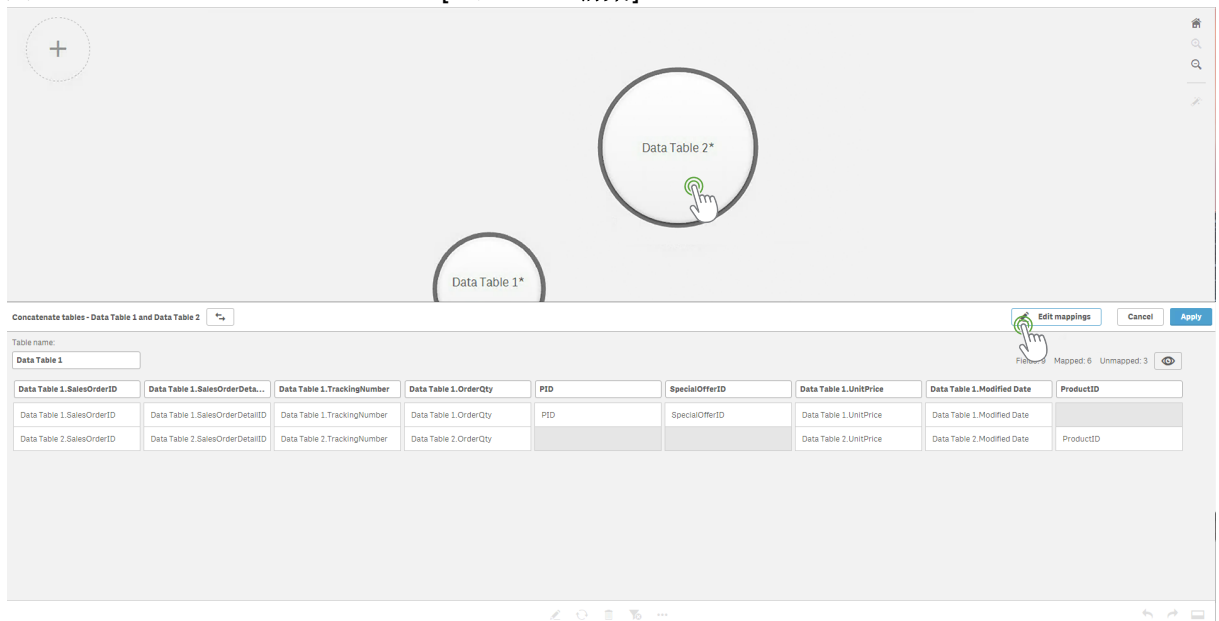
次の手順を実行します。

1. データ マネージャーの **[関連付け]** ビューで、バブルをクリックして1つのテーブルを選択します。**[**]** をクリックし、**[連結または結合]** を選択します。

3 データマネージャーを使用したデータのロードと管理



2. 別のテーブルのバブルをクリックしてから、[マッピングの編集] をクリックします。



3. 必要に応じて、ここで以下を行うことができます。
- [テーブル名] で、テーブルの結合時に作成されるテーブルの名前を変更します。
 - 項目のラベルをドラッグアンドドロップして、項目を結合します。
 - 項目の名前を変更します。
 - 項目用の [✕] をクリックして、項目を削除します。

3 データマネージャーを使用したデータのロードと管理

Concatenate tables - Data Table 1 and Data Table 2

Table name: Data Table 1

Fields: 9 Mapped: 6 Unmapped: 3

Data Table 1.SalesOrderID	Data Table 1.SalesOrderData...	Data Table 1.TrackingNumber	Data Table 1.OrderQty	PID	SpecialOfferID	Data Table 1.UnitPrice	Data Table 1.Modified Date	ProductID
43659	1	4E6A-4F69-AE	1	799	1	4,667.85	2013-07-01 00:00:00	
43660	2	4911-483C-98	2	711		5.13	2013-07-02 00:00:00	
43661	3	6431-4D57-83	3	712		18,167.85	2013-07-02 00:00:00	
	4	4E6A-4F69-AE	4	714		25,956.36	2013-07-04 00:00:00	
	5	4911-483C-98	6	715		377,513.81	2013-07-05 00:00:00	
	6	6431-4D57-83		716		643,233.67	2013-07-06 00:00:00	
	7			743		728,784	2013-07-07 00:00:00	
	8			745		787,314.6	2013-07-08 00:00:00	
	9			747		1,896.9	2013-07-09 00:00:00	
	10			758		1,822,494.6		

この例では、以下を行いました。

- テーブルの名前を **Data Table** に変更しました。
- ProductID** のラベルと項目を **PID** 項目の下にドラッグし、項目を結合しました。
- PID** 項目の名前を **Product ID** に変更しました。
- SpecialOfferID** 項目を削除しました。

このテーブルは次のようになっています。

Concatenate tables - DataTable_Tab1 and DataTable_Tab2

Table name: Data Table

Fields: 7 Mapped: 7 Unmapped: 0

Data Table_Tab1.SalesOrderID	Data Table_Tab1.SalesOrderData...	Data Table_Tab1.TrackingNu...	Data Table_Tab1.OrderQty	Product ID	Data Table_Tab1.UnitPrice	Data Table_Tab1.Modified Date
Data Table_Tab1.SalesOrd...	Data Table_Tab1.SalesOrd...	Data Table_Tab1.TrackiN...	Data Table_Tab1.OrderQty	PID	Data Table_Tab1.UnitPrice	Data Table_Tab1.Modified ...

- [選択アクション]** ドロップダウンで、**[連結]**、**[適用]** の順にクリックします。テーブルはマッピングされた項目上で連結されます。「*」が付いているのは、まだアプリにロードされていないデータです。



5. [データのロード] をクリックします。データが正しくロードされたことを示すメッセージが表示されます。[シートの編集] をクリックし、データを使用したビジュアライゼーションを作成します。

その先の手順 - 新しいテーブルの追加とデータ項目の連結

上記のサンプル データは同じ Excel ファイルの 2 つのタブにペーストされました。ただし、項目を連結する場合、テーブルが同じファイル内に存在する必要はありません。テーブルは、アプリに追加される別のファイル内に存在していてもかまいません。後から別のテーブルを追加することもできます。例えば、最初に June からテーブルを追加し、後から July から 2 つめのテーブルを追加することも可能です。

この例では、同じ項目を持つ別のテーブルを、上記で作成した連結テーブルに追加します。

以下にサンプル データを示します。ここでは、テーブルを持つタブに「Data Table_Newest」という名前を付けました。データファイルには「Concatenate_Data2.xlsx」という名前を付けました。

DataTable_Newest

DataTable_Newest

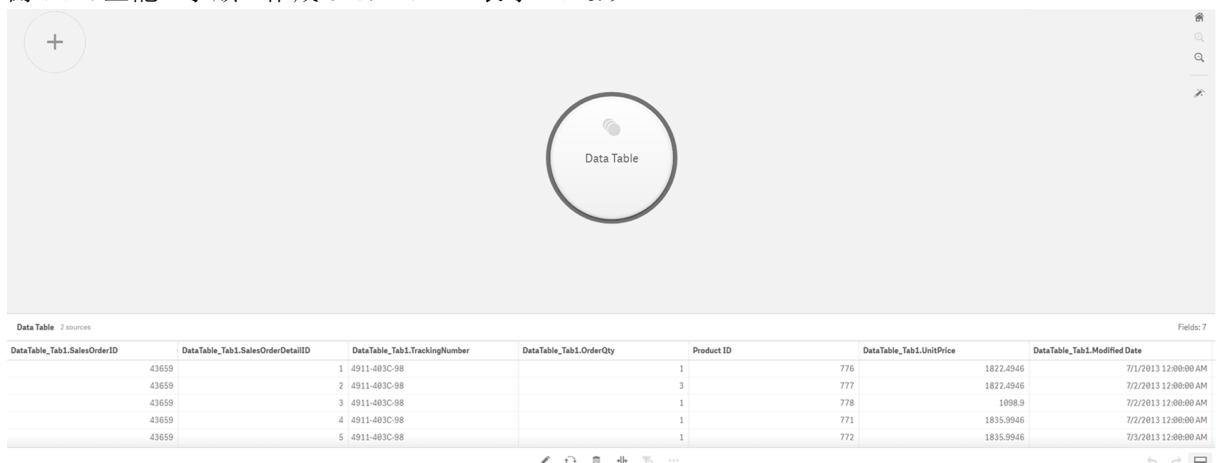
SalesOrder ID	SalesOrderDetailID	TrackingNumber	ZIP	OrderQty	ID	UnitPrice	ModifiedDate
43666	34	568E-472E-9C	20012	3	764	377.51301	7/12/2013 0:00
43666	34	568E-472E-9C	23220	5	770	377.51301	7/12/2013 0:00
43667	35	AB6C-4FF9-9D	30004	2	730	165.54438	7/13/2013 0:00
43668	36	AB6C-4FF9-9D	11215	4	754	787.3146	7/14/2013 0:00
43668	36	AB6C-4FF9-9D	55401	3	725	165.54438	7/14/2013 0:00

3 データマネージャーを使用したデータのロードと管理

SalesOrder ID	SalesOrderDetailID	TrackingNumber	ZIP	OrderQty	ID	UnitPrice	ModifiedDate
43668	36	AB6C-4FF9-9D	20003	5	762	377.51301	7/14/2013 0:00
43668	36	AB6C-4FF9-9D	15213	3	765	377.51301	7/14/2013 0:00
43669	37	C618-4998-BE	33125	2	768	377.51301	7/15/2013 0:00
43669	37	C618-4998-BE	11215	1	753	1932.2658	7/15/2013 0:00

次の手順を実行します。

1. Qlik Sense ハブで、上記の手順で作成したアプリをクリックします。アプリが開きます。
2. 最上部のツールバーのドロップダウンリストで、**[データマネージャー]**を選択します。データマネージャーが開かれ、上記の手順で作成したテーブルが表示されます。



Data Table_SalesOrderID	Data Table_Tab1.SalesOrderDetailID	Data Table_Tab1.TrackingNumber	Data Table_Tab1.OrderQty	Product ID	Data Table_Tab1.UnitPrice	Data Table_Tab1.Modified Date
43659	1	4911-493C-98	1	776	1822.4946	7/1/2013 12:00:00 AM
43659	2	4911-493C-98	3	777	1822.4946	7/2/2013 12:00:00 AM
43659	3	4911-493C-98	1	778	1098.9	7/2/2013 12:00:00 AM
43659	4	4911-493C-98	1	771	1835.9946	7/2/2013 12:00:00 AM
43659	5	4911-493C-98	1	772	1835.9946	7/3/2013 12:00:00 AM

3. **+** ボタンをクリックしてデータを追加します。
4. **[このアプリにファイルを添付]** ダイアログにドラッグして、新しい Excel ファイルをアプリに追加します。**[データの追加]** ウィンドウが開きます。
5. **[データの追加]** をクリックして、データテーブルをアプリに追加します。新しいテーブルがアプリに追加されます。

3 データ マネージャーを使用したデータのロードと管理



DataTable_Tab1.SalesOrderID	DataTable_Tab1.TrackingNumber	DataTable_Tab1.OrderQty	Product ID	DataTable_Tab1.UnitPrice	DataTable_Tab1.Modified Date
43659	1 4911-493C-98		1	776	1822.4946
43659	2 4911-493C-98		3	777	1822.4946
43659	3 4911-493C-98		1	778	1098.9
43659	4 4911-493C-98		1	771	1835.9946
43659	5 4911-493C-98		1	772	1835.9946

6. これで、テーブルを連結し、マッピングを編集し、データをロードすることができます。

3.17 [データ マネージャー] でのスクリプト化されたテーブルの同期

デフォルトでは、データロードエディターに追加された、スクリプト化されたテーブルでは、[データ マネージャー] で使用可能なツールを使用できません。

例えば、スクリプト化されたテーブルを [データ マネージャー] 内の他のテーブルに関連付けたり、データプロファイリング カードを使用して、スクリプト化されたテーブルの項目を変換したりすることはできません。スクリプト化されたテーブルを [データ マネージャー] に同期する場合、[データ マネージャー] 内のスクリプト化されたテーブルを、管理対象のスクリプト化されたテーブルに置換することができます。これらのテーブルでは、以下のように、[データ マネージャー] に追加されているテーブルと同じツールすべてを使用できます。

- テーブルの編集 (計算された項目の追加など)。
- 項目の変換 (項目の種類の変更、データプロファイリング カードを使用した項目の変換など)。
- テーブルの変換 (テーブルのピボット解除や連結など)。

同期および管理対象のスクリプト化されたテーブルには次の制限があります。

- スクリプト化されたテーブルを管理対象のスクリプト化されたテーブルとして同期するには、データロードスクリプトの **自動生成セクション** の前に配置する必要があります。データロードスクリプトの **自動生成セクション** の後のテーブルは同期されません。
- 管理対象のスクリプト化されたテーブル内の項目の選択を変更する場合に、[ソースからデータを選択] は使用できません。



データロードスクリプトに **Exit** ステートメントまたは動的項目が含まれている場合は、スクリプト化されたテーブルを同期しないでください。

スクリプト化されたテーブルを管理対象のスクリプト化されたテーブルに変換するには、スクリプト化されたテーブルを [データ マネージャー] で同期します。同期によって、次のことが行われます。

- 同期されたすべてのスクリプト化されたテーブルが、管理対象のスクリプト化されたテーブルに置き換わる。

3 データマネージャーを使用したデータのロードと管理

- データロードスクリプト内でスクリプト化されたテーブルが削除されている、管理対象のスクリプト化されたテーブルを削除する。
- データロードスクリプト内でソーステーブルが変更されている、管理対象のスクリプト化されたテーブルを更新する。



テーブルを同期した場合、**[データマネージャー]**を別のタブで開いた状態で、データロードエディターで変更を行わないでください。



すでに**[データマネージャー]**で同期されているテーブルのデータロードスクリプトを変更することは避けてください。データロードエディターで項目を削除または変更する場合は、同期されたテーブル内の自動取得項目や関連付けを削除またはやり直する必要があります。計算された項目または**[分割]**カードによって作成された項目など、削除または変更された項目を使用する自動取得項目には、**NULL**値が表示されます。

同期後、**[データマネージャー]**で管理対象のスクリプト化されたテーブルを他のテーブルのように使用できます。**[データマネージャー]**では、管理対象のスクリプト化されたテーブルとソースのスクリプト化されたテーブルとの間で差異を検出した場合、再度同期するようプロンプトが表示されます。

管理対象のスクリプト化されたテーブルを元のスクリプト化されたテーブルに変更する場合は、**[データマネージャー]**で削除します。再度同期する場合は、繰り返し削除する必要があります。

スクリプト化されたテーブルの同期


次の手順を実行します。

1. **[データマネージャー]**でをクリックします***。
または、スクリプト化されたテーブルを選択します。
2. **[スクリプト化されたテーブルの同期]**をクリックします。

管理対象のスクリプト化されたテーブルが、**[データマネージャー]**内のすべてのスクリプト化されたテーブルを置換します。

管理対象のスクリプト化されたテーブルの削除

次の手順を実行します。

1. **[データマネージャー]**で**[テーブル]**表示を選択します。
2. 削除する管理対象のスクリプト化されたテーブルで、をクリックします .
3. **[データのロード]**をクリックします。

管理対象のスクリプト化されたテーブルが、元のスクリプト化されたテーブルに変更されます。

3.18 データ関連付けの管理

Qlik Sense はデータのプロファイリングを行って、テーブルどうしの関連付けを作成しやすくします。関連付けは、Qlik Sense Insight Advisor による分析に基づいた推奨事項から選択するか、または独自に作成することができます。

データを関連付ける場合は、データプロファイリングを有効にして[dデータの追加] オプションを使用するようお勧めします。これはデフォルトのオプションです。この設定を検証するには、[データの追加] ページ右下隅の[データの追加] ボタンの横にある***をクリックします。

データマネージャーの[関連付け] ビューでは、データがバブルで図示されます。各バブルはデータテーブルを表しています。バブルのサイズはテーブル内のデータの量を表しています。バブル間のリンクは、テーブル間の関連付けを表しています。2つのテーブル間に関連付けが存在する場合は、リンク内のボタンをクリックして、関連付けを表示または編集することができます。

データマネージャーの関連付けビュー

The screenshot displays the Qlik Sense Data Manager interface. At the top, there are tabs for 'Prepare Data manager', 'Analyze Sheet', and 'Narrate Storytelling'. The main area shows a bubble chart with five nodes: 'Customer', 'Sales*', 'Cities', 'Sales rep', and 'Item master*'. The 'Sales*' node is the largest. A sidebar on the right shows 'Recommended associations' with 'Total tables: 5', 'Unassociated tables: 1', and 'Recommendations: 1'. Below the chart, a data table for 'Sales' is visible, showing columns like %KEY, Cost, Sales.Custo..., Date, GrossSales, Invoice Date, Invoice Num..., Item Desc, Sales.Item N..., Margin, Order Number, Promised D..., and Sales. The table contains 5 rows of data.

%KEY	Cost	Sales.Custo...	Date	GrossSales	Invoice Date	Invoice Num...	Item Desc	Sales.Item N...	Margin	Order Number	Promised D...	Sales	
1	-513.15	10012715	12/31/2012	-573.3835	12/31/2012	118960	Cutting Edge Sliced Ham	10696		-60.23	215785	12/31/2012	-573.38
2	-105.93	10012715	12/31/2012	-204.6638	12/31/2012	118960	Washington Cranberry Juice	10009		-98.73	215785	12/31/2012	-204.66
3	-88.07	10012715	12/31/2012	-165.8016	12/31/2012	118960	Moms Sliced Ham	10385		-77.73	215785	12/31/2012	-165.8
4	-43.12	10012715	12/31/2012	-118.3703	12/31/2012	118960	Tip Top Lox	10215		-75.25	215785	12/31/2012	-118.37
5	-37.98	10012715	12/31/2012	-102.3319	12/31/2012	118960	Just Right Beef Soup	10965		-64.35	215785	12/31/2012	-102.33




ほとんどの場合、テーブル関連付けをモデルビュー内で編集するのが簡単な方法ですが、テーブル編集ビューで[関連付け] オプションを使用して、単一テーブル形式の関連付けを編集することもできます。

詳細については、「テーブルエディタでデータを関連付ける (page 39)」を参照してください。

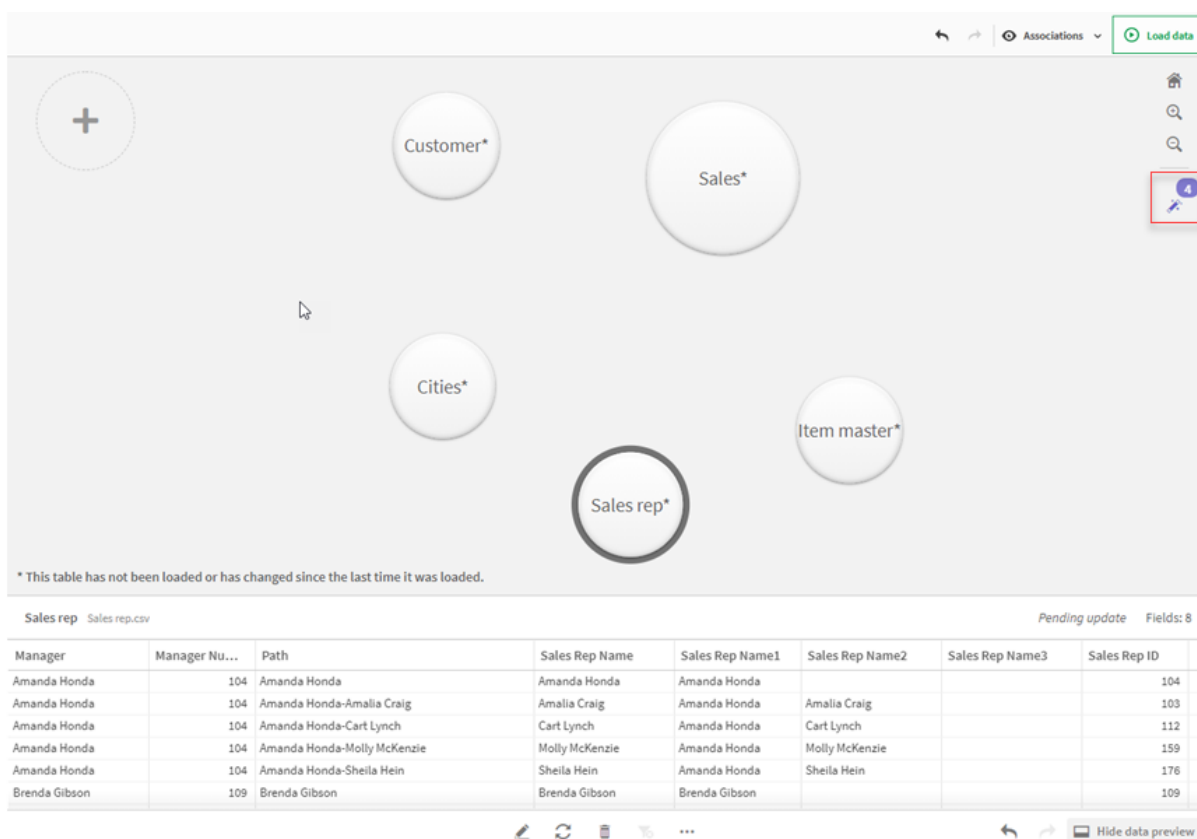
[Recommended associations (推奨される関連付け)] パネルの使用によるテーブルの関連付け

多くの場合、Qlik Sense Insight Advisor ではデータテーブル間の関連付けを推奨します。[Recommended associations (推奨される関連付け)] パネルでは、これらの推奨事項を閲覧し、適用することができます。

デフォルトでは、テーブルが存在する場合 [Recommended associations (推奨される関連付け)] パネルが開きます。パネルは  をクリックすると、閉じたり再び開いたりできます。


パネルが閉じていて、推奨事項が存在する場合は、 の上部に推奨事項の数が表示されます。

推奨される関連付けを開く



Manager	Manager Nu...	Path	Sales Rep Name	Sales Rep Name1	Sales Rep Name2	Sales Rep Name3	Sales Rep ID
Amanda Honda	104	Amanda Honda	Amanda Honda	Amanda Honda			104
Amanda Honda	104	Amanda Honda-Amalia Craig	Amalia Craig	Amanda Honda	Amalia Craig		103
Amanda Honda	104	Amanda Honda-Carl Lynch	Carl Lynch	Amanda Honda	Carl Lynch		112
Amanda Honda	104	Amanda Honda-Molly McKenzie	Molly McKenzie	Amanda Honda	Molly McKenzie		159
Amanda Honda	104	Amanda Honda-Sheila Hein	Sheila Hein	Amanda Honda	Sheila Hein		176
Brenda Gibson	109	Brenda Gibson	Brenda Gibson	Brenda Gibson			109

次の手順を実行します。


1. [Recommended associations (推奨される関連付け)] パネルが閉じている場合は、関連付けビューの右上隅の  をクリックします。
右側にパネルが表示されます。
2. 次の情報が表示されます。
 - [Total tables (総テーブル数)]: テーブルの総数。
 - [Unassociated tables (関連付けのないテーブル数)]: 関連付けのないテーブルの総数。
 - [Recommendations (推奨事項)]: 推奨事項の総数。

3 データマネージャーを使用したデータのロードと管理

- **[Recommended association details (推奨される関連付けの詳細)]**: 推奨される関連付けの名前、テーブル名、項目名をコロンで区切って表示します。
3. いずれかの推奨事項をクリックすると、プレビューがダークブルー色で表示されます。
 4. 一部の推奨事項のみ承諾するには、必要な推奨事項の**[適用]** ボタンをクリックします。
 5. **[Preview all (すべてプレビュー)]** をクリックすると、推奨される関連付けがデータテーブルにどのように影響するかが表示されます。プレビュー中の関連付けは、ハイライトされます。
推奨される関連付けパネル

The screenshot displays the Data Manager interface with a central diagram of tables: Customer*, Sales*, Cities*, Sales rep*, and Item master. The Sales* table is highlighted. To the right, a panel titled 'Recommended associations' lists several associations with 'Apply' buttons. Below the diagram, a table titled 'Sales rep' is shown with columns for Manager, Manager Nu..., Path, Sales Rep Name, Sales Rep Name1, Sales Rep Name2, Sales Rep Name3, and Sales Rep ID.

Manager	Manager Nu...	Path	Sales Rep Name	Sales Rep Name1	Sales Rep Name2	Sales Rep Name3	Sales Rep ID
Amanda Honda	104	Amanda Honda	Amanda Honda	Amanda Honda			104
Amanda Honda	104	Amanda Honda-Amalia Craig	Amalia Craig	Amanda Honda	Amalia Craig		103
Amanda Honda	104	Amanda Honda-Cart Lynch	Cart Lynch	Amanda Honda	Cart Lynch		112
Amanda Honda	104	Amanda Honda-Molly McKenzie	Molly McKenzie	Amanda Honda	Molly McKenzie		159
Amanda Honda	104	Amanda Honda-Sheila Hein	Sheila Hein	Amanda Honda	Sheila Hein		176
Brenda Gibson	109	Brenda Gibson	Brenda Gibson	Brenda Gibson			109

6. **[Apply all (すべて適用)]** をクリックすると、それぞれの推奨される関連付けが適用されます。承諾された関連付けは、ライトグレー色で表示されます。
画面下部の  をクリックすると、テーブルがどのように変更されるかが表示されます。

データについてのビジュアライゼーションを開始できるようになります。

手動によるテーブルの関連付け

テーブルは、当該テーブルをドラッグすることにより、手動で関連付けることができます。テーブルバブルは、相互にドラッグするとき、緑色、オレンジ色、または赤色のストライプが表示されます。

- 緑: データマネージャーは、関連付ける項目について非常に信頼しています。たとえば、2つのテーブルに "Sales Region" というラベルの項目が存在する場合、データマネージャーは、これらの項目が関連付けられているとみなします。

3 データマネージャーを使用したデータのロードと管理

- オレンジ: データマネージャーは、これらのテーブルが関連付け可能であると、ある程度信頼しています。たとえば、ラベルの異なる別々の2つの項目が存在し、それらの項目に一桁のデータが含まれている場合、データ型が同じであるため、データマネージャーはこれらの項目にオレンジ色のフラグを設定します。
- 赤: データマネージャーは、これらのテーブルを関連付けする方法を認識していません。**[Associate tables (テーブルの関連付け)]** エディターで、関連付けるテーブルと項目を選択する必要があります。

緑色またはオレンジ色のテーブルを手動で関連付けたい

次の手順を実行します。

1. テーブルを、緑色またはオレンジ色のマークが付けられている、いずれかのテーブルにドラッグします。
2. 関連付けは自動的に適用されます。

これで、テーブルは、推奨されている項目を使用して関連付けられます。

赤色のテーブルを手動で関連付けたい

この場合にテーブルを関連付けるには、カスタム関連付けを作成します。

次の手順を実行します。

1. テーブルを、赤色のマークが付けられているテーブルのうちの1つにドラッグします。
[テーブルの関連付け] エディタが開きます。
2. 左側のテーブルで、関連付けに使用する項目を選択します。
単一の項目または複数の項目を選択できます。また、区切り文字を追加すると、データの解釈、またはすでに存在している項目との照合が簡単化されます。プレビューでは、関連付けデータの内容が確認できます。
3. 右側のテーブルでは、左側のテーブルで行った選択内容と照合するために使用する項目を選択します。
4. 作成されるキー項目の名前を**[名前]**に入力します。
新しい項目名に、どちらかのテーブルで使用されている項目名と同じものを使用することはできません。
5. **[関連付け]** をクリックします。

これで、テーブルは、カスタム関連付けを使用して関連付けられます。

関連付けの解除

データモデルに適さない関連付けを解除するには2つの方法があります。

次の手順を実行します。



- 関連付けテーブルのうちの1つをクリックします。そのテーブルを、関連付けが解除されるまでもう一方のテーブルから離れた場所にドラッグします。または、次の方法で解除できます。
- 2つのバブルの間のリンクをクリックし、下部のパネルにある**[削除]** ボタンをクリックします。

2つのテーブルの関連付けが解除されます。

関連付けの編集

データモデルを調整する必要がある場合は、2つのテーブル間の既存の関連付けを編集することができます。

次の手順を実行します。


1. 関連付けられているテーブルの間の円をクリックし、データパネルを開きます。
パネルが開き、関連付け項目内にデータがプレビューされます。
2.  をクリックします。
1つ以上のボタンが表示されます。それぞれのボタンは、緑色、オレンジ色、または赤色のマークが付けられています。緑色は、**データマネージャー**が関連付けを非常に信頼していること、オレンジ色はある程度信頼していること、赤色は信頼できるかどうか分からないことを示しています。現在の関連付けには、グレーのマークが付けられています。
3. 関連付けボタンのいずれか1つをクリックします。
 - 推奨されている関連付けをクリックして選択します。
 - 既存のカスタム関連付け  をクリックして、関連付けで使用する項目を編集します。
 - **[カスタムの関連付け]** をクリックして、新しい関連付けを作成します。
このボタンが使用できるのは、テーブルペアに対して推奨されている関連付けが存在する場合のみです。
カスタム関連付けは、1つの項目を含むことも、複数の項目を含むこともできます。

これで、テーブルペア間の関連付けが変更されています。

データのプレビュー

データを適切に理解するために、関連付けビューでテーブルをプレビューすることができます。

次の手順を実行します。

1. テーブルを選択します。
2. ビューの下部にある  をクリックします。

プレビューパネルに、テーブルデータが表示されます。

合成キー

複数のデータテーブル間に複数の項目が共通して含まれている場合、複合キーの関係が存在することを意味します。Qlik Senseは、自動的に合成キーを作成し、これに対処しています。合成キーとは、複合キーが発生するすべての組み合わせを表す匿名項目です。

詳細については、「[合成キー \(page 127\)](#)」を参照してください。

テーブルを追加すると次のいずれかの状況に至る場合には、プロファイリングを有効にしたデータ追加しかできません。

- 6項目以上の項目を含む合成キーが作成される。
- 合成キーが11以上作成される。
- ネストされた合成キー(中に別の合成キーが含まれている形の合成キー)が作成される。

このようなケースとなる場合、問題を解決するためにデータテーブルを修正する必要があります。

制限事項

ロードされたテーブルの構造やテーブル内のデータが原因で、関連付けの提案が表示されない場合もあります。そのような場合は、テーブル エディタで関連付けを調整する必要があります。


- 多対多の関係。
- データを含む項目ペアが両方向で十分に一致しません。小さなテーブルの数項目の値が大きなテーブルのある項目と100%一致している一方で、逆の方向では一致の度合いが著しく低い場合などが、これに当たります。
- 複合キーの関連付け。

また、**データマネージャー**は、**[データの追加]**で追加されたテーブルのみを分析します。データロードスクリプトを使って追加されたテーブルは、**データマネージャー**と同期していない限り、関連付けに関する推奨事項には含まれません。


詳細については、「**[データマネージャー]**でのスクリプト化されたテーブルの同期 (page 88)」を参照してください。

変更の適用とデータのリロード




[データマネージャー]で行った変更は、データをリロードするまでアプリでは使用できません。データをリロードすると、変更が適用され、追加したすべての新規データが外部データソースからロードされます。以前ロードしたデータはリロードされません。

[データマネージャー]フッターの  ボタンを使用して、外部データソースからすべてのデータをリロードすることができます。



 ボタンを使用すると、選択したテーブルのすべてのデータがリロードされます。アプリ内のすべてのテーブルのすべてのデータはリロードされません。

[データマネージャー]内のデータがアプリのデータと同期していない場合は、**[データのロード]** ボタンが緑色で表示されます。**[関連付け]**表示では、新規作成または更新されたテーブルにはすべて*が付き、削除されたテーブルは薄いグレーで表示されます。**[テーブル]**表示では、新規作成および更新、削除されたテーブルはすべて青でハイライトされ、テーブルのステータスを示すアイコンが表示されます。

- **[削除保留]**  とマークされているテーブルは削除されます。
- **[更新保留]**  とマークされているテーブルは、項目の追加または名前変更、削除、あるいはテーブル名の変更が反映した状態で更新されます。
- **[追加保留]**  とマークされているテーブルは追加されます。

変更の適用

次の手順を実行します。

- [データのロード] をクリックして、アプリで変更をロードします。

これで [データ マネージャー] での変更がアプリのデータに反映されます。

4 スクリプトを使用したデータのロードと変換

ここでは、データロードスクリプトを使用してQlik Senseにデータをロードする方法を簡単に説明します。

Qlik Sense では、データロードエディタで管理するデータロードスクリプトを使用し、さまざまなデータソースに接続してデータを取得します。データソースは、例えば Excel ファイルや .csv ファイルなどのデータファイルの場合があります。また、データソースは Google BigQuery や Salesforce データベースなどのデータベースの場合もあります。

データマネージャーを使用してデータをロードすることもできますが、データロードスクリプトの作成、編集、実行には、データロードエディターを使用します。

ロード対象の項目とテーブルは、スクリプト内で指定します。スクリプトは主にデータソースからロードするデータを指定する目的で使用されます。データ構造は、スクリプトステートメントを使用して操作することも可能です。

データのロード中、Qlik Senseは異なるテーブル間で共通する項目(キー項目)を特定し、データを関連付けます。アプリにおける結果データのデータ構造は、データモデルビューアで監視できます。項目名を変更し、テーブル間に異なる関連付けを設定することで、データ構造を変更できます。

Qlik Sense にロードされたデータは、アプリに保存されます。

Qlik Sense の分析は常に、アプリがデータソースに直接接続されていない間に行われます。よって、データを更新するには、スクリプトを実行してデータをリロードする必要があります。

4.1 [データマネージャー] およびデータロードスクリプト間の操作

[データマネージャー] でデータテーブルを追加すると、データロードスクリプトコードが生成されます。データロードエディタの [自動生成セクション] でスクリプトコードを表示できます。生成されたスクリプトコードをロック解除して編集することもできますが、その場合、データテーブルは [データマネージャー] で管理されなくなります。

デフォルトでは、ロードスクリプトで定義されたデータテーブルは、[データマネージャー] で管理されません。これは、データの概要でテーブルを閲覧することはできても、[データマネージャー] 内でテーブルを削除したり編集したりはできないことを意味します。また、スクリプトでロードしたテーブルに対しては、関連付けは提案されません。ただし、スクリプトテーブルを [データマネージャー] と同期する場合、そのテーブルはマネージドスクリプトテーブルとして [データマネージャー] に追加されます。



テーブルを同期した場合、[データマネージャー] を別のタブで開いた状態で、データロードエディターで変更を行わないでください。

スクリプトセクションの追加や、[データマネージャー] で作成したデータモデルを強化したり、通信やコードの開発は可能ですが、注意を必要とするエリアがあります。ユーザー記述のスクリプトコードは、例えば、次のような場合に [データマネージャー] のデータモデルを妨害し、問題を生じさせる可能性があります。

4 スクリプトを使用したデータのロードと変換

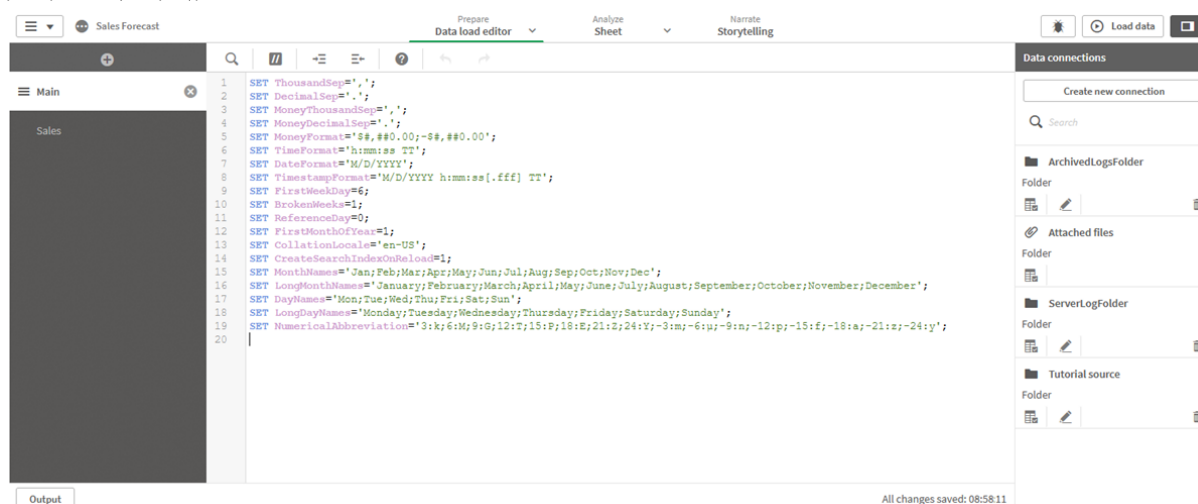
- [データマネージャー] を使って追加したテーブルの名前をスクリプトで変更したりテーブルをドロップした場合。
- [データマネージャー] を使って追加したテーブルから項目をドロップした場合。
- [データマネージャー] を使って追加したテーブルとスクリプトにロードしたテーブルを連結した場合。
- [データマネージャー] を使って追加したテーブル内にある項目と **Qualify** ステートメントを併用した場合。
- スクリプトで **Resident** を使用している [データマネージャー] を使って追加したテーブルをロードした場合。
- 生成されたコードセクションの後にスクリプトコードを追加した場合。データモデルの変更結果は、[データマネージャー] に反映されません。

4.2 データロードエディタの使用

データロードエディターからデータロードスクリプトを作成して実行し、データソースへの接続を作成します。データがロードされると、アプリで分析できます。

ナビゲーションバーの [準備] タブにある [データロードエディター] をクリックし、データロードエディターを開きます。

データロードエディター



ツールバー

データロードエディターで最も頻繁に使用されるコマンドを含むツールバーです。

データ接続

[データ接続] の下に、よく使用するデータソース(データベースまたはリモートファイル)へのショートカットを保存できます。ここからロードするデータを選択できます。

テキストエディタ

テキストエディタではスクリプトコードを記述、編集することができます。スクリプトの各行には番号が振られ、スクリプトは構文のコンポーネント別に色分けされます。テキストエディタツールバーには、[検索と置換]、[ヘルプモード]、[元に戻す]、[やり直す] のコマンドがあります。最初のスクリプトには、事前定義された地域変数設定の一

4 スクリプトを使用したデータのロードと変換

部がすでに含まれており(たとえば、SET ThousandSep=)、通常、編集する必要はありません。

セクション

スクリプトをセクションに分けて、読みやすく、管理しやすくします。セクションは上から下に実行されます。

[データの追加] を使用してデータを追加した場合、データをロードするのに必要なスクリプトコードを含む [自動生成セクション] というデータロードスクリプトセクションが表示されます。

アウトプット

[アウトプット] には、自動保存ステータスとスクリプトの実行中に生成されたすべてのメッセージが表示されます。

データロードエディターでのデータソースへの接続

データロードエディターのデータ接続によって、よく使用するデータソース(データベースやローカルファイル、リモートファイル)へのショートカットを保存できます。

[データ接続] には、アルファベット順で保存したデータ接続がリストアップされます。検索ボックスを使用すれば、リストを特定の名前またはタイプの接続に絞り込むことができます。



自分が所有しているか、アクセス権が付与されているデータ接続のみを表示できます。アクセス権を取得する必要がある場合は、Qlik Sense システム管理者にお問い合わせください。

データ接続の新規作成

次の手順を実行します。

1. [接続の新規作成] をクリックします。
2. ドロップダウンリストから作成したいデータソースのタイプを選択します。
選択したデータソースの種類に応じた設定ダイアログが開きます。
3. データソース設定を入力して [作成] をクリックし、データ接続を作成します。
一意の名前になるよう、接続名にはユーザー名とドメインが付加されます。

これで作成者を既定の所有者とするデータ接続が作成されます。他のユーザーがサーバー環境で接続を利用できるようにする場合、Qlik Management Console で接続のアクセス権を編集する必要があります。




データソースの設定が変更されても、作成した接続の設定は自動的に更新されません。つまり、Windows の統合セキュリティと DSN のデータベースログインとの設定を変更する場合には特に、ユーザー名とパスワードの保存に注意する必要があります。



[接続の新規作成] が表示されていない場合、データ接続の追加に必要なアクセス権がないことを意味します。アクセス権を取得する必要がある場合は、Qlik Sense システム管理者にお問い合わせください。

データ接続の削除

次の手順を実行します。


1. 削除したいデータ接続で  をクリックします。
2. 接続を削除することを確認します。

これでデータ接続が削除されました。


  が表示されていない場合、データ接続の削除に必要なアクセス権がないことを意味します。アクセス権を取得する必要がある場合は、**Qlik Sense** システム管理者に問い合わせてください。



データ接続の編集

次の手順を実行します。

1. 編集したいデータ接続で  をクリックします。
2. データ接続の詳細を編集します。接続の詳細は接続のタイプごとに異なります。接続の資格情報を提供する必要がある場合があります。


これでデータ接続が更新されます。

 データ接続名を編集する際、同じ接続の参照を維持したい場合は、スクリプトに含まれる参照 (`lib://`) をすべて編集する必要があります。


  が表示されていない場合、データ接続の更新に必要なアクセス権がないことを意味します。必要に応じて **Qlik Sense** システム管理者に連絡してください。

接続文字列の挿入

ほとんどの接続では、接続文字列が必要になります。フォルダー接続とWeb ファイル接続の場合にのみ、接続文字列は不要です。

 この機能は **Qlik Sense SaaS** では使用できません。

次の手順を実行します。


- 接続文字列を挿入したい接続で  をクリックします。

選択されたデータ接続の接続文字列がデータロードエディタの現在の位置に挿入されます。

データ接続からのデータの選択

データ接続からデータを選択してアプリでロードする場合、次の項目を実行します。

4 スクリプトを使用したデータのロードと変換

1. データソースにリンクする[接続の新規作成]を実行します(データ接続が既に存在していない場合)。
2.  接続からデータを選択します。

スクリプト内でのデータ接続の参照

パス付きのファイル名を参照する場合など、ステートメント内のデータソースやスクリプト内の関数を参照する際は、データ接続を使用できます。

ファイル参照の構文: 'lib://(接続名)/(パスを含むファイル名)'

例: DataFiles 接続からのファイルのロード

Folder データ接続で定義されている場所から *orders.csv* ファイルをロードします。これは、例えば管理者が Qlik Sense サーバー上に作成したフォルダです。

```
LOAD * FROM 'lib://DataSource/orders.csv';
```

例: サブフォルダーからのファイルのロード

DataSource データ接続 フォルダーから *Customers/cust.txt* ファイルをロードします。Customers は、データ接続で定義された場所にあるサブフォルダーです。

```
LOAD * FROM 'lib://DataSource/Customers/cust.txt';
```

例: Web ファイルからのロード

実際の URL へのリンクを含む PublicData Web ファイル データ接続からテーブルをロードします。

```
LOAD * FROM 'lib://PublicData' (html, table is @1);
```

例: データベースからのロード

DataSource データベース接続から *Sales_data* テーブルをロードします。

```
LIB CONNECT TO 'DataSource';
```

```
LOAD *;
```

```
SQL SELECT * FROM `Sales_data`;
```

データ接続の保存場所

接続は Qlik Sense Repository Service を使用して保存されます。また、データ接続は、Qlik Sense サーバー デプロイメントの Qlik Management Console で管理できます。Qlik Management Console では、データ接続の削除やアクセス権の設定、その他のシステム管理タスクを実行できます。



Qlik Sense Desktop では、接続はすべて、暗号化なしでアプリに保存されます。この中には、接続の作成時に入力したユーザー名やパスワード、ファイルパスに関する可能な詳細が含まれます。そのため、別のユーザーとアプリを共有した場合、これらの情報がテキスト形式で表示される可能性があります。共有を目的にしたアプリを設計する際には、この点について考慮してください。

データロードエディタでデータを選択


データロードエディタで**[データを選択]**を使用して、ファイルまたはデータベーステーブルからどの項目をロードするかや、どのデータソースを表示するかを選択できます。

このダイアログでは項目の選択に加え、項目の名前を変更することもできます。項目の選択が終了したら、生成したスクリプトコードをスクリプトに挿入できます。

データソースには、CSVファイルなど、単独のテーブルのみ含まれるものもあれば、Microsoft Excel スプレッドシートやデータベースなど、多くのテーブルが含まれるものもあります。




[データマネージャー]で、同じ名前および同じ列でスクリプト化されたテーブルとして既に追加済みのテーブルは、**[データロードエディター]**に追加しないでください。

データロードエディタのデータ接続でをクリックすると、**[データの選択]**ダイアログが開きます。

データベースからのデータを選択

データベースからデータを選択する際、データソースには複数のテーブルが含まれる場合があります。

次の手順を実行します。

1. **データロードエディター**を開きます。
2. 左側の**[データ接続]**で、データベース接続のをクリックします。
選択したデータダイアログが表示されます。
3. ドロップダウンリストから**[データベース]**を選択します。
一部の選択ダイアログでは、接続の構成時にデータベース名が入力されているため、**[データベース]**ドロップダウンリストはありません。
4. データベースの**[所有者]**を選択します。
[テーブル]のリストには、選択したデータベースで使用可能なビューとテーブルが自動的に反映されます。
一部のデータベースでは、データ選択プロセス時に、所有者を指定する必要はありません。
5. テーブルを選択します。
6. 各項目名の横にあるボックスにチェックを付けて、ロードする項目を選択します。
テーブル名の横にあるボックスにチェックを付けるとテーブルのすべての項目を選択できます。



既存の項目名をクリックして新しい名前を入力すると、項目名を編集できます。規定値では共通項目に加わるため、その他のテーブルにテーブルがリンクする方法に影響が及ぶ場合があります。

7. 追加のテーブルからデータを追加する場合は、そのテーブルを選択します。



項目の検索中に、データ選択ウィザードで項目名を変更することはできません。まず、テキストボックスの検索文字列を消去する必要があります。

4 スクリプトを使用したデータのロードと変換



また、同じテーブル内の2つの項目に同じ名前を付けることはできません。

- データ選択が終了したら、次のことを行います。
 - [スクリプトを挿入]** をクリックします。
データ選択ウィンドウが閉じ、選択内容に従って **LOAD /SELECT** ステートメントがスクリプトに挿入されます。

Microsoft Excel スプレッドシートからのデータの選択

Microsoft Excel スプレッドシートからデータを選択する際、ファイルには複数のシートが含まれる場合があります。各シートは個別のテーブルとしてロードされます。例外となるのは、シートに別のシートやロード済みのテーブルと同じ項目や列構造がある場合であり、この場合はテーブルが連結されます。

次の手順を実行します。

- データロードエディタの適切なフォルダー接続で、 をクリックします。
[ファイルを選択] ダイアログが表示されます。
- このフォルダー接続でアクセス可能なファイルのリストからファイルを選択します。
- データを選択する最初のシートを選択します。シート名の横にあるボックスにチェックを付けるとシートのすべての項目を選択できます。
- シートが正しく設定されていることを確認します。

テーブルデータの正しい解釈に役立つ設定

UI 項目	説明
項目名	テーブルに [埋め込まれた項目名] が含まれているか、[項目名が存在しない] か指定します。通常、Excel スプレッドシートでは、最初の行には埋め込まれた項目名が含まれます。[項目名なし] を選択すると、項目名は A,B,C... となります。
ヘッダーサイズ	テーブルヘッダーとして無視する行数を設定します。通常は、列形式になっていない一般情報が含まれる行です。

例

スプレッドシートは次のようになっています。

スプレッドシートの例

マシン:	AEJ12B	-	-
[日付]:	2015-10-05 09	-	-
タイムスタンプ	順序	演算子	生産額
2015-10-05 09:22	00122344	A	52
2015-10-05 10:31	00153534	A	67
2015-10-05 13:46	00747899	B	86

4 スクリプトを使用したデータのロードと変換

この場合、最初の2行を無視して、Timestamp、Order、Operator、およびYieldの項目を含むテーブルをロードしたいと考えるでしょう。これを行うには、次の設定を使用します。

最初の2行を無視して項目をロードするための設定

UI項目	説明
ヘッダーサイズ	2 これは、最初の2行がヘッダーデータとみなされて、ファイルをロードするときに無視されることを意味します。この場合、Machine: および Date: で開始する2行はテーブルデータの一部ではないため無視されます。
項目名	[埋め込まれた項目名] これは、読み込まれる最初の行が、それぞれの列の項目名として使用されることを意味します。この場合は、最初の2行がヘッダーデータになっているため、読み込まれる最初の行は3行目となります。

5. 各項目名の横にあるボックスにチェックを付けて、ロードする項目を選択します。



既存の項目名をクリックして新しい名前を入力すると、項目名を編集できます。テーブルは既定では共通項目により結合されているため、他のテーブルにリンクされる方法に影響することがあります。

6. データ選択が終了したら、次のことを行います。

- [スクリプトを挿入] をクリックします。

データ選択ウィンドウが閉じ、選択内容に従ってLOAD /SELECT ステートメントがスクリプトに挿入されます。



ODBC インターフェイスを使い、Microsoft Excel ファイルをデータソースとして使用することも可能です。その場合は、[すべてのファイル] データ接続ではなく、ODBC データ接続を使用する必要があります。


テーブル ファイルからのデータの選択

多数のデータファイルから、データを選択できます。

- テキストファイル (項目のデータがコンマ、タブ、セミコロンなどで区切られたもの。コンマ区切りCSV ファイル)
- HTML テーブル
- XML ファイル
- KML ファイル
- Qlik ネイティブ QVD および QVX ファイル
- 固定長レコードファイル
- DIF ファイル (Data Interchange Format)

次の手順を実行します。

4 スクリプトを使用したデータのロードと変換

1. データロードエディタの適切なフォルダー接続で、 をクリックします。
[ファイルを選択] ダイアログが表示されます。
2. このフォルダー接続でアクセス可能なファイルのリストからファイルを選択します。
3. [ファイル形式] で、ファイルの種類が正しく選択されていることを確認します。
4. ファイルが正しく設定されていることを確認します。ファイル設定は、ファイルの種類によって異なります。
5. 各項目名の横にあるボックスにチェックを付けて、ロードする項目を選択します。シート名の横にあるボックスにチェックを付けるとファイルのすべての項目を選択できます。



既存の項目名をクリックして新しい名前を入力すると、項目名を編集できます。テーブルは既定では共通項目により結合されているため、他のテーブルにリンクされる方法に影響することがあります。

6. データ選択が完了したら、[スクリプトの挿入] をクリックします。
7. データ選択ウィンドウが閉じ、選択内容に従って **LOAD /SELECT** ステートメントがスクリプトに挿入されます。



スペース内での編集権限を持つユーザーは、そのスペースで **DataFiles** の読み取り、書き込み、およびロードを行うことができます。その他のユーザーは **DataFiles** を表示することはできません。

ファイルの種類の設定の選択

区切り記号付きテーブル ファイル

これらの設定は、各レコードが改行で区切られた単一テーブルや、各項目が区切り文字で区切られた (CSV ファイルなど) の単一テーブルを含む、区切り記号付きテーブル・ファイルで検証されます。

区切り記号付きテーブル ファイルのファイル形式の設定

UI 項目	説明
区切り記号付きテーブル ファイルのファイル形式	[区切り記号付き] または [固定長レコード] に設定します。 選択すると、データ選択ダイアログが選択したファイル形式に応じて変化します。
項目名	テーブルに [埋め込まれた項目名] が含まれているか、[項目名が存在しない] か指定します。
区切り記号	テーブル ファイルで使用する [区切り記号] 文字を設定します。

4 スクリプトを使用したデータのロードと変換

UI 項目	説明
引用符	引用符の取り扱い方法を指定します。 [なし]= 引用符の使用不可 [標準]= 標準的な引用符 (項目値の最初および最後の文字として使用できます) [MSQ]= 新しいスタイルの引用符 (項目に複数行の内容を含めることができます)
ヘッダーサイズ	テーブルヘッダーとして省略される行数を設定します。
文字セット	テーブルファイルで使用される文字セットを設定します。
コメント	データファイルでは、レコード間にコメントを含めることができます。1つ以上の特殊文字 (// など) で始まる行はコメントを意味します。 ここでは、コメント行を示す文字を1つ以上指定します。Qlik Sense は、ここで指定された文字で始まる行をロードしません。
EOF を無視	項目値の一部として、データにファイルの終端を意味する文字が含まれている場合、[EOF を無視する] を選択します。

固定長レコードのデータファイル

固定長レコードのデータファイルには単一テーブルが含まれています。このテーブルでは通常、各レコード(データの行)に、空白やタブ記号が埋め込まれた固定項目サイズの列が多数含まれています。

項目の区切り位置の設定には、2通りの方法があります。

- [項目の区切り位置] に手動でコンマを入力して区切ります。各位置が項目の開始点となります。

1,12,24

- [項目の区切り] を有効にして、項目データのプレビューで項目の区切り位置を編集します。[項目の区切り位置] が、選択した位置により更新されます。次の操作が可能です。
 - 項目データのプレビュー内をクリックして、項目を区切ります。
 - 項目の区切りをクリックして、その区切りを削除します。
 - 項目の区切りをドラッグして、その区切りを移動させます。

固定長レコードのデータファイルのファイル形式の設定

UI 項目	説明
項目名	テーブルに [埋め込まれた項目名] が含まれているか、[項目名が存在しない] か指定します。
ヘッダーサイズ	テーブルヘッダーとして省略される行数に、[ヘッダーサイズ] を設定します。
文字セット	テーブルファイルで使用される文字セットに設定します。
タブサイズ	1つのタブ文字がテーブルファイルで示すスペースの数に設定します。

4 スクリプトを使用したデータのロードと変換

UI 項目	説明
レコード行のサイズ	1つのレコードがテーブル ファイルで占めるライン数に設定します。デフォルトは 1 です。

HTML ファイル

HTML ファイルには複数のテーブルを含めることができます。Qlik Sense は、<TABLE> タグにより、1つのテーブルとしてすべての要素を解釈します。

HTML ファイルのファイル形式の設定

UI 項目	説明
項目名	テーブルに [埋め込まれた項目名] が含まれているか、[項目名が存在しない] が指定します。
文字セット	テーブル ファイルで使用される文字セットを設定します。

XML ファイル

XML 形式で保存されているデータをロードできます。

XML ファイルには特定のファイル形式の設定はありません。

QVD ファイル

QVD 形式で保存されているデータをロードできます。QVD はネイティブの Qlik 形式で、Qlik Sense または QlikView でしか書き込みと読み出しを行えません。ファイル形式は、Qlik Sense スクリプトからデータを読み取る際の速度について最適化されていますが、それでも非常にコンパクトです。

QVD ファイルには特定のファイル形式の設定はありません。

QVX ファイル

Qlik data eXchange (QVX) 形式で保存されたデータをロードできます。QVX ファイルは、Qlik QVX SDK で開発されたカスタム コネクタにより作成されます。

QVX ファイルには特定のファイル形式の設定はありません。

KML ファイル

KML 形式で保存されているマップ ファイルをロードし、マップ ビジュアライゼーションで使用できます。

KML ファイルには特定のファイル形式の設定はありません。

スクリプトのプレビュー表示

挿入されるステートメントはスクリプトプレビューで表示されますが、[スクリプトをプレビュー] をクリックすると非表示になります。

LOAD ステートメントを含める

[LOAD ステートメントを含める] を選択すると、SELECT ステートメントは、入力として SELECT ステートメントを使用する LOAD ステートメントの後に生成されます。

4 スクリプトを使用したデータのロードと変換



テーブル内の項目名を変更する場合、**LOAD** ステートメントは、この設定に関係なく自動的に挿入されます。

インライン ロード

インライン ロードを使用して、データロードエディタでデータをロードできます。基本的なインライン ロードはテーブルを作成し、データ項目とレコードを挿入します。例:

```
MyTable:
Load * Inline [
Country, Year, Sales
Argentina, 2014, 66295.03
Argentina, 2015, 140037.89
Austria, 2014, 54166.09
Austria, 2015, 182739.87
];
```

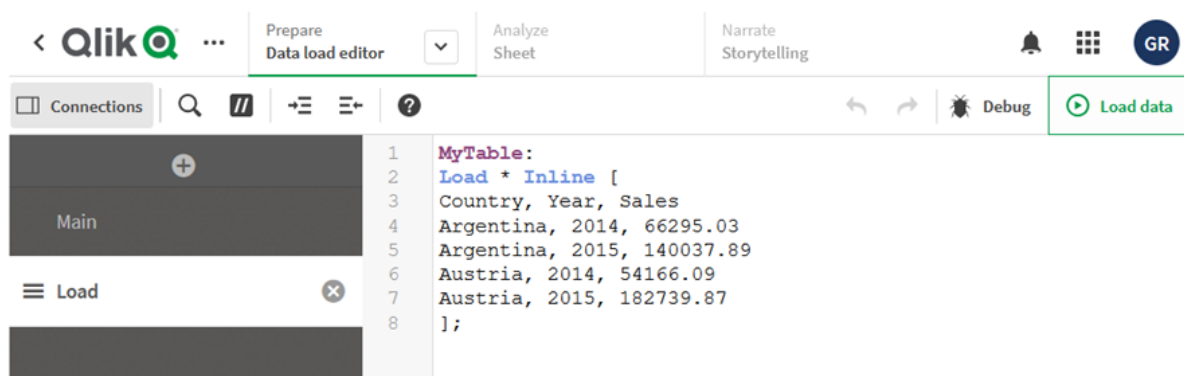
上記のインライン ロードには、次の構文が使用されます。

- **MyTable:** はデータのテーブルを作成します。
- **Load * Inline** は、インライン (データロードエディタ内) のデータロードを示します。
- 角括弧でデータを囲みます。
- **Load** ステートメントの最初の行は、データ項目を定義します。
- コマはデータ項目とレコードを区切ります。
- セミコロンは **Load** ステートメントを閉じます。

データは **[データのロード]** ボタンを使用してロードされます。

次の画像は、**Load** という新しいスクリプトセクションでのインライン ロードを示しています。

データロードエディタでのインライン ロード



データロード スクリプトの編集

ユーザーは、**データロードエディター**のテキストエディタでスクリプトを記述できます。ここで、データ選択時に生成した **LOAD** または **SELECT** ステートメントを手作業で変更し、新規スクリプトを入力できます。

4 スクリプトを使用したデータのロードと変換

Qlik Sense スクリプト構文を用いて記述する必要のあるスクリプトは、要素の区別がしやすいように、コメントは緑、**Qlik Sense** 構文のキーワードは青でハイライトされます。スクリプトの各行に番号が付けられます。


エディタには、ロードスクリプトの開発を支援するための多くの機能があります。この機能については、本セクションで説明します。

コマンドおよび関数の構文ヘルプへのアクセス

いくつかの方法で、**Qlik Sense** 構文キーワードの構文ヘルプにアクセスできます。

ヘルプ ポータルへのアクセス

2つの異なる方法で、**Qlik Sense** ヘルプポータルの詳細なヘルプにアクセスできます。

- ツールバーの  をクリックして、構文のヘルプモードに入ります。ヘルプモードで、構文のキーワード(青色でハイライト、下線付き)をクリックすると、構文のヘルプにアクセスできます。
- キーワードの内側または末尾にカーソルを置き、**Ctrl+H** を押します。



構文のヘルプモードでは、スクリプトの編集はできません。

オートコンプリート機能の使用

Qlik Sense スクリプトのキーワードの入力を開始すると、フォーム選択の一致するキーワードがオートコンプリートリストに表示されます。入力続けると、リストが絞られてきます。推奨される構文とパラメータが含まれるテンプレートから選択することができます。ツールヒントには、パラメータ、追加のステートメント、およびステートメントまたは関数のヘルプポータル説明へのリンクを含む関数の構文が表示されます。



また、キーボードショートカット **Ctrl+Space** キーを使用するとキーワードリスト、**Ctrl+Shift+Space** キーを使用するとツールヒントが表示されます。

設定済みのテストスクリプトの挿入

本製品にはインラインデータ項目一式をロードするテストスクリプトが事前に設定されています。このテストスクリプトを使用すると、テスト目的のデータセットを迅速に作成できます。

次の手順を実行します。



- **Ctrl + 00** を押します。

スクリプトにテストスクリプトコードが挿入されます。

コードのインデント

読みやすくするため、コードにはインデントを設定できます。

次の手順を実行します。

1. インデントを変更する行を1つまたは複数選択します。
2.  をクリックしてテキストをインデント(インデントを増大)するか、 をクリックしてテキストのインデントを解除します(インデントを減少)。



以下のキーボードショートカットも利用可能です。

Tab (インデント)





Shift+Tab (インデント解除)

テキストの検索および置換

スクリプトセクションのテキストを検索および置換できます。

テキストの検索

データロードエディターを開きます。次の手順を実行します。





1. ツールバーで  をクリックします。
検索ドロップダウンダイアログが表示されます。
2. 検索ボックスに、検索したいテキストを入力します。
検索結果がスクリプトコードの現在のセクションでハイライトされます。また、見つかったテキストインスタンスの数がセクションラベルの隣に示されます。
3.  と  をクリックして、結果リストを移動することができます。
4. ツールバーの  をクリックして、検索ダイアログを閉じます。



また、**[すべてのセクションを検索]** を選択して、あらゆるスクリプトセクションで検索を行うことができます。見つかったテキストインスタンスの数が各セクションラベルの隣に示されます。**[大文字小文字を区別する]** を選択すると、大文字と小文字を区別して検索できます。

テキストの置換

次の手順を実行します。

1. ツールバーで  をクリックします。
検索ドロップダウンダイアログが表示されます。
2. 検索ボックスに、検索したいテキストを入力します。
3. 置換ボックスに置換後の文字列を入力し、**[置換]** をクリックします。
4.  をクリックして、検索テキストの次のインスタンスを見つけ、以下のいずれかを行います。
 - **[置換]** をクリックしてテキストを置き換えます。
 -  をクリックして次のインスタンスを見つけます。
5. ツールバーの  をクリックして、検索ダイアログを閉じます。



[選択範囲をすべて置換] をクリックすると、現在のスクリプトセクション内で、検索テキストのすべてのインスタンスが置き換わります。置換機能では大文字と小文字が区別され、置換ボックスに入力されたテキスト通りに置き換えられます。置換が完了するとメッセージが表示され、置換されたインスタンスの数を確認できます。


スクリプトへのコメント入力

スクリプトコードにコメントを挿入したり、コメントマークを使用してスクリプトコードの一部を無効にすることができます。`//` (2本のスラッシュ) の右側に続く行のテキストはすべてコメントと見なされ、スクリプトを実行しても実行されません。

データロードエディタのツールバーには、コメント入力またはコメント解除コードへのショートカットがあります。関数はトグルとして機能します。つまり、選択したコードにコメントが付いていなければコメントが挿入され、コメントが付いていれば解除されます。

コメント入力


次の手順を実行します。

1. コメントアウトされていないコードの1行または複数の行を選択するか、行の先頭にカーソルを置きます。
2.  をクリックするか、または **Ctrl + K** を押します。

これで選択したコードがコメントアウトされます。

コメント解除

次の手順を実行します。

1. コメントアウトされたコードの1行または複数の行を選択するか、コメントされた行の先頭にカーソルを置きます。
2.  をクリックするか、または **Ctrl + K** を押します。

これで、選択されたコードがスクリプトの残りの部分で実行されます。



この他にも、スクリプトコードにコメントを挿入する方法には次のものがあります。

- **Rem** ステートメントを使用する。
- コードのセクションを `/*` と `*/` で囲む。

```
Rem This is a comment ;
```

```
/* This is a comment
```

```
that spans two lines */
```

```
// This is a comment as well
```

すべてのコードを選択

現在のスクリプトセクションですべてのコードを選択できます。

4 スクリプトを使用したデータのロードと変換

次の手順を実行します。

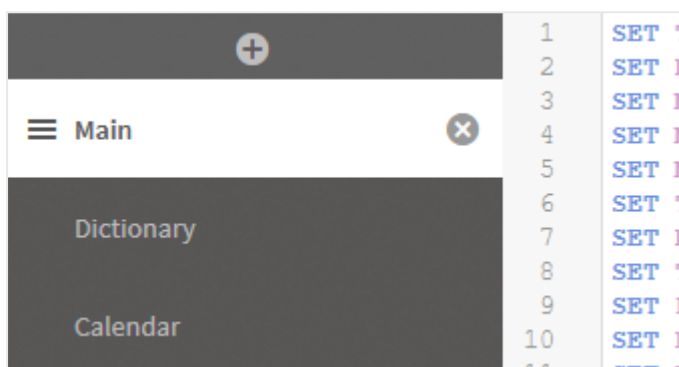
- **Ctrl + A** を押します。

現在のセクション内のすべてのスクリプトが選択されます。

スクリプト コードの整理

スクリプトをセクションごとに分割して、構造を整理することができます。スクリプトは、上から下に、セクションの順番通りに実行されます。スクリプトには、1つ以上のセクションが含まれる必要があります。

データロードエディター内の *Main*、*Dictionary*、および *Calendar* セクション



[データの追加] を使用してデータを追加した場合、データをロードするのに必要なスクリプトコードを含む [自動生成セクション] というデータロードスクリプトセクションが表示されます。

スクリプト セクションの操作

スクリプトセクションを整理するには、いくつかの方法があります。

- **+** をクリックして、コードを構造化するために、新しいスクリプトセクションを挿入できます。現在の選択行の後に、新しいセクションが挿入されます。
- セクションタブの横にある **×** をクリックして、削除します。削除することを確認します。



スクリプトセクションは一度削除すると元に戻すことができません。

- セクション名をクリックして、名前を入力して編集します。終了したら **Enter** を押すか、セクションの外側をクリックします。
- カーソルを **≡** に合わせてバーをドラッグし、セクションをドラッグして順番を並べ替えます。


データロードスクリプトのデバッグ

デバッグユーティリティをデータロードエディターで使用し、ブレイクポイントを使用してスクリプトを実行できます。それにより、変数の値と出力をスクリプト実行から調べることができます。

[出力] および [変数]、[ブレイクポイント] の一部または全部を表示するかどうかを選択できます。

4 スクリプトを使用したデータのロードと変換

デバッグ パネルを表示するには以下を行ってください。

- データロードエディタツールバーの  をクリックします。
デバッグ パネルがデータロードエディターの最下部に開きます。







デバッグ モードの実行中は、接続の作成や編集、データの選択、スクリプトの保存、データのロードは行えません。デバッグモードはデバッグ実行により開始され、スクリプトを実行、または実行が終了するまで続きます。


デバッグ ツールバー


データロードエディターのデバッグ パネルには、デバッグの実行を制御するための以下のオプションを含むツールバーがあります。

デバッグ ツールバーのオプション

UI 項目	説明
[ロード数の制限]	これを有効にすると、各データソースからロードするデータの行数を制限できます。これは、データソースが大きい場合に実行時間を短縮する上で役に立ちます。 ロードしたい行の数を入力します。  これは物理的なデータソースにのみ適用されます。例えば、自動的に生成されたインラインロードは制限されません。
	次のブレイクポイントに達するまでデバッグ モードで実行を開始または継続することはできません。
	次のコードのラインに進みます。
	ここで実行を終了します。すべてのコードが実行される前に終了すると、結果のデータモデルには実行済みの行までしか表示されません。

アウトプット

[アウトプット] には、デバッグの実行中に生成されたすべてのメッセージが表示されます。新しいメッセージが表示されたら、 をクリックしてアウトプットのスクロールをロックするよう選択できます。

さらに、アウトプットメニュー () には以下のオプションも含まれています。

アウトプット メニューのオプション

UI 項目	説明
[クリア]	クリックすると、アウトプットのメッセージがすべて削除されます。
[すべてのテキストを選択]	クリックすると、アウトプットのメッセージがすべて選択されます。
[最下部までスクロール]	クリックすると、アウトプットの最後のメッセージまでスクロールします。

変数

[変数]には、保留変数とシステム変数、スクリプトで定義された変数がすべてリストアップされ、スクリプト実行中の現在の値が表示されます。

変数をお気に入りに設定

特定の変数をお気に入りとして設定しておけば、実行中にその変数を調べることができます。お気に入りの変数は変数リストの一番上に表示され、黄色い星印が付きます。変数をお気に入りとして設定するには、以下を行ってください。

- 変数の隣にある★をクリックします。
★が黄色になり、変数が変数リストの一番上に移動します。

変数のフィルター

変数メニューで以下のオプションを使用すると、フィルターが適用され、選択したタイプの変数だけが表示されます。☰

変数メニューのオプション

UI項目	説明
[すべての変数を表示]	クリックすると、あらゆるタイプの変数が表示されます。
[システム変数を表示]	クリックすると、システム変数が表示されます。 システム変数は Qlik Sense で定義されますが、変数の値はスクリプトで変更できません。
[保留変数を表示]	クリックすると、保留変数が表示されます。 保留変数は Qlik Sense で定義された変数で、値は変更できません。
[ユーザー定義変数を表示]	クリックすると、ユーザー定義変数が表示されます。 ユーザー定義変数は、スクリプトで定義した変数です。

ブレイクポイント

スクリプトにブレイクポイントを追加して、特定のコードのラインでデバッグの実行を中止し、その時点での変数値とアウトプットのメッセージを調べることができます。ブレイクポイントに達すると、実行を中止するか、次のブレイクポイントまで続けるか、次のコードのラインに進むか選択できます。スクリプトのブレイクポイントがすべて、セクションとライン番号への参照とともにリストアップされます。

ブレイクポイントの追加

コードのラインにブレイクポイントを追加するには、以下のいずれかを行ってください。



- スクリプトで、ブレイクポイントを追加したいライン番号のすぐ右にある領域をクリックします。
ライン番号の隣にある●は、このラインにブレイクポイントがあることを示します。



デバッグパネルが閉じている場合でもブレイクポイントを追加できます。


ブレイクポイントの削除


以下のいずれかを行うとブレイクポイントを削除できます。

- スクリプトで、ライン番号の隣にある  をクリックします。
- ブレイクポイントのリストで、ブレイクポイントの隣にある  をクリックします。

 をクリックして、**[すべて削除]** を選択し、あらゆるブレイクポイントをスクリプトから削除することも可能です。

ブレイクポイントの有効化と無効化


ブレイクポイントを作成するとデフォルトで有効になり、ブレイクポイントのリストで当該ブレイクポイントの隣に  と表示されます。リストで個々のブレイクポイントを選択または選択解除すると、そのブレイクポイントを有効または無効にできます。

また、ブレイクポイントメニューには以下のオプションもあります。

- **[すべて有効化]**
- **[すべて無効化]**

ロードスクリプトの保存


スクリプトを保存すると、アプリ全体が保存されますが、データは自動的にリロードされません。

スクリプトが保存されても、アプリにはまだ前のリロードによる古いデータが残っており、ツールバーにその旨が表示されます。アプリを新しいデータで更新する場合は、データロードエディターのツールバーで  をクリックします。データロードが完了すると、スクリプトはアプリに自動的に保存されます。


データロードエディターは、ロードスクリプトに変更を加えると、その変更結果を自動的に保存します。**CTRL+S** を押して強制的に保存できます。



スクリプトは自動的に *Qlik Sense Desktop* に保存されません。手動でスクリプトを保存する必要があります。

スクリプトを保存する際、構文エラーが自動的にチェックされます。構文エラーはコード内でハイライトされ、構文エラーを含むすべてのスクリプトセクションのセクションラベルの隣に  が表示されます。

スクリプト実行によるデータのロード

ツールバーで  をクリックしてスクリプトを実行し、アプリにデータをリロードします。アプリはデータのロード前に、自動的に保存されます。

[データのロード進捗状況] ダイアログが表示されます。ここでは、データのロードを **[中止]** できます。データロードが完了すると、ダイアログのステータスが更新され (**[成功]** または **[失敗]** のいずれか)、エラーの可能性や合成キーに関する警告などの要約が表示されます。ダイアログを閉じた後も、要約は **[アウトプット]** で確認することができます。

4 スクリプトを使用したデータのロードと変換



実行に成功した場合、常に[データのロード進捗状況]ダイアログが自動的に閉じるよう設定するには、[終了後、閉じる]を選択します。

[データロードエディタ]でのキーボードショートカット

[データロードエディタ]環境で効果的かつ簡単に作業を行うために使用できるショートカットが数多く用意されています。

キーボードショートカット



キーボードのショートカットは、**Windows** を想定しています。**Mac OS** の場合は、**Ctrl** の代わりに **Cmd** を使用してください。

[データロードエディタ]でのキーボードショートカット

ショートカット	アクション
Ctrl+0, Ctrl+0	サンプルデータを生成します。
Alt+1	出力パネルを表示、または表示されている場合は非表示にします。
Alt+2	デバッグツールがオンであれば、変数パネルを表示するか、または表示されている場合には非表示にします。
Alt+3	デバッグツールがオンであれば、ブレイクポイントパネルを表示するか、または表示されている場合には非表示にします。
Alt+F5	デバッグツールを表示、または表示されている場合は非表示にします。
Alt+F6	デバッグツールがオンであればデバッグを実行します。
Alt+F7	デバッグツールがオンであればデバッガーの次のステップに進みます。
Alt+F8	デバッグツールがオンであればデバッガーを停止します。
F9	デバッグのブレイクポイントの挿入と削除を切り替えます。
Alt+F10	右側のパネルを表示、または表示されている場合は非表示にします。
Alt+F11	スクリプトエディタを全画面に拡大します。
Ctrl+C	選択されたアイテムをクリップボードにコピーします。
CTRL+F	[検索エントリを検索]フィールドを表示、またはこのフィールドが開いている場合は非表示にします。
Ctrl+H	データロードエディターまたは数式エディター内で、現在の選択機能に関連するオンラインヘルプを開きます。
Ctrl+K	コードコメント行の挿入と削除を切り替えます。
Ctrl+P	現在のビューまたはアクティブなシートストーリーを印刷します。

4 スクリプトを使用したデータのロードと変換

ショートカット	アクション
Ctrl+S	変更を保存します。
Ctrl+V	クリップボードから直前にコピーされたアイテムを貼り付けます。
Ctrl+X	選択されたアイテムを切り取り、クリップボードにコピーします。Google Chrome ブラウザを使用している場合、データロードエディターまたは数式エディターでカーソルが列の前にある場合、何も選択しなくても、行全体が切り取られます。
Ctrl+Z	操作を元に戻します。以前の操作を元に戻すには、このショートカットを繰り返し押します。
Alt+Insert	スクリプトに新しいセクションを挿入します。
Alt+PgUp	前のセクションに移動します。
Alt+PgDn	次のセクションに移動します。
Ctrl+Shift+Enter/Return	データをリロードします。
Ctrl+Shift+Space	ツールヒントを表示します。(Qlik Sense Desktop では対応していません)
Ctrl+Space	自動テキスト文字列を使用してオートコンプリートします。
検索フィールドでの Enter/Return	検索文字列の次のインスタンスを検索します。
置換フィールドでの Enter/Return	検索文字列の選択したインスタンスを置換します。
検索フィールドまたは置換 フィールドでの Esc	[検索 エントリを検索] フィールドを閉じます。
Shift+Tab	スクリプトで行のインデントを解除します。
Tab	スクリプトで行をインデントします。

4.3 スクリプト構文とデータ構造を理解する

抽出、変換、ロード

一般的に、アプリにデータをロードする方法は、抽出、変換、ロードプロセスで説明できます。

- 抽出
最初のステップは、データソースシステムからのデータ抽出です。スクリプトで、**SELECT** または **LOAD** ステートメントを使用してこれを定義します。この 2 つのステートメントには、以下のような違いがあります。
 - SELECT** は、ODBC データソースあるいは OLE DB プロバイダからのデータ選択に使用します。**SELECT SQL** ステートメントは、Qlik Sense ではなく、データプロバイダによって評価されます。
 - LOAD** は、ファイルのデータやスクリプトで定義されたデータ、以前にロードされたテーブル、ウェブページ、後続の **SELECT** ステートメントの結果からロードされたデータ、または自動的に生成されたデータのロードに使用します。
- 変換

4 スクリプトを使用したデータのロードと変換

変換段階では、スクリプト関数とルールを使用してデータを操作し、目的のデータモデル構造を取得します。標準的な操作は以下のとおりです。

- 新しい値の計算
- コード化された値の変換
- 項目名の変更
- テーブルの結合
- 値の集計
- ピボット化
- データ検証

- ロード

最後のステップで、アプリに定義したデータモデルをロードするスクリプトを実行します。

Qlik Sense でデータを効率良く取り扱えるデータモデルを構築することが目標です。これは通常、循環参照しない、ある程度標準的なスタースキーマあるいはスノーフレイクスキーマを作り、各エンティティが個々のテーブルで維持されるようなモデルを作成することを意味します。典型的なデータモデルは、以下のようになります。

- (ユニット数、売上金額、予算額などの) メジャーの計算に使用される、軸および数値のキーを含む中心ファクトテーブル。
- (製品、顧客、カテゴリ、カレンダー、サプライヤなど) 属性をすべて備えた軸を含む周辺のテーブル。



多くの場合、集計などのタスクは、ロードスクリプトでより高度なデータモデルを構築するか、チャート式で集計を実行することで解決できます。一般的に、ロードスクリプトにデータ変換が含まれていると、より高いパフォーマンスを実現できます。



データモデルを紙にスケッチすることも役立ちます。構造を目で確認することで、どのデータを抽出し、どの変換を実行すべきかを判断しやすくなります。

データロードのステートメント

データは、**LOAD** または **SELECT** ステートメントによりロードされます。各ステートメントは、1つの内部テーブルを生成します。テーブルは常にデータのリストとして表示されます。各レコード(行)には、オブジェクトタイプの新しいインスタンス、各項目(列)には、オブジェクトの特定の属性またはプロパティが含まれます。

この2つのステートメントには、以下のような違いがあります。

- **SELECT** は、ODBC データソースあるいは OLE DB プロバイダからのデータ選択に使用します。**SELECT SQL** ステートメントは、**Qlik Sense** ではなく、データプロバイダによって評価されます。
- **LOAD** は、ファイルのデータやスクリプトで定義されたデータ、以前にロードされたテーブル、ウェブページ、後続の **SELECT** ステートメントの結果からロードされたデータ、または自動的に生成されたデータのロードに使用します。

ルール

以下の規則はデータを **Qlik Sense** にロードする際に適用されます。

4 スクリプトを使用したデータのロードと変換

- **Qlik Sense** では、**LOAD** ステートメントで生成されるテーブルと**SELECT** ステートメントで生成されるテーブルに違いはありません。つまり、複数のテーブルをロードする場合、**LOAD** と**SELECT** のいずれか一方のステートメントを使用するか、これら2つを組み合わせるかは問題にはなりません。
- **Qlik Sense** ロジックでは、ステートメント内またはデータベース内の元のテーブルの項目の順序は任意です。
- 項目名は、項目を識別したり、関連付けを作成する処理で使用されます。項目名には大文字と小文字の区別があり、そのために項目名を変更する必要がしばしば生じます。

スクリプトの実行

典型的な **LOAD** または **SELECT** ステートメントでは、イベントの順序は概ね次のようになります。

1. 数式の評価
2. **as** による項目名の変更
3. **alias** による項目名の変更
4. 項目名の修飾
5. 項目名が一致する場合はデータのマッピング
6. 内部テーブルへのデータの保存

項目

項目とは、**Qlik Sense** の主要なデータ伝送エンティティです。項目には通常、項目値と呼ばれる値がいくつか含まれています。データベース用語では、**Qlik Sense** によって処理されるデータは、データファイルから読み込まれたものであると定義されます。ファイルは複数の項目から構成されており、各データエントリがレコードとなります。ファイル、項目、レコードという用語は、それぞれテーブル、列、行と同等です。**Qlik Sense AQL** ロジックは、項目とその項目値でのみ機能します。

項目データは、**LOAD** または **SELECT**、**Binary** ステートメントを介して、スクリプトにより取得されます。項目のデータを変更する唯一の方法は、スクリプトを再実行することです。ユーザーがレイアウトやオートメーションを使用して項目値を操作することはできません。**Qlik Sense** に読み込まれた項目値は、論理選択および計算目的のためにのみ、表示および使用することができます。

項目値には、数値データや英数字 (テキスト) データを含めることができます。数値は、実際にはデュアル値、つまり数値とその時点の書式設定されたテキスト表現を持ちます。シートオブジェクトなどにはテキスト表現が表示されます。

項目の内容は、フィルターパネルに表示されます。

自動取得項目

関連項目のグループがある場合、または軸またはメジャーの作成時に項目の情報を関連性のあるパーツに分解できる場合、自動取得項目の生成に使用できる項目定義を作成できます。たとえば、年、月、週番号、曜日といった複数の属性を取得する元となる日付項目を作成できます。これらのすべての属性は **Qlik Sense** 日付関数を使用して軸の数式で計算できますが、日付タイプのすべての項目に共通のカレンダー定義を作成することもできます。項目定義は、データロードスクリプトに記載します。



Qlik Sense のデフォルトのカレンダー項目定義は、**データマネージャー**を使用してロードされたデータ項目の **autoCalendar** に含まれています。詳細については、「**アプリへのデータの追加 (page 16)**」を参照してください。

カレンダー項目定義の宣言

自動取得項目の定義を作成するには、**Declare** ステートメントを使用します。このステートメントでは、項目のさまざまな属性を定義することができます。この場合は、日付関連の属性を定義します。各項目は、

<expression> **As** field_name **tagged** tag のように記述されます。タグ (複数可) の設定は任意ですが、自動取得項目のソート順に影響を与える場合があります。自動取得項目の生成元のデータ項目を参照するには **\$1** を使用します。



カレンダー項目定義の名前として **autoCalendar** は使用しないでください。この名前は、自動生成されるカレンダーテンプレートとして予約済みです。

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');
```

Derive を使用したカレンダーへのデータ項目のマッピング

次に、**Derive** ステートメントを使用して、既存のデータ項目をカレンダーにマッピングします。これで、自動取得項目が作成されます。これは、データロードスクリプトで以下の 3 つの方法で実行できます。

- 項目名で特定の項目をマッピングする。
`DERIVE FIELDS FROM FIELDS OrderDate,ShippingDate USING Calendar;`
- 1 つまたは複数の特定の項目タグを持つすべての項目をマッピングする。
`DERIVE FIELDS FROM EXPLICIT TAGS ('$date') USING Calendar;`
- 項目定義のタグのうちの 1 つによってタグ付けされたすべての項目をマッピングする (上の例の **\$date**)。
`DERIVE FIELDS FROM IMPLICIT TAG USING Calendar;`

この場合、ここで示す 3 つの例のいずれかを使用できます。

ビジュアライゼーションでの自動取得日付項目の使用

カレンダー定義を作成済みで項目が上の例のようにマッピングされている場合、**Qlik Sense** が自動取得日付項目を認識します。自動取得日付項目は、**[項目]** アセットパネルの **[日付および時刻項目]** セクションで利用できます。すべての自動取得項目は、数式エディタでも確認できます。また軸の作成、編集時にも確認できます。

4 スクリプトを使用したデータのロードと変換

項目 タグ

項目 タグを使用して、データモデルの項目にメタデータを追加できます。項目 タグには、次の 2 つの異なるタイプがあります。

- システム項目 タグ
システム項目 タグは、スクリプトが実行され、データがロードされると自動的に生成されます。一部のタグはスクリプトで操作できます。システム タグの前には、常に \$ 記号が付きます。
- カスタム項目 タグ
Tag ステートメントを使用して、データロードスクリプトの項目にカスタム タグを追加できます。カスタム タグに、システム タグと同じ名前を使用することはできません。

システム項目 タグ

次のシステム フィールドはデータのロード時に自動的に生成されます。

システム項目 タグ

Tag	説明	スクリプトで操作が可能かどうか
\$system	スクリプト実行時にQlik Senseによって生成されるシステム フィールド。	いいえ
\$key	複数のテーブル間のリンクを提供するキー項目。	いいえ
\$keypart	この項目は、1 つ以上の合成キーの一部です。	いいえ
\$syn	合成キー	いいえ
\$hidden	非表示項目。つまり、ビジュアライゼーション、軸、またはメジャーの作成時に項目選択リストに表示されません。数式で非表示項目を使用できますが、項目名を入力する必要があります。 HidePrefix および HideSuffix システム変数を使用して、非表示にする項目を設定することができます。	はい
\$numeric	項目内のすべての値 (NULL以外)は数値です。	はい
\$integer	項目内のすべての値 (NULL以外)は整数です。	はい
\$text	項目内の値は数値ではありません。	はい
\$ascii	項目の値には標準のASCII 文字列のみが含まれます。	はい
\$date	項目内のすべての値 (NULL以外)は日付 (整数)として解釈されます。	はい
\$timestamp	項目内のすべての値 (NULL以外)はタイムスタンプとして解釈されます。	はい
\$geoname	項目値には、ポイント項目 (\$geopoint) またはエリア項目 (\$geomultipolygon) に関連する地理的な場所の名前が含まれます。	はい

4 スクリプトを使用したデータのロードと変換

Tag	説明	スクリプトで操作が可能かどうか
\$geopoint	項目値には、マップ上のポイントを[経度、緯度]形式で表す地理的なポイントデータが含まれます。	はい
\$geomultipolygon	項目値には、マップ上のエリアを表す地理的なポリゴンデータが含まれます。	はい

自動取得項目タグ

次のタグは、自動取得項目の宣言時に、折れ線グラフの連続した軸上で、項目の使用方法と表示方法を指定するのに使用できます。タグは、データロードスクリプトで操作できます。

自動取得項目タグ

Tag	説明
\$axis	\$axisタグは、チャートの連続した軸上で、項目が目盛を生成するように指定するのに使用されます。
\$qualified \$simplified	軸ラベルの修飾済みかつ簡素化されたバージョンを指定するには、2つの異なる項目を取得します。修飾済みの項目は、軸が深いレベルにズームしたときに、全コンテキストを示すようにラベルとして表示されます。 例えば、四半期別にデータを表示する場合には、次の2つの項目を生成できます。 簡素化された項目。「Q1」のように四半期を表す \$simplified タグと、「2016-Q1」のように年と四半期を表す \$qualified タグが付けられています。 時間軸をズームアウトすると、軸には、簡素化された項目を使用して、年 (2016) と四半期 (Q1) の2つのレベルでラベルが表示されます。ズームインすると、軸には四半期と月のラベルが表示され、修飾済みの項目 (2016-Q1) を使用して、四半期の完全な年のコンテキストが提供されます。
\$cyclic	\$cyclicタグは、周期的な項目に使用されます。例えば、四半期または月を、デュアルデータで表現します。

システム項目

データソースから抽出された項目に加えて、Qlik Sense はシステム項目を生成します。システム項目は「\$」で始まり、ビジュアライゼーション (フィルター パネルやテーブルなど) では通常の項目のように表示できます。システムフィールドは、データのロード時に自動的に作成され、主にアプリ設計を支援する目的で使用されます。

利用可能なシステム項目

次のシステム項目を使用できます。

4 スクリプトを使用したデータのロードと変換

利用可能なシステム項目

システム フィールド	説明
\$Table	ロードされるすべてのテーブルが含まれます。
\$Field	ロードされるテーブルのすべての項目が含まれます。
\$Fields	各テーブルの項目数が含まれます。
\$FieldNo	テーブル内での項目の位置が含まれます。
\$Rows	テーブル内の行数が含まれます。

どのシステム フィールドもスクリプトで操作することはできません。

ビジュアライゼーションにおけるシステム項目の使用

システム フィールドデータは関連付けられます。例えば、2つのフィルターパネルを追加する場合 (1つは\$Tableにより、もう1つは\$Fieldにより) や、テーブルを選択する場合は、\$Field選択したテーブルのフィールドが絞込値としてフィルターパネルに表示されます。

システム フィールドは、アセットパネルの項目リストには含まれていません。数式エディタに含まれています。アセットパネルでシステム フィールドを使用したい場合は、手動で入力して参照する必要があります。

アセットパネルの軸の場合

=*\$Field*

項目名の変更

必要な関連付けを取得するために、項目名の変更が必要となる場合があります。項目名を変更する主な理由は3つあります。

- 2つの項目は同じことを意味しているにも関わらず、異なる名前が付けられています。

- *ID* テーブルの項目 *Customers*
- *CustomerID* テーブルの項目 *Orders*

2つの項目は特定の顧客 ID コードを示しているため、同じ名前 (*CustomerID* など) を付ける必要があります。

- 2つの項目は同じ名前ですが、実際には異なるものを表しています。

- *Date* テーブルの項目 *Invoices*
- *Date* テーブルの項目 *Orders*

この2つの項目は、*InvoiceDate* と *OrderDate* などに名前を変更した方がよいでしょう。

- データベース内にスペルミスがある場合や、大文字と小文字に関する規則が異なっている可能性があります。

項目名はスクリプト内で変更できるため、元のデータを変更する必要はありません。例示されているように、項目の名前を変更する方法は2通りあります。

Example 1: alias ステートメントの使用

LOAD または **SELECT** ステートメントの前に **alias** ステートメントを使用します。

4 スクリプトを使用したデータのロードと変換

```
Alias ID as CustomerID;  
LOAD * from Customer.csv;
```

Example 2: as 指定子の使用

LOAD または **SELECT** ステートメントに **as** 指定子を含めます。

```
LOAD ID as CustomerID, Name, Address, Zip, City, State from Customer.csv;
```

論理 テーブル

LOAD ステートメントまたは **SELECT** ステートメントは、それぞれ 1 つのテーブルを生成します。通常 Qlik Sense は、各ステートメントの結果を 1 つの論理テーブルとして処理します。ただし、この規則にはいくつかの例外があります。

- 2 つ以上のステートメントにより、同一の項目名を持つテーブルが生成された場合は、これらのテーブルが連結されて、1 つの論理テーブルとして処理されます。
- **LOAD** または **SELECT** ステートメントの前に次のいずれかの修飾子が付いている場合、データが変更されるか、別々に処理されます。

論理 テーブル

テーブル	説明
concatenate	このテーブルは、別の名前のテーブルまたは最後に作成された論理テーブルに連結 (追加) されます。
crosstable	このテーブルではピボットは無効です。つまり、このテーブルは、クロス集計形式から列形式に変換されます。
generic	このテーブルは、他のいくつかの論理テーブルに分割されます。
info	このテーブルは、論理テーブルとしてではなく、ファイル、サウンド、URL などの外部情報を含む info テーブルとしてロードされます。
intervalmatch	このテーブル (厳密に 2 つの列を含む) は、数値間隔として解釈されます。これらは、指定された項目内の離散数値に関連付けられます。
join	このテーブルは Qlik Sense によって、共通項目において別の名前のテーブルまたは最後に作成された論理テーブルと結合されます。
keep	このテーブルは別の名前のテーブルまたは最後に作成された論理テーブルにより、共通する項目に縮小されます。
mapping	このテーブル (厳密に 2 つの列を含む) は、マッピング テーブルとして読み取られます。このテーブルが他のテーブルに関連付けられることはありません。
semantic	このテーブルは、論理テーブルとしてではなく、結合されない関係 (先行や後続など、同じタイプの他のオブジェクトに対するその他の参照) を含むセマンティックテーブルとしてロードされます。

データがロードされると、論理テーブルが関連付けられます。

テーブル名

Qlik Sense テーブルは、**Qlik Sense** データベースに保存されるときに名前が付けられます。テーブル名は、**LOAD** ステートメントの **resident** 節や **peek** 関数を含む数式などで使用され、レイアウト内の **\$Table** システム項目に表示されます。

テーブルの名前は、次の規則に従って付けられます。

1. ラベルが **LOAD** または **SELECT** ステートメントの直前にある場合は、そのラベルがテーブル名として使用されます。ラベルの後にはコロンが必要です。

Table1:

```
LOAD a,b from c.csv;
```

2. ラベルが指定されていない場合、**LOAD** または **SELECT** ステートメント内のキーワード **FROM** の直後にあるファイル名またはテーブル名が使用されます。最大 32 文字まで使用されます。ファイル名が使用される場合、拡張子は省かれます。
3. インラインでロードされたテーブルには、**INLINExx** という名前が付けられます。xx は番号です。最初のインラインテーブルは **INLINE01** という名前になります。
4. 自動生成されるテーブルには、**AUTOGENERATExx** という名前が付けられます。xx は番号で、最初の自動生成テーブルは **AUTOGENERATE01** という名前になります。
5. 上の規則に従って生成されたテーブル名が既存のテーブル名と競合する場合は、名前に **-x** が付けられます。x は番号で、競合がなくなるまで増え続けます。たとえば、「**Budget**」という名前のテーブルが 3 つある場合には、それぞれ **Budget**、**Budget-1**、および **Budget-2** という名前が付けられます。

テーブル名には **section access**、**section application**、およびマッピング テーブルという 3 つの個別のドメインがあります。**section access** および **section application** 内で生成されるテーブル名は個別に扱われます。参照されるテーブル名がセクション内でない場合、**Qlik Sense** は他のセクションも検索します。マッピング テーブルは個別に扱われ、他の 2 つのドメインのテーブル名とは一切関係がありません。

テーブル ラベル

LOAD ステートメントの **resident** 節または **peek** 関数を含む数式などが後で参照するために、テーブルにラベルを付けることができます。ラベルは数値または文字で構成される任意の文字列で、テーブルを作成する最初の **LOAD** または **SELECT** ステートメントの前に配置します。ラベルの後にはコロン「:」が必要です。

空白を含むラベルは、単一引用符や二重引用符、または角括弧などで囲む必要があります。

Example 1:

Table1:

```
LOAD a,b from c.csv;
```

```
LOAD x,y from d.csv where x=peek('a',y,'Table1');
```

Example 2: 空白を含むテーブル ラベル

```
[All Transactions]:
```

4 スクリプトを使用したデータのロードと変換

```
SELECT * from Transtable;  
LOAD Month, sum(Sales) resident [All Transactions] group by Month;
```

論理テーブル間の関連付け

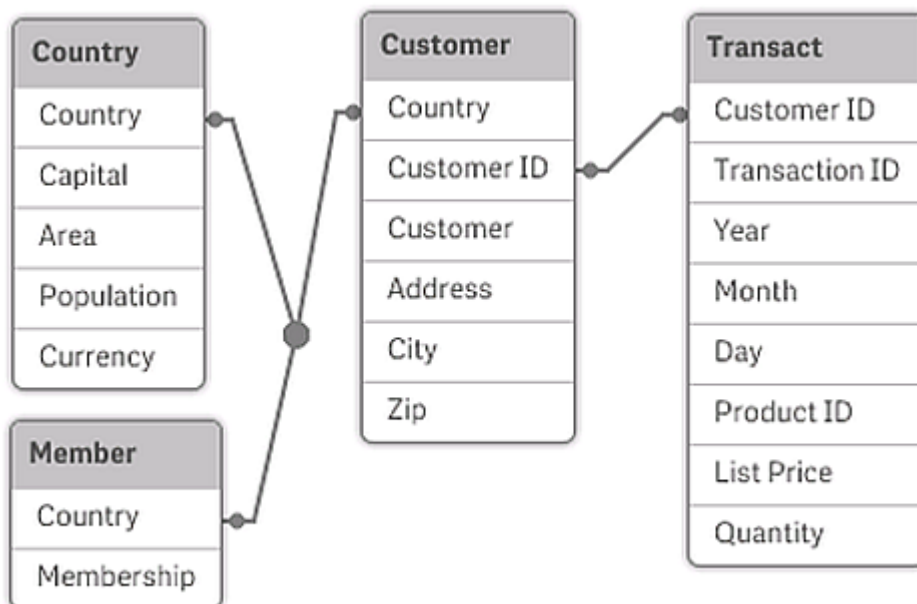
1つのデータベースに複数のテーブルを含めることができます。各テーブルは、何かのリストと見なすことができます。つまり、リスト内の各レコードは、何らかの種類のオブジェクトのインスタンスを表します。

2つのテーブルが異なる対象を扱うリストの場合であっても(例えば、1つは顧客のリストで、もう1つが請求書のリストなど)、この2つのテーブルに顧客番号などの共通する項目があるなら、多くの場合、それはこの2つのテーブルの間に関係が存在することを示しています。標準のSQLクエリツールでは、ほとんどの場合、この2つのテーブルは結合されます。

Qlik Sense スクリプトで定義されたテーブルは、論理テーブルと呼ばれます。**Qlik Sense** は項目名に基づいてテーブルを関連付け、選択が行われると(例えば、フィルターパネルの項目値を選択すると)結合を実行します。

つまり、関連付けは結合とほとんど同じです。唯一の違いは、結合はスクリプトが実行されるときに実行され、この結合の結果として論理テーブルが生成されます。それに対して、関連付けは論理テーブルの作成後に実行され、常に論理テーブル間で行われます。

上記の例では、国 (*Country*)、顧客 (*Customer*)、取引 (*Transaction*)、およびメンバーシップ (*Membership*) の4つのリストは、*Country* と *CustomerID* の項目によって互いに関連付けられています。



Qlik Sense の関連付けとSQL 自然外部結合の違い

Qlik Sense の関連付けは **SQL** 自然外部結合に類似しています。ただし、**SQL** の外部結合では通常、あるテーブルから別のテーブルへの一方向の関係となるため、関連付けの方がより一般的です。関連付けでは、常に完全な(双方向)自然外部結合が形成されます。

関連付け項目のレコード数情報

ほとんどの関連付け項目 (2 つ以上のテーブル間に共通する項目) の使用には、いくつかの制限があります。項目が複数のテーブルに存在する場合、Qlik Sense では、データのレコード数の計算に使用するテーブルを特定することが困難になります。

Qlik Sense は、データを分析することにより、計算に含めるメインテーブルを特定するための確実な方法が存在するかどうかを確認しますが (存在する場合もあります)、ほとんどの場合は推測することしかできません。間違った推測は致命的な結果を引き起こす可能性があるため (Qlik Sense が計算ミスをしたように見えます)、関連付け項目のデータ解釈があいまいな場合、プログラムは特定の操作が許可されないように設計されています。

項目関連付けの制限

1. 項目を表示するフィルターパネルで、レコード数情報を表示することはできません。
2. 項目の統計ボックスでは、ほとんどの統計ボックスに n/a が表示されます。
3. チャートでは、**Distinct** 修飾子をアクティブにしない限り、項目のレコード数情報に基づく関数 (Sum、Count、Average など) を含む数式を作成することはできません。Qlik Sense はリロードするたびにすべてのチャート式をスキャンして、データ構造の変更により、あいまいさが発生していないか確認します。あいまいな数式が見つかった場合は、警告ダイアログが表示され、その数式は無効になります。問題が修正されるまで、その数式を有効にすることはできません。ログファイルが有効にされている場合は、すべてのあいまいな数式がログにリストされます。

回避方法

これらの制限に対しては、簡単な対処方法があります。レコード数が作成されるテーブルから、項目を新しい名前でも余分にロードします。次に、レコード数を含むフィルターパネル、統計ボックス、またはチャートの計算で、その新しい項目を使用します。

合成キー

Qlik Sense は、2 つ以上のデータテーブルに 2 つ以上の共通の項目がある場合に合成キーを作成します。合成キーとは、複合キーが発生するすべての組み合わせを表す匿名項目です。

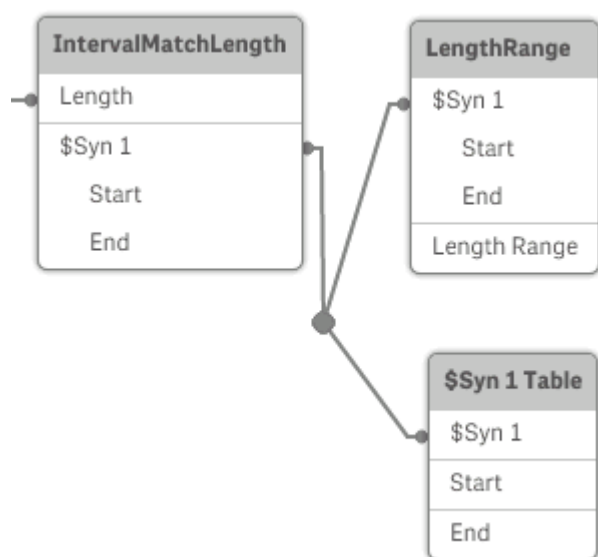
データのロード中に、合成キーに関する警告が表示された場合は、データモデルビューアでデータ構造を確認する必要があります。データモデルが正しいかどうかを、自分で確認してください。データモデルが正しくても、ほとんどの場合、合成キーが多すぎるとスクリプトでエラーが発生する原因となります。

複数の合成キーが不正なデータモデルの兆候となることはよくあります。ですが、他の合成キーに基づく合成キーがある場合、これは不正なデータモデルの明らかな兆候です。



データ量やテーブル構成、その他の要因により合成キーが増加すると、Qlik Sense はそれらを効率的に処理できなくなる場合があります。結果として大量の時間とメモリを消費してしまいます。そうした場合は、すべての合成キーを削除して、スクリプトの再構築が必要となります。

合成キー \$Syn 1 に関連付けられた 3 つのテーブル。



合成キーの処理

合成キーを回避する必要がある場合は、データのロードスクリプトで次のような方法を使って解決できます。

- キーとして使用されている論理的にリンクしている2つのテーブルの項目のみをチェックする。
 - “Comment” や “Remark”、“Description” のような項目が互いに関係のない複数のテーブルに存在する場合、キーとして使用できない。
 - “Date” や “Company”、“Name” のような項目が複数のテーブルに存在し、一意の値を持っているがロールが異なる場合 (Order Date/Shipping Date、Customer Company/Supplier Company) は、キーとして使用できない。
- 冗長項目が使用されておらず、必要な項目のみが接続されていることを確認する。たとえば、日付をキーとして使用する場合は、**year**、**month**、または **day_of_month**などを複数の内部テーブルから同じ日付がロードされていないことを確認してください。
- 必要に応じて、通常は **AutoNumber** スクリプト関数内で文字列連結を使用している独自の非複合キーを作成する。

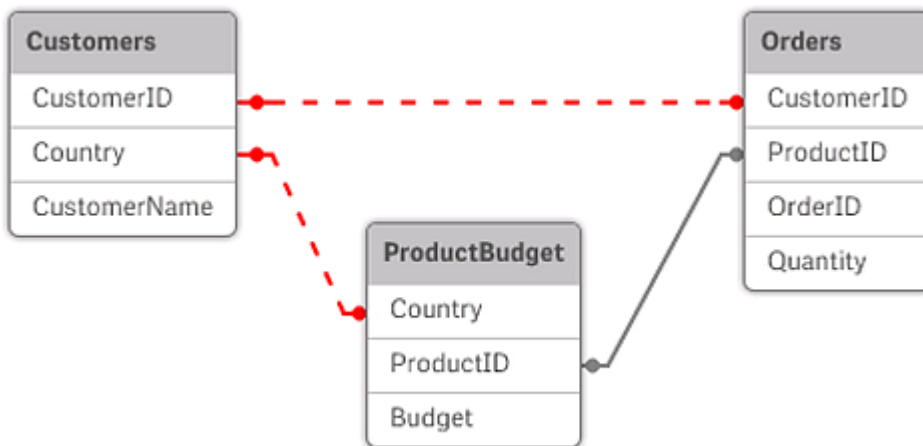
循環参照を理解し解決

データ構造に循環参照 (「ループ」) が存在する場合、テーブルは2つの項目間に複数の関連付けパスが存在するように関連付けられます。

このようなデータ構造は、データの解釈にあいまいさが生じるため、できる限り避ける必要があります。

2つの項目間の関連付けのパスが2つ以上存在するため、循環参照を含む3つのテーブル。

4 スクリプトを使用したデータのロードと変換



Qlik Sense は、疎結合テーブルを使用してループを壊すことにより、循環参照の問題を解決します。Qlik Sense がロードスクリプトの実行中に循環データ構造を見つけると、警告ダイアログが表示され、1つ以上のテーブルが疎結合として設定されます。通常、Qlik Sense は、ループ内の最も長いテーブルを疎結合しようとします。多くの場合、トランザクションテーブルが最も長いため、通常はこのテーブルが疎結合されます。データモデルビューアでは、疎結合テーブルは、他のテーブルと赤い点線でリンクされています。

次の情報を含む 3 つのテーブルからデータがロードされます。

- 国代表サッカーチームの名前
- 地方のサッカークラブ
- ヨーロッパ諸国の都市

Excel のソースデータテーブルの外観

NationalTeams	
Country	Team
Germany	Die Mannschaft
Italy	Azzurri
Spain	La Roja

Clubs	
City	Team
Barcelona	Barcelona
Hamburg	Altona
Madrid	Real Madrid
Milano	Milan
Munich	Bayern München
Rome	Lazio
Turin	Juventus

Cities	
Country	City
Germany	Hamburg
Germany	Munich
Italy	Milano
Italy	Rome
Italy	Turin
Spain	Barcelona
Spain	Madrid

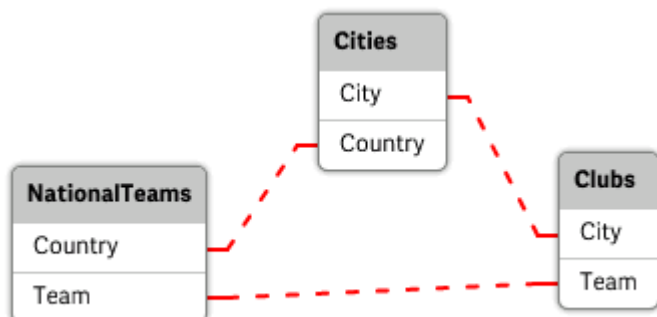
このデータ構造は、項目名 **Team** が国代表チームと地方クラブの 2 つの異なる目的で使用されているため、あまりよくありません。テーブル内のデータで非現実的なロジック状況が発生します。

テーブルを Qlik Sense にロードする際、Qlik Sense はどのデータ接続が最も重要でないかを判定し、このテーブルを疎結合にします。

4 スクリプトを使用したデータのロードと変換

[データモデルビュー] を選択して、Qlik Sense がデータ接続の関連性をどのように解釈しているかを確認します。

循環参照を赤い点線で示した状態




これで、所在する都市と国が含まれているテーブルが、異なる国の代表チームのテーブルと異なる都市の地方クラブのテーブルに対して疎結合になります。

循環参照の解決

循環参照が発生した場合は、同一の名前を持つ項目のいずれかに一意の名前を割り当てて、データロードスクリプトを編集する必要があります。

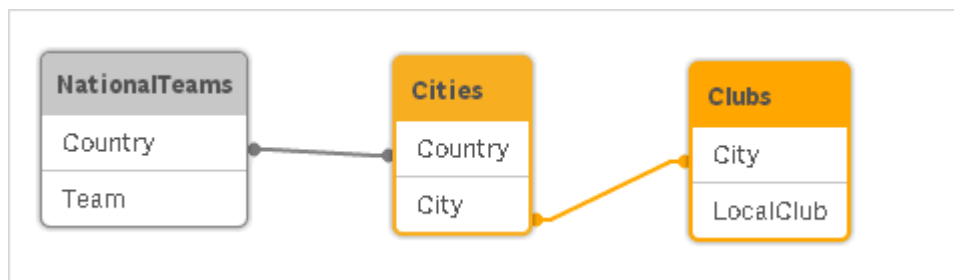
次の手順を実行します。

1. データロードエディターを開きます。
2. 名前が重複する項目のどちらか一方で **LOAD** ステートメントを編集します。
この例では、地方チームとその都市が含まれているテーブルの **LOAD** ステートメントに、**Team** の新しい名前が含まれています (例: **LocalClub**)。更新された **LOAD** ステートメントは以下のようになります。
`LOAD City, Team as LocalClub`
3. ツールバーの  をクリックして、アプリにデータをリロードします。

これですべてのテーブルで機能するロジックが完成しました。この例では、**Germany** が選択された場合、ドイツの代表チームとドイツの都市、各都市の地元クラブが関連付けられます。

Country	Team	City	LocalClub
Germany ✓	Die	Hamburg	Altona
Italy	Azzurri	Munich	Barcelona
Spain	LaRoja	Barcelona	
		Madrid	

[データモデルビュー] を開くと、疎結合された結びつきが通常の結びつきに置き換えられていることがわかります。



テーブルの連結

連結は、2つのテーブルを1つにまとめる操作です。

2つのテーブルは単純に足し合わされます。つまり、データは変更されず、作成されるテーブルには元の2つのテーブルの合計と同じ数のレコードが含まれます。複数の連結操作を連続で実行して、複数のテーブルを連結したテーブルを作成できます。

自動連結

ロードされた複数のテーブルの項目名と項目数が完全に一致する場合、Qlik Sense はさまざまなステートメントのコンテンツを1つのテーブルに連結します。

```
LOAD a, b, c from table1.csv;
```

```
LOAD a, c, b from table2.csv;
```

その結果の内部テーブルには、項目 **a**、**b**、および **c** が含まれています。レコード件数は、テーブル 1 とテーブル 2 の合計レコード数になります。



項目の数と名前は、完全に同じでなければなりません。2つのステートメントの順序は任意に指定できます。

強制連結

複数のテーブルにまったく同じ項目セットが存在しない場合でも、Qlik Sense は2つのテーブルを強制的に連結できます。強制連結は、スクリプトで **concatenate** プレフィックスを使用すると実行できます。このプレフィックスは、特定のテーブルを別の名前のテーブルまたは最後に作成されたテーブルに連結します。

```
LOAD a, b, c from table1.csv;
```

```
concatenate LOAD a, c from table2.csv;
```

その結果の内部テーブルには、項目 **a**、**b**、および **c** が含まれています。結果として生成されるテーブルのレコード件数は、テーブル 1 とテーブル 2 の合計レコード数になります。テーブル 2 から取得したレコードの項目 **b** の値は、NULL です。



先にロードされたテーブルの名前が **concatenate** ステートメントで指定されている場合を除き、**concatenate** プレフィックスは最後に作成されたテーブルを使用します。そのため、2つのステートメントの順序を任意に指定することはできません。

連結の防止

ロードされた複数のテーブルの項目名と項目数が完全に一致する場合、Qlik Sense はさまざまなステートメントのコンテンツを1つのテーブルに連結します。**noconcatenate** ステートメントを使用すると、この動作を防止することができます。関連付けられた **LOAD** または **SELECT** ステートメントでロードされたテーブルは、既存のテーブルに連結されません。

```
LOAD a, b, c from table1.csv;
noconcatenate LOAD a, b, c from table2.csv;
```

事前にロードされているテーブルからのデータのロード

すでにロードされているテーブルからデータをロードし変換するには、2つの方法があります。

- **Resident LOAD** では、**Resident** 述語を以後の **LOAD** ステートメントで使用し、新規テーブルをロードします。
- 先行するロード-先行する **LOAD** または **SELECT** ステートメントから、ソースを指定せずにロードします。

Resident と先行する LOAD の違い

大半の場合、いずれの方法を使用しても結果は同じです。先行する **LOAD** は一般に高速なオプションですが、代わりに **Resident LOAD** を使用する必要がある場合もあります。

- **Order_by** 節を使用して、**LOAD** ステートメントを処理する前にレコードをソートしたい場合。
- 先行する **LOAD** がサポートされていない以下のプレフィックスのいずれかを使用したい場合。
 - **Crosstable**
 - **Join**
 - **Intervalmatch**

Resident LOAD

LOAD ステートメントで **Resident** 述語を使用すると、以前ロードしたテーブルからデータをロードできます。日付や数値の処理といった Qlik Sense 関数を使用するオプションがない場合でも、**SELECT** ステートメントを使用してロードされたデータで計算を行う場合に役立ちます。

この例では、最初の **Crosstable LOAD** では日付の変換を実行できないため、**Resident** ロードで変換を行います。

```
PreBudget:
Crosstable (Month, Amount, 1)
LOAD Account,
    Jan,
```

4 スクリプトを使用したデータのロードと変換

```
Feb,  
Mar,  
...  
From Budget;  
  
Budget:  
Noconcatenate  
LOAD  
    Account,  
    Month(Date#(Month, 'MMM')) as Month,  
    Amount  
Resident PreBudget;  
  
Drop Table PreBudget;
```



Resident を使用する一般的なケースは、計算やフィルターで一時的なテーブルを使う場合です。目的を達成した後は、**Drop table** ステートメントを使用して一時的なテーブルをドロップする必要があります。

先行するLOAD

先行するロード機能を使用すると、1つのパスでテーブルをロードし、複数の連続した変換を定義できます。基本的に、これは以下の **LOAD** または **SELECT** ステートメントからロードする **LOAD** ステートメントのことで、通常指定する **From** や **Resident** などのソース修飾子を指定せずに実行します。この方法では、**LOAD** ステートメントをいくつでもスタックできます。ステートメントの評価は、最下部のステートメントから順番に、上に向かって実行されます。

Resident を使用しても同じ結果を得られますが、ほとんどの場合、先行する **LOAD** の方が時間がかかりません。

先行する **Load** のもう1つの利点は、計算を1か所に維持して、その上に配置された **LOAD** ステートメントで再利用できるという点にあります。

Example 1: SELECT ステートメントでロードされたデータの変換

SELECT ステートメントを使用してデータベースからデータをロードすると、**Qlik Sense** 関数を使用して **SELECT** ステートメントでデータを解釈することができなくなります。この問題を解決するには、データの変換を行う場所、**SELECT** ステートメントの上に **LOAD** ステートメントを追加します。

この例では、前の **SELECT** ステートメントをソースとして使用し、**LOAD** ステートメントで **Qlik Sense** 関数 **Date#** を使用する文字列として、保存された日付を解釈します。

```
LOAD Date#(OrderDate, 'YYYYMMDD') as OrderDate;  
SQL SELECT OrderDate FROM ... ;
```

Example 2: 計算の再利用によるスクリプトの簡素化

この例では、スクリプトで計算を複数回使用しています。

```
LOAD Age(FromDate + IterNo() - 1, BirthDate) as Age,  
    Date(FromDate + IterNo() - 1) as ReferenceDate  
Resident Policies  
    While IterNo() <= ToDate - FromDate + 1 ;
```

4 スクリプトを使用したデータのロードと変換

最初のパスで計算を導入すると、先行する **LOAD** で **Age** 関数を再利用できます。

```
LOAD ReferenceDate,
    Age(ReferenceDate, BirthDate ) as Age;
LOAD *,
    Date(FromDate + IterNo() - 1 ) as ReferenceDate
Resident Policies
    While IterNo() <= ToDate - FromDate + 1 ;
```

先行する **LOAD** の制限

- **Join**、**Crosstable**、**Intervalmatch** のプレフィックスは、先行する **LOAD** とは併用できません。
- **distinct** を使用して一意のレコードをロードしている場合、**distinct** はロード先のテーブルにのみ反映されるので、**Load** ステートメントの先頭に **distinct** を配置する必要があります。

Qlik Sense のデータ型

Qlik Sense では、テキスト文字列、数字、日付、時刻、タイムスタンプ、通貨を正しく処理し、さまざまな書式でソートおよび表示したり、計算に使用することができます。つまり、日付と時刻、タイムスタンプを相互に加算/減算できることを意味します。

Qlik Sense 内のデータ表現

Qlik Sense でのデータの解釈と数値書式を理解するには、プログラムの内部でデータがどのように保存されているかを知る必要があります。Qlik Sense にロードされるすべてのデータは、文字列または数値の 2 種類の方法で表現できます。

1. 文字列表現は常に使用でき、リストボックスやその他のシートオブジェクトに表示されます。リストボックス内のデータの書式設定 (数値書式) は、文字列表現にのみ影響を与えます。
2. 数値表現は、データが有効な数値として解釈可能な場合に限り使用できます。数値表現は、すべての数値計算と数値のソートで使用できます。

1 つの項目に読み取られた複数のデータ項目の数値表現が同じ場合、それらは同じ値として処理され、検出された最初の文字列表現を共有します。例: 数値 1.0、1、および 1.000 がこの順序で読み取られた場合、これらの数値の数値表現はすべて 1 となり、最初の 1.0 が文字列表現として使用されます。

データ型変換

数字、通貨、日付などを含むデータをロードした場合、データの種類が定義されているかどうかに応じてデータ型が変換されます。このセクションでは、2 つの事例におけるデータ型の変換方法について解説します。

型情報を持つデータ

ODBC を使用してロードされたデータベースで、データ型が定義されている数値を含む項目は、それぞれの書式に従って Qlik Sense によって処理されます。それらの文字列表現は、適切な書式が適用された数値になります。

Qlik Sense は、数値書式がプロパティパネルの **[数値書式]** でメジャー用に変更された場合でも、項目の元の数値書式を記憶します。

各種データ型のデフォルトの書式は次のとおりです。

4 スクリプトを使用したデータのロードと変換

- 整数、浮動小数点数: 数値のデフォルト設定
- 通貨: 通貨のデフォルト設定
- 時刻、日付、タイムスタンプ: ISO 標準書式

数値と通貨のデフォルト設定は、スクリプトデータ型変換変数またはオペレーティングシステム設定 ([コントロールパネル]) で定義されます。

型情報を持たないデータ

ソースに特定の書式情報がないデータ(テキストファイルのデータや一般的な書式の ODBC データなど) では、状況が複雑になります。最終的な結果は、少なくとも次の 6 つの要因で決まります。

- ソースデータベースにデータが書き込まれた方法
- 数値、時刻、日付などに対するオペレーティングシステムの設定 ([コントロールパネル])。
- スクリプトでのオプションのデータ型変換変数の使用
- スクリプトでのオプションの変換関数の使用
- スクリプトでのオプションのフォーマット関数の使用
- ドキュメント内の数値書式設定 コントロール

Qlik Sense は、入力データを数値や日付、時刻として変換しようとします。データでシステムのデフォルト設定が使用されている場合は、変換と表示書式設定は Qlik Sense によって自動的に行われるため、自分でスクリプトや Qlik Sense の設定を変更する必要はありません。

デフォルトでは、完全な一致が見つかるまで、次の手順で書式が検索されます。(デフォルトの書式とは、オペレーティングシステム ([コントロールパネル]) またはスクリプト内のデータ型変換変数で指定された小数点記号、年、月、日の順序などの書式です。)

Qlik Sense は、データを次のように解釈します。

1. 数値のデフォルト書式に従った数値。
2. 日付のデフォルト書式に従った日付。
3. 時刻と日付のデフォルト書式に従ったタイムスタンプ。
4. 時刻のデフォルト書式に従った時刻。
5. の書式に従った日付。yyyy-MM-dd。
6. の書式に従ったタイムスタンプ。YYYY-MM-DD hh:mm[:ss[:fff]]。
7. の書式に従った時刻。hh:mm[:ss[:fff]]。
8. 通貨のデフォルト書式に従った通貨。
9. 小数点記号と桁区切り記号がどちらも「,」に設定されていない場合は、「.」を小数点記号、「,」を桁区切り記号とする数値。
10. 小数点記号と桁区切り記号がどちらも「,」に設定されていない場合は、「.」を小数点記号、「,」を桁区切り記号とする数値。
11. テキスト文字列。この最後の条件が成り立たないことはありません。データを読み取ることができた場合は、常に文字列として解釈することができます。

4 スクリプトを使用したデータのロードと変換

テキストファイルから数値をロードすると、変換の問題が発生する場合があります。例えば、不適切な桁区切り記号や小数点記号によって、Qlik Sense が数値を間違えて解釈する可能性があります。そこで最初に、スクリプト内のデータ型変換変数が正しく定義されているかどうか、[コントロールパネル]のシステム設定が正しいかどうかを確認する必要があります。

Qlik Sense で日付または時刻として解釈されたデータは、ビジュアライゼーションのプロパティパネルで日付または時刻の別の書式に変更可能です。

データに書式が定義されていないため、別のレコードには、同じ項目に別の書式のデータが含まれる場合があります。つまり、1つの項目に有効な日付、整数、およびテキストが入っている可能性もあります。その場合、データは書式設定されず、元の形式で表示されます。

日付と時刻の変換

Qlik Sense では、データ内の日付、時刻、タイムスタンプが日付シリアル値として保存されます。日付シリアル値は、日付と時刻、タイムスタンプのほか、日付と時刻のエンティティに基づく算術計算で使用されます。これにより、日付や時刻を加算/減算したり、時間間隔を比較することが可能になります。

日付シリアル値は、1899年12月30日からの経過日数(実数値)です。Qlik Senseの書式は、1900年3月1日から2100年2月28日までの範囲については、Microsoft Excelなどのプログラムで使用される1900日付システムと同じです。例えば、33857は1992年9月10日に相当します。この範囲外では、Qlik Senseはグレゴリオ暦に拡張した同じ日付システムを使用します。



この項目に1980年1月1日より前の日付が含まれている場合、この項目に **\$date** または **\$timestamp** システムタグが含まれることはありません。この項目は、Qlik Sense では引き続き日付項目として認識されますが、これらのタグが必要な場合は、データロードスクリプトで **Tag** ステートメントを使用して手動で追加できます。

時刻シリアル値は、0から1の間の数になります。シリアル値 0.00000 は 00:00:00 に相当し、0.99999 は 23:59:59 に相当します。これらを混合することで、日付と時刻を示すことができます。例えばシリアル値 2.5 は、1900年1月1日の午後 12:00 を表します。

ただし、データは文字列の書式に従って表示されます。デフォルトでは、[コントロールパネル]の設定が使用されます。また、スクリプトでデータ型変換変数を使用したり、書式設定関数を使用して、データの書式を設定することもできます。さらに、シートオブジェクトのプロパティシートでデータの書式を再設定できます。

Example 1:

- 1997-08-06 の保存形式 35648
- 09:00 の保存形式 0.375
- 1997-08-06 09:00 の保存形式 35648.375

およびその他の回避法:

- 35648 の 'D/M/YY' 数値書式による表示 6/8/97
- 0.375 の 'hh.mm' 数値書式による表示 09.00

Qlik Sense は一連の規則に従って日付や時刻、その他のデータ型を解釈しようとします。ただし、最終的な結果は、既に説明したように多くの要素の影響を受けます。

4 スクリプトを使用したデータのロードと変換

Example 2:

このセクションの例は、次のデフォルト設定を前提としています。

- 小数点記号:
- 短い日付書式: YY-MM-DD
- 時間書式: hh:mm

以下のテーブルは、スクリプト内で特別なデータ型変換関数を使用することなく、Qlik Sense にデータを読み込んだ場合のさまざまな表示形式を示しています。

スクリプト内で特別なデータ型変換関数を使用することなく、データが読み込まれるときのテーブル

ソースデータ	Qlik Sense のデフォルト解釈	'YYYY-MM-DD' 日付書式	'MM/DD/YYYY' 日付書式	'hh:mm' 時間書式	'###0.00' 数値書式
0.375	0.375	1899-12-30	12/30/1899	09:00	0.38
33857	33857	1992-09-10	09/10/1992	00:00	33 857.00
97-08-06	97-08-06	1997-08-06	08/06/1997	00:00	35 648.00
970806	970806	4557-12-21	12/21/4557	00:00	970 806.00
8/6/97	8/6/97	8/6/97	8/6/97	8/6/97	8/6/97

以下のテーブルは、スクリプト内で `date#(A, 'M/D/YY')` 変換関数を使用して、Qlik Sense にデータを読み込んだ場合のさまざまな表示形式を示しています。

スクリプト内で `date#(A, 'M/D/YY')` 変換関数を使用するときのテーブル

ソースデータ	Qlik Sense のデフォルト解釈	'YYYY-MM-DD' 日付書式	'MM/DD/YYYY' 日付書式	'hh:mm' 時間書式	'###0.00' 数値書式
0.375	0.375	0.375	0.375	0.375	0.375
33857	33857	33857	33857	33857	33857
97-08-06	97-08-06	97-08-06	97-08-06	97-08-06	97-08-06
970806	970806	970806	970806	970806	970806
8/6/97	8/6/97	1997-08-06	08/06/1997	00:00	35 648.00

ドル記号展開

ドル記号展開はスクリプトや数式で使用されるテキスト置換の定義です。このプロセスは、たとえ新しいテキストが短くても展開と呼ばれます。置換はスクリプトステートメントや数式が評価される直前に行われます。技術的に言えば、これはマクロ展開です。

展開は、常に「\$(」から始まり、「)」で終わります。例:\$(variablename)。角括弧で囲まれた内容が、テキストの置換方法を定義します。



ドル記号展開は、計算できる展開の数に制限があります。ネストされた展開が1,000レベルを超える展開は計算されません。

変数を使用したドル記号展開

ロードスクリプトまたはチャートの数式で、ドル記号展開の変数を使用して次のことを行います。

- テキストを参照
- 数値を参照

テキスト変数

スクリプトや数式でのテキスト置換に変数を使用する場合は、次の構文が使用されます。

`$(variablename)`

`$(variablename)` により、変数内の値に展開されます。`variablename` が存在しない場合は、空の文字列に展開されます。

例: テキスト変数 ロードスクリプト

例 1: ロードスクリプト

ロードスクリプト

次のデータをインラインロードとしてデータロードエディタにロードします。

```
Set x = 'red';           // Assign the value "red" to variable x Set y = 'blue';           // Assign the value
"blue" to variable y Set z = '$(x) $(y)'; // Expands x and y, returns "red blue" in
variable z // Expand x and y, return "red green blue" in variable MyString Let
MyString='$(x)('&' green '&'$(y)'; // Create table MyTable, load variable values for x, y, z
into fields X, Y, Z // Concatenate with variable MyString into field NewString MyTable: Load
'$(x)' as X, '$(y)' as Y, '$(z)' as Z, '$(MyString)' as NewString autogenerate 1;
```

説明

この例では、次のことを示しています。

- 変数割り当てで変数を展開する方法。
- テキスト操作と組み合わせて変数を展開する方法。

これは、変数の内容と静的文字列を組み合わせた動的ラベルと一般的なテキスト文字列を作成するための便利な設定です。

アウトプット

Qlik Sense で次のテーブルを作成します:

テーブル - ロードスクリプトからの出力

X	Y	Z	NewString
red	blue	red blue	red blue green

4 スクリプトを使用したデータのロードと変換

例 2: ロードスクリプト

ロードスクリプト

次のデータをインラインロードとしてデータロードエディタにロードします。

```
Set vFunction = 'upper'; // Assign the string "upper" to variable vFunction
Set vField = 'String'; // Assign the string "String" to variable vField
Let vEvaluate = '$(vFunction)' & ('&' & $(vField) & '&'); // The variable vEvaluate returns the value "upper(string)"
MyTable: // Create table called MyTable
Load *, $(vEvaluate) as Upper; // vEvaluate expanded as a dynamic expression
Load *, '$(vEvaluate)' as Expression; // vEvaluate expanded as string
Load * inline [ ID, String 1, abc 2, def 3, ghi 4, jkl ];
```

説明

Set ステートメントとLet ステートメントは、ロードスクリプトの変数に値を割り当てるために使用されます。2つの違いは、Set ステートメントが変数に文字列を割り当てるのに対し、Let ステートメントは結果の値を変数に割り当てる前に文字列の内容を評価することです。この例のload インラインテーブルには、変数vEvaluateのさまざまな評価をテキスト文字列と対応する数式の両方として視覚化するために使用される2つの先行するload ステートメントが追加されています。

アウトプット

Qlik Sense で次のテーブルを作成します:

テーブル - ロードスクリプトからの出力

ID	String	Expression	Upper
1	abc	upper(String)	ABC
2	DEF	upper(String)	DEF
3	ghi	upper(String)	HIJ
4	jkl	upper(String)	JKL

例: テキスト変数のチャートの数式

例: ロードスクリプト

ロードスクリプト

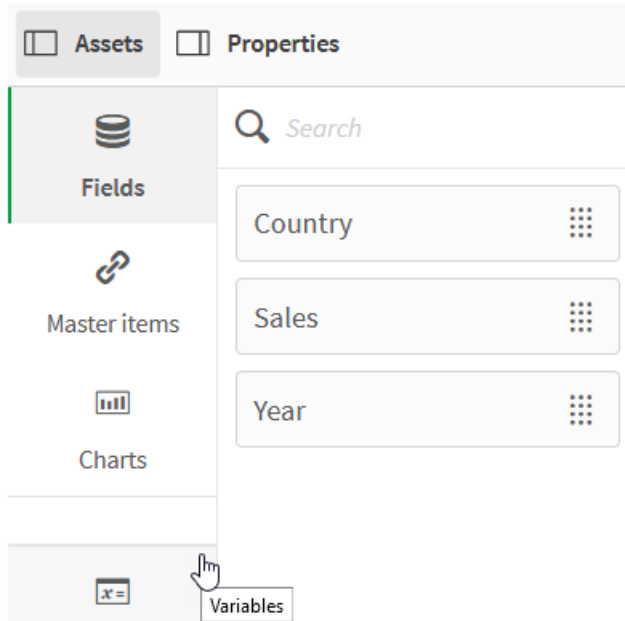
次のデータをインラインロードとしてデータロードエディタにロードします。

```
// Create table SalesByCountry
SalesByCountry: 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 ];
```

変数

編集モードのシートで、[アセット] パネルから[変数] ダイアログを開きます。

変数を開くダイアログ



次の変数を作成します。

名前	作成する変数 定義
vSales	Sum(Sales)
vSales2014	Sum({<Year={2014}>}Sales)
vSales2015	Sum({<Year={2015}>} Sales)
vSalesAllYears	\$(vSales2014) +\$(vSales2015)
vSalesDifference	\$(vSales2015)/\$(vSales2014) - 1

変数

4 スクリプトを使用したデータのロードと変換

Variables ×

Create new Delete

	Name ▼	Definition	Value	Tags ▼	
<input type="checkbox"/>	vSales	Sum(Sales)	Sum(Sales)		...
<input type="checkbox"/>	vSales2014	Sum({<Year={2014}>}S...	Sum({<Year={...		...
<input type="checkbox"/>	vSales2015	Sum({<Year={2015}>} ...	Sum({<Year={...		...
<input type="checkbox"/>	vSalesAllYears	\$(vSales2014) +\$(vSal...	\$(vSales2014...		...
<input type="checkbox"/>	vSalesDifference	\$(vSales2015)/\$(vSale...	\$(vSales2015...		...

Showing: 1-5 of 5. |< < > >|

Done

KPI チャートを作成して、展開を確認します。

ドル記号展開を使用する KPI

4 スクリプトを使用したデータのロードと変換

The screenshot shows a Qlik Sense dashboard with a grid of KPI cards. The top card displays '\$(vSales)' with a value of '1.65M'. Below it are two cards for '\$(vSales2014)' (856.8k) and '\$(vSales2015)' (788.6k). The bottom left card shows '\$(vSalesAllYears)' (1.65M). The bottom right card, '\$(vSalesDifference)', shows '-8%' and is highlighted with a green border. A configuration panel on the right shows the 'Measures' section with 'First KPI' set to '\$(vSalesDifference)'. The 'Expression' and 'Label' fields both contain '\$(vSalesDifference)'. The 'Number formatting' is set to 'Number', and the 'Formatting' is set to 'Simple' with a '12%' dropdown.

数値の変数展開

数値の変数展開では、次の構文が使用されます。

(#variablename)

展開は、常に変数の数値を有効な少数で表現します。指数表記が使用される場合もあります(非常に大きい数字または小さい数字の場合)。variablenameが存在しないか、数値が含まれていない場合、これは NULLではなく0に展開されます。

例: 数値変数 ロードスクリプト

例 1: ロードスクリプト

ロードスクリプト

次のデータをインラインロードとしてデータロードエディタにロードします。

```
Set DecimalSep = ','; // Set decimal comma as separator for this example. Let X = 7/2;
// Assign the expression 7/2 to variable X. MyTable: // Create an inline table labeled
"MyTable" Load 1 as ID, * inline [ DecimalComma DecimalPoint $(X) $(#X) ] (delimiter is
'\t');
```

4 スクリプトを使用したデータのロードと変換

説明

#vvariable 展開は、常に変数の数値を有効な少数で表現します。これは、ポイントではなくコンマが小数点記号として使用され、コンマ区切りのリストと競合するリスクがある場合に役立ちます。

ロードインラインテーブルでこれらの変数を展開する主な理由は、\$(x) の追加の引用符が必要ないためです。

アウトプット

Qlik Sense で次のテーブルを作成します:

テーブル - ロードスクリプトからの出力

DecimalComma	DecimalPoint
3,5	3.5

例 2: ロードスクリプト

ロードスクリプト

次のデータをインラインロードとしてデータロードエディタにロードします。

```
// The three Set statements below are required to mimic and initialize // Format specifiers
that are relevant to this particular example Set ThousandSep=' '; // Set example thousand
separator Set DecimalSep=','; // Set example decimal separator Set TimestampFormat='YYYY-MM-
DD hh:mm:ss'; // Set example date format Let vRaw = today()-1/1440; // Timestamp minus
one minute Let vFormat = timestamp$(#vRaw)); // Formatted as timestamp // Create MyTable as
an inline table to expand variables as field values MyTable: Load
DecimalComma DecimalPoint FormattedNumber $(vRaw) $(#vRaw) $(vFormat) ] (delimiter is '\t');
```

説明

#vvariable 展開は、常に変数の数値を有効な少数で表現します。これは、ポイントではなくコンマが小数点記号として使用され、コンマ区切りのリストと競合するリスクがある場合に役立ちます。変数が正しい小数点記号なしで展開されると、小数部分が切り捨てられるため、数値の精度が影響を受けることに注意することも重要です。

ロードインラインテーブルでこれらの変数を展開する主な理由は、\$(x) の追加の引用符が必要ないためです。

アウトプット

Qlik Sense で次のテーブルを作成します:

テーブル - ロードスクリプトからの出力

DecimalComma	DecimalPoint	FormattedNumber
44 470,00	44469.999305556	2021-09-18 23:59:00

4 スクリプトを使用したデータのロードと変換

例 3: ロードスクリプト

ロードスクリプト

次のデータをインラインロードとしてデータロードエディタにロードします。

```
// The three Set statements below are required to mimic and initialize // format specifiers
that are relevant to this particular example Set ThousandSep=' '; // Set example thousand
separator Set DecimalSep=','; // Set example decimal separator Set TimestampFormat='YYYY-MM-
DD hh:mm:ss'; // Set example date format // Assign a numerical value and a valid format
specifier to vStart Let vStart = timestamp#('2021-03-23 12:34:56','$(TimestampFormat)'); //
Calculate timestamp (vStart + 3 hours) with valid decimal separator: "." Let vStop =
timestamp(#vStart)+1/8,'YYYY-MM-DD hh:mm:ss'); // Create MyTable as an inline table to
expand variables as field values MyTable: Load * inline [ StartTime StopTime $(vStart)
$(vStop) ] (delimiter is '\t'); // This is a tab delimited inline table // Tab delimited
tables are useful for avoiding conflicting list separators
```

説明

#variable 展開は、常に変数の数値を有効な少数で表現します。これは、ポイントではなくコンマが小数点記号として使用され、コンマ区切りのリストと競合するリスクがある場合に役立ちます。変数が正しい小数点記号なしで展開されると、小数部分が切り捨てられるため、数値の精度が影響を受けることに注意することも重要です。

ロードインラインテーブルでこれらの変数を展開する主な理由は、\$(x) の追加の引用符が必要ないためです。

アウトプット

Qlik Sense で次のテーブルを作成します:

テーブル - ロードスクリプトからの出力

StartTime	StopTime	
2021-03-23 12:34:56	2021-03-23 15:34:56	

並列ステートを参照する変数の展開

変数には 1 つの値のみが含まれ、この値がすべての代替ステートで使用されます。変数を展開しても、この値は変わりません。変数が作成された場所やオブジェクトのステートは、この値に影響しません。

変数が計算による変数の場合、つまり定義が等号で始まる場合、変数の定義で代替ステートを指定しない限り、計算は既定のステートで実行されます。

例えば、名前が MyState のステート、vmyvar の名前の変数がある場合:

```
vmyvar: =only({MyState}MyField)
```

変数の定義で代替ステート名への参照が明示されているので、変数のコンテンツを評価するステートがこの参照により決定されます。

パラメータを使用したドル記号展開

ドル記号展開ではパラメータを使用できます。これにより、変数の動的評価が可能になります。

4 スクリプトを使用したデータのロードと変換

拡張で使用するために作成された変数には、仮パラメータが含まれている必要があります。正式なパラメータは、評価への入力のプレースホルダーであり、ドル記号で記述されます: $\$1$ 、 $\$2$ 、 $\$3$ 。番号はパラメータ番号を示します。

変数を使用する場合、実際のパラメータはコンマ区切りのリストに記載する必要があります。

例: ドル記号展開のパラメータ

2つの数値間の乗算を定義する場合は、次のように記述できます。

```
Set MUL= $1*$2 ;
```

これは、 $\$1$ と $\$2$ を互いに乗算する必要があることを定義します。この変数を使用する場合は、数式に $\$1$ と $\$2$ の値を挿入してドル展開する必要があります。

```
Set x= $( MUL(3,7) ) ;
```

値 (3 と 7) は、展開で $\$1$ と $\$2$ を置き換える実際のパラメータです。展開は、**set** ステートメントが解析および実行される前に行われます。つまり、スクリプトパーサーは次のことを認識します。

```
Set x= 3*7 ;
```

その結果、変数 x には文字列値: $3*7$ が割り当てられます。

Set ステートメントの代わりに Let を使用する場合:

```
Let x= $( MUL(3,7) ) ;
```

パーサーには次のように表示されます。

```
Let x= 3*7 ;
```

ここで評価が行われ、 x には数値 21 が割り当てられます。

パラメータの数

仮パラメータの数を実パラメータの数より多い場合は、実パラメータに対応する仮パラメータだけが展開されます。実パラメータの数か仮パラメータの数より多い場合は、余分な実パラメータが無視されます。

例: 仮パラメータと実パラメータ

```
Set MUL= '$1*$2' ;
```

```
Set x= $(MUL) ; // returns $1*$2 in x
```

```
Set x= $(MUL(10)) ; // returns 10*$2 in x
```

```
Let x= $(MUL(5,7,8)) ; // returns 35 in x
```

$\$0$ パラメータ

パラメータ $\$0$ は、呼び出しで実際に渡されるパラメータの数を返します。

例: エラー処理を導入する方法

```
Set MUL= If($0=2, $1*$2, 'Error') ;
```

数式を使用したドル記号展開

ドル記号展開では数式を使用できます。

角括弧で囲まれた内容は、等号で始まる必要があります。

```
$(=expression)
```

4 スクリプトを使用したデータのロードと変換

数式が評価され、値が展開に使用されます。例:

```
// returns a string with the current year $(=Year(Today()));
```

```
// returns the year before the selected one $(=Only(Year)-1);
```

例: チャートの数式でメジャーを参照してメジャーを選択します

例: ロードスクリプト

ロードスクリプト

次のデータをインラインロードとしてデータロードエディタにロードします。

```
// Load string "=MinString(Measure)" into variable vSelectMeasure Let vSelectMeasure =
' =MinString(Measure)'; MyTable1: // Create table and load values for Dim and Sales Load *
inline [ Dim, Sales A, 150 A, 200 B, 240 B, 230 C, 410 C, 330 ];
MyTable2: // Create table and load aggregations as field values for Measure Load * Inline [
Measure avg(Sales) sum(Sales) count(distinct Dim)];
```

説明

このスクリプトを使用すると、チャートの数式でメジャーを参照してメジャーを選択できます。チャートの数式には変数展開 `$(=MinString(Measure))` が含まれています。`MinString()` は数式内の文字列値を検索し、アルファベット順にソートされた最初のテキスト値 (この場合は `avg(Sales)`) を返します。これにより、オブジェクトのプロパティにアクセスして操作することなく、ビジュアライゼーションで使用する数式をインタラクティブに決定 (および選択) することができます。

アウトプット

以下を使用して、Qlik Sense で次のテーブルを作成します。

- 軸: Dim
- 数式:
 - `'$(vSelectMeasure)'`
 - `$(=MinString(Measure))`
 - `Avg(Sales)`

テーブル - 数式を使用したドル記号展開

Dim	<code>=\$(vSelectMeasure)'</code>	<code>\$(=MinString(Measure))</code>	<code>Avg(Sales)</code>
合計	-	260	260
ライン番号の隣にある	<code>avg(Sales)</code>	175	175
B	<code>avg(Sales)</code>	235	235
C	<code>avg(Sales)</code>	370	370

4 スクリプトを使用したデータのロードと変換

例: 2つの軸間のクォータを計算します

例: ロードスクリプト

ロードスクリプト

次のデータをインラインロードとしてデータロードエディタにロードします。

```
Let vDivision = '=MinString(Numerator) / MinString(Denominator)'; Load recno() as Numerator  
autogenerate 100; Load recno() as Denominator autogenerate 100;
```

説明

このスクリプトは、それぞれ 100 個の値を含む項目のペアを生成します。これらの項目は、2つの間のクォータの計算で分子と分母として機能します。チャートメジャーには変数展開 `='${(vDivision)}` が含まれているため、計算するクォータをインタラクティブに決定 (および選択) できます。

アウトプット

以下を使用して、Qlik Sense で次のテーブルを作成します。

- 軸:
 - Numerator
 - Denominator
- メジャー: `='${(vDivision)}`

結果のクォータ(合計)は、項目で選択した値の影響を受けます。Numerator (= 5) と Denominator (= 3) であり、いずれかの項目で新しい選択が行われるたびに再計算されます。

テーブル - 数式としてのドル記号展開

分子	分母	<code>='\${(vDivision)}</code>
合計	-	1,6666666666667
5	3	1,6666666666667

ファイル インクルード

ファイル インクルードは、ドル記号展開を使用して作成されます。構文は次のとおりです。

```
$(include=filename)
```

上記のテキストは、等号記号の後に指定されたファイルのコンテンツで置き換えられます。この機能は、テキストファイルのスクリプト全体、またはその一部を保存する場合に便利です。

ドル展開と代替ステート

通常、ドル展開では代替ステートは考慮されません。ただし、ドル展開に数式がある場合は、ドル展開が行われた対象に関連する状態で評価されます。

例:

```
$(=Sum(Amount))
```

この計算は、選択に基づく合計値をオブジェクトのステートで返します。

スクリプトでの引用符の使用

さまざまな方法により、スクリプトステートメントで引用符を使用することができます。

LOAD ステートメントの内側での使用

LOAD ステートメントでは、引用符として次の記号を使用する必要があります。

項目名

説明	シンボル	コードポイント	例
二重引用符	""	34	"string"
角括弧	[]	91, 93	[string]
アクセント	``	96	`string`

文字列リテラル

説明	シンボル	コードポイント	例
単一引用符	''	39	'文字列'

SELECT ステートメントの場合

ODBC ドライバにより解釈される **SELECT** ステートメントの場合、使用はさまざまです。通常、項目名やテーブル名にはストレート二重引用符 (**Alt + 0034**) を使用し、リテラルにはストレート単一引用符 (**Alt + 0039**) を使用して、アクセントの使用は避けるようにします。ただし、一部の ODBC ドライバでは、アクセントを引用符として使用できるだけでなく、むしろ優先するものもあります。このような場合、生成される **SELECT** ステートメントには、アクセントの引用符が含まれます。

Microsoft Access での使用例

Microsoft Access ODBC Driver 3.4 (Microsoft Access 7.0 に付属) では、**SELECT** ステートメントを分析する際に以下の引用符を使用できます。

項目名およびテーブル名:

- []
- ""
- ``

文字列リテラル:

- ''

データベースによっては、異なる規則が適用される場合があります。

LOAD ステートメントの外側での使用

LOAD ステートメントの外側で、**Qlik Sense** が数式を予測する場所では、二重引用符は項目参照ではなく変数参照を意味します。二重引用符を使用すると、囲まれた文字列は変数として解釈され、変数の値が使用されます。

コンテキスト外の項目参照およびテーブル参照

一部のスクリプト関数は、作成済みの項目あるいは **LOAD** ステートメントのアウトプットに含まれる項目を参照します (**Exists()**や**Peek()**など)。**LOAD** ステートメントの入力テーブルのような、コンテキスト内の項目を参照するソース項目参照に対して、これらの項目参照はコンテキスト外項目参照と呼ばれます。

コンテキスト外項目参照およびテーブル参照は、リテラルと見なす必要があるため、単一引用符が必要となります。

名前とリテラルの違い

次の数式を比較すると、名前とリテラルの違いがよりはっきりします。

'Sweden' as Country

この数式が **LOAD** ステートメントまたは **SELECT** ステートメントの項目リストの一部として使用された場合、**Qlik Sense** の項目 "Country" には、テキスト文字列 *Sweden* が項目値としてロードされます。

"land" as country

この数式が **LOAD** ステートメントまたは **SELECT** ステートメントの項目リストの一部として使用された場合、**Qlik Sense** の項目 "Country" には、"*land*" という名前のデータベース項目またはテーブル列のコンテンツが項目値としてロードされます。つまり、*land* は項目参照として扱われます。

数値と文字列リテラルの違い

次の数式を比較すると、数値と文字列リテラルの違いがよりはっきりします。

'12/31/96'

この文字列が数式の一部として使用された場合、まずテキスト文字列 "12/31/96" として解釈され、次に日付形式が 'MM/DD/YY' の場合には日付として解釈されます。この場合、数値表現とテキスト表現の両方を持つデュアル値として保存されます。

12/31/96

この文字列が数式の一部として使用された場合、 $12 \div 31 \div 96$ と数値的に解釈されます。

文字列における引用符の使用

引用符として使用可能な文字が文字列に含まれる場合、その文字列を引用するときは文字列の先頭と末尾を明確に示すことが重要です。文字列が正しく引用されていない場合、スクリプトがエラーになったり、データが正しくロードされなかったりします。

引用符を含む文字列を引用するには2つの方法があります。

文字列の引用に特定の引用符を使用

文字列内で使用されていない引用符を選択し、その引用符を使用して文字列全体を引用します。Qlik Senseは特定の引用符を使用して文字列の先頭と末尾を決定します。

以下のいずれかの引用符を文字列全体の引用に使用することができます。

- 二重引用符 ""
- 角括弧 []
- アクサングラフ ``
- 単一引用符 ''

[テーブル '1 "2"]

文字列の引用に角括弧が使用されています。文字列は次のようにロードされます。テーブル '1 "2"

'文字列 `名前 1` "名前 2'

文字列の引用に単一引用符が使用されています。文字列は「文字列 `名前 1` "名前 2」のようにロードされます。

エスケープ文字の使用

エスケープ文字は、文字列の引用に使用する引用符の追加インスタンスです。エスケープ文字は、文字列内に登場する引用符の各インスタンスとは別に追加する必要があります。すべての引用符が文字列内で使用されている場合、その文字列の引用で使用されているものと同じ種類の引用符の他にエスケープ文字を追加する必要があります。文字列内ですでに使用されている引用符を使用したい場合にもエスケープ文字を使用できません。

次の記号のみエスケープ文字として使用できます。

- 二重引用符 ""
- 角括弧 []
- 単一引用符 ''

"マイケルは ""美しい日だ"" と言った。"

二重引用符 "" を使用している文字列を引用する場合、文字列内で使用されているそれぞれの二重引用符とは別に二重引用符を追加する必要があります。

4 スクリプトを使用したデータのロードと変換

この文字列は「マイケルは "美しい日だ" と言った。」としてロードされます。エスケープ文字 "" を使用することで、Qlik Sense データロードエディターは、文字列の一部を表す二重引用符と文字列の末尾を表す引用符を理解します。短縮形 *It's* で使用される単一引用符 ' は、文字列の引用に使用する記号ではないため、エスケープ文字は必要ありません。

```
'マイケルは"美しい日だ"と言った.'
```

単一引用符を使用している文字列を引用する場合、文字列内で使用されているそれぞれの単一引用符とは別に単一引用符を追加する必要があります。

この文字列は「マイケルは "美しい日だ" と言った。」としてロードされます。「マイケルは ~ と言った」の内容の引用に使用される二重引用符 " は、文字列の引用に使用する記号ではないため、エスケープ文字は必要ありません。

```
[マイケルは ["美しい日だ"]と言った。]
```

角括弧 [] は他の 2 種類の引用符とは動作が異なります。括弧をエスケープ文字として使用する場合、右角括弧] にのみ角括弧を追加する必要があり、左角括弧 [には追加する必要はありません。

この文字列は「マイケルは ["美しい日だ"] と言った。」としてロードされます。右角括弧] のみエスケープされています。文字列内で使用されている一重引用符 ' および二重引用符 " は、文字列の引用で使用されていない場合はエスケープする必要はありません。

データ内のワイルドカード

データ内のワイルドカードを使用することができます。ワイルドカードには、この項目のすべての値として解釈されるスターマークと、この項目の残りのすべての値として解釈されるオプションシンボルの 2 種類があります。

スター マーク

スターマークは、この項目内のすべての値 (リストされている値)、つまりこのテーブル内の任意の場所にリストされている値として解釈されます。スクリプトのアクセスセクションでロードされたテーブル内のシステム項目 (*USERID*, *PASSWORD*, *NTNAME*, *SERIAL*) の 1 つで使用される場合は、この項目のすべての可能な値 (リストされていない値も含む) として解釈されます。

明示的に指定しない限り、スターマークは使用されません。

OtherSymbol

テーブル内のその他すべての値、つまりロードされたテーブルで明示的に検出されなかったすべての値を表す必要が生じることがあります。これは、**OtherSymbol** と呼ばれる特殊な変数を使用して表します。「その他のすべての値」として処理される **OtherSymbol** を定義するには、次の構文を使用します。

```
SET OTHERSYMBOL=<sym>;
```

この構文を **LOAD** または **SELECT** ステートメントの前に置きます。<sym> には任意の文字列を指定できません。

4 スクリプトを使用したデータのロードと変換

定義されたシンボルが内部テーブル内に出現すると、Qlik Sense はそのシンボルが検出された項目にそれまでロードされていないすべての値として、そのシンボルを定義します。**OtherSymbol** の出現後に項目で検出された値は無視されます。

この機能をリセットするには、次の構文を使用します。

```
SET OTHERSYMBOL=;
```

Table Customers

CustomerID	Name
1	ABC Inc.
2	XYZ Inc.
3	ACME INC
+	Undefined

Table Orders

CustomerID	OrderID
1	1234
3	1243
5	1248
7	1299

スクリプト内で、上記の最初のテーブルがロードされる前の場所に、次のステートメントを挿入します。

```
SET OTHERSYMBOL=+;
```

1、2、3 以外の *CustomerID* への参照 (例えば、*OrderID 1299* をクリックした場合) は、*Name* が *Undefined* になります。



OtherSymbol は、テーブル間の外部結合の作成で使用することを意図したものではありません。

NULL 値の処理

データベース クエリまたはテーブル間の結合の結果、特定の項目にデータが生成されない場合、通常、結果は NULL 値になります。

概要

Qlik Sense ロジックは、以下を実際の NULL 値として処理します。

- ODBC接続から返されたNULL値
- データロードスクリプトのテーブルの強制連結の結果として作成されたNULL値

4 スクリプトを使用したデータのロードと変換

- データロードスクリプトで行われた結合の結果として作成された **NULL** 値
- テーブルに表示される項目値の組み合わせを生成した結果として作成された **NULL** 値



NullAsValue ステートメントが使用されている場合を除き、これらの **NULL** 値を関連付けや選択に使用することはできません。

定義用のテキストファイルに **NULL** 値を含めることはできません。

ODBC からの **NULL** 値の関連付け/選択

ODBC データソースから **NULL** 値を関連付けたり、選択することができます。この目的のために、スクリプト変数が定義されています。次の構文を使用できます。

```
SET NULLDISPLAY=<sym>;
```

データ入力の最下位レベルにある ODBC データソースのすべての **NULL** 値の代わりにシンボル **<sym>** が使用されます。**<sym>** には任意の文字列を指定できます。

この機能をデフォルト解釈にリセットするには、次の構文を使用します。

```
SET NULLDISPLAY=;
```



NULLDISPLAY の使用は、ODBC データソースのデータにのみ影響します。

ODBC 接続から返された **NULL** 値を空の文字列として Qlik Sense ロジックに解釈させるには、**SELECT** ステートメントの前に次の構文を追加します。

```
SET NULLDISPLAY=";
```



" は、間に何も挟まない 2 つの単一引用符です。

テキストファイルからの **NULL** 値の作成

テキストファイルまたは **inline** 節に存在する場合に、実 **NULL** 値として解釈されるシンボルを定義することができます。次のステートメントを使用します。

```
SET NULLINTERPRET=<sym>;
```

シンボル **<sym>** は、**NULL** として解釈されます。**<sym>** には任意の文字列を指定できます。

この機能をデフォルト解釈にリセットするには、次の構文を使用します。

```
SET NULLINTERPRET=;
```



NULLINTERPRET の使用は、テキストファイルおよび **inline** 節のデータにのみ影響します。

数式における **NULL** 値の伝達

NULL 値は、いくつかの論理的および妥当な規則に基づいて、数式を介して伝達されます。

4 スクリプトを使用したデータのロードと変換

関数

一般的な規則としては、関数が定義されている範囲からパラメータが外れると、関数は **NULL** を返します。

例

数式	結果
asin(2)	の戻り値: NULL
log(-5)	の戻り値: NULL
round(A,0)	の戻り値: NULL

上記の結果として、評価に必要なパラメータのいずれかが **NULL** の場合、関数は原則として **NULL** を返します。

例

数式	結果
sin(NULL)	の戻り値: NULL
chr(NULL)	の戻り値: NULL
if(NULL, A, B)	の戻り値: B
if(True, NULL, A)	の戻り値: NULL
if(True, A, NULL)	の戻り値: A

2番目の規則の例外として、型をテストするための論理関数があります。

例

数式	結果
isnull(NULL)	の戻り値: True (-1)
isnum(NULL)	の戻り値: False (0)

算術および文字列演算子

これらの演算子のいずれかの側に **NULL** が発生した場合は、**NULL** が返されます。

例

数式	結果
A + NULL	の戻り値: NULL
A - NULL	の戻り値: NULL
A / NULL	の戻り値: NULL
A * NULL	の戻り値: NULL
NULL / A	の戻り値: NULL

4 スクリプトを使用したデータのロードと変換

数式	結果
0 / NULL	の戻り値: NULL
0 * NULL	の戻り値: NULL
A & NULL	returns A

関係演算子

関係演算子のいずれかの側に **NULL** が発生した場合は、特別な規則が適用されます。

例

数式	結果
NULL (任意の関係演算子) NULL	の戻り値: NULL
A <> NULL	の戻り値: True (-1)
A < NULL	の戻り値: False (0)
A <= NULL	の戻り値: False (0)
A = NULL	の戻り値: False (0)
A >= NULL	の戻り値: False (0)
A > NULL	の戻り値: False (0)

4.4 データおよび項目のガイドライン

Qlik Sense には、把握しておかなければならない規則や制限がいくつかあります。例えば、データテーブルや項目の数には上限があり、**Qlik Sense** にロードできるデータの量も限られています。

ロードされるデータ量のガイドライン

Qlik Sense にロード可能なデータ量は、主にコンピュータのメインメモリの容量によります。

データテーブルと項目の上限

非常に大きなアプリを作成する際は、**Qlik Sense** アプリの 1 つの項目内に許可されている固有の値数は、最大 **2,147,483,648** であることに注意してください。

ロード可能な項目やデータテーブル、テーブルのセルおよび列の数は **RAM** のみによって制限されます。

ロードスクリプトセクションの推奨制限

ロードスクリプトセクションあたりの文字数は、**50,000** 文字以内に収めることをお勧めします。

数値や日付書式の説明

多くの変換関数および書式設定関数では、書式コードを使用して数値や日付の書式を設定できます。このトピックでは、数値、日付、時刻、タイムスタンプ関数の形式について説明します。これらの形式は、スクリプト関数とチャート関数の両方に該当します。

4 スクリプトを使用したデータのロードと変換

数値書式

数値の桁数を指定するには、各桁に記号 "0" を使用します。

小数点の左側で可能な桁を示すには、"#" 記号を使用します。

桁区切りや小数点の位置に記号を入れるには、適用可能な桁区切り記号と小数点記号を使用します。

書式コードは、区切り記号の位置を定義します。書式コードで区切り記号を設定することはできません。スクリプトで **DecimalSep** と **ThousandSep** の変数を使用して、設定します。

桁区切り記号を使用して、任意の位置で桁をグループ化できます。例えば、書式文字列 "0000-0000-0000" (桁区切り記号は "-") を使用して、12 桁の部品番号を "0012-4567-8912" のように表示できます。

数値書式の例

数値書式	説明
###0	区切り記号付きの 3 桁の整数として数値を表示します。この例では、" " は千の位をわかりやすくするために使用されています。
###0	3 桁区切りの記号がない整数で数値を表示します。
0000	4 桁以上の整数として数値を表示します。例えば、数値 123 は、0123 と表示されます。
0.000	小数点以下 3 桁で数値を表示します。この例では、「.」は小数点の記号として使用されています。

特殊な数値書式

Qlik Sense では、2 進数、8 進数、16 進数を含む 2 ~ 36 の基数を解釈し、書式設定できます。また、ローマ数字も処理可能です。

特殊な数値書式

形式	説明
2 進形式	2 進数の書式コードを表すには、(bin) または (BIN) で開始します。
8 進形式	8 進数の書式コードを表すには、(oct) または (OCT) で開始します。
16 進形式	16 進数の書式コードを表すには、(hex) または (HEX) で開始します。大文字の A-F は、形式表示に使用されます (14FA など)。小文字の場合は、形式表示に a-f が使用されます (14fa など)。書式コードが大文字でも小文字でも正しく変換されます。
10 進形式	10 進形式であることを示すために (dec) または (DEC) を使用することはできますが、必要ではありません。

4 スクリプトを使用したデータのロードと変換

形式	説明
カスタム基数形式	2～36の基数を示すには、書式コードを(rx)または(Rxx)で開始します。ここで、 xx は、使用する基数を示す2桁の数値です。大文字の R を使用すると、基数が10より大きい場合にQlik Senseの形式設定で大文字が使用されます(14FAなど)。小文字の r の場合は、形式設定に小文字が使用されます(14faなど)。書式コードが大文字でも小文字でも正しく変換されます。(r02)は(bin)と同じで、(R16)は(HEX)と同じです。
ローマ数字形式	ローマ数字の書式コードを表すには、(rom)または(ROM)で開始します。大文字を使用する場合、大文字が形式表示に使用されます(MMXVIなど)。小文字の場合は、書式に小文字が使用されます(mmxvi)。書式コードが大文字でも小文字でも正しく変換されます。ローマ数字の場合、負の数値はマイナス記号、ゼロは0で示されます。小数点は無視されます。

特殊な数値書式の例

例	結果
num(199, '(bin)')	の戻り値: 11000111
num(199, '(oct)')	の戻り値: 307
num(199, '(hex)')	の戻り値: c7
num(199, '(HEX)')	の戻り値: C7
num(199, '(r02)')	の戻り値: 11000111
num(199, '(r16)')	の戻り値: c7
num(199, '(R16)')	の戻り値: C7
num(199, '(R36)')	の戻り値: 5J
num(199, '(rom)')	の戻り値: cxcix
num(199, '(ROM)')	の戻り値: CXCIX

日付

日付形式には以下の記号が使用可能です。任意の区切り記号を使用できます。

日付形式に使用する記号

記号	説明
D	日を表示するには、各桁に記号"D"を使用します。

4 スクリプトを使用したデータのロードと変換

記号	説明
M	<p>月の数字表示には、記号 "M" を使用します。</p> <p>1桁表示には "M" を、2桁表示には "MM" を使用します。</p> <p>"MMM" は、オペレーティング システム、またはスクリプトのシステム優先変数 MonthNames で定義されている月の短い方の名前を文字列で表示します。</p> <p>"MMMM" は、オペレーティング システム、またはスクリプトのシステム優先変数 LongMonthNames で定義されている月の長い方の名前を文字列で表示します。</p>
Y	<p>年を表示するには、各桁に記号 "Y" を使用します。</p>
W	<p>曜日を表示するには、記号 "W" を使用します。</p> <p>"W" は、曜日を1桁の数字として返します (月曜日は 0 など)。</p> <p>"WW" は、2桁の数字を返します (水曜日は 02 など)。</p> <p>"WWW" は、オペレーティング システム、またはスクリプトのシステム優先変数 DayNames で定義されている曜日の短いバージョン (Mon など) を示します。</p> <p>"WWWW" は、オペレーティング システム、またはスクリプトのシステム優先変数 LongDayNames で定義されている曜日の長いバージョン (Monday) を示します。</p>

(例として 2013 年 3 月 31 日を使用します)

日付形式の例

例	結果
YY-MM-DD	13-03-31 と表示。
YYYY-MM-DD	3/31/13 と表示。
YYYY-MMM-DD	2013-Mar-31 と表示。
DD MMMM YYYY	31 March 2013 と表示。
M/D/YY	3/31/13 と表示。
W YY-MM-DD	6 13-03-31 と表示。
WWW YY-MM-DD	Sat 13-03-31 と表示。
WWWW YY-MM-DD	Saturday 13-03-31 と表示。

時刻

時刻形式には以下の記号を使用できます。任意の区切り記号を使用できます。

4 スクリプトを使用したデータのロードと変換

時刻形式に使用する記号

記号	説明
h	時を表示するには、各桁に記号 "h" を使用します。
m	分を表示するには、各桁に記号 "m" を使用します。
s	秒を表示するには、各桁に記号 "s" を使用します。
f	秒の小数部を表示するには、各桁に記号 "f" を使用します。
tt	時刻をAM/PM 書式で表示するには、時刻の後に記号 "tt" を使用します。

(例として 18.30 を使用します)

時刻形式の例

例	結果
hh:mm	18:30 と表示。
hh.mm.ss.ff	18.30.00.00 と表示。
hh:mm:tt	06:30:pm と表示。

タイムスタンプ

上記の日付と時刻の表記の法則は、タイムスタンプにも当てはまります。

(例として 2013 年 3 月 31 日 18:30 を使用します)

タイムスタンプ形式の例

例	結果
YY-MM-DD hh:mm	13-03-31 18:30 と表示。
M/D/Y hh.mm.ss.ffff	3/31/13 18.30.00.0000 と表示。

4.5 QVD ファイルでの作業

QVD (QlikView Data) ファイルは、Qlik Sense からエクスポートされたデータのテーブルを含むファイルです。QVD はネイティブの Qlik 形式で、Qlik Sense または QlikView でしか書き込みと読み出しを行えません。ファイル形式は、スクリプトからデータを迅速に読み取れるように最適化されており、サイズもコンパクトです。QVD ファイルからのデータの読み取りは、他のデータソースから読み取る場合よりも、一般に 10 ~ 100 倍速くなっています。

QVD ファイルは、標準 (高速) と最適化 (超高速) の 2 つのモードで読み取ることができます。使用されるモードは、スクリプトエンジンによって自動的に決定されます。

最適化されたロードに関して、いくつかの制限があります。項目の名前は変更できますが、ここで言及されている操作のいずれかによって、最適化されているロードが無効になり、標準のロードとなります。

4 スクリプトを使用したデータのロードと変換

- ロードされた項目に対する変換。
- **where** 節を使用すると、**Qlik Sense** でレコードが解凍されます。
- ロードされた項目でのマップの使用。

QVD ファイルの使用目的

QVD ファイルは多数の目的に使用することができ、どのような状況でも複数の目的を適用できます。少なくとも主に4つの目的で使用されます。

- ロード速度の向上
QVD ファイルの入力データの変更されない部分または変化が遅い部分をバッファリングすることで、大きなデータセットに対するスクリプトの実行が大幅に高速化します。
- データベース サーバーの負荷の減少
外部データソースから取得するデータ量を大幅に削減できます。これにより、外部データベースおよびネットワークトラフィックの負荷が減少します。さらに複数のスクリプトが同じデータを共有する場合は、ソースデータベースからQVDファイルに一度データをロードするだけで済みます。他のアプリケーションでも、このQVDファイルを利用することで同じデータを活用できます。
- 複数アプリケーションからのデータ統合
binary スクリプトステートメントでは、単一のアプリケーションからのみ別のアプリケーションにデータをロードできます。しかし、QVDファイルを使用すれば、スクリプトは任意の数のアプリケーションからデータを統合できます。これにより、さまざまな部署の類似データを統合するなど、アプリケーションの可能性が広がります。
- 増分
多くの場合、QVDの機能を活用することで、増大するデータベースから新しいレコードだけをロードする増分ロードを簡単に実行できます。

QVD ファイルの作成

QVD ファイルは2つの方法で作成できます。

- スクリプトで **store** コマンドを使用した明示的な作成と命名。スクリプトで、以前に読み取ったテーブルまたはその一部を、選択した場所にある明示的に指定されたファイルにエクスポートするように記述します。
- スクリプトからの自動作成とメンテナンス。**LOAD** または **SELECT** ステートメントの前に **buffer** プレフィックスを付けると、**Qlik Sense** は自動的に QVD ファイルを作成します。この QVD ファイルは一定の条件下で、データのリロード時に元のデータソースの代わりに使用できます。

読み込み時間など、生成された QVD ファイルに違いはありません。

QVD ファイルからのデータのロード

次の方法で、QVDファイルの読み込みやアクセスを実行します。

- 明示的なデータソースとしての QVD ファイルのロード。QVDファイルは、その他のテキストファイル (**csv**、**fix**、**dif**、**biff**など) と同様にスクリプトの**LOAD**ステートメントで参照できます。
例:

4 スクリプトを使用したデータのロードと変換

- `LOAD * from xyz.qvd (qvd)`
- `LOAD Name, RegNo from xyz.qvd (qvd)`
- `LOAD Name as a, RegNo as b from xyz.qvd (qvd)`
- バッファ済みの QVD ファイルの自動ロード。LOAD または SELECT ステートメントで **buffer** プレフィックスを使用する場合、読み取り用の明示的なステートメントは必要ありません。Qlik Sense は元の LOAD ステートメントまたは SELECT ステートメントを使用してデータを取得するのではなく、QVD ファイルからどの範囲のデータを使用するかを決定します。
- スクリプトからの QVD ファイルへのアクセス。(qvd で始まる) いくつかのスクリプト関数を使用して、QVD ファイルの XML ヘッダーにあるデータのさまざまな情報を取得できます。

QVD 形式

QVD ファイルは、厳密に 1 つのデータテーブルを保持し、次の 3 つの部分で構成されます。

- ヘッダー



QVD ファイルが QlikView で作成された場合、ヘッダーはテーブル内の項目や、後続情報およびその他のメタデータのレイアウトを適格に記述する XML ヘッダー (UTF-8 文字セット) です。

- バイト埋め込み形式のシンボル テーブル。
- ビット埋め込み形式の実際のテーブル データ。

4.6 における分析接続の構成 Qlik Sense Desktop

分析接続を使用すると、外部分析をビジネスディスカバリと統合することができます。分析接続は、外部計算エンジンの呼び出し (このとき外部計算エンジンはサーバーサイド拡張 (SSE) として動作) によって、ロードスクリプトおよびチャートで使用可能な数式を拡張します。たとえば、R への分析接続を作成し、データのロード時に統計数式を使用できます。

Qlik Sense Desktop の場合、*Settings.ini* ファイルで構成を実行する必要があります。

次の手順を実行します。

1. *Settings.ini* ファイルを開きます。
Qlik Sense Desktop の場合、`C:/Users/<ユーザー ID>/Documents/Qlik/Sense/` または `C:/Users/AppData/Local/Programs/Qlik/Sense/Engine` にあります。
Qlik Sense の場合、次の場所にあります。`C:/ProgramData/Qlik/Sense/Engine/`
2. 次の構成を追加します (最後に空の行を挿入します)。
[Settings 7]
SSEPlugin=<PluginConfig>[;<PluginConfig>...]

ここで、<PluginConfig> は次を含む構成要素のコンマ区切りのリストです。

```
<EngineName>,<Address>[,<PathToCertFile>,<RequestTimeout>,<ReconnectTimeout>]
```



新しい接続を追加するか、または既存の接続を変更したら、変更内容を有効にするために **Qlik Sense Desktop** を再起動する必要があります。



サーバー側の拡張 (**SSE**) プラグインサーバーが稼働している状態で **Qlik Sense** を起動する必要があります。そうしない場合、接続は確立されません。

Qlik オープン ソース SSE リポジトリ

次の2つの Qlik SSE リポジトリはオープンソースです。

- <https://github.com/qlik-oss/server-side-extension>
SSE プロトコル、一般的なマニュアル、Python および C++ で書かれた例が含まれます。
- <https://github.com/qlik-oss/sse-r-plugin>
C# で書かれた R プラグイン (ソースコードのみ) が含まれます。使用できる状態にするには、プラグインを作成する必要があります。

要素の説明

<EngineName>: プラグイン関数 (例えば、Python プラグインの場合の **SSEPython**) により、アプリの数式内から使用されるプラグインへのマッピング/エイリアス。

<Address>: 2つの要素から構成されるコロン区切りのリスト。

- <Host>: プラグインの DNS 名 (または IP アドレス)。
- <Port>: プラグインの待ち受けポート (通常は 50051)。

<PathToCertFile>: プラグインとのセキュアな通信に必要なクライアント証明書を含むフォルダーへのファイル システムパス。オプション。省略されている場合は、セキュアでない通信が呼び出されます。このパスは、証明書が配置されているフォルダーを指し示しているだけです。証明書が実際にこのフォルダーにコピーされていることを確認する必要があります。3つの証明書の名前は、**root_cert.pem**、**sse_client_cert.pem**、**sse_client_key.pem** とする必要があります。許可されているのは相互認証のみです (サーバーとクライアント間の認証)。

<RequestTimeout>: 整数値 (秒)。オプション。既定値は 0 (無期限) です。メッセージ経過時間のタイムアウト。

<ReconnectTimeout>: 整数値 (秒)。オプション。既定値は 20 (秒) です。クライアントが、プラグインへの接続が失われてから再接続を試みるまでの時間。

- 1つの SSE プラグインサーバーの定義例: `SSEPlugin=SSEPython,localhost:50051`
- 2つの SSE プラグインサーバーの定義
例: `SSEPlugin=SSEPython,localhost:50051;R,localhost:50053`
- 証明書のパスは存在しないものの、タイムアウトは設定されている1つの SSE プラグインサーバーの定義
例: `SSEPlugin=SSEPython,localhost:50051,,0,20`

4 Section Access によるデータセキュリティの管理

Section Access は、アプリケーションのセキュリティを制御するために使用されます。これは基本的にデータロードスクリプトの一部であり、セキュリティテーブルを追加して、誰が何を表示するかを定義します。**Qlik Sense** はこの情報を使用して、ユーザーがアプリケーションを開いたときにデータを適切なスコープに縮小します。つまり、アプリ内のデータの一部は、ユーザーの ID に基づいてユーザーから非表示になります。**Section Access** はアプリ内のデータと緊密に統合されており、アクセスを制御するためにデータに依存しています。この形式の動的データ削減は、テーブルの行、列、または両方の組み合わせを対象にすることができます。

4.7 ロードスクリプトのセクション

データのアクセス制御は、データが通常ロードされるのと同じ方法でロードされる 1 つまたは複数のセキュリティテーブルを通じて管理されます。これにより、これらのテーブルを標準のデータベースまたはスプレッドシートに保存できます。セキュリティテーブルを管理するスクリプトステートメントは、スクリプトの中で **Section Access** ステートメントで開始される承認セクション内に指定します。スクリプト内で承認セクションが定義されている場合、アプリのデータをロードする部分のスクリプトを **Section Application** で開始される別のセクション内に配置する必要があります。

```
Section Access;
Load * INLINE [
    ACCESS,  USERID,          REDUCTION
    USER,    AD_DOMAIN\ADMIN, *
    USER,    AD_DOMAIN\A,     1
    USER,    AD_DOMAIN\B,     2
    USER,    AD_DOMAIN\C,     3
    ADMIN,   INTERNAL\SA_SCHEDULER,
];
Section Application;
T1:
Load *,
NUM AS REDUCTION;
LOAD
Chr(RecNo()+ord('A')-1) AS ALPHA,
RecNo() AS NUM
AUTOGENERATE 3;
```

ロードスクリプトに変更を加えた後、変更を有効にするためには、常にデータをリロードする必要があります。

セクションアクセスのシステム項目

アクセスレベルは、スクリプトの **Section Access** 部分内にロードされた 1 つ以上のセキュリティテーブルのユーザーに割り当てられます。これらのテーブルには、少なくとも 2 つのシステムフィールドが含まれている必要があります。アクセスレベルを定義する項目である **ACCESS**、および **USERID** または **USER**、**EMAIL**。使用例に応じて、オプションのシステムフィールドを追加できます。**Section Access** のすべてのシステム項目は次のとおりです。

ACCESS

対応するユーザーに与えられるアクセス権を定義します。

指定したユーザーに対して、Qlik Sense アプリへのアクセスを承認できます。セキュリティテーブルでは、アクセスレベルの **ADMIN** または **USER** をユーザーに割り当てることができます。管理者権限を持つユーザーは、セキュリティテーブルによって制限されない限り、アプリ内の全データにアクセスできます。**USER** 権限を持つユーザーは、セキュリティテーブルで指定されたデータにのみアクセスできます。有効なアクセスレベルが割り当てられていないユーザーは、アプリを開くことができません。

Section Access がリロードシナリオで使用される場合、スケジューラサービスユーザーである **INTERNAL\SA_SCHEDULER** は、リロードを実行するために **ADMIN** アクセス権を必要とします。例:

```
Section Access;
LOAD * inline [
    ACCESS, USERID
    ADMIN, INTERNAL\SA_SCHEDULER
];
```

INTERNAL\SA_SCHEDULER アカウントを使用したくない場合は、別の方法について **なりすましを使用したデータのリロード (page 169)** を参照してください。

テンプレートアプリの **On-demand アプリ生成 (ODAG)** シナリオで **Section Access** が使用されている場合、**INTERNAL\SA_API** ユーザーは、**Section Access** テーブルに **ADMIN** として含まれる必要があります。例:

```
Section Access;
LOAD * inline [
    ACCESS, USERID
    ADMIN, INTERNAL\SA_API
];
```

USERID

Qlik Sense ドメイン名とユーザー名に対応する文字列が含まれます。**Qlik Sense** はプロキシ サービスからログイン情報を取得し、その情報をこの項目の値と比較します。

ワイルドカード文字 (*) は、セキュリティテーブルで指定された追加の条件に従って、すべてのユーザーとして解釈されます。例えば、次のセキュリティテーブルでは、**Qlik Sense Tenant Admins** にいるユーザーは、リストされているすべての **REDUCTION** 値を確認できます。

```
Section Access;
LOAD * INLINE [
    ACCESS, USERID, GROUP, REDUCTION
    ADMIN, *, Qlik Sense Tenant Admins, *
    USER, QLIK-POC\SOMEOTHERUSER1, *, 1
    USER, QLIK-POC\SOMEOTHERUSER2, *, 2
    ...
];
```



USERID と **NTNAME** はどちらも同じ認証情報を使用するため、セキュリティテーブルの同じ行で両方をチェックする必要はありません。2つの項目の違いは、**NTNAME** がグループもチェックすることです。

NTNAME



NTNAME は古い **QlikView** 項目なので、**QlikView** が同じセキュリティテーブルを使用していない場合は **USERID** を使用することを推奨します。

WindowsNT ドメインのユーザー名またはグループ名に対応する文字列を含む必要がある項目。別の認証システムを使用する場合は、認証されたユーザーの名前を含める必要があります。**Qlik Sense** は OS からログイン情報を取得し、その情報をこの項目の値と比較します。

GROUP

Qlik Sense 内のグループに対応する文字列を含みます。**Qlik Sense** は、プロキシ サービスから提供されるユーザーを、このグループに基づいて解決します。

SERIAL



SERIAL はレガシー **QlikView** 項目であり、**Qlik Sense** のみを使用している場合は使用されません。

プラットフォームに対応する文字列が含まれます。項目に文字列 '**QLIKSENSE**' またはワイルドカード '*' が含まれている場合、セキュリティテーブルの他の項目に応じて、アクセスが許可される場合があります。



項目 **SERIAL** にライセンス番号が含まれている場合、セクション アクセス行はドキュメントへのアクセスを拒否します。この設定は **QlikView** でのみ有効です。

OMIT

この特定のユーザーに対して省略する項目の名前を含みます。ワイルドカードを使用することができます。空にすることもできます。



OMIT をキー項目に適用しないことをお勧めします。省略されたキー項目はデータモデルビューアーに表示されますが、コンテンツは使用できないので、ユーザーを混乱させる可能性があります。さらに、ビジュアライゼーションで使用されるキー項目に **OMIT** を適用すると、省略された項目へのアクセス権を持っていないユーザーにはビジュアライゼーションが不完全なものになります。

4.8 アプリへのユーザーアクセスの管理

最も単純な形式のセクションアクセスを使用して、特定のユーザーによるアプリへのアクセスを制限できます。ユーザーは、除外によってアプリへのアクセスを拒否されます。つまり、特定のユーザー ID がセキュリティテーブルにリストされていない場合、そのユーザーはアプリにアクセスできません。この規則の唯一の例外は、セキュリティテーブルの行の 1 つにある **USERID** 項目にワイルドカード (*) が割り当てられている場合です。この場合、ワイルドカードは、認証されたすべてのユーザーがアプリにアクセスできることを意味します。ユーザー ID のリストを含むセキュリティテーブルの例を次に示します。

```
Section Access;  
LOAD * inline [  
    ACCESS,  USERID  
    ADMIN,   AD_DOMAIN\ADMIN  
    USER,    AD_DOMAIN\A  
    USER,    AD_DOMAIN\B  
];  
Section Application;
```

4.9 アプリ内の特定のデータへのユーザーアクセスの管理

動的データ削減は、ユーザーがアプリ自体へのアクセスを許可された後、Qlik Sense アプリ内のデータテーブルの行と列へのアクセスを制限します。

行レベルのデータへのアクセスの管理

ロードスクリプトのアクセスセクションのセキュリティテーブルにデータ削減列を追加して、行レベルのデータへのアクセスを制限します。**Section Access** データを実際のデータにリンクすることにより、特定のレコード(行)をユーザーから隠すことができます。表示されるまたは除外されるデータの選択は、スクリプトの **Section Access** 部分と **Section Application** 部分に1つ以上の縮小項目に共通の名前を付けることで制御されます。ユーザーログインの後、Qlik Sense はアクセスセクションの削減項目にある選択と、同じ項目名(項目名は大文字で入力する必要があります)のアプリケーションセクションにある任意の項目を照合します。選択が行われると、Qlik Sense は選択によって除外されたすべてのデータをユーザーに表示しなくなります。データ削減列の項目値としてワイルドカード(*)が使用されている場合、ユーザーは、セキュリティテーブル内で選択されているすべての削減項目に関連付けられたレコードにアクセスできると解釈されます。



データ削減列のワイルドカード文字*は、セキュリティテーブルのすべての値のみを参照します。**Section Application** に、セキュリティテーブルの削減列には存在しない値がある場合、その値は削減されます。



アクセスセクションでは、すべての項目名と項目値がデフォルトで大文字に変換されるので、上で説明した転送で使用するすべての項目名とそれらの項目のすべての項目値を大文字にする必要があります。



既定では、Qlik Management Console タスクでスクリプトのリロードを有効にする場合は、ADMIN アクセス権を付与されている INTERNALISA_SCHEDULER アカウントユーザーが必要とされます。INTERNALISA_SCHEDULER アカウントを使用したくない場合は、別の方法についてなりすましを使用したデータのリロード (page 169) を参照してください。

ユーザーIDに基づく行レベルのデータ削減

```
Section Access;  
Authorization:  
LOAD * inline [  
    ACCESS,  USERID,          REDUCTION  
    ADMIN,   AD_DOMAIN\ADMIN, *  
];
```

4 Section Access によるデータセキュリティの管理

```
USER,    AD_DOMAIN\A,      1
USER,    AD_DOMAIN\B,      2
USER,    AD_DOMAIN\C,      *
ADMIN,    INTERNAL\SA_SCHEDULER, *
```

```
];
Section Application;
T1:
LOAD *,
NUM AS REDUCTION;
LOAD
RecNo() AS NUM
AUTOGENERATE 3;
```

この例では、現在、項目 REDUCTION (大文字) が Section Access と Section Application の両方にあります (項目値はすべて大文字)。2つの項目は通常別のものに分かれています。Section Access を使用すると、この項目はリンクし、ユーザーに表示されるレコードの数が減ります。

結果は次のようになります。

- ユーザー ADMIN は、REDUCTION = 1 または REDUCTION = 2 のときに他のユーザーに表示されるレコードとすべての項目を見ることができます。
- User A は、すべての項目と、REDUCTION=1 に関連付けられたレコードを見ることができます。
- User B は、すべての項目と、REDUCTION=2 に関連付けられたレコードを見ることができます。
- ユーザー C は、REDUCTION = 1 または REDUCTION = 2 のときに他のユーザーに表示されるレコードとすべての項目を見ることができます。

列レベルのデータへのアクセスの管理

OMIT システム フィールドを Section Access スクリプトのセキュリティテーブルに追加して、行レベルのデータへのアクセスを制限します。次の例は、行のデータ削減が既に行われている前の例に基づいています。

ユーザー ID に基づく列のデータ削減

```
Section Access;
LOAD * inline [
ACCESS,  USERID,          REDUCTION,  OMIT
ADMIN,   AD_DOMAIN\ADMIN,  *,
USER,    AD_DOMAIN\A,      1,
USER,    AD_DOMAIN\B,      2,          NUM
USER,    AD_DOMAIN\C,      3,          ALPHA
ADMIN,   INTERNAL\SA_SCHEDULER, *,
];
Section Application;
T1:
LOAD *,
NUM AS REDUCTION;
LOAD
Chr( RecNo()+ord('A')-1) AS ALPHA,
RecNo() AS NUM
AUTOGENERATE 3;
```

Section Access の項目 OMIT で、ユーザーに表示しない項目を定義します。

結果は次のようになります。

4 Section Access によるデータセキュリティの管理

- ユーザー ADMIN は、REDUCTION が 1 または 2、3 の場合にその他のユーザーに表示されるレコードとすべての項目を見ることができます。
- User A は、すべての項目と、REDUCTION=1 に関連付けられたレコードを見ることができます。
- User B は、NUM 以外のすべての項目と、REDUCTION=2 に関連付けられたレコードを見ることができます。
- User C は、ALPHA 以外のすべての項目と、REDUCTION=3 に関連付けられたレコードを見ることができます。



一部のビジュアライゼーションには、レンダリングするために満たす必要がある最小データ要件があります。その結果、列レベルの項目をユーザーのデータビューから省略すると「未完了のビジュアライゼーション」と表示される場合があります。

ユーザーグループへのアクセスの管理

Section Access には、グループメンバーシップを通じてユーザーに表示されるデータの範囲を制限するオプションがあります。ユーザーグループを使用してデータを制限するには、アクセスセクションのセキュリティテーブルに GROUP 項目名を追加し、GROUP 項目の値を定義します。

ユーザーグループに基づくデータ削減

```
Section Access;
LOAD * inline [
    ACCESS,  USERID,          GROUP,  REDUCTION,  OMIT
    USER,    *,              ADMIN,  *,
    USER,    *,              A,     1,
    USER,    *,              B,     2,          NUM
    USER,    *,              C,     3,          ALPHA
    USER,    *,              GROUP1, 3,
    ADMIN,   INTERNAL\SA_SCHEDULER, *,
];
section application;
T1:
LOAD *,
NUM AS REDUCTION;
LOAD
Chr( RecNo()+ord('A')-1) AS ALPHA,
RecNo() AS NUM
AUTOGENERATE 3;
結果は次のようになります。
```

- ADMIN グループに属するユーザーは、すべてのデータとすべての項目を見ることができます。
- A グループに属するユーザーは、すべての項目において REDUCTION=1 に関連付けられたデータを見ることができます。
- B グループに属するユーザーは、REDUCTION=2 に関連付けられたデータを見ることができますが、NUM 項目のデータは見ることができません。
- C グループに属するユーザーは、REDUCTION=3 に関連付けられたデータを見ることができますが、ALPHA 項目のデータは見ることができません。

4 Section Access によるデータセキュリティの管理

- GROUP1 グループに属するユーザーは、すべての項目において REDUCTION=3 に関連付けられたデータを見ることができます。

Qlik Sense はユーザーを UserID と比較し、そのユーザーがテーブル内のグループに含まれているか確認します。ユーザーがアクセス権のあるグループに属する場合、あるいはユーザーが一致した場合、アプリにアクセスできます。

4.10 なりすましを使用したデータのリロード

既定では、内部システム アカウント SA_SCHEDULER がリロードタスクの実行に使用されます。このアカウントには昇格された権限があり、技術的には、任意のデータソースを使用できます。ただし、QMC には、内部システム アカウントではなくアプリ所有者の権限を使用してリロードタスクを実行するための実装を使用する設定があります。この設定を構成すると、SA_SCHEDULER ではなくアプリ所有者がリロードに使用されます。つまり、Section Access テーブルに追加するのは SA_SCHEDULER ではなく、アプリ所有者になります。タスクチェーン内において、アプリは異なる所有者およびソースに対する権限を有することができますが、これは各所有者のアクセス権に依存します。

4.11 マルチクラウド環境におけるユーザーアクセスの管理

Qlik Sense マルチクラウド環境には、ユーザー認証メカニズムが混在しています。通常、Qlik Sense Enterprise on Windows を使用すると、Section Access セキュリティテーブルの USERID がプロキシサービスによって検証されます。Qlik Sense SaaS では、ID プロバイダーがその認証のロールを引き受けます。したがって、Qlik Sense Enterprise on Windows などのオンプレミス環境用に設定された Section Access は、クラウド環境では機能しません。

OIDC ID プロバイダー (Qlik IdP またはカスタム IdP) を Qlik Sense SaaS で使用する場合、[subject claim] はログイン時のユーザー識別に使用されます。Section Access では、セキュリティテーブルの USERID 項目の値が、[subject claim] の値と比較されます。テナントを設定するとき、SAM アカウント名が ID プロバイダーの [subject claim] にマッピングされていることを確認してください。したがって、例えば、SAM アカウント名が AD_DOMAIN\Dev の場合、[subject claim] を AD_DOMAIN\Dev に設定します。IdP の [subject claim] の値を確認する場合、ブラウザーのテナント URL に「/api/v1/diagnose-claims」を追加します (例: *your-tenant.us.qlikcloud.com/api/v1/diagnose-claims*)。JSON 応答では、[subject claim] は [sub] と呼ばれます。

SAM アカウント名を使用できない場合は、別のユーザー認証方法があります。メールアドレスは異なる環境でも変更されない傾向があるため、セキュリティテーブルでは、USERID の代わりに USER.EMAIL 項目を使用できます。セキュリティテーブルがどのように見えるかの例を次に示します。

ACCESS	USERID	USER.EMAIL	Comment	COUNTRY
USER	ABC\Joe	*	Access-on-prem	米国
ユーザー	*	joe.smith@example.com	Access-in-cloud	United States
USER	ABC\Ursula	*	Access-on-prem	Germany
USER	*	ursula.schultz@example.com	Access-in-cloud	Germany
USER	ABC\Stefan	*	Access-on-prem	Sweden
USER	*	stefan.svensson@example.com	Access-in-cloud	Sweden

承認スクリプト:

```
Section Access;
LOAD * INLINE [
    ACCESS,   USERID,      USER.EMAIL,      COUNTRY
    USER,     ABC\Joe,      *,               United States
    USER,     *,           joe.smith@example.com,   United States
    USER,     ABC\Ursula,  *,               Germany
    USER,     *,           ursula.schultz@example.com, Germany
    USER,     ABC\Stefan,  *,               Sweden
    USER,     *,           stefan.svensson@example.com, Sweden
];
```

各ユーザーには 2 つのレコードがあることに注意してください。1 つはオンプレミス アクセス用で、もう 1 つはクラウド アクセス用です。ワイルドカードは、関連する認証項目のみが使用されることを保証します。この例では、COUNTRY がデータ削減項目として使用されています。

4.12 セクションアクセスと Insight Advisor Chat の使用

セクションアクセスを使用するアプリを Insight Advisor Chat で利用できるようにするには、次のサービスユーザーがセクションアクセススクリプトで管理者アクセス権を持っていることを確認する必要があります。

- **INTERNAL/sa_repository:** これにより、ユーザーアクセスを制御するためのリポジトリサービスでセクションアクセススクリプトを利用できるようになります。
- **INTERNAL/sa_scheduler:** これにより、アプリは QMC タスクを使用してリロードできます。



アプリ名、項目名、またはマスターアイテム名に機密情報がある場合、セクションアクセスを使用するアプリを Insight Advisor Chat で利用できるようにすることで、これらが公開される可能性があります。クエリのアプリの提案には、ユーザーがアクセスできるストリーム内のアプリが含まれます。これらには、ユーザーがアプリのセクションアクセスでアクセスできないアプリが含まれる場合があります。ただし、これらのアプリを選択しても何も起こりません。**[軸]** または **[メジャー]** をクリックして、セクションアクセスを使用してアプリから利用可能なアイテムを表示すると、ユーザーにはアクセスできないアイテムが表示される場合があります。ただし、これらのアイテムをクリックしても、ユーザーにデータは提供されません。

例:

```
Section Access;
LOAD * inline [
    USERID ,ACCESS
    INTERNAL\s_a_repository ,ADMIN
    INTERNAL\s_a_scheduler ,ADMIN
    DOMAINNAME\user1 ,ADMIN
    DOMAINNAME\user2 ,USER
    DOMAINNAME\user3 ,USER
];
```

これらのユーザーが Section Access スクリプトに入ると、アプリを Insight Advisor Chat で利用できるようになります。アプリがリロードされると、アプリは Insight Advisor Chat で利用できるようになります。

4.13 Section Access を使用するためのガイドラインとヒント

Section Access について知っておくべきいくつかの重要な事実と役立つヒントを次に示します。

- アクセスセクション内の **LOAD** または **SELECT** ステートメントにリストする項目は、すべて大文字で記述する必要があります。データベース内の小文字を含む項目名は、**LOAD** または **SELECT** ステートメントで読み取る前に、**Upper** 関数で大文字に変換されます。
- 項目名として一覧に表示される **Section Access** システム フィールド名をデータモデルで使用できません。
- **Section Access** コントロールが適用される前に、アプリは公開されていなければなりません。アプリをリロードすると、新規または変更済み **Section Access** スクリプトは適用されません。
- スナップショットには、スナップショットを取得するユーザーのアクセス権限に従ってデータが表示され、そのスナップショットは 1 つのストーリーで共有することができます。ただし、ユーザーがアプリでライブデータを見るためにストーリーからビジュアライゼーションに戻ると、それらのスナップショットは、それ独自のアクセス権限によって制限されます。
- セクションアクセスを使用する場合、または極秘データを扱う場合は、値が色構成により公開される可能性があるため、マスター軸の値に色を割り当てないでください。
- 制限されているデータを公開しないようにするには、アプリを公開する前に、セクションアクセス設定が含まれるすべての添付ファイルを削除します。添付ファイルはアプリの公開時に含まれます。公開済みアプリがコピーされると、添付ファイルがそのコピーに含まれます。ただし、添付したデータファイルにセクションアクセス制限が適用されている場合、ファイルがコピーされるときにセクションアクセス設定が保持されません。そのため、コピーしたアプリのユーザーは、添付ファイルのすべてのデータを表示することができます。
- ワイルドカード(*) は、テーブル内の項目に含まれるすべての値として解釈されます。スクリプトのアクセスセクションでロードされたテーブル内のシステム項目 (**USERID**、**GROUP**) の 1 つで使用すると、この項目のすべての可能な値 (リストされていない値も含む) として解釈されます。
- セキュリティ項目は、さまざまなテーブルに配置できます。
- **QVD** ファイルからデータをロードする場合、**Upper** 関数はロード速度を遅くします。
- セクションアクセスを設定することで、アプリから自分自身をロックしている場合、データなしでアプリを開き、データロードスクリプトでアクセスセクションを編集することができます。そのためには、データロードスクリプトの編集およびリロードにアクセスできる必要があります。
- バイナリロードを使用すると、新しい **Qlik Sense** アプリにアクセス制限が継承されます。

5 On-demand アプリによるビッグデータの管理

On-demand アプリを使用すると、Qlik Sense Enterprise でビッグデータソースをロードし、分析することができます。ビッグデータのストア全体を一度に分析しようとするのは非常に非効率的です。それでも、代表値によるビジュアライゼーションを行うには、すべてのデータをディスカバリ可能にする必要があります。Qlik Sense On-demand アプリを使用することによりユーザーは、ビッグデータストアの集計ビューを表示し、データの関連サブセットを特定およびロードして詳細な分析を行うことができます。

On-demand アプリによって、データディスカバリが可能となる対象が拡大するため、ビジネスユーザーはいつでも大規模なデータソースで連想分析を実施できます。ユーザーはまず、知見を探索したいデータを選択し、次に、対話的に On-demand アプリを生成できます。アプリでは、Qlik の全インメモリ機能を使用してデータを分析できます。

5.1 On-demand アプリの構成要素

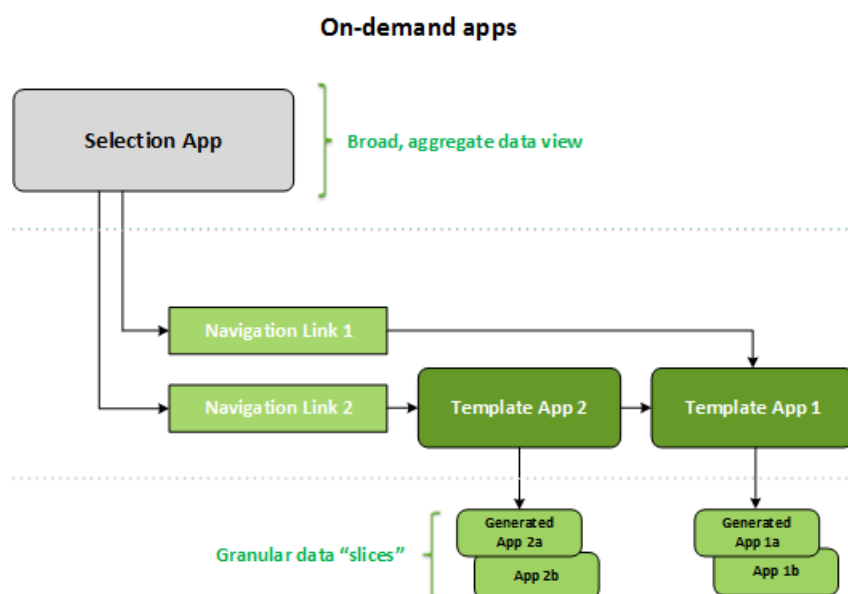
Qlik Sense では、ビッグデータの集計ビューを表示するとともに、ユーザーがズームインして細分化されたデータを分析できるようにする選択アプリにより、ビッグデータソースのロードを管理します。各選択アプリには、On-demand アプリを作成するための基盤として使用される、1つ以上のテンプレートアプリへの On-demand アプリナビゲーションリンクが埋め込まれています。ナビゲーションリンクとテンプレートアプリのプロパティにより、On-demand アプリにロードされるデータの形式と量を厳密に制御できます。

アプリはテンプレートアプリから繰り返し生成して、頻繁に変更されるデータセットを追跡できます。データは選択アプリでの選択に従ってフィルタリングされますが、On-demand アプリのコンテンツは基盤となるデータソースから動的にロードされます。同じ On-demand アプリを数回生成して、データの変更に応じて新しく分析を行うことが可能です。



On-demand アプリの生成は、On-demand アプリサービスによって制御されます。このサービスは既定で無効に設定されているため、選択を行ってテンプレートアプリをリンクし、On-demand アプリを生成するには、設定を有効にする必要があります。On-demand アプリサービスは Qlik Management Console で管理します。

On-demand アプリのコンポーネント間の関係



5.2 On-demand アプリの作成

On-demand 選択アプリとテンプレートアプリには特別なロードスクリプトが必要なため、通常は、**Qlik Sense** ロードスクリプトの作成経験のあるユーザーが作成します。On-demand 選択アプリは例えば、適度なレベルの軸の粒度でデータをロードする必要があります。On-demand テンプレートアプリには、データバインディング数式が入力されたロードスクリプトが含まれます。この数式は、データソースに対するクエリの作成に使用されます。

選択アプリは複数のテンプレートアプリにリンクでき、単一のテンプレートアプリは複数の選択アプリにリンクできます。ただし、テンプレートアプリのデータバインディング数式は、リンクされている選択アプリの項目に対応する必要があります。そのため、選択アプリとテンプレートアプリは、大抵は経験豊富な同スクリプト開発者によって併せて作成される傾向にあります。



Qlik Sense Enterprise インストール (`ProgramData\Qlik\Examples\OnDemandApp\sample`) には、サンプルの On-demand 選択とテンプレートアプリがあります。

ナビゲーションリンクを作成する場合も、テンプレートアプリに対応するバインディングを持つ選択アプリの項目を理解する必要があります。各ナビゲーションリンクには、詳細なレコードの合計数を計算する数式が必要なためです。その合計数は、選択アプリの選択状態を使ってアクセスできる集計レコードを表します。その数式を作成するには、選択アプリで利用可能な項目を使って、テンプレートアプリのレコード合計数を計算する方法を理解する必要があります。

選択アプリを使って On-demand アプリを生成する場合は、ロードスクリプトの知識は不要です。On-demand アプリのナビゲーションリンクを作成したら、そのナビゲーションリンクを選択アプリの **[アプリのナビゲーション]** バーにドラッグして、アプリナビゲーションポイントを作成できます。作成後、On-demand アプリがナビゲーションポイントから生成されます。

5 On-demand アプリによるビッグデータの管理

ナビゲーションリンクの数式による最大行の計算が必要範囲内になると、On-demand アプリの生成にナビゲーションポイントを利用できます。この時点で、ユーザーは On-demand アプリを生成できます。また、一連の別の選択を行って、それを基に追加のアプリを生成できます。

ナビゲーションリンクでは、リンクから生成できる On-demand アプリの数が制限されています。最大数のアプリが生成されると、ナビゲーションポイントからアプリを生成しているユーザーは、既存のアプリの1つを削除しないと、新しい On-demand アプリを生成できません。生成済みアプリの最大数は、On-demand アプリのナビゲーションリンクに適用されます。ナビゲーションリンクから On-demand アプリのナビゲーションポイントを1つ作成すると、そのナビゲーションポイントから最大数のアプリを作成できます。同じナビゲーションリンクから複数のナビゲーションポイントを作成した場合、これらのナビゲーションポイントはひとまとまりとして、ナビゲーションリンクに設定された最大数に制限されます。

ナビゲーションリンクによって、生成済みアプリの保持期間も設定されます。保持期間が終了すると、On-demand アプリは自動的に削除されます。

5.3 On-demand アプリの公開

ほとんどのユーザーは、公開後の On-demand アプリと選択アプリを使用します。選択アプリをストリームに公開すると、そのストリームへの適切な権限を持つユーザーはこれらのアプリを使用して集計を選択し、選択アプリに含まれるナビゲーションポイントから On-demand アプリを生成できます。すべての公開済みアプリと同じく、これらのアプリは公開後に変更することはできません。ナビゲーションポイントを追加するには、例えば、選択アプリのコピーを作成する必要があります。

多くの場合、ユーザーは生成済み On-demand アプリしか使用しません。生成済みアプリはそれぞれ個別に公開できます。実際、アプリナビゲーションリンクは、自身から生成されたアプリが特定のストリームに自動的に公開されるように指定できます。ユーザーは、アプリの公開先ストリームの生成済み On-demand アプリでロードしたデータのスライスを選択し、探索することができます。

5.4 On-Demand アプリの利点

On-Demand アプリを使用すると、ビジネスユーザーとIT部門はビッグデータ環境からさまざまな方法で値を引き出すことができます。On-Demand アプリ:

- 期間、顧客セグメント、または地区などのデータのサブセットをアプリに対話的に入力できる「ショッピングリスト」のような体験を実現します。
- メモリでホストされている潜在的なサブセットに対し、完全な Qlik Sense の機能を提供します。その一方で、大規模なデータソースも管理できる Direct Discovery は、すべての関連データをメモリ内で保持するわけではありません。Direct Discovery の場合、メジャーデータは実行されるまでソースにあります。
- IT はアプリのサイズを制御し、データ量または軸の選択に基づいてアプリを起動することができます。
- Teradata Aster、MapR、SAP BEx、PLACEHOLDER 関数 (SAP HANA) など、SQL 以外のデータソースにアクセスできます。その一方で、Direct Discovery は SQL 以外のクエリを実行できず、SQL データソースしか一緒に使用することはできません。
- カスタマイズ可能な SQL とロードスクリプトの生成を実行できます。
- どのような場合でも、セクションアクセスが可能です。

5.5 制限事項

On-Demand アプリで Qlik NPrinting を使用することはできません。

5.6 On-demand 選択 アプリの作成

On-demand 選択 アプリを使用すると、大規模なデータセットのサブセットを選択できるため、Qlik associative engine で関連付けを効率的、効果的に作成できるようになります。データ量が非常に多い環境では、選択 アプリで適度なレベルの粒度の軸のみをロードすることをお勧めします。たとえば、四半期、地域、および製品 カテゴリ別に集計された売上データがそのデータの基となっている選択 アプリでは、次のような **SQL SELECT** ステートメントを使用できます。

```
SELECT SUM(S.UNIT_COST) AS TOTAL_UNIT_COST, SUM(S.QUANTITY) AS TOTAL_QUANTITY, SUM(S.UNIT_PRICE * S.QUANTITY) AS TOTAL_SALE, SUM( (S.UNIT_PRICE - S.UNIT_COST) * QUANTITY) AS TOTAL_PROFIT, SUM(1) AS TOTAL_LINE_ITEMS, S.REGION, S.YEARQUARTER, S.PRODCAT, FROM SALE_DETAIL S GROUP BY S.REGION, S.YEARQUARTER, S.PRODCAT
```

通常、On-demand メジャー数式プロパティは、データのロードに使用される SQL GROUP BY クエリから計算された集計結果に基づいています。選択 アプリでは **GROUP BY** クエリを使用して SALE_DETAIL レコードを集計することから、メジャー項目の UNIT_COST および QUANTITY と、計算値の TOTAL_SALE および TOTAL_PROFIT で集計関数 (この場合は **SUM**) を使用する必要があります。

SUM(1) AS TOTAL_LINE_ITEMS を使用すると、地域、四半期、および製品 カテゴリのすべての異なる組み合わせごとに、販売の項目数の合計を正確に測定することができます。On-demand アプリの生成に使用するリンクの作成時には、On-demand アプリにロードされるレコードの数を制御する方法として、メジャー数式を指定する必要があります。SALE_DETAIL の例では、ユーザーが複数の製品 カテゴリ、地域、または四半期 (あるいはそのすべて) を選択したときに、TOTAL_LINE_ITEMS で合計数を計算し、選択数が On-demand アプリのレコード制限数を超過していないかどうかを判定することができます。



Qlik Sense Enterprise on Windows インストール

(ProgramData\Qlik\Examples\OnDemandApp\sample) に、サンプルの On-demand 選択 アプリがあります。

レコード制限数は、選択 アプリをテンプレート アプリにリンクしてアプリのナビゲーション リンクを作成するときに指定します。各 アプリナビゲーション リンクには、レコード制限数が指定されています。選択 アプリから複数のナビゲーション リンクを作成できます。複数の アプリナビゲーション リンクは、データのビューを複数生成するため、一般に、選択 アプリを別々のテンプレート アプリにリンクして作成します。

個々の On-demand アプリのナビゲーション リンクは、選択 アプリに含めて公開することができます。アプリのナビゲーション リンクを選択 アプリに含めると、そのリンクを使用してアプリのナビゲーション ポイントが 1 つ以上作成され、特定のシートのユーザーがそのリンクのテンプレート アプリに基づいて On-demand アプリを作成できるようになります。

5.7 On-demand テンプレート アプリの作成

On-demand テンプレート アプリは通常の Qlik Sense アプリですが、1 つ重要な違いがあります。そのロードスクリプトには、データソースに対するクエリの作成に使用されるデータバインディング数式が含まれます。それらのデータバインディング数式は、選択 アプリの選択 ステートから値を読み取り、その値をテンプレート アプリスクリプトにバインドして、生成されたアプリがユーザー制御のデータサブセットと一緒にロードされるようにするために、On-demand アプリの生成時に使用されます。

テンプレート アプリは、通常、選択 アプリと同じデータソースに接続されます。選択 アプリのロードスクリプトは通常、集計データをロードしてデータ量を減らしつつ、重要な軸やメジャーのインタラクティブなビジュアライゼーションを可能にします。テンプレート アプリのロードスクリプトは、制御されたより詳細なデータサブセットをロードするクエリを使用します。



On-Demand テンプレート アプリは、データを直接ロードしません。テンプレート アプリ接続は有効でなければならず、正しく接続されているかどうかをテストするには、**On-Demand** アプリを生成しなければなりません。**On-Demand** アプリが生成されると、**On-Demand** アプリサービスによってロードスクリプトが修正され、**On-Demand** 選択 アプリの選択 ステートがロードされます。**On-Demand** アプリがエラーなしで生成されれば、テンプレート アプリの接続が正常であるとわかります。

テンプレート アプリの構造

テンプレート アプリは、**On-Demand** アプリのナビゲーション リンクを使用して選択 アプリにリンクされます。アプリのナビゲーション リンクには、**On-demand** アプリがデータと一緒にロードされるときにクエリされるレコードの数を制御するためのプロパティが含まれます。テンプレート アプリのロードスクリプトにはデータバインディング数式が含まれています。その数式により、**On-demand** アプリにデータをロードするときに発行されるクエリを作成するために使用される、選択 アプリの項目データが指定されます。



Qlik Sense Enterprise on Windows インストール (`ProgramData\Qlik\Examples\OnDemandApp\sample`) に、サンプルの **On-Demand** テンプレート アプリがあります。



データバインディング数式の新しい構文が **2020 年 6 月** に導入されました。以前の構文とプレフィックス `od_`、`ods_`、`odo_`、`odso_`、`odb_` は、数量制約、`_n` サフィックス、フォーマット指定を含めて、以前と同じように動作します。アプリが **2020 年 6 月** より前の **Qlik Sense** バージョンで動作する場合は、古い構文を使用してください。古い構文については、[\[On-demand テンプレート アプリの作成 \(古いバージョン\)\]](#) を参照してください。

新しいプレフィックスを使用する場合、`_n` サフィックスはサポートされません。

データバインディング数式の形式は次のとおりです。

\$(odag_FIELDNAME)

od_プレフィックスは、選択アプリの選択ステートを On-demand アプリのロードスクリプトにバインドするために使用され、テンプレートアプリをコピーして作成されます。データバインディング数式で **od_**プレフィックスに続く部分は、選択アプリの項目と一致する名前にする必要があります。On-demand アプリが生成されると、選択アプリの現在の選択ステートを使用して、各項目をバインドするための適切な値が取得されます。新規作成された On-demand アプリのロードスクリプトに含まれる各 **\$(odag_FIELDNAME)** 数式は、選択アプリの選択ステートの対応する項目において選択された値のリストに置き換えられます。

より特化したデータバインディング用の他のプレフィックスが利用可能です。特殊なケースの調整とロードステートメントの最適化の詳細については、*On-demand* テンプレートアプリのバインディング数式 ([page 178](#)) を参照してください。

On-demand バインディングは、ロードスクリプトの **SELECT** ステートメントと **WHERE** ステートメントに直接挿入できます。バインディングを **WHERE** ステートメントに直接追加すると、ステートメント内の他の条件と簡単に組み合わせることができます。

ロードスクリプトを作成するときに、プレースホルダー変数 **\$(odagActive)** を追加できます。これにより、サンプルデータをテンプレートアプリにロードすることができるため、すべてのデータをロードしなくてもダイナミックビューのマスターチャートを作成できます。

次の例は、サンプルの On-demand テンプレートロードスクリプトを示しています。

サンプルデータの追加

この例では、サンプル値を追加して、バインディングが完了していてもアプリをロードできるようにします。

```
IF '$(odagActive)'='' THEN trace ODAG variables not inserted! Loading sample data.; SET 'odag_Origin Code' = ''LAX''; SET 'odag_Destination Code' = ''JFK''; SET odagn_Year = 2015; SET odag_Quarter = ''1''; SET 'odag_Ticket Carrier Code' = ''CA''; SET 'odag_Fare Class' = ''X''; END IF;
```

テンプレートアプリでのデータのロード

以下は、サンプルデータをロードし、生成された **odag_FIELDNAME** バインディングでフィルタリングするためのサンプルロードスクリプトです。**odagn_<Field Name>** バインディングは、デュアル値を選択し、既定では引用符なしで使用します。

```
SQL SELECT * FROM FlightDB.Flights WHERE "Origin Code" IN ($(odag_Origin Code)) AND "Destination Code" IN ($(odag_Destination Code)) AND "Year" IN ($(odagn_Year)) AND "Quarter" IN ($(odag_Quarter)) AND "Ticket Carrier Code" IN ($(odag_Ticket Carrier Code)) AND "Fare Class" IN ($(odag_Fare Class));
```

シングルサインオン (SSO)

On-Demand アプリでは、SSO をサポートするデータソースでシングルサインオン (SSO) を使用できます。エンジンとデータソースは、SSO を許容するように構成する必要があります。

エンジンおよびデータソースを SSO 用に構成したら、次の構文をテンプレートアプリのスクリプトに追加することによって、テンプレートアプリで SSO を有効にする必要があります。

```
///!ODAG_SSO
```

On-demand アプリサービスは、On-Demand アプリが生成されたとき、およびそのアプリがリロードされるごとに、スクリプトを解析します。

5 On-demand アプリによるビッグデータの管理

On-Demand アプリが SSO でロードされると、エンドユーザーの ID がデータソースに送信されます。エンドユーザーには、テンプレートアプリのデータ接続で使用されるソースへのアクセス権が必要です。大きなデータセットが選択された場合でも、それらのソースで、そのユーザーがアクセス権を持っているデータのみがロードされます。



シングルサインオン (SSO) を使用するテンプレートアプリから生成された On-Demand アプリは公開できません。

テンプレートアプリのノードのリロード

管理者は、テンプレートアプリに対して負荷分散ルールを設定することにより、マルチノード環境において On-demand アプリをリロードする場所を制御できます。負荷分散ルールを設定する場合、テンプレートアプリへのリンクから生成されるアプリはすべて、テンプレートアプリに適用されるルールに従ってロードされます。

On-demand テンプレートアプリのバインディング数式

テンプレートアプリにおけるデータバインディングでは、On-demand アプリへのデータロード時に発行されるクエリの作成に使用される、テンプレートに対応した選択アプリのデータが指定されます。

バインディング数式の基本的な形式 `--$(odag_FIELDNAME)--` は、数式をバインドするために使用できる唯一の方法ではありません。他のプレフィックスを使用して、選択を絞り込み、テンプレートアプリがデータを正しくロードできます。



最初から On-demand App Generation の Qlik Sense 拡張を使用して作成されたテンプレートアプリは、以下に説明する手法を使って項目から大量の選択をバインディングするよう変更する必要があります。

利用可能なバインディングプレフィックス


一般的なプレフィックス形式は `odag[s|o][n][cnt]` です。

- **s** - 選択した値のみを含む
- **o** - オプションの値のみを含む
- **n** - 既定では引用符で囲まれていない数値バージョンを選択
- **cnt** - 実際の値の代わりに値の数を挿入

次のテーブルに、使用可能なバインディングプレフィックスのすべてのバージョンのリストを示します。例では、選択された値 (緑の値) として 1、2、3、オプションの選択された値 (白の値) として 4 の *MyField* という名前の項目を想定しています。

5 On-demand アプリによるビッグ データの管理

バインディングプレフィックス

プレフィックス	説明	例	以下に置換済み
odag_	選択した値 (緑) とオプションの値 (白) で置き換えられます。値のテキストバージョンを選択します。これは文字列値の標準のプレフィックスです。	<code>\$(odag_ MyField)</code>	'1','2','3','4'
odagcnt_	対応する odag_ バインディングの値の数に置き換えられます。このプレフィックスは、クエリの最適化に使用されます。	<code>\$(odagcnt_ MyField)</code>	4
odagn_	<p>選択した値 (緑) とオプションの値 (白) で置き換えられます。値の数値バージョンを選択します。これは数値の標準のプレフィックスです。</p> <div style="border: 1px solid gray; padding: 10px; margin-top: 10px;"> <p> データモデルに項目の選択された値またはオプションの値がない場合は、数式に <code>noValue</code> を指定する必要があります。例: <code>\$(odagn_MyField){"noValue": "-99999"}</code>。</p> <p>詳細については、「値の引用符と区切り文字の変更 (page 182)」を参照してください。</p> </div>	<code>\$(odagn_ MyField)</code>	1,2,3,4
odagncnt_	対応する odagn_ バインディングの値の数に置き換えられます。これはクエリの最適化のためです。	<code>\$(odagncnt_ MyField)</code>	4
odago_	<p>オプションの値 (白) で置き換えられます。値のテキストバージョンを選択します。これはクエリの最適化のためです。</p> <p>大規模データベースの最適化 (page 180).</p>	<code>\$(odago_ MyField)</code>	'4'
odagocnt_	対応する odago_ バインディングの値の数に置き換えられます。これはクエリの最適化のためです。	<code>\$(odagocnt_ MyField)</code>	1
odagon_	<p>オプションの値 (白) で置き換えられます。値の数値バージョンを選択します。これはクエリの最適化のためです。</p> <p>大規模データベースの最適化 (page 180).</p>	<code>\$(odagon_ MyField)</code>	4
odagoncnt_	対応する odagon_ バインディングの値の数に置き換えられます。これはクエリの最適化のためです。	<code>\$(odagoncnt_ MyField)</code>	1

プレフィックス	説明	例	以下に置換済み
odags_	選択した値 (緑) で置き換えられます。値のテキストバージョンを選択します。これはクエリの最適化のためです。 大規模データベースの最適化 (page 180).	$\$(odags_MyField)$	'1','2','3'
odagscnt_	対応する odags_ バインディングの値の数に置き換えられます。これはクエリの最適化のためです。	$\$(odagscnt_MyField)$	3
odagsn_	選択した値 (緑) で置き換えられます。値の数値バージョンを選択します。これはクエリの最適化のためです。 大規模データベースの最適化 (page 180).	$\$(odagsn_MyField)$	1,2,3
odagsncnt_	対応する odagsn_ バインディングの値の数に置き換えられます。これはクエリの最適化のためです。	$\$(odagsncnt_MyField)$	3



空の値は、テキストバージョンでは除外されます。数値バージョンでは、非数値とNaN値は除外されます。

大規模データベースの最適化

odags_ および odagsn_ プレフィックスは、クエリの最適化を目的としています。バインド項目に選択項目がない場合、odag_ にはすべての値が含まれ、odags_ には値が含まれません。場合によっては、odags_ および odagscnt_ プレフィックスを使用する方が効率的です。これにより、値のセットが空かどうかをテストできます。例えば、odag_MyField のすべての値をテストするよりも、MyField で選択を行わない場合のほうが効率的です。

WHERE ($\$(odagscnt_MyField)=0$ OR MyColumn IN ($\$(odags_MyField)$))

オンデマンド項目ではない選択アプリで選択できる代替項目がある場合、odags_ は使用できません。例えば、ユーザーが CountryName で選択を行ったが、バインディング数式が関連項目 CountryCode にある場合、odags_ は使用できませんでした。このような場合は、代わりに odago_ を使用できます。odago_ バインディングに値がない場合は、すべての値を含めるか、値を含めないことを意味する可能性があります。

数値のバインディング

On-demand アプリにバインドするデータが文字列ではなく数値で構成されている場合は、数値項目を引用符で囲む動作を無効にすると効果的です。たとえば、売上レコードに数値の DAY_OF_WEEK 列が含まれており、選択アプリのユーザーが DAY_OF_WEEK の任意の組み合わせを選択できるようにする場合は、選択アプリをロードするために使用される集計クエリを増やして、SELECT リストと GROUP BY リストの両方に DAY_OF_WEEK を含めます。データベースで文字列から数値への自動型変換機能がサポートされていない場合、DAY_OF_WEEK の値を選択するときに引用符で囲まれていると、実行時にクエリエラーが発生する可能性があります。

この状況を処理するには、バインディング数式サフィックスの数値バージョンを使用できます。これにより、項目バインディングは文字列値ではなく、選択アプリからの数値を使用するようになります。次の数値バージョンを使用できます。

- odagn_
- odagon_
- odagsn_

数値バージョンを使用することにより、選択された値を格納するデュアル値の数値部分から値が取得され、既定では値が引用されません。

一定数の選択の要求

状況によっては、On-demand アプリのクエリに、特定の項目に関する、特定の数または範囲の値を含める必要があることがあります。たとえば、On-demand アプリのクエリに開始日から終了日までの全売上高を取得するための **BETWEEN** 節が含まれている場合、YEARQUARTER 項目のバインド数式には、YEARQUARTER の値を正確に 2 つ選択するように要求するサフィックス構文 [2] を以下に示すように含めることができます。

`$(odag_YEARQUARTER)[2]`

選択アプリの [On-Demand アプリのナビゲーションポイント] は、選択された YEARQUARTER の値が正確に 2 つにならない限り無効のままになります。YEARQUARTER の値を正確に 2 つ選択する必要があることを示すメッセージが表示されます。

選択数量の制約により、[選択アプリ] と [On-Demand アプリ] の間に前提条件の連鎖が作成されます。これは、数量の制約を使用しないバインド式とは異なります。たとえば、[テンプレートアプリ] のスクリプトに、以下のよう数量の制約なしのバインド数式が含まれているとします。

`$(odag_MYFIELD)`

[選択アプリ] に MYFIELD という名前の項目を含めたり、その項目に選択値 (存在する場合) を含めたりするように求める要件はありません。[選択アプリ] に MYFIELD という名前の項目が含まれていない場合、またはユーザーがそのアプリから何らかの選択を行うことを単に無視した場合でも、他の選択によってレコード制限値の条件が満たされているときは、[On-Demand アプリのナビゲーションポイント] が有効となる可能性があります。

一方、次のようなバインド数式があるとします。

`$(odag_MYFIELD)[1+]`

この場合は、以下の 2 つの要件が [選択アプリ] に課されます。

- [選択アプリ] には、MYFIELD という名前の項目を含める必要があります。
- ユーザーは、MYFIELD の値を少なくとも 1 つ選択する必要があります。

この種類のバインド数式では、[テンプレートアプリ] と一緒に使用できる [選択アプリ] が制限されるため、注意して使用する必要があります。[テンプレートアプリ] にリンクするすべての [選択アプリ] に対して選択数量の要件を確実に課したい場合を除き、この数量の制約を [テンプレートアプリ] に課さないでください。

データバインドプロセスを実行するため、On-Demand アプリサービスは、スクリプト内のコメントに影響されない、文字列代替手法を使用します。このため、アプリ生成後のバインド値のリストをコメントに含めたい場合を除き、コメント内ではバインド数式を使用しないでください。

数量に関するその他の制約を設定することも可能です。次の表に、選択数量の制約に関するさまざまな組み合わせの概要を示します。

5 On-demand アプリによるビッグ データの管理

選択数量の制約に関するさまざまな組み合わせ

制約のパターン	選択要件
<code>\$(odag_YEARQUARTER)[2]</code>	YEARQUARTER の値を正確に 2 個選択する必要があります。
<code>\$(odag_YEARQUARTER)[2-4]</code>	YEARQUARTER の値を 2 ~ 4 個選択する必要があります。
<code>\$(odag_YEARQUARTER)[2+]</code>	YEARQUARTER の値を 2 個以上選択する必要があります。
<code>\$(odag_YEARQUARTER)[2-]</code>	YEARQUARTER の値を最大 2 個選択する必要があります。



テンプレート アプリにおけるすべての数量制約が満たされているかどうかを確認するチェックは、アプリ生成プロセス中に実行されます。数量制約違反があった場合、アプリ生成リクエストは拒否され、エラーメッセージが表示されます。

値の引用符と区切り文字の変更

選択アプリで選択された項目値のリストがテンプレートアプリのスクリプトに挿入される場合、その値は単一引用符で囲まれ、コンマで区切られます。これらは、引用符と区切り記号の既定の文字です。これらの値は、各項目のバインディング ステートメントに付加される構文で変更できます。例:

```
$(odag_ORIGIN){"quote": "|", "delimiter": ";"}
```

新しい設定値は、選択アプリから取得したバインド値のリストを次に作成するときに使用されます。例えば、選択した値が年の最初の 3 か月の場合、リストは以下のように作成されます。

```
|January|;|February|;|March|
```



引用符と区切り文字の既定値は、ほとんどの標準的な SQL データベースに対応しています。ただし、一部の SQL データベースでは機能しない場合があり、また NoSQL や REST を始めとする多くの動的データソースには対応していません。そうしたソースの場合は、このバインディング数式を付加して、引用符と区切り文字を変更する必要があります。

次のテーブルに、引用符と区切り文字を変更するためのフォーマット パラメーターの概要を示します。

パラメーターの書式

パラメーター	既定値	定義
引用符	テキストプレフィックスの場合は '(一重引用符) 数値プレフィックスの場合は空	各値の前後に追加されます
区切り記号	カンマ	すべての値の間に追加されます

5 On-demand アプリによるビッグ データの管理


パラメーター	既定値	定義
quoteReplace	"(二重一重引用符)	<p>値が空でなく、引用符も空でない場合、値内のすべての引用符の出現は、指定された文字列に置き換えられます。</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p> quoteReplace は、<i>odagn_</i> などの数値プレフィックスのパラメーターとしてサポートされていません。 quoteReplace は数値プレフィックスによって無視されます。</p> </div>
noValue	空	<p>項目に値が選択されていない場合は、代わりにこの値が使用されます。このパラメーターは、選択範囲内の特定のフィールドの値がない場合に役立ちます。</p> <p>値は、ソースデータに存在しない値に設定する必要があります。例えば、数値の場合、データベース内のすべての値が正の場合は負の値を使用します。</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p> 引用符で囲まれていない値の場合、項目の選択した値を空のセットにできる場合は、noValue を指定する必要があります。</p> </div>

次のテーブルは、*odag_* および *odagn_* のフォーマット仕様と生成された値の概要を示しています。生成された挿入値は、VAL1、VAL2 の既定のデータ値に基づいています。

odag_ の例のフォーマット仕様と生成された値

書式指定	説明	生成された挿入値
指定なし	'で囲まれた値のカンマ区切りのリスト。	'VAL1','VAL2'
{"quote": "", "delimiter": ""}	連結値	VAL1VAL2
{"quote": "X", "delimiter": "Y"}	Xで囲まれ、Yで区切られた値。	XVAL1XYXVAL2X
{"quote": "XX", "delimiter": "YY"}	XXで囲まれ、YYで区切られた値。	XXVAL1XXYYXXVAL2XX
{"quote": "X"}	Xで囲まれ、,(既定)で区切られた値。	XVAL1X,XVAL2X
{"delimiter": "YY"}	'(既定)で囲まれ、YYで区切られた値	'VAL1'YY'VAL2'
{"quote": ""}	,で区切られた、引用符で囲まれていない値。	VAL1,VAL2

書式指定	説明	生成された挿入値
{ "quote": "A", "quoteReplace": "\A" }	A で囲まれ、コンマ(既定)で区切られた値。項目内の値は \A に置き換えられます。	AVAL1A,AVAL2A



この例では、\ が json 形式のエスケープ文字であるため、二重の \ が必要です。

odagn_ の例のフォーマット仕様と生成された値

書式指定	説明	生成された挿入値
指定なし	引用符で囲まれていない値のカンマ区切りのリスト。数値が使用されることに注意してください。	VAL1,VAL2
{ "delimiter": "YY" }	YY で区切られた、引用符で囲まれていない値	VAL1YYVAL2
{ "quote": "A", "quoteReplace": "\A" }	odag_ プレフィックスと比較すると、quoteReplace パラメータは無視されます。	AVAL1A,AVAL2A

個々の値の処理

項目値の個別処理が必要な場合は、インラインメソッドを使用して変数 **Values** に値を生成し、**Replace** または別の関数で任意の処理を実行できます。以下の例では、**Replace** がプレースホルダー値とともに使用されています。

```
MyTempBindingData:
LOAD * INLINE [VAL
$(odag_MyField){ "quote": "", "delimiter": "\n" }
];

_TempTable:
LOAD Concat(chr(39) & Replace(text, from_str, to_str) & chr(39), ',') as CombinedData Resident
MyTempBindingData;
LET Values = Peek('CombinedData',0,'_TempTable');
drop table _TempTable;
drop table MyTempBindingData;
```

5.8 On-demand アプリの構築

On-demand アプリには、選択アプリによって集計形式でロードされたデータサブセットがロードされます。On-demand アプリは、On-demand 選択アプリを On-demand テンプレートアプリにリンクして作成します。選択アプリとテンプレートアプリは、On-demand アプリを構成する基本要素です。

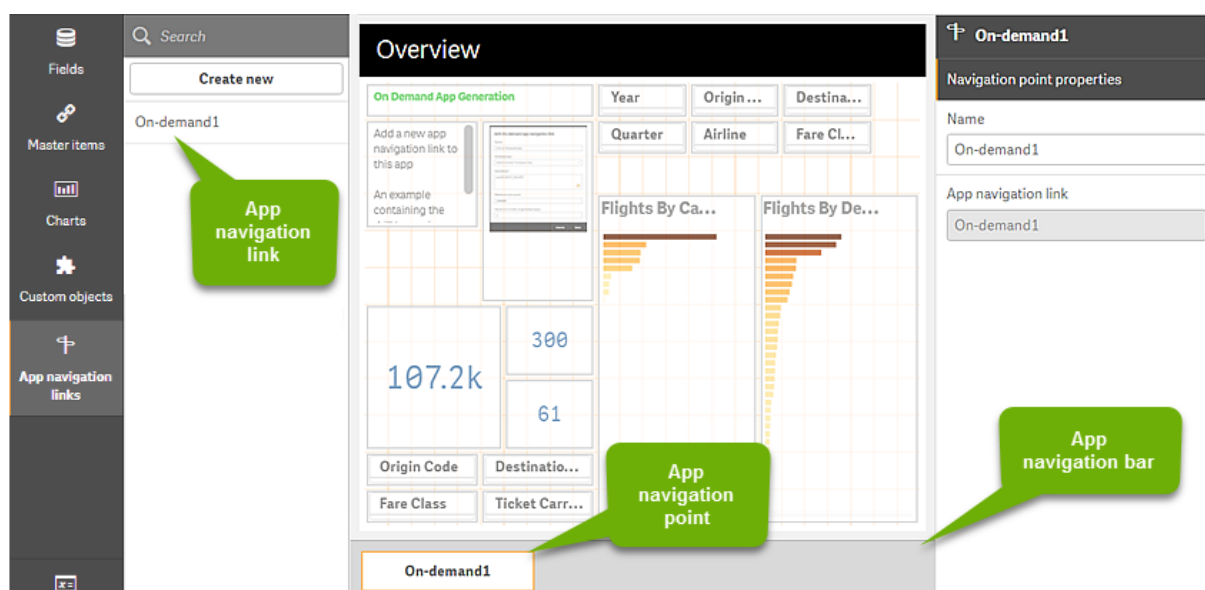
On-demand アプリを構築するには、一緒にリンクできる選択アプリとテンプレートアプリを最初に作成します。リンクを可能にするには、選択アプリとテンプレートアプリに、一緒にバインドできる共通のデータ項目を設けます。

5 On-demand アプリによるビッグデータの管理

選択アプリは複数のテンプレートアプリにリンクでき、単一のテンプレートアプリは複数の選択アプリにリンクできます。ただし、テンプレートアプリのデータバインディング数式は、リンクされている選択アプリの項目に対応する必要があります。

On-demand アプリのナビゲーションリンクによって、選択アプリがテンプレートアプリに追加されます。On-demand アプリのナビゲーションリンクは選択アプリ内で作成されます。ナビゲーションリンクを定義したら、選択アプリの[アプリのナビゲーション]バーにOn-demand ナビゲーションポイントとして追加できます。アプリの各シートには、自身の[アプリのナビゲーション]バーが含まれます。次に、アプリナビゲーションポイントからOn-demand アプリを生成します。

表示を構築するOn-demand アプリの例。



選択したデータの別々の組み合わせが含まれる、複数のOn-demand アプリを同じアプリナビゲーションポイントから生成できます。

1つのアプリナビゲーションリンクに対するポインターは、同じ選択アプリの複数のシートに追加できます。また、シートには、複数のアプリナビゲーションリンクから作成された複数のアプリナビゲーションポイントを含めることができます。

選択アプリがナビゲーションリンクとナビゲーションポイントで完了したら、On-demand アプリを生成できます。

次の手順を実行します。

1. On-demand 選択アプリを開き、[編集]を選択します。
2. 左側のパネルから、[アプリのナビゲーションリンク]を選択します。
3. [新規作成] ボタンをクリックして [On-demand アプリのナビゲーションリンクの新規作成] ダイアログを開きます。
4. 新しい On-demand アプリのナビゲーションリンクに名前を付けます。
5. [On-demand テンプレートアプリ]を選択します。

[**テンプレート アプリ**] ドロップダウン リストには、有効ではないテンプレート アプリも含まれています。テンプレート アプリとして作成されており、そのデータ バインディング 数式が、使用している選択 アプリの各項目に対応しているアプリを選択する必要があります。それ以外のアプリを選択すると、アプリのナビゲーション リンクから生成された **On-demand** アプリでエラーが生成されます。

6. 選択 アプリの選択 ステートを使ってアクセス可能な集計 レコードによって表される、詳細なレコードの合計数を計算する数式を入力します。
その数式では、通常、**SUM** 関数を使用して、選択したレコードの合計数を得ます。その結果は、ロードするデータ量が **On-demand** アプリを生成する場合の指定範囲内にあるかどうかを判定するために使用されます。
7. [**最大行数**] を指定します。
[**最大行数**] の値により、**On-demand** アプリでロード可能なレコード数の上限が設定されます。レコードの数は、[**数式**] エントリの関数によって計算されます。選択 アプリの行予測数式によって計算されるレコードの数が [**最大行数**] の値より大きい場合、**On-demand** アプリを生成することはできません。アプリは、行予測数式によって計算されたレコードの数が、[**最大行数**] の値によって設定された上限値以下になっている場合にのみ生成可能です。
[**最大行数**] に使用する数式を作成するには、選択 アプリで使用可能な項目から合計レコード数を計算する方法が分かっている必要があります。
8. [**生成されるアプリの最大数**] を指定します。
選択 アプリの [**アプリのナビゲーション**] バーにある同じ **On-demand** アプリのナビゲーション ポイントから、複数の **On-demand** アプリを生成できます。複数のアプリを生成する理由は、各アプリに異なる選択データを取り込むことができるからです。最大数のアプリが生成されると、ナビゲーション ポイントからアプリを生成しているユーザーは、既存のアプリの 1 つを削除しないと、新しい **On-demand** アプリを生成できません。
生成済みアプリの最大数は、**On-demand** アプリのナビゲーション リンクに適用されます。ナビゲーション リンクから **On-demand** アプリのナビゲーション ポイントを 1 つ作成すると、そのナビゲーション ポイントから最大数のアプリを作成できます。ただし、同じナビゲーション リンクから複数のナビゲーション ポイントが作成される場合、これらのナビゲーション ポイントから生成された **On-demand** アプリの総数は、[**生成されるアプリの最大数**] の設定に制限されます。
9. ナビゲーション リンクから生成されたアプリを削除するまで保持する期間について、[**保存期間**] フィールドに数値を入力します。
10. [**保存期間**] フィールドの右のドロップダウン メニューから、保存期間の時間単位を選択します。
保存期間のオプションは、時間、日数、または [**無期限**] です。
ナビゲーション リンクから生成されたすべての **On-demand** アプリが、この設定に従って保持されます。生成された **On-demand** アプリの存在期間は、現在の時間および最後のデータロードの時間の差に相当します。**On-demand** アプリの存在期間のこの計算は、公開アプリと未公開アプリで同じです。**On-demand** アプリを生成してから手動で公開した場合、存在期間の計算は同じです。生成されたアプリの最後のデータロードに基づきます。



匿名ユーザーが生成したアプリに対して適用される、**On-demand** アプリサービスの保存期間設定もあります。この設定は、自身の ID でログインしているユーザーにとっての保存期間には影響しません。匿名ユーザーが生成したアプリの場合、ナビゲーション リンクにおける [**保存期間**] 設定および **Qlik Management Console** で設定する **On-demand** アプリサービス設定は、より短くなります。

11. **[オープン時のデフォルト表示]** ドロップダウン メニューで、ナビゲーション リンクから生成されたアプリが開かれたとき、最初に表示するシートを選択します。
[アプリ概要] を選択することも、ナビゲーション リンクが作成された選択アプリのシートの 1 つにすることもできます。
12. **[公開先]** ドロップダウン メニューからストリームを選択します。ナビゲーション リンクから生成されたアプリはこのストリームに公開されます。
選択したストリームに公開する権限が必要です。選択したストリームに対して公開権限がない場合、ナビゲーション リンクからの On-Demand アプリ生成の試行は失敗します。
生成されたアプリを公開するストリームを選択する場合、On-Demand アプリの対象ユーザーに、ストリームに対する読み取り権限があることを確認する必要があります。
また、**[非公開 (ワークスペースに保存)]** を選択して、生成されたアプリを公開せずに、ユーザーのワークスペースに保存することもできます。



匿名ユーザーが公開された選択アプリを使用できる場合、On-demand アプリのナビゲーション リンクは、匿名ユーザーがアクセス可能なストリームに対して公開するよう構成します。ナビゲーション リンクから生成された On-demand アプリが自動で公開されない場合、匿名ユーザーがそれらのアプリを生成しようとするとエラー メッセージが表示されます。

アプリの生成後、手動で公開できます。

13. **[作成]** をクリックします。**[アプリのナビゲーション リンク]** のリストに新しい On-demand アプリのナビゲーション リンクが表示されます。
14. アプリのナビゲーション リンクを選択アプリの **[アプリのナビゲーション]** バーにドラッグします。
アプリのナビゲーション リンクを選択アプリにドラッグすると、On-demand アプリのナビゲーション ポイントが作成されます。新しい On-demand アプリのナビゲーション ポイントのプロパティが右側のパネルに表示されます。必要に応じて、そのパネルでナビゲーション ポイントの名前を変更できます。
15. シートエディターで **[完了]** をクリックします。
これで、On-demand 選択アプリを使用または公開する準備ができました。これで、On-demand 選択アプリを使用または公開する準備ができました。選択アプリのユーザーは、選択アプリの **[アプリのナビゲーション]** バーのナビゲーション ポイントから On-demand アプリを生成できます。

5 ダイナミックビューによるデータの管理

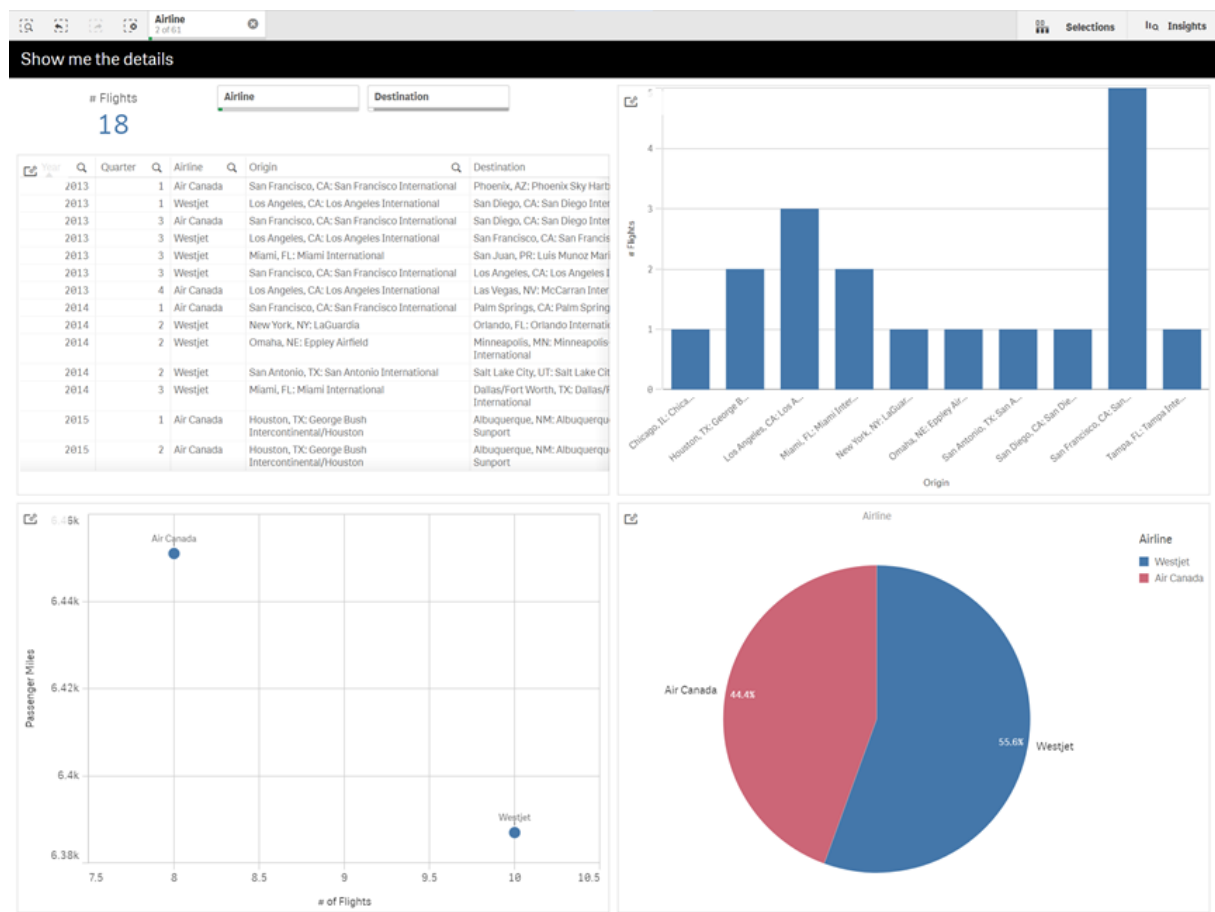
ダイナミックビューにより、ユーザーは参照したい分析ソースと、ビジュアライゼーションでデータを更新する時期の両方を直接コントロールできます。

ダイナミックビューにより、選択を決定するとともに動的に更新される、チャート内の大型データセットの関連するサブセットをクエリしたり、表示したりできるようになります。これにより、大量のデータまたはデータが急速に変化するシナリオにおける最新のビジュアライゼーションが可能になります。

5.9 ダイナミックビューの概要

ダイナミックビューで、ベースアプリを別のアプリに接続できます。そうするとベースアプリで、接続したアプリからのマスタービジュアライゼーションの使用が可能です。これによりアプリ作成者は、テンプレートアプリからのマスタービジュアライゼーションをその他のアプリでダイナミックチャートとして使用できるようになります。ベースアプリに追加できるダイナミックビューの数に制限はありません。

ダイナミックビューとダイナミックチャートによるシートビュー



ダイナミックビューは、次の3つの主要なコンポーネントを基にしています。

- ダイナミックビュー: テンプレートアプリに接続するベースアプリに追加されたメカニズムで、アプリ作成者は、テンプレートアプリからのマスター ビジューライゼーションをベースアプリに追加できるようになります。
- ダイナミックビューのテンプレートアプリ: クラウドデータベースなどのデータソースへの接続が含まれる Qlik Sense アプリです。
- ダイナミックチャート: ダイナミックビューのテンプレートアプリにあるマスター ビジューライゼーションで、ベースアプリに追加したり、ユーザにより手動で更新したりできます。

テンプレートアプリとベースアプリで同じデータを使用する必要はありません。顧客の購入データセットがある場合、気象データを含むテンプレートアプリをダイナミックビューに追加して、相関性を調べることができます。

ベースアプリの値を使用して、テンプレートアプリのソースからクエリされたデータをフィルターできる場合、テンプレートアプリのスクリプトでバインド数式を使用できます。これによりダイナミックビューは、テンプレートアプリのデータソースからのベースアプリにある選択に、特に関連のあるデータのサブセットのみをクエリできるようになります。

このサブセット機能は、ベースアプリが集計データを含んでおり、ダイナミックビューのデータは同じソースではあるものの、ベースアプリのデータよりも粒度が細かい(例: ベースアプリには月次の売上高と製品のブランドが含まれ、テンプレートアプリには日次の売上高と製品名が含まれる)場合に有用です。テンプレートアプリへのバインド数式の追加に関する詳細については、「[On-demand テンプレートアプリのバインディング数式 \(page 178\)](#)」を参照してください。

ダイナミックビューでは、あらゆる種類のデータが使用できます。ダイナミックビューは、大量のデータまたはデータが急速に変化するシナリオの取り扱いといった、データベースでデータ集計を実行したほうが良い場合に特に有用です。これは、データソースからのデータ転送の遅延を避けるのに役立ちます。

ダイナミックビューは、[\[アセット\]](#) パネルからアクセスできます。ダイナミックビューは、QMC の管理者により有効化できます。ダイナミックビューの有効化については、「[On-Demand アプリの管理](#)」を参照してください。

ダイナミックビューは、On-demand アプリの生成と類似しています。両方とも On-demand データを提供するためにテンプレートアプリを使用しますが、ダイナミックビューでは、On-demand アプリ全体を生成するのではなく、シート内で個別のチャートが使用できるようにします。On-demand アプリも使用している場合は、On-demand テンプレートアプリを使用してダイナミックビューを作成できます。On-demand アプリ生成の詳細については、「[On-demand アプリによるビッグデータの管理 \(page 172\)](#)」を参照してください。

ダイナミックビュー

ダイナミックビューを作成する際、テンプレートアプリを選択し、オプションとして行を制限する数式を適用してダイナミックビューがアクセスするデータ量を管理します。ダイナミックビューが作成された後、シートにテンプレートアプリからのマスター ビジューライゼーションを追加できます。

複数のダイナミックビューで、同一のテンプレートアプリを使用できます。各ダイナミックビューは、個別に更新されます。ダイナミックビューのテンプレートアプリスクリプトでバインド数式が使用されている場合、そのテンプレートアプリを使用する各ダイナミックビューにどのデータをロードするかをベースアプリ内の選択によりコントロールできます。たとえば、同一のテンプレートアプリを使用する2つのダイナミックビューがあるとします。そして、ベースアプリの *SaledDate* から2018年1月1日を選択し、一方のダイナミックビューを更新します。それから選択を2019年1月1日へと変更し、もう一方のダイナミックビューを更新すると、ダイナミックチャートを比較できます。

ユーザーがダイナミックビューを含むアプリにアクセスすると、**On-demand** アプリが**[作業]**に追加されます。このアプリには、現在のデータを含むダイナミックビューテンプレートアプリが含まれており、基本アプリにダイナミックビューを設定するために使用されます。ダイナミックビューが更新されるたびに、新しいバージョンに置き換えられます。ユーザーがダイナミックビューテンプレートアプリの所有者でない場合、ロードスクリプトは削除されます。これらのアプリは、最後の更新から**24**時間後に削除されます。

ダイナミックビューの作成と編集については「[ダイナミックビューによるデータの管理 \(page 188\)](#)」を参照してください。

ダイナミックビューの使用については、「[ダイナミックビューとダイナミックチャートの使用 \(page 192\)](#)」を参照してください。

ダイナミックビューのテンプレートアプリ

ダイナミックビューのテンプレートアプリは、データとマスタービジュアライゼーションを供給するために使用される**Qlik Sense** アプリです。

ダイナミックビューのテンプレートには、ベースアプリ内の選択に基づいてデータソース上でクエリを作成するために使用する、バインディング数式を含んだロードスクリプトをもたせることができます。バインディング数式は通常、**Qlik Sense** ロードスクリプトを書いた経験のあるユーザーによって作成されます。テンプレートアプリには、ダイナミックチャートのアクティベーション中に供給される入力パラメーターに基づいた、クエリのフィルター条件をもたせることができます。

ダイナミックビューのテンプレートアプリのデータモデルが完了すると、マスタービジュアライゼーションにテンプレートアプリを追加できます。こうしたマスタービジュアライゼーションは、ダイナミックビューを経由してアクセスしたり、他のアプリでダイナミックチャートとして追加したりできます。

テンプレートアプリの作成については、「[On-demand テンプレートアプリの作成 \(page 176\)](#)」を参照してください。

ダイナミックチャート

ダイナミックチャートは、ダイナミックビューのテンプレートアプリのマスターチャートから抽出されます。ダイナミックチャートは、ダイナミックビューを使用して他のアプリのシートに追加できます。他の**Qlik Sense** チャートと異なり、ユーザーはチャートの更新オプションを使用して、ダイナミックビューのソースデータを更新するタイミングをコントロールできます。ダイナミックビューのデータがバインド数式によりコントロールされている場合は、**Qlik Sense** がベースアプリの選択ステートを追跡します。ベースアプリの選択ステートが変更されるたびに、ダイナミックビューのバインド項目の新しい値のセットは、ビューの前回更新時に使用された値と合致しなくなるため、ダイナミックビューの各チャートには古いデータを意味するアイコンが表示されます。

ダイナミックチャートの使用については、「[ダイナミックビューとダイナミックチャートの使用 \(page 192\)](#)」を参照してください。

5.10 ダイナミックビューの制限

ダイナミックビューには、次の制限があります。

- ストーリーは、ダイナミックビューに非対応です。ダイナミックチャートのスナップショットはストーリーに追加できますが、ダイナミックチャートでソースに移動することはできません。
- **Qlik NPrinting** は、ダイナミックビューに非対応です。

- ダイナミックビューは、ダッシュボードとビジュアライゼーションのバンドル拡張に対応しています。その他の拡張は非対応です。
- アプリの所有権でダイナミックビューの所有権が変更されることはありません。
- 管理スペースにあるアプリ内に、ダイナミックビューを作成することはできません。
- Qlik Sense Desktop では、ダイナミックビューが非対応です。
- 動的チャートをコンテナに配置することはできません。
- Trellis コンテナおよびコンテナは、動的ビューのマスターチャートとしてサポートされていません。

5.11 ストリームとダイナミックビュー

アクセスできる任意のアプリに対してダイナミックビューを作成できます。ストリームで公開されているアプリと[作業]で未公開のアプリに対してダイナミックビューを作成できます。また、アクセスできるストリーム内の他のユーザーが所有する公開アプリに対してダイナミックビューを作成できます。

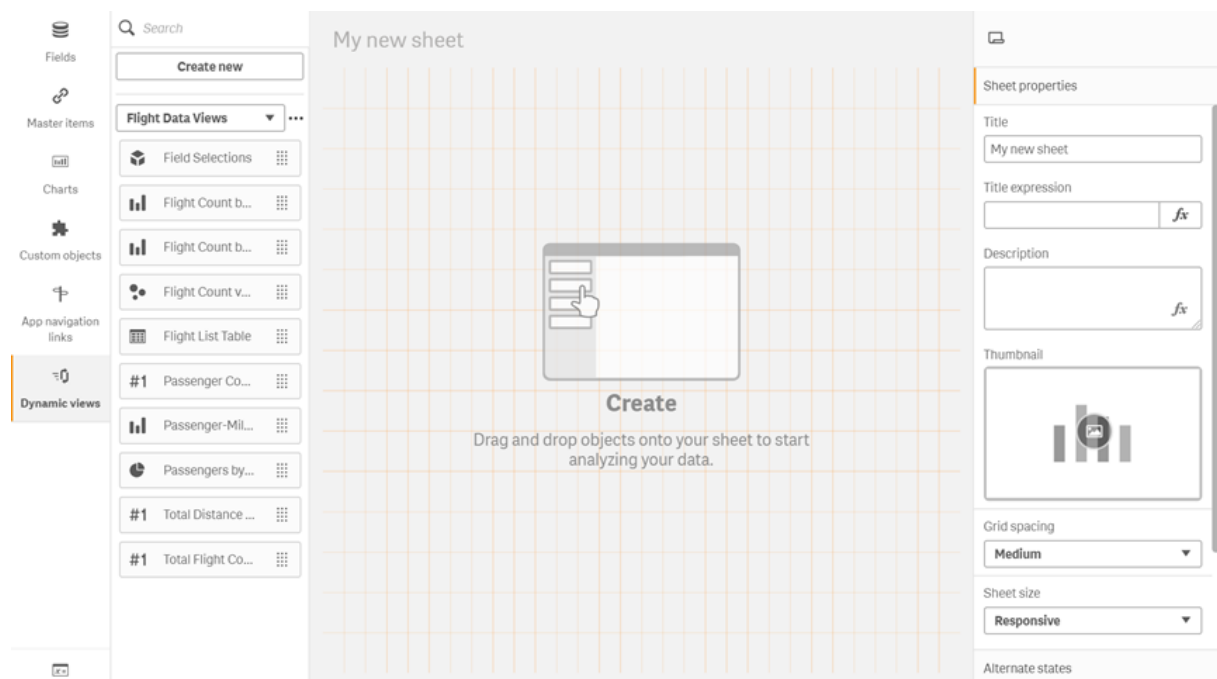
ダイナミックビューを持つアプリにアクセスできるユーザーは、テンプレートアプリにアクセスできない場合でも、ダイナミックビューを使用できます。

公開されたアプリにダイナミックビューを追加することはできません。アプリを複製して再公開し、新しいダイナミックビューを追加できます。

5.12 ダイナミックビューとダイナミックチャートの作成

ダイナミックビューは、シートビューの[アセット]パネルから追加できます。ダイナミックビューからのダイナミックチャートをシートに追加できます。

新しいシートの[アセット]パネルにあるダイナミックビュー



ダイナミックビューを作成

ダイナミックビューを作成する時に、ダイナミックビューの一部として持ち込まれる行の数を制限することができます。これにより、ダイナミックビューが非常に大きなデータセットに接続している時に、返す行が多くなり過ぎることを防止できます。

次の手順を実行します。

1. シートの **[編集]** モードで、**[ダイナミックビュー]** をクリックします。
2. **[新規作成]** をクリックします。
3. **[名前]** の後に、ビューの名前を入力します。
4. **[テンプレートアプリ]** の後に、テンプレートアプリを選択します。
5. **[行数制限]** の後に、オプションとして **[行数制限数式]** の使用と、**[最大行数]** の設定を選択できます。
6. **[作成]** をクリックします。

シートにダイナミックチャートを追加

ダイナミックチャートは、**[アセット]** パネルからシートに追加できます。


テンプレートアプリでの編集の権限がある場合は、ダイナミックチャートの基となるマスターチャートの編集が可能です。

ダイナミックチャートを編集するには、シートの **[編集]** モードでダイナミックチャート、**[ソースアプリの編集]** の順に選択します。

ダイナミックビューの編集

ダイナミックビューの編集と行数制限の変更ができます。ダイナミックビューで使用されるテンプレートアプリの変更はできません。

次の手順を実行します。

1. ダイナミックビューのドロップダウンからダイナミックビューを選択します。
2. **⋮** をクリックして  をクリックします。

5.13 ダイナミックビューとダイナミックチャートの使用

ダイナミックチャートは、その他の Qlik Sense ビジュアライゼーションと同様にインタラクティブです。ユーザーはダイナミックビューで、データを更新するタイミングを直接制御することもできます。ダイナミックビューのテンプレートアプリにより、ロードスクリプトでバイト数式が使用される場合、ダイナミックビューのデータを更新するクエリには通常、ベースアプリの選択ステートで選択した値を使用したフィルターが条件が含まれます。ユーザーがダイナミックビューを更新すると、そのビューに関連するすべてのダイナミックチャートが新しいデータで更新されます。

スナップショットを取得したり、探索メニューを開いたりするため、チャートオプションのコンテキストメニューにアクセスするには、チャートを右クリックし、**[ダイナミックチャート]** を選択します。

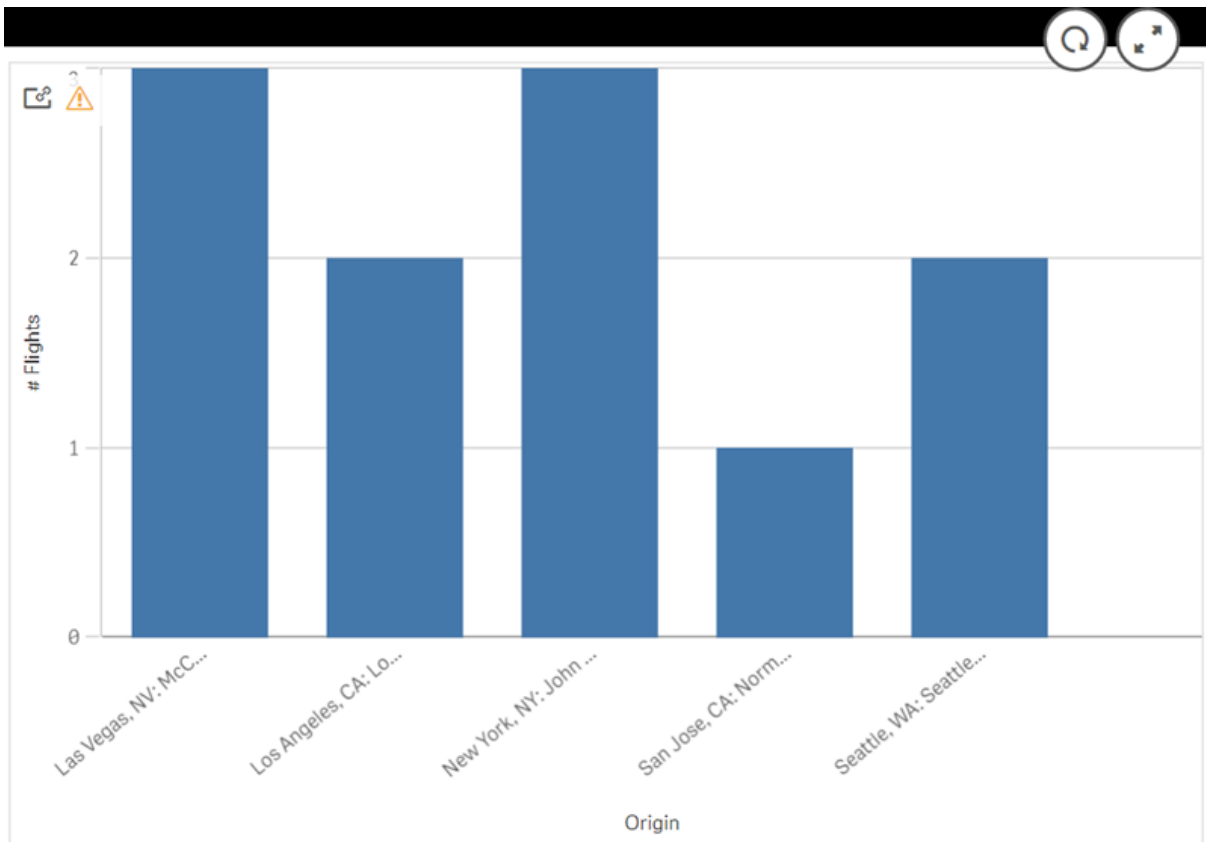
ダイナミックチャート経由でダイナミックビューとインタラクションできます。ダイナミックチャートを右クリックし、**[ダイナミックビュー]**を選択すると、チャートの更新、制約の表示、およびダイナミックビューの以前の更新で使用したダイナミックビューのバインド項目値の表示ができます。

ダイナミックビューでの選択

ダイナミックチャートでは、標準 Qlik Sense チャートと同様の連想選択モデルを使用します。ダイナミックチャートでの選択は、ベースアプリの選択に影響を及ぼすことなく、ベースアプリの選択バーには表示されません。ダイナミックビューでの選択をクリアするには、そのビューからダイナミックチャートを右クリックし、**[ダイナミックビュー]**、**[選択をクリア]**の順に選択します。

ダイナミックビューのテンプレートアプリにバインド数式がある場合、ベースアプリで選択した条件は、ダイナミックビューにロードされたデータの次回更新時に影響を与えます。Qlik Sense は、各ダイナミックビューの前回更新時に存在した選択ステートを常に追跡します。選択により、ダイナミックビューが前回更新された時から、ダイナミックビューのいずれかのバインド項目で使用する値が変更されるたびに、ダイナミックビューのデータは古いものとみなされます。データが古くなるたびに、すべてのダイナミックビューのチャートには古いデータを意味するアイコンが表示されます。古いチャートを更新すると、ビューの各バインド項目の更新済みの選択した値のセットを使用して、ダイナミックビューからのすべてのチャートを更新します。ベースアプリの選択ステートを更新する代わりに、ダイナミックビューが前回更新された時の状態に合わせて復元したい場合は、そのダイナミックビューのいずれかのチャートでコンテキストメニューを使用し、**[ダイナミックビュー]**、**[前回の選択を表示]**の順に選択し、**[適用]**をクリックします。

古いデータの通知を伴うダイナミックチャート



5 ダイナミックビューによるデータの管理

行数が現在の選択の結果で最大行数を超えるなど、ダイナミックビューのいずれかの制約が満たされていない場合、ビューのいずれのチャートにおいてもデータは表示されません。すべての制約を満たすようにベースアプリの選択状態が変更されるまで、ダイナミックビューの更新はできません。

選択が制約を超えているダイナミックチャート



Your current selections exceed the constraints set for this view.


ダイナミックビューの詳細表示

更新履歴、制約、およびダイナミックビューで使用される選択は、**[ダイナミックビューの詳細]**で表示できます。詳細にアクセスするには、ダイナミックチャートの隅にある**[ダイナミックビューの詳細]**アイコンをクリックします。

[更新]では、前回ダイナミックビューのデータが更新された時を表示します。

[ダイナミックビューの詳細]における詳細の更新

Dynamic view of Flight Details with 70k Limit


 Linked from ODAG Sample Detail

Refresh	Constraints	Selections
<p>Last completed run: Nov 25, 2019, 2:59:46 PM</p> <p><input type="button" value="Refresh"/></p>		

制約では、ダイナミックビューからコンテンツを生成する際に適用される項目と行の制約を表示します。

[[ダイナミックビューの詳細](#)]における現在の制約

Dynamic view of Flight Details with 70k Limit

 Linked from ODAG Sample Detail

Refresh
Constraints
Selections

Field and row constraints for generating an app from this dynamic view.

Row count

	Current	Constraint
✓	18	< 70000


Fields and value count

Field	Current	Constraint
Fare Class	✓ 3	None
Origin Code	✓ 10	None
Destination Code	✓ 13	None

選択では、このビューのデータを生成するために適用された選択を表示します。ダイナミックビューのデータが変更されるように新しい選択をして、ダイナミックビューを更新していない場合は、**[適用]** をクリックしてダイナミックビューの元の選択を復元できます。


[ダイナミックビューの詳細] における現在の選択

Dynamic view of Flight Details with 70k Limit

 Linked from ODAG Sample Detail

Refresh	Constraints	Selections
		<div style="border: 1px solid #ccc; padding: 5px; display: inline-block;">Apply</div>
Selections made when this app was generated		
Field	Values	
Airline	Air Berlin PLC and CO, Iberia Air Lines Of Spain, Japan Air Lines Co. Ltd., Korean Air Lines Co. Ltd., Klm Royal Dutch Airlines ...(2 more)	
Destination Name	Atlanta, GA: Hartsfield-Jackson Atlanta International, Billings, MT: Billings Logan International, Boston, MA: Logan International, Burlington, VT: Burlington International, Baltimore, MD: Baltimore/Washington International Thurgood Marshall ...(1 more)	

ダイナミックビューの更新

ダイナミックビューのデータを更新するには、ビューからダイナミックチャートを選択してクリックします 。ダイナミックチャートの更新オプションを使用すると、同じダイナミックビューからのすべてのチャートを更新します。更新により、ダイナミックビューのダイナミックチャート内で選んだあらゆる選択が削除されます。

6 データソースへの接続

Qlik Sense により、データが保管されている場所を問わず、さまざまな Qlik コネクタやその他のデータ接続タイプを使用してデータに接続することができます。

データ接続を作成する場合、データ接続は Qlik Sense に保存されます。そのため、よく使用するデータソースのデータをすばやく選択したり、ロードしたりできます。データベース、ソーシャル メディア データ、ローカル ファイル、リモートファイル、Web ファイルに接続できます。

6.1 接続の作成


データソースからデータを選択するため、新しいデータ接続を作成するか、または保存したデータ接続を使用できます。以下より、データ接続を作成したり、保存したデータ接続にアクセスしたりできます。

- データマネージャーの [データの追加]。
新しいデータをアプリに即座に追加して、関連付けを簡単に作成できます。
- データロードエディターの [データ接続]。
新規または既存のデータ接続からデータを選択するか、またはスクリプトエディターを使用してデータ接続からデータをロードします。既存のデータ接続を編集することもできます。



ここでは、自分が所有するデータ接続か、読み取りまたは更新のアクセス権を付与された接続だけが表示されます。アクセス権を取得する必要がある場合は、Qlik Sense システム管理者に問い合わせてください。

6.2 データ接続の種類

Qlik Sense により、データが存在する場所に関係なく、データにアクセスすることができます。Qlik Sense では、以下の種類のデータ接続を利用できます。Qlik Sense とともに使用するコネクタを  [Qlikダウンロードサイト](#) からダウンロードできます。

これらのデータソースにアクセスするコネクタの多くは Qlik Sense に組み込まれており、他にもコネクタを追加することができます。各タイプのデータ接続には、構成が必要な特別の設定があります。

添付ファイル

ドラッグ アンド ドロップにより、アプリに直接データファイルを添付します。

データベース コネクタ



Qlik Sense Enterprise のみ。

設定済みの ODBC データベース コネクタを使用し、ODBC データソースに接続します。

- Amazon Athena
- Amazon Redshift
- Apache Drill
- Apache Hive
- Apache Phoenix
- Apache Spark
- Azure SQL
- Azure Synapse
- Cloudera Impala
- Databricks
- Google BigQuery
- IBM DB2
- Microsoft SQL Server
- MongoDB
- MySQL Enterprise
- Oracle
- PostgreSQL
- Presto
- Snowflake
- Sybase ASE
- Teradata

Essbase

Essbase データセットに接続します。

ローカル ファイルまたはネットワーク ファイル



Qlik Sense Enterprise のみ。

ローカル ファイルまたはネットワーク ファイルに接続し、データを選択、ロードします。

DSN 経由の ODBC 接続



Qlik Sense Enterprise のみ。

データベース管理システム (DBMS) に接続します (ODBC 使用)。問題の DBMS の ODBC ドライバーをインストールし、データソース DSN を作成する。

Qlik Web コネクタ

ソーシャル メディアまたは Web ベースのデータソースに接続します。個別にインストールする必要があります。

- Amazon S3
- AYLIEN News V2
- AYLIEN Text Analysis
- Azure Storage
- Bitly V2
- Box
- Dropbox
- Facebook Insights
- GitHub
- Google Ad Manager
- Google AdWords
- Google Analytics
- Google Calendar
- Google Drive and Spreadsheets
- Google Search Console
- Helper Connector
- JIRA
- Mailbox IMAP
- Mailbox POP3
- MeaningCloud
- Microsoft Dynamics CRM V2
- OData
- Outlook 365
- Office 365 SharePoint
- RegEx Connector
- Repustate
- Sentiment140
- SMTP Connector
- Strava
- SugarCRM
- SurveyMonkey
- Watson Natural Language Understanding
- Twitter
- YouTube Analytics

REST

REST データソースに接続します。REST コネクタは REST データソース専用ではなく、REST API 経由で公開された任意のデータソースへの接続に使用できます。

Salesforce



Qlik Sense Enterprise のみ。

Salesforce.com アカウントに接続します。

SAP



Qlik Sense Enterprise のみ。

SAP NetWeaver からデータをロードします。

Web ファイル



Qlik Sense Enterprise only

Web URL 上の Web ベースデータソースに接続します。

Web ストレージプロバイダコネクタ

Web ストレージプロバイダから、ファイルベースのデータに接続することができます。

- Dropbox
- Google Drive

サードパーティコネクタ

サードパーティコネクタを使用して、**Qlik Sense** が直接サポートしていないデータソースに接続できます。サードパーティコネクタには、**QVX SDK** を使用して開発されたもの、またはサードパーティ開発者製のものがあります。**Qlik Sense** の標準インストールにサードパーティコネクタは含まれていません。

6.3 データ接続の保存場所

接続は、**Qlik Sense Repository Service** によってリポジトリデータベースに保存されます。**Qlik Sense** サーバー展開では、**Qlik Management Console** を使用してデータ接続を管理します。**Qlik Management Console** では、データ接続の削除やアクセス権の設定、その他のシステム管理タスクを実行できます。

Qlik Sense Desktop では、接続はすべて、暗号化なしでアプリに保存されます。



Qlik Sense Desktop 接続には、接続の作成時に入力したユーザー名、パスワードおよびファイルパスに関する詳細が保存されているため、アプリを別のユーザーと共有するときに、これらの格納済みの詳細をテキスト形式で使用できます。共有を目的としたアプリを設計する場合には、この点について考慮してください。

6.4 ファイルからのデータのロード

Qlik Sense では、さまざまなファイル形式からデータを読み込むことができます。

ファイル形式

次のデータファイル形式に対応しています:

- テキストファイル: 項目のデータは、コンマ、タブ、セミコロンなどの区切り文字で区切る必要があります。
例: コンマ区切り変数 (CSV) ファイル。
- HTML テーブル
- Excel ファイル:



パスワードで保護されている Excel ファイルまたは Excel Binary Workbook ファイル (.xlsb) からデータをロードすることはできません。

詳細については、「[Microsoft Excel スプレッドシートからのデータのロード \(page 204\)](#)」を参照してください。

- XML ファイル
- Qlik ネイティブ QVD および QVX ファイル
- 固定長レコードファイル
- Data Interchange Format (DIF) ファイル: DIF ファイルはデータロードエディターでのみロード可能です。

接続の種類

複数の種類のデータ接続からファイルをロードできます。

- ローカルとネットワークのファイル フォルダ: 詳細については、「[ローカルおよびネットワークのファイル フォルダからのファイルのロード \(page 203\)](#)」を参照してください。
- [添付ファイル] フォルダ: このフォルダを削除または編集することはできません。このフォルダにはアップロードしてアプリに添付されたファイルが入ります (Qlik Sense Desktop では利用できません)。詳細については、「[アプリへのデータの追加 \(page 16\)](#)」を参照してください。
- ウェブ上のファイル: 詳細については、「[Web リソースからのファイルのロード \(page 203\)](#)」を参照してください。



DataFiles 接続のファイル拡張子は大文字と小文字を区別します。例: .qvd。


ファイルからデータをロードする方法

ファイルからデータをロードする方法は、次のようにいくつかあります。



スペース内での編集権限を持つユーザーは、そのスペースで DataFiles の読み取り、書き込み、およびロードを行うことができます。その他のユーザーは DataFiles を表示することはできません。

データロードエディターでデータ接続からデータを選択する

[データ接続] に移動し、 **データを選択** ダイアログを使用して、ロードするデータを選択します。

スクリプトコードを作成してファイルからデータをロードする

ファイルはスクリプト内の **LOAD** ステートメントによりロードされます。**LOAD** ステートメントには、スクリプト式一式を含めることができます。別の Qlik Sense アプリからデータを読み込むには、**Binary** ステートメントを使用します。

ローカルおよびネットワークのファイルフォルダーからのファイルのロード

フォルダ接続を使用して、ローカルおよびネットワークのファイルフォルダからファイルをロードすることができます。

データ接続の設定

UI 項目	説明
パス	<p>データファイルを含むフォルダへのパス。次のいずれかの方法が可能です。フォルダを選択し、有効なローカルパスを入力するか、UNC パスを入力します。</p> <p>有効なパスの例: <code>C:\data\DataFiles\</code></p> <p>UNC パスの例: <code>\\myserver\filedir\</code></p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;">  パス内にあるマッピングされたネットワークドライブは使用できません。 </div>
名前	データ接続の名前。

Web リソースからのファイルのロード

Web ファイルデータ接続を使用して、FTP、HTTP、HTTPS などの Web リソースからファイルをロードすることができます。ファイルは、Qlik Sense に対応している任意のタイプにすることができます。

Web ファイルデータ接続の設定

UI 項目	説明
URL	<p>接続する Web ファイルの完全な [URL] (プロトコル識別子を含む)。</p> <p><code>http://unstats.un.org/unsd/demographic/products/socind/Dec.%202012/1a.xls</code></p> <p>FTP ファイルに接続する場合は、URL のユーザー名およびパスワードに「:」や「@」などの特殊文字を使用しなければならないことがあります。このような場合、特殊文字をパーセント記号 (%) およびその文字に対応する ASCII 16 進数コードに置換する必要があります。たとえば、「:」は「%3a」、「@」は「%40」に置き換えます。</p>
名前	データ接続の名前。

Web ファイルデータ接続における URL セットは、既定では静的ですが、書式指定設定である **URL is** により、この URL を上書きすることができます。これは、動的に作成された URL からデータをロードする必要のある場合に便利です。

動的に作成された URL からのデータのロード

この例では、Qlik Community の New to Qlik Sense というフォーラムの最初の 10 ページからフォーラムへの投稿をロードします。フォーラム ページには、1 ページにつき 20 件の投稿があり、URL の最後のパラメータである **start** により、ページの最初の投稿として表示される投稿が設定されます。この例の URL では、投稿番号 20 番から 20 個の投稿がページに表示されます。

```
https://community.qlik.com/community/qlik-sense/new-to-qlik-sense/content?filterID=contentstatus%5Bpublished%5D~objecttype~objecttype%5Bthread%5D&itemView=detail&start=20
```

カウンター *i* では、1 ステップあたり 20 で 180 までページを進めます。これは、**For** ループが 10 回実行されることを意味します。

ページをロードするには、**URL is** 設定の URL の最後で、スタートページを $\$(i)$ に置き換えます。

```
For i = 0 to 180 step 20
LOAD
    Title1,
    "Author",
    F6 As Replies,
    Views,
    "Latest activity"
FROM [lib://x2]
(URL IS [https://community.qlik.com/community/qlik-sense/new-to-qlik-sense/content?filterID=contentstatus%5Bpublished%5D~objecttype~objecttype%5Bthread%5D&itemView=detail&start=$(i)], html, utf8, embedded labels, table is @1);
Next i;
```

これにより、フォーラムの直近の投稿 200 個が、タイトル、投稿者、返信数、閲覧数、最新アクティビティの日時とともに、テーブルにロードされます。

Microsoft Excel スプレッドシートからのデータのロード

Qlik Sense は、Microsoft Excel スプレッドシートからデータを読み込むことができます。サポートされているファイル形式は、**XLS**、**XLSX**、**XLW**、**XLSM** です。データマネージャーで [データの追加] を使用するか、データロードエディタでデータを選択することができます。どちらの場合も、スプレッドシートファイルのシートの名前付き領域、1 つのシート、必要なシート、またはすべてのシートを選択できます。各シートは、項目の構造が同じ場合を除いて個別のテーブルとしてロードされ、構造が同じ場合は 1 つのテーブルに連結されます。



Microsoft Excel スプレッドシートをロードする場合は、**Qlik Sense** アプリのデータソースとしてスプレッドシートを使用します。つまり、**Microsoft Excel** シートは、**Qlik Sense** アプリのシートではなく、**Qlik Sense** のテーブルになります。

スプレッドシートをロードする前に、Microsoft Excel でいくつかの変更を行うと使用しやすくなります。

Microsoft Excelシートからのデータの選択

Microsoft Excel シートからデータを選択する場合には、テーブルのデータを正しく解釈するのに役立ついくつかの設定を選択できます。

テーブルデータの正しい解釈に役立つ設定

UI 項目	説明
項目名	テーブルに [埋め込まれた項目名] が含まれているか、[項目名が存在しない] が指定します。通常、Excel スプレッドシートでは、最初の行には埋め込まれた項目名が含まれます。[項目名なし] を選択すると、項目名は A,B,C... となります。
ヘッダーサイズ	テーブルヘッダーとして無視する行数を設定します。通常は、列形式になっていない一般情報が含まれる行です。

例

スプレッドシートは次のようになっています。

スプレッドシートの例

マシン:	AEJ12B	-	-
[日付]:	2015-10-05 09	-	-
タイムスタンプ	順序	演算子	生産額
2015-10-05 09:22	00122344	A	52
2015-10-05 10:31	00153534	A	67
2015-10-05 13:46	00747899	B	86

この場合、最初の 2 行を無視して、Timestamp、Order、Operator、および Yield の項目を含むテーブルをロードしたいと考えましょう。これを行うには、次の設定を使用します。

最初の 2 行を無視して項目をロードするための設定

UI 項目	説明
ヘッダーサイズ	2 これは、最初の 2 行がヘッダーデータとみなされて、ファイルをロードするときに無視されることを意味します。この場合、Machine: および Date: で開始する 2 行はテーブルデータの一部ではないため無視されます。
項目名	[埋め込まれた項目名] これは、読み込まれる最初の行が、それぞれの列の項目名として使用されることを意味します。この場合は、最初の 2 行がヘッダーデータになっているため、読み込まれる最初の行は 3 行目となります。

Qlik SenseでロードしやすくするためのMicrosoft Excelスプレッドシートの準備

Microsoft ExcelスプレッドシートをQlik Senseにロードする場合は、データロードスクリプトでさまざまな関数を使用してデータの変換やクリアを行うこともできますが、Microsoft Excelスプレッドシートファイルでソースデータを直接準備した方が効率的です。このセクションでは、最小限のスクリプトコードでスプレッドシートをQlik Senseにロードするためのヒントを紹介します。

列見出しの使用

Microsoft Excelで列見出しを使用している場合、それらの列見出しは、Qlik Senseでデータを選択するときに**[埋め込まれた項目名]**を選択すると項目名として自動的に使用されます。ラベルには改行を挿入せず、ヘッダーはシートの1行目に配置するようにしてください。

データの書式設定

コンテンツをテーブル内の生データとして設定しておく、Qlik SenseへのMicrosoft Excelファイルのロードがしやすくなります。以下はなるべく避けてください。

- 合計やカウントなどの集計。集計はQlik Senseで定義、計算できます。
- ヘッダーの重複。
- データの一部ではない追加情報(コメントなど)。Qlik Senseにファイルをロードする際にスキップできるコメント用の列を作成することをお勧めします。
- クロス集計データレイアウト。月ごとに列を1つずつ作成するのではなく、「Month」という名前の列を1つ作成し、各月につき1行割り当て、合計12の行に同じデータを記述するようにしてください。こうすることで、Qlik Senseでクロス集計の表示が可能となります。
- 例えば、「Department A」という名前の後にDepartment Aに関する行が続く(中間ヘッダーがある)とします。このような場合、代わりに「Department」という名前の列を作成し、適切な部署名を入力します。
- セルの結合。セルは結合せず、各セルにセル値をリストしてください。
- 前の値により默示的に値が定義される空白セル。すべてのセルにデータ値が含まれるよう、同じ値が繰り返されている場合もその値を入力する必要があります。

名前付き領域の使用

シートの一部のみを読み取る場合、行または列の領域を選択し、Microsoft Excelの名前付き領域として定義することができます。Qlik Senseでは名前付き領域とシートの両方からデータをロードできます。

通常、名前のあるデータとして生データを定義し、名前のあるエリア外のすべての追加コメントと説明を保存できます。この方法では、Qlik Senseへのデータロードをより簡単に行えます。

パスワード保護の削除

パスワードで保護されたファイルはQlik Senseによってサポートされないため、Qlik Senseにロードする前にスプレッドシートからパスワード保護を解除する必要があります。

Excel Binary Workbook ファイルのロード(.xlsb)

Excel Binary Workbook ファイル(.xlsb)を直接Qlik Senseにロードすることはできません。回避策は、ODBC接続を使用することです。

6.5 データベースからのデータのロード

次のコネクタを使用して、商用データベース システムから Qlik Sense にデータをロードすることができます。

- ライセンス付き ODBC ドライバーを介して、DSN 接続なしでデータベースからデータを直接ロードするために特別に開発されたコネクタ。詳しくは、「[Qlik Connectors: Database](#)」を参照してください。
- Microsoft ODBC インターフェースまたは OLE DB を使用するコネクタ。ODBC を使用するには、DBMS に対応したドライバーをインストールし、Windows の [コントロール パネル] の [ODBC データソース アドミニストレーター] でデータベースを Microsoft ODBC データソースとして構成する必要があります。

ODBC データベースからのデータのロード

データベースからデータをロードする方法は、2 種類あります。

Qlik のライセンス付き ODBC ドライバーのいずれかを介してデータベースに直接接続するには、Qlik コネクタのヘルプサイトで Database コネクタに関する説明を参照してください。


詳細については、[ODBC Connector Package](#) を参照してください。

Qlik のライセンス付き ODBC ドライバーは、次のデータベースをサポートしています。

- Amazon Redshift
- Apache Drill
- Apache Hive
- Apache Phoenix
- Apache Spark
- Azure SQL
- Cloudera Impala
- Google BigQuery
- Microsoft SQL Server
- MongoDB
- MySQL Enterprise
- Oracle
- PostgreSQL
- Presto
- Sybase ASE
- Teradata

Microsoft ODBC インターフェースを使用するには、次の手順に従います。

1. アクセスしたいデータベースに対応する ODBC データソースが必要です。これは、Windows [コントロール パネル] の [ODBC データソース アドミニストレータ] で構成されます。データソースがない場合は追加して、例えば Microsoft Access データベースを指すように構成する必要があります。
2. データロード エディターを開きます。

3. [ODBC] データ接続を作成し、ステップ 1 の ODBC 接続を指すようにします。
4. データ接続の  をクリックして、データ選択ダイアログを開きます。

これでデータベースからデータを選択し、データのロードに必要なスクリプトコードを挿入できます。

ODBC

ODBC データソースからのデータのロード

- 最も一般的な ODBC ソースに対応する、Qlik ODBC Connector Package の Database コネクタを使用できます。これにより、Microsoft Windows [ODBC データソースアドミニストレータ] を使用せずに、Qlik Sense でデータソースを定義できます。ODBC Connector Package 内の Qlik ライセンス付き ODBC ドライバのいずれかを介してデータベースに直接接続するには、Qlik Connectors のヘルプサイトで Database コネクタに関する説明を参照してください。
- 問題の DBMS の ODBC ドライバーをインストールし、データソース DSN を作成できます。このオプションはこのセクションで説明します。

Qlik Sense により ODBC を介して DBMS (Database Management System) にアクセスできません。



[接続の新規作成 (ODBC)] ダイアログには、設定されている [ユーザー DSN] 接続が表示されます。Qlik Sense Desktop を使用している場合、DSN 接続のリストには、ODBC Connector Package に含まれる ODBC ドライバが表示されます。これらは、名前に付けられる「Qlik-」によって識別されます (例: Qlik-db2)。これらのドライバは、新しい ODBC 接続の作成には使用できません。これらのドライバは、ODBC Connector Package のデータベースコネクタによってのみ使用されます。ODBC Connector Package の ODBC ドライバは、サーバー環境で Qlik Sense を使用している場合は表示されません。

別の方法として、データベースのデータを Qlik Sense で読み取り可能なファイルにエクスポートすることもできます。

通常、一部の ODBC ドライバは、Microsoft Windows とともにインストールされます。また、ソフトウェア販売店のウェブサイトや DBMS 製造元から追加ドライバを購入することもできます。無料で再配布されているドライバもあります。



サーバー環境内では、Microsoft Access Database ドライバーに制限があります。問題を避けるために、SQL Server Express Edition を使用してください。

[Microsoft Access Database Engine 2016 再頒布可能コンポーネント](#)

ここでは、クライアントコンピュータ上にある ODBC インターフェースについて説明します。ネットワークサーバー上のマルチユーザーリレーショナルデータベースへのアクセスに ODBC を使用する場合は、クライアントがサーバー上のデータベースにアクセスできるように追加の DBMS ソフトウェアが必要となる場合があります。必要なソフトウェアの詳細については、DBMS サプライヤに問い合わせてください。

ODBC データ接続の設定

ODBC データ接続の設定

UI 項目	説明
ユーザー DSN システム DSN	どのタイプの DSN を接続するかを選択します。 [ユーザー DSN] ソースで 32 ビット ドライバを使用する場合は、[32 ビット互換モード] を指定する必要があります。 [システム DSN] 接続は、[32 ビット] または [64 ビット] によりフィルター可能です。
Single Sign-On	または SAP HANA データソースへの接続時には、Single Sign-On (SSO) を有効にすることができます。 このオプションが選択されていない場合は、[ユーザー名] と [パスワード] で資格情報を指定しない限り、Engine Service ユーザーの資格情報が使用されます。 このオプションが選択されている場合は、Windows へのログオンでエンジン サービス ユーザーまたは [ユーザー名]/[パスワード] 資格情報が使用され、その後の SAML (SAP HANA) へのログオンでは現在のユーザーの資格情報が使用されます。
[ユーザー名]	データソースで必要な場合、接続するためのユーザー名。 Engine Service ユーザーの資格情報を使用する場合、またはデータソースで資格情報が必要とされない場合は、この項目を空白のままにします。
[パスワード]	データソースで必要な場合、接続するためのパスワード。 Engine Service ユーザーの資格情報を使用する場合、またはデータソースで資格情報が必要とされない場合は、この項目を空白のままにします。
名前	データ接続の名前。

ODBC ドライバの追加

Qlik Sense によるデータベースへのアクセスを可能にするには、DBMS (DataBase Management System) 用の ODBC ドライバが必要です。詳細については、DBMS のマニュアルを参照してください。

Qlik Sense によるデータベースへのアクセスを可能にするには、DBMS 用の ODBC ドライバが必要です。これは外部のソフトウェアです。そのため下記の手順は、すべてのベンダーのソフトウェアに当てはまらない場合があります。詳細については、DBMS のマニュアルを参照してください。

次の手順を実行します。

1. [コントロールパネル] の [管理ツール] アイコンをダブルクリックします。
2. [データソース (ODBC)] アイコンをダブルクリックします。
[ODBC データソースアドミニストレーター] ダイアログが表示されます。
3. Qlik Sense で使用するデータベースを選択します。
4. [データソース] ダイアログの [ドライバ] タブを選択します。

[**ドライバ**] タブに、インストール済みの ODBC ドライバ一覧が表示されます。使用する DBMS が含まれていない場合はドライバをインストールする必要があります。ODBC ドライバに付属するインストールプログラム (例: Microsoft ODBC インストールプログラム) を実行します。

64 ビットおよび 32 ビットバージョンでの ODBC 設定

Microsoft Windows オペレーティング システムの 64 ビットバージョンには、Microsoft Open DataBase Connectivity (ODBC) Data Source Administrator ツール (*Odbcad32.exe*) の次のバージョンが含まれています。

- *Odbcad32.exe* ファイルの 32 ビットバージョンは、`%systemdrive%\Windows\SysWOW64` フォルダにあります。
- *Odbcad32.exe* ファイルの 64 ビットバージョンは、`%systemdrive%\Windows\System32` フォルダにあります。

ODBC データソースの作成

アクセスしたいデータベース向けに ODBC データソースを作成する必要があります。データソースの作成は、ODBC のインストール時に行うことも、インストール後に行うこともできます。



データソースを作成する前に、データソースを [**ユーザー DSN**] と [**システム DSN**] (推奨) のどちらにするか決定します。ユーザーデータソースにアクセスするには、正しいユーザー認証情報が必要です。サーバーインストール環境では、他のユーザーとデータソースを共有するため、システム データソースを作成する必要があります。

次の手順を実行します。

1. *Odbcad32.exe* を開きます。
2. [**システム DSN**] タブでシステム データソースを作成します。
3. [**追加**] をクリックします。
[**データソースの新規作成**] ダイアログが表示され、インストールされている ODBC ドライバのリストが表示されます。
4. 正しい ODBC ドライバが表示されている場合は、それを選択して [**完了**] をクリックします。
選択したデータベース ドライバに固有のダイアログが表示されます。
5. [**Microsoft Access Driver (*.mdb, *.accdb)**] ドライバを選択し、[**完了**] をクリックします。



リストでこのドライバを見つけることができない場合は、Microsoft のダウンロードサイトからダウンロードしてインストールできます。

6. データソースに名前を付け、必要なパラメータを設定します。
7. データソースに *Scripting tutorial ODBC* という名前を付けます。
8. [**データベース**] で [**選択**] をクリックします。
9. [**ディレクトリ**] で、*Sales.accdb* ファイル (チュートリアル用 サンプル ファイル) が保存されている場所までナビゲートします。

10. テキストボックスに **Sales.accdb** ファイルが表示されている場合は、このファイルをクリックしてデータベースの名前にします。
11. **[OK]** を 3 回 クリックし、すべてのダイアログを閉じます。
12. **[OK]** をクリックします。

ODBC データ接続使用時のベストプラクティス

ODBC データ接続を含むアプリの移動

Qlik Sense サイトと Qlik Sense Desktop インストール環境の間でアプリを移動する場合、データ接続も含まれます。アプリに ODBC データ接続が含まれている場合、関連する ODBC データソースが新たな展開先にも存在している必要があります。これらの ODBC データソースは同じ名前と構成で、同じデータベースまたはファイルを指している必要があります。

ODBC データ接続ベースでファイルに接続する場合のセキュリティ面

ファイルベースのドライブを使用した ODBC データ接続では、接続文字列に接続済みデータファイルへのパスが表示されます。このパスは、接続の編集時にデータ選択ダイアログまたは特定の SQL クエリに表示されます。

これが心配な場合は、可能であればフォルダデータ接続を使用してデータファイルに接続することをお勧めします。

テーブル内にある大きなデータセットのプレビューの停止

大きなデータセットがあり、**[データマネージャー]** または **[データロードエディター]** に ODBC のデータソースを追加している間、プレビューが表示されることを望まない場合は、**Shift** キーを押しながら ODBC のデータ接続を選択します。



OLE DB

Qlik Sense は、外部データソース向けの OLE DB (Object Linking and Embedding, Database) インタフェース接続をサポートしています。OLE DB 経由で、多数の外部データベースにアクセスすることができます。

OLE DB データ接続の設定

OLE DB データ接続の設定

UI 項目	説明
[プロバイダ]	利用可能なプロバイダのリストから [プロバイダ] を選択します。新規接続の作成時にのみ利用できます。

UI 項目	説明
[データソース]	<p>接続する[データソース]の名前を入力します。通常はサーバー名、または一部のケースではデータベースファイルへのパスを入力しますが、これはどのOLE DBプロバイダーを使用しているかにより異なります。新規接続の作成時にのみ利用できます。</p> <p>Microsoft Office 12.0 Access Database Engine OLE DB Providerを選択した場合、完全なファイルパスを含むAccessデータベースファイルの名前を入力します。</p> <p><code>C:\Users\{user}\Documents\Qlik\Sense\Apps\Tutorial source files\Sales.accdb</code></p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;">  データソースへの接続に失敗した場合、警告メッセージが表示されます。 </div>
[接続文字列]	データソースへの接続時に使用する接続文字列。この文字列には、[プロバイダ]と[データソース]への参照が含まれます。接続の編集時にのみ利用できます。
[Windowsの統合セキュリティ]	このオプションでは、Qlik Sense サービスを実行しているユーザーの既存のWindows資格情報を使用します。
[特定のユーザー名とパスワード]	このオプションでは、データソースのログイン資格情報として、[ユーザー名]と[パスワード]が必要です。
[ユーザー名]	<p>データソースが必要な場合、接続するためのユーザー名。</p> <p>[Windows 統合セキュリティ]を使用する、またはデータソースで資格情報が必要ない場合、この項目には何も入力しないでください。</p>
[パスワード]	<p>データソースが必要な場合、接続するためのパスワード。</p> <p>[Windows 統合セキュリティ]を使用する、またはデータソースで資格情報が必要ない場合、この項目には何も入力しないでください。</p>
[ロード] [データベースを選択...]	<p>接続をテストするには、[ロード]、[データベースを選択...]の順にクリックし、データ接続の確立に使用するデータベースを指定します。</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;">  データ接続からデータを選択する場合でも、データソースの利用可能なその他すべてのデータベースを引き続き使用できます。 </div>
[名前]	データ接続の名前。

OLE DB データ接続ベースでファイルに接続する場合のセキュリティ面

ファイルベースのドライバを使用したOLE DBデータ接続では、接続文字列に接続済みデータファイルへのパスが表示されます。このパスは、接続の編集時にデータ選択ダイアログまたは特定のSQLクエリに表示されます。

問題がある場合、可能な際にはフォルダデータ接続を使用してデータファイルに接続することが推奨されます。

テーブル内にある大きなデータセットのプレビューの停止

大きなデータセットがあり、[データマネージャー] または [データロードエディター] に OLE DB のデータソースを追加している間、プレビューが表示されることを望まない場合は、Shift キーを押しながら OLE DB のデータ接続を選択します。

データベースのロジック

Qlik Sense ロジックには、データベース アプリケーションのテーブルが同時に複数含まれている場合があります。1 つまたは複数のテーブルに特定の項目が存在する場合、これをキー項目としてテーブルを論理的にリンクします。

値が選択されていると、その選択と互換性のあるすべての値が選択可能な値として表示されます。その他の値はすべて、除外値として表示されます。

複数の項目で値が選択されると、論理積ANDと見なされます。

同じ項目で複数の値が選択されると、論理和ORと見なされます。

単一の項目内の選択に論理積ANDを設定できる場合もあります。

6.6 を利用した膨大なデータセットへのアクセスDirect Discovery

Direct Discovery では、シンプルなスター スキーマ構造を持つ特定の SQL ソースからビッグ データセットをロードし、それらをインメモリデータと結合できます。

インメモリデータと**Direct Discovery** データ上での選択が可能で、関連付けを示す **Qlik Sense** の色 (緑、白、およびグレー) でデータセットの連想結果を見ることができます。特定のビジュアライゼーションで両方のデータセットからデータを分析できますが、このアプローチには多数の制限があります。この設計は、リアルタイムで解決できるようにはなっていません。



この制限を取り除くために **Direct Discovery** の新しい開発に着手する予定はありません。

Qlik Sense の On-Demand アプリでは、ビッグ データソースのロードおよび分析に、より柔軟にアプローチできるようになっています。

クエリを集計して膨大なデータセットとインメモリデータをシームレスに結び付け、追加のソースデータにアクセスすることで、**Qlik Sense** インメモリデータモデルの連想機能を拡張します。**Direct Discovery** は、ビジネスユーザーがビッグ データソース上で制約を受けずに連想分析できるように、機能が向上されています。インメモリデータと**Direct Discovery** データ上での選択が可能で、関連付けを示す **Qlik Sense** の色 (緑、グレー、白) でデータセットの連想結果を見ることができます。ビジュアライゼーションでは、両方のデータセットから同時にデータを分析できます。

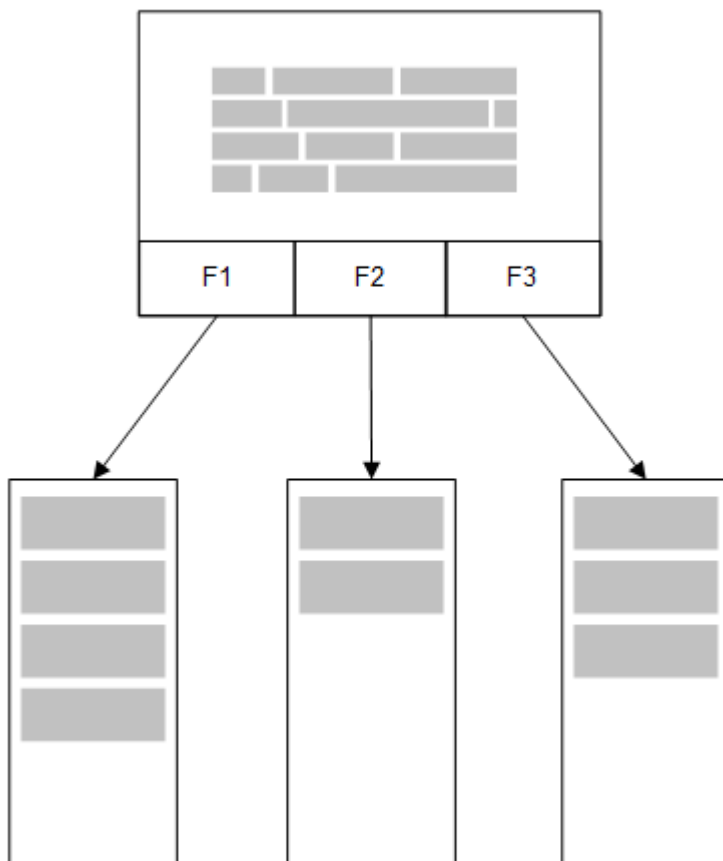
Direct Discovery のデータは **DIRECT QUERY** という特殊なスクリプト構文を用いて選択します。Direct Discovery 構造が完成すると、Direct Discovery 項目とインメモリデータを使用して Qlik Sense オブジェクトを作成できるようになります。Qlik Sense オブジェクトで Direct Discovery 項目を使用すると、外部データソースで自動的に SQL クエリが実行されます。

On-Demand アプリは、大規模データセットにアクセスするための別の方法を提供します。Direct Discovery とは異なり、On-Demand アプリは、メモリ内にホストされた潜在的サブセットに対し、Qlik Sense のすべての機能を提供します。

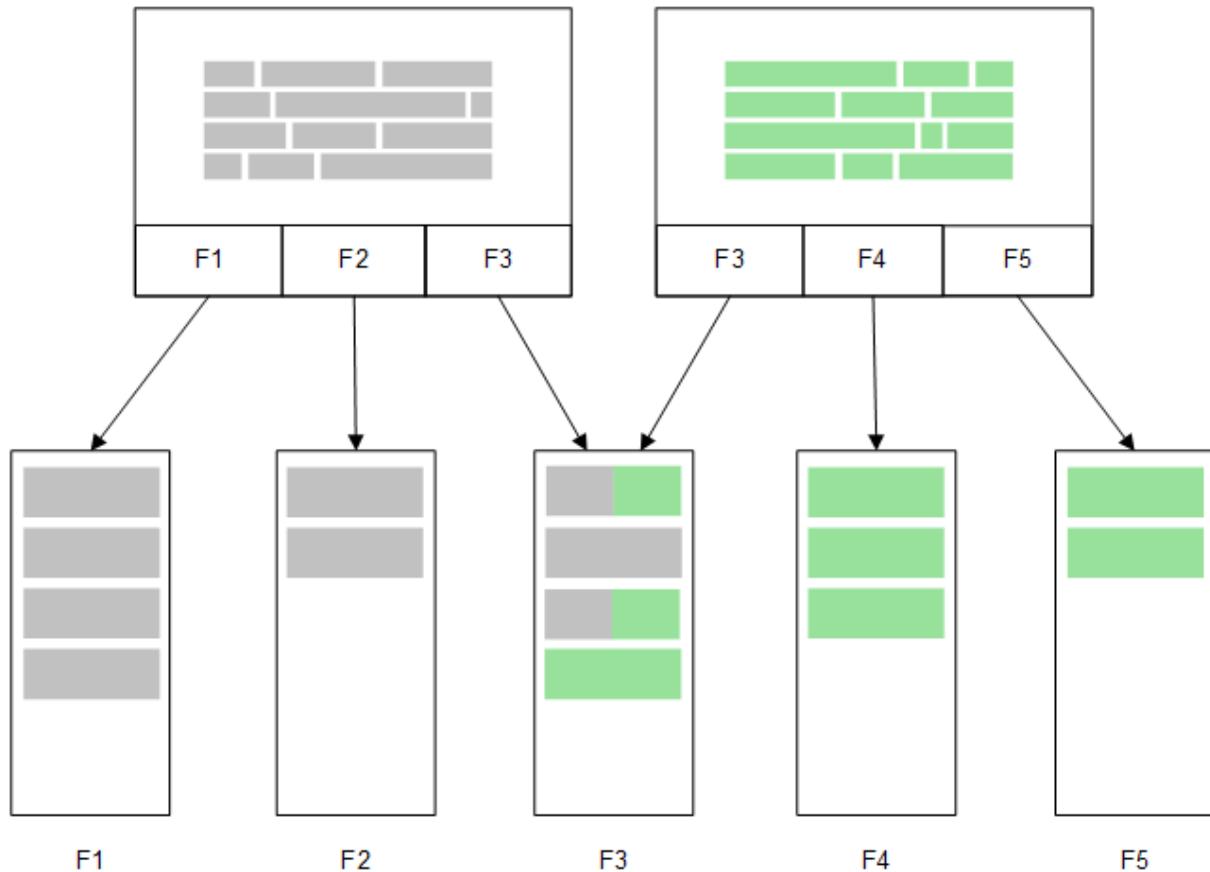
Direct Discovery とインメモリデータとの違い

インメモリモデル

Qlik Sense のインメモリモデルでは、ロードスクリプトのテーブルから選択された項目のユニーク値が項目構造にすべてロードされ、それと同時に連想データはテーブルにロードされます。項目データおよび連想データは、すべてメモリに保持されています。

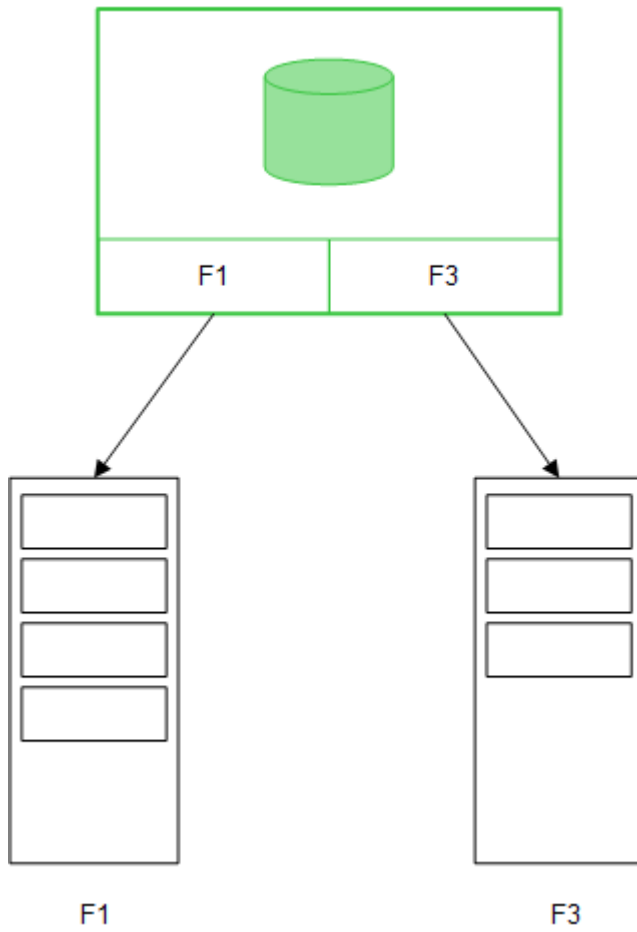


メモリにロードされたもう一つの関連テーブルは、共通の項目を共有することがあります。そしてそのテーブルは、共通の項目に新しい一意の値を追加したり、既存値を共有したりする場合があります。

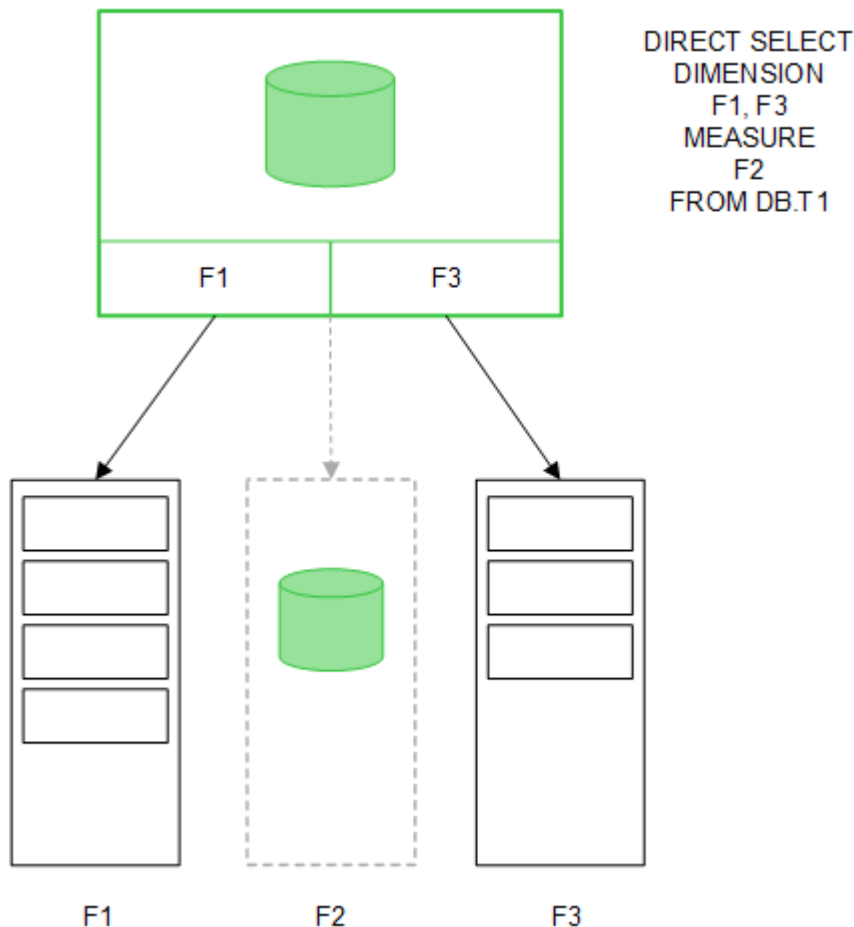


Direct Discovery

テーブル項目が **Direct Discovery LOAD** ステートメント (**Direct Query**) でロードされると、**DIMENSION** 項目のみで類似テーブルが作成されます。インメモリ項目では、**DIMENSION** 項目のユニーク値がメモリにロードされます。ただし、項目間の関係性はデータベースに残ります。



MEASURE 項目値もまた、データベースに残ります。



Direct Discovery 構造が確立すると、特定の ビジュアライゼーション オブジェクトと共に Direct Discovery 項目を使用できます。そして、インメモリ項目を使った関連付けのためにもこの項目を使用できます。Direct Discovery 項目が使用されている場合、Qlik Sense は適切な SQL クエリを自動作成し、外部データ上で実行します。選択を行うと、データベースのクエリの WHERE 条件で Direct Discovery 項目の関連データ値が使用されます。

Direct Discovery 項目を含むビジュアライゼーションは、各選択条件に従い再計算されます。そして、Qlik Sense で作成された SQL クエリを実行することにより、元のデータベース テーブルに計算結果が表示されます。演算実行条件の機能を使用すれば、ビジュアライゼーションを再計算するタイミングを指定できます。条件が合うまで、Qlik Sense はビジュアライゼーションを再計算するクエリを送信しません。

インメモリ項目と Direct Discovery 項目 とのパフォーマンスの違い

インメモリでの処理スピードは、ソースデータベースでの処理スピードよりも常に速くなります。Direct Discovery のパフォーマンスは、Direct Discovery クエリを処理するデータベースを稼働しているシステムのパフォーマンスにより異なります。

Direct Discovery のベストプラクティスを引き出す標準的なデータベースおよびクエリを使用することができます。パフォーマンスの調整はすべて、ソースデータベース上で行われます。Direct Discovery は、Qlik Sense アプリからのクエリのパフォーマンス調整には対応していません。ただし、接続プーリング機能を使用することで、データベースへの非同期・同時呼び出しを行うことはできます。プーリング機能を設定するロードスクリプト構文は以下の通りです。

```
SET DirectConnectionMax=10;
```

Qlik Sense キャッシュにより、ユーザー エクスペリエンス全体が改善されます。以下 キャッシュおよび *Direct Discovery* (page 219) を参照してください。

一部項目の関連付けを解除することでも、**DIMENSION** 項目を含む **Direct Discovery** のパフォーマンスを改善できます。これは、**DIRECT QUERY** 上で **DETACH** キーワードを用いて実行されました。関連付けが解除された項目では、関連付けのクエリは実行されませんが、フィルターの一部として残っているので、選択時間を短縮できます。

Qlik Sense のインメモリ項目と **Direct Discovery** の **DIMENSION** 項目の両方の全データがメモリ内に保持されている間、項目へのロード方法の違いは、メモリへのロード速度に影響を及ぼします。同じ値を持つ複数のインスタンスが存在する場合、**Qlik Sense** のインメモリ項目は、項目値のコピーを1つだけ保持します。ただし、全項目データをロードして、重複データが抽出されます。

DIMENSION 項目も項目値のコピーを1つだけ保持しますが、重複値はメモリへのロードに先立ち、データベースで抽出されます。**Direct Discovery** を使用する際と同様に、大量のデータを取り扱う場合は、インメモリ項目の **SQL SELECT** ロードよりも、**DIRECT QUERY** ロードを実行した方が素早くデータをロードできます。

データ インメモリとデータベース データとの違い

インメモリデータで関連付けを行う場合、**DIRECT QUERY** では大文字と小文字が区別されます。**Direct Discovery** は、クエリ対象のデータベースフィールドについて大文字と小文字の区別を行って、ソースデータベースからデータを選択します。大文字と小文字を区別しないデータベースフィールドの場合、**Direct Discovery** のクエリは、インメモリのクエリが返さないデータを返す場合があります。大文字と小文字を区別しないデータベースに以下のデータが存在する場合、値 "Red" の **Direct Discovery** クエリは以下 4 つの行をすべて返します。

テーブルの例

ColumnA	ColumnB
赤色	1
赤色	2
赤色	3
赤色	4

一方、"Red," のインメモリの選択は、以下のみを返します。

Red two

Qlik Sense は、データベースが合致しない選択データで一致する値が生成されるまで、データを正規化します。その結果、インメモリクエリは、**Direct Discovery** クエリよりも一致度が高い値を生成する場合があります。例えば、以下のテーブルでは、前後のスペースの位置により、数値 "1" に対する値が変化します。

数値の前後のスペースの位置が異なるために、数値「1」で異なる値を持つテーブル

ColumnA	ColumnB
'1'	space_before

ColumnA	ColumnB
'1'	no_space
'1'	space_after
'2'	2

標準 Qlik Sense インメモリにデータがある ColumnA の [フィルター パネル] で "1" を選択すると、最初の 3 行が以下に関連付けられます。

Associated rows

ColumnA	ColumnB
'1'	space_before
'1'	no_space
'1'	space_after

[フィルター パネル] に Direct Discovery データが含まれる場合、"1" のデータ選択は "no_space" のみを関連付ける可能性があります。返される Direct Discovery データとの一致は、データベースにより異なります。例えば、"no_space" のみ返す場合もあれば、SQL Server のように "no_space" と "space_after" を返す場合もあります。

キャッシュおよび Direct Discovery

Qlik Sense のキャッシュは、クエリの選択状態および関連付けられたクエリの結果をメモリに保存します。同じ種類の選択が行われると、Qlik Sense は、ソースデータのクエリを実行せずに、キャッシュのクエリを利用します。異なる選択を行うと、データソースで SQL クエリを実行します。キャッシュされている結果は、ユーザー間で共有されます。

1. 最初の選択を適用した場合
SQL は、基底のデータソースに渡されます。
2. 選択をクリアし、最初の選択と同じ選択を適用した場合
キャッシュ結果が返され、SQL は基底データソースには渡されません。
3. 異なる選択を適用した場合
SQL は、基底のデータソースに渡されます。

DirectCacheSeconds システム変数を使用して、キャッシュに制限時間を設定できます。制限時間が経過すると、Qlik Sense は、前の選択で生成された Direct Discovery クエリの結果のキャッシュをクリアします。そして、Qlik Sense は選択用にソースデータのクエリを実行し、指定した制限時間でキャッシュを再び作成します。

Direct Discovery クエリ結果のキャッシュ時間はデフォルトで 30 分間に設定されていますが、**DirectCacheSeconds** のシステム変数が使用されている場合は該当しません。

Direct Discovery 項目の種類

Direct Discovery には、次の 3 種類のデータ項目があります。DIMENSION、MEASURE、DETAIL。この 3 種類の項目は、ロードスクリプトの **Direct Query** ステートメントを使用して Direct Discovery の選択が行われた際に、データ項目に設定されます。

Direct Discovery 項目はすべてインメモリ項目との組み合わせが可能です。通常、軸として使用する離散値を含む項目は、DIMENSION キーワードでロードする必要がありますが、集計関数でのみ使用する数値データは、MEASURE 項目を使用してロードします。

以下の表には、Direct Discovery 項目の種類の特性と使用方法が要約されています。

Direct Discovery;項目の種類

項目の種類	インメモリ	関連付け	チャート式での使用
DIMENSION	はい	はい	はい
MEASURE	いいえ	いいえ	はい
DETAIL	いいえ	いいえ	いいえ

DIMENSION 項目

DIMENSION 項目はメモリにロードされ、インメモリデータと Direct Discovery 項目のデータの関連付けに使用できます。また、Direct Discovery DIMENSION 項目は、チャートの軸の値を定義する際にも使用します。

MEASURE 項目

MEASURE 項目が「メタレベル」で認識される一方で、MEASURE 項目はメモリにロードされず、データモデルビューアーにも表示されません。これは MEASURE 項目のデータ集計をメモリではなく、データベースで行うようにするためです。しかし、MEASURE 項目は数式で使用され、数式構文は変更されません。その結果、データベースの Direct Discovery 項目の使用は、エンドユーザーには透過的に行われます。

MEASURE 項目で使用可能な集計関数は次のとおりです。

- Sum
- Avg
- Count
- Min
- Max

DETAIL 項目

DETAIL 項目には、表示するがチャート式では使用しない情報や詳細が含まれています。通常、DETAIL として指定されている項目には、実際に集計することができないデータ(コメントなど)が含まれています。

どのような項目も DETAIL 項目に指定できます。

でサポートされているデータソースDirect Discovery

Qlik SenseDirect Discovery では、以下のデータソースを利用できます (32 ビット接続および 64 ビット接続)。

- ODBC/OLEDB データソース - すべての ODBC/OLEDB ソースは、SQL Server、Teradata、Oracle を含めサポートされています。
- SQL 対応のコネクタ - SAP SQL コネクタ、SQL 準拠のデータストア用カスタム QVX コネクタ。

32 ビットおよび 64 ビット接続の両方に対応しています。

SAP

SAP の場合、Direct Discovery は Qlik SAP SQL Connector のみと併用可能で、**SET** 変数には以下のパラメータが必要となります。

```
SET DirectFieldColumnDelimiter=' ';  
SET DirectIdentifierQuoteChar=' ';
```

SAP は OpenSQL を使用しており、コンマでなくスペースで列を区切ります。そのため上記の SET ステートメントは、ANSI SQL と OpenSQL 間の相違に対応するために代替が行われます。

Google Big Query

Direct Discovery は Google Big Query と共に使用できますが、SET 変数には以下のパラメータが必要となります。

```
SET DirectDistinctSupport=false;  
SET DirectIdentifierQuoteChar='[]';  
SET DirectIdentifierQuoteStyle='big query'
```

Google Big Query は **SELECT DISTINCT** や引用符で囲まれた列/テーブル名をサポートしていません。また、**[]** を使用する非 ANSI 引用符の設定を採用しています。

MySQL また、Microsoft Access

Direct Discovery は MySQL、Microsoft Access と共に使用できますが、これらのソースで引用符文字が使用されているため、SET 変数に次のパラメータが必要となる場合があります。

```
SET DirectIdentifierQuoteChar='`';
```

DB2、Oracle、および PostgreSQL

Direct Discovery は DB2、Oracle、PostgreSQL と共に使用できますが、これらのソースで引用符文字が使用されているため、SET 変数に次のパラメータが必要となる場合があります。

```
SET DirectIdentifierQuoteChar='\"';
```

Sybase また、Microsoft SQL Server

Direct Discovery は Sybase、Microsoft SQL Server と共に使用できますが、これらのソースで引用符文字が使用されているため、SET 変数に次のパラメータが必要となる場合があります。

```
SET DirectIdentifierQuoteChar='[]';
```

Apache Hive

Direct Discovery は、Apache Hive と共に使用できますが、これらのソースで引用符文字が使用されているため、SET 変数に次のパラメーターが必要となる場合があります。

```
SET DirectIdentifierQuoteChar='';
```

Cloudera Impala

Direct Discovery は、Cloudera Impala と共に使用できますが、これらのソースで引用符文字が使用されているため、SET 変数に次のパラメーターが必要となる場合があります。

```
SET DirectIdentifierQuoteChar='[]';
```

Cloudera Impala Connector を Qlik ODBC Connector Package で使用する場合、このパラメーターが必要です。DSN から ODBC を使用する場合は不要なことがあります。

を使用する際の制限事項 Direct Discovery



この制限を取り除くために *Direct Discovery* の新しい開発に着手する予定はありません。

対応しているデータ型

Direct Discovery はすべてのデータ型に対応しているわけではありません。特定のソースデータ形式を Qlik Sense で定義する必要があることもあります。"SET Direct...Format" 構文を使用して、ロードスクリプトでデータ形式を定義します。以下の例は、Direct Discovery のソースとして使用される、ソースデータベースの日付書式の定義方法を示しています。

```
SET DirectDateFormat='YYYY-MM-DD';
```

また、生成された SQL ステートメントで Direct Discovery が通貨の値の形式を制御する方法についての、次の 2 つのスクリプト変数もあります。

```
SET DirectMoneyFormat (default '#.0000')
SET DirectMoneyDecimalSep (default '.')
```

この 2 つの変数の構文は、**MoneyFormat** および **MoneyDecimalSep** の構文と同じですが、使用する際には次の 2 つの重要な違いがあります。

- これは表示形式ではないので、通貨記号や桁区切りは含まれません。
- 既定値はロケールによって引き出されるのではなく、値に組み込まれています。(ロケール固有の書式設定には通貨記号が含まれています。)

Direct Discovery では、SQL Server などの一部のデータベースで必要とされる、拡張された文字列リテラル (N'<extended string>') に対して SQL 標準形式を使用することで、拡張 Unicode データを選択することができます。この構文は、スクリプト変数 **DirectUnicodeStrings** を用いることで Direct Discovery での使用が可能になります。この変数を "true" に設定すると、文字列リテラルの前で "N" を使用できるようになります。

セキュリティ

セキュリティにおけるベストプラクティスに影響を与える可能性のある次の動作を、**Direct Discovery** を使用する際に考慮に入れる必要があります。

- **Direct Discovery** 機能を伴う同一のアプリを使用するすべてのユーザーは、同じ接続を使用します。パスワード認証やユーザーごとの資格認証には対応していません。
- セクションアクセスはサーバーモードでのみサポートされています。
- セクションアクセスは、高基数の結合には対応していません。
- **NATIVE** キーワードの数式で、データベースのカスタム **SQL** ステートメントを実行できます。そのため、ロードスクリプトで設定されているデータベース接続は、データベース読み取り専用のアクセス権を持つアカウントを使用する必要があります。
- **Direct Discovery** にはログ機能はありませんが、**ODBC** のトレース機能を使用できます。
- データベースにクライアントの要求が殺到することがあり、
- その場合、サーバーログファイルから詳細なエラーメッセージが発行される可能性があります。

サポート対象外の Qlik Sense 機能

Direct Discovery の双方向的な **SQL** 構文特有の性質により、一部の機能には対応していません。

- 高度な計算 (**Set** 分析、複雑な数式)
- 計算軸
- **Direct Discovery** 項目を使用するオブジェクトの比較分析 (並列ステート)
- **Direct Discovery** の **MEASURE** および **DETAIL** 項目は、スマート検索には対応していません
- **Direct Discovery** **DETAIL** 項目の検索
- **Direct Discovery** テーブルにアクセスしているアプリケーションからのバイナリロード
- **Direct Discovery** テーブルの合成キー
- スクリプトにおけるテーブルの名前付けは、**Direct Discovery** テーブルには適用されません。
- ロードスクリプトで **DIRECT QUERY** キーワードの後にあるワイルドカード記号*

(**DIRECT QUERY** *)

- **LONG** データタイプ列のある **Oracle** データベース テーブル。
- 科学的記数法における、[-9007199254740990, 9007199254740991] の範囲外の大きな整数。このような整数は、丸めエラーや定義されていない動作を引き起こす可能性があります。
- **Snowflake** データベース スキーマ
- データマネージャーでのデータの準備
- **Qlik Sense Enterprise SaaS**
- **Microsoft Excel** にダウンロード
- オフライン モバイル **iOS** アプリ
- 高度な分析統合
- エクステンション
- **Qlik GeoAnalytics**
- マスター軸およびマスターメジャーへの色の割り当て

- Qlik Sense Enterprise on Windows June 2017 以降に含まれている新しいビジュアライゼーション
- SQL 以外のソースおよび SQL 以外のステートメント(例: SAP HANA の PLACEHOLDER 関数)
- 以下のコネクタには対応していません。
 - Qlik Salesforce コネクタ
 - Qlik REST コネクタ
 - Qlik Web コネクタ
 - SAP NetWeaver で使用する Qlik コネクタ
- Direct Discovery クエリで生成された SQL の最適化。
- インメモリテーブルとの高基数の結合によって、データソースの SQL バッファ制限を超える大きな IN 節が生成される可能性があります。
- Qlik Visualization および Dashboard バンドル オブジェクト
- インサイトアドバイザー
- アラート
- ダイナミック ビュー
- カスタム ツール ヒント

のマルチテーブル サポート Direct Discovery

Direct Discovery を使用すると、ANSI SQL 結合機能を使って複数のテーブルやビューをロードできます。

単一チャートの中のメジャーは、すべて、Qlik Sense 内の同じ論理テーブルから派生したものでなければなりません。派生元には、join ステートメントを介してリンクされたソースからの複数のテーブルの組み合わせを使用できます。ただし、軸については、同じチャートの中で他の複数のテーブルから取得した複数の軸を使用できます。

例えば、Where 節または Join 節のいずれかを使用して、Direct Discovery でロードしたテーブルをリンクできます。

- 大きなデータセットを使用するメモリシナリオでは、Direct Discovery を 1 事実/複数軸で展開できます。
- Direct Discovery は、以下の基準のいずれかを満たす複数のテーブルで使用できます。
 - 結合に使用するキー項目の濃度が低い。
 - 結合に使用するキー項目の濃度が高く、DirectEnableSubquery が true に設定されており、すべてのテーブルが Direct Discovery で結合されている。
- Direct Discovery は、すべてのテーブルが Direct Discovery 形式の第 3 正規形シナリオでの展開には適していません。

Where 節を使用した Direct Discovery テーブルのリンク

このスクリプト例では、AW2012 データベースからデータをロードします。Product と ProductSubcategory の 2 つのテーブルは、共通の ProductSubCategoryID 項目を使用して Where 節でリンクされています。

```
Product_Join:
DIRECT QUERY
DIMENSION
  [ProductID],
  [AW2012].[Production].[Product].[Name] as [Product Name],
  [AW2012].[Production].[ProductSubcategory].[Name] as [Sub Category Name],
  Color,
```



```

[AW2012].[Production].[Product].ProductSubcategoryID as [SubcategoryID]
MEASURE
  [ListPrice]
FROM [AW2012].[Production].[Product],
     [AW2012].[Production].[ProductSubcategory]
WHERE [AW2012].[Production].[Product].ProductSubcategoryID =
      [AW2012].[Production].[ProductSubcategory].ProductSubcategoryID ;

```

Join On 節を使用した Direct Discovery テーブルのリンク

Direct Discovery テーブルは、**Join On** 節を使用してリンクすることもできます。このステートメント例では、SalesOrderID 項目を介して SalesOrderHeader テーブルを SalesOrderDetail テーブルに結合し、さらに Customer ID 項目を介して Customer テーブルを SalesOrderHeader テーブルに結合します。

この例では、同じ論理テーブルからメジャーを作成するため、それらのメジャーを同じチャートに使用できます。例えば、SubTotal と OrderQty をメジャーとしてチャートを作成することができます。

Sales_Order_Header_Join:

```

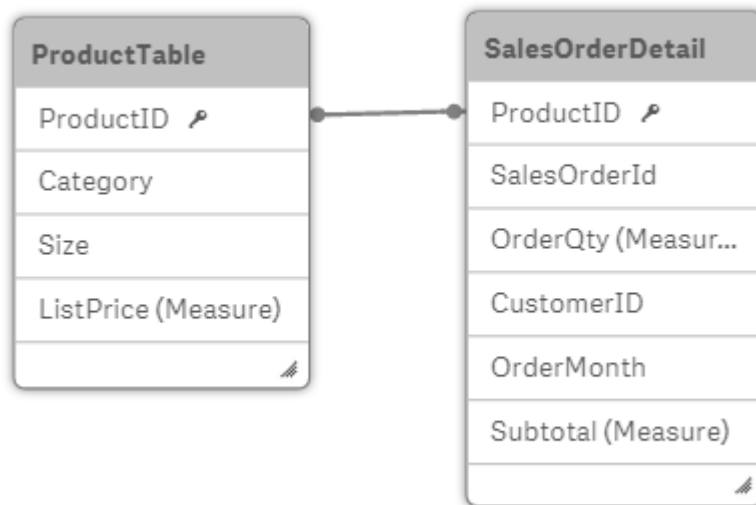
DIRECT QUERY
DIMENSION
  AW2012.Sales.Customer.CustomerID as CustomerID,
  AW2012.Sales.SalesOrderHeader.SalesPersonID as SalesPersonID,
  AW2012.Sales.SalesOrderHeader.SalesOrderID as SalesOrderID,
  ProductID,
  AW2012.Sales.Customer.TerritoryID as TerritoryID,
  OrderDate,
  NATIVE('month([OrderDate])') as OrderMonth,
  NATIVE('year([OrderDate])') as OrderYear
MEASURE
  SubTotal,
  TaxAmt,
  TotalDue,
  OrderQty
DETAIL
  DueDate,
  ShipDate,
  CreditCardApprovalCode,
  PersonID,
  StoreID,
  AccountNumber,
  rowguid,
  ModifiedDate
FROM AW2012.Sales.SalesOrderDetail
JOIN AW2012.Sales.SalesOrderHeader
ON (AW2012.Sales.SalesOrderDetail.SalesOrderID =
    AW2012.Sales.SalesOrderHeader.SalesOrderID)
JOIN AW2012.Sales.Customer
ON(AW2012.Sales.Customer.CustomerID =
    AW2012.Sales.SalesOrderHeader.CustomerID);

```

でのサブクエリの使用 Direct Discovery

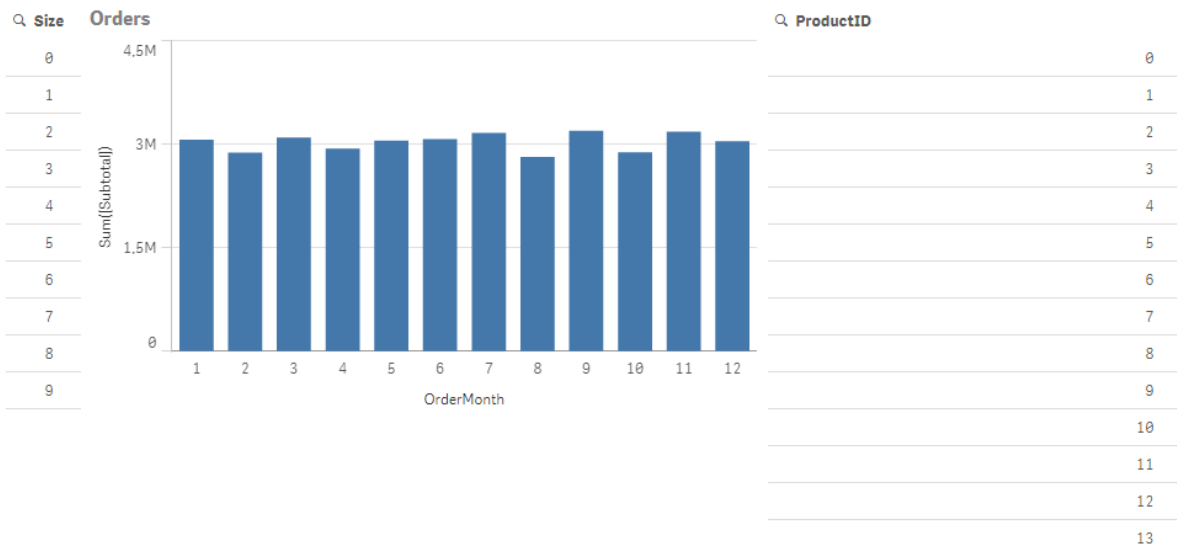
テーブルを結合するキー項目の濃度が高い場合、つまり、キー項目が多くのユニーク値を含む場合、Qlik Sense での選択によっては、**WHERE key_field IN** 節に多くの値が含まれることがあるため、非常に大きな SQL ステートメントが生成される可能性があります。この場合、Qlik Sense がサブクエリを作成するようにして問題を解決することができます。

説明のために、商品テーブル (ProductTable) が販売注文テーブル (SalesOrderDetail) に商品 ID (ProductID) を使用してリンクされており、両方のテーブルが Direct Discovery モードで使用されている例を利用します。



OrderMonth が軸、Sum(Subtotal) がメジャーで、フィルター ボックスで Size を選択できるチャートを作成します。

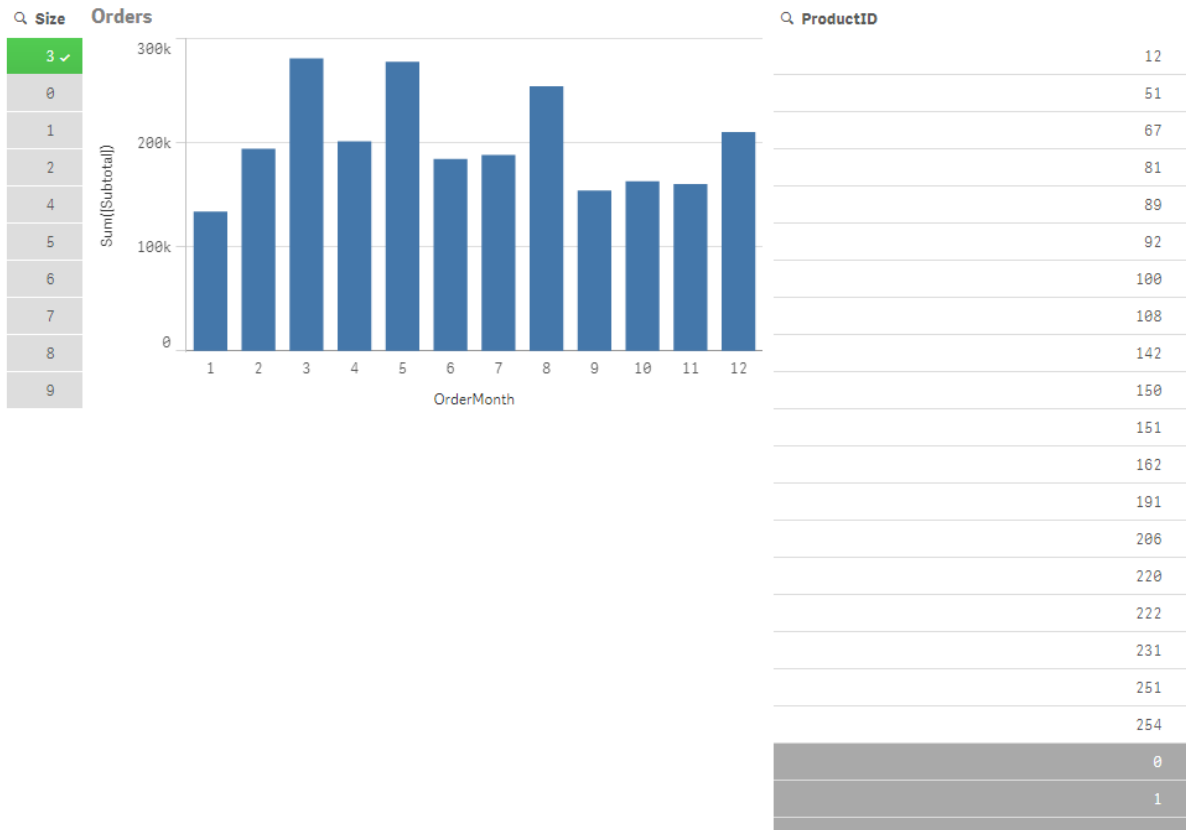
Orders



シナリオ 1: 低濃度

このシナリオでは、商品テーブルには、266 のユニークな商品が含まれ、その数は多くはありません。Size で選択を行うと、Direct Discovery が、選択されたサイズと一致する商品 ID (この場合は、19 商品) を含む WHERE ProductID IN 節を使用して、データを返す SQL ステートメントを生成します。

Orders



生成される SQL は、次のようになります。

```
SELECT ProductID, month([OrderDate]), SUM(OrderQty), SUM(SubTotal)
FROM SalesTable
WHERE ProductID IN ( 12, 51, 67, 81, 89, 92, 100, 108, 142, 150, 151, 162, 191, 206, 220, 222,
251, 254)
GROUP BY ProductID, month([OrderDate])
```

シナリオ 2: サブクエリの使用

サンプル例に多くのユニーク商品、たとえば 20,000 商品が含まれている場合、軸フィルター Size を選択したとすると、何千もの商品 ID が含まれる WHERE ProductID IN 節を持つ SQL ステートメントが生成されることになります。その結果、ステートメントが大きすぎて、メモリやパフォーマンスの制限または問題で、データソースでステートメントを処理できなくなる場合もあります。

これを解決するには、DirectEnableSubquery を true に設定して、Qlik Sense でサブクエリが作成されるようにします。生成される SQL ステートメントは、次のようになります。

```
SELECT ProductID, month([OrderDate]), SUM(OrderQty), SUM(SubTotal)
FROM SalesTable
```

```
WHERE ProductID IN  
( SELECT DISTINCT "AW2012"."dbo"."PRODUCT"."PRODUCTID" WHERE "AW2012"."dbo"."PRODUCT"."SIZE"  
  IN (3))  
GROUP BY ProductID, month([OrderDate])
```

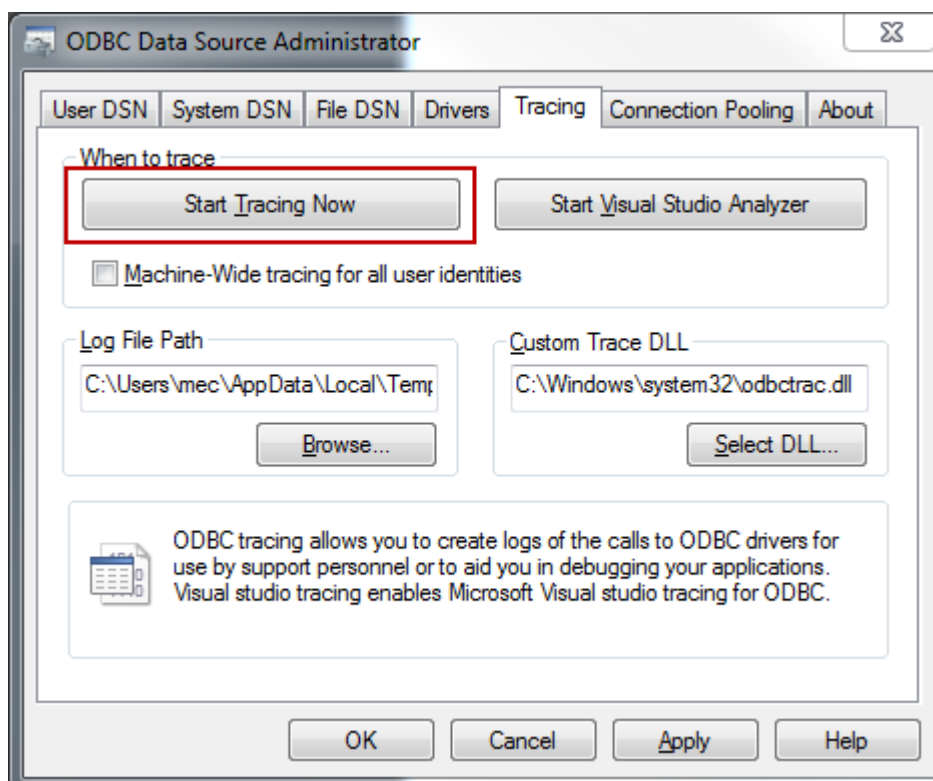
WHERE ProductID IN 節のサイズが、選択の結果としてのキーの数の影響を受けることはありません。

サブクエリを使用する場合は、次の制限が適用されます。

- サブクエリ構文は、別のテーブルからのデータを使用してチャートのフィルタリングをする必要のあるデータを
 選択した場合のみ、呼び出されます。
- キーの数ではなく、キー内のデータの量が、決定要因になります。
- サブクエリは、関係するすべてのテーブルが **Direct Discovery** モードの場合のみ、呼び出されます。メモリ
 モードに含まれるテーブルからのデータを使用して、チャートをフィルタリングする場合は、**IN** 節が生成され
 ます。

Direct Discovery アクセスの ログ

データソースに渡される **Direct Discovery SQL** ステートメントは、データベース接続のトレース ファ
イルに記録されます。標準的な ODBC 接続の場合、**[ODBC データソース管理者]** がトレースを
開始します。



結果のトレースファイルには、ユーザーの選択内容やインタラクションを通じて生成された **SQL** ステートメントの
詳細が記されています。

7 データモデルの表示および変換

データモデルビューアーは、アプリのデータ構造の概要です。テーブルや項目に関する詳細なメタデータを表示できます。軸とメジャーをデータ項目から作成することもできます。

ナビゲーションバーの[準備]タブにある[データモデルビューアー]をクリックし、データモデルビューアーを開きます。

各データテーブルはボックスとして表示され、各ボックスにはテーブル名とすべての項目が含まれています。テーブル間の関係は線で示されます(点線は循環参照を表しています)。テーブルまたは項目を選択すると関係がハイライトされるため、項目とテーブルの関連を一目で把握できます。[Q]をクリックすると、特定のテーブルと項目を検索できます。

データモードでは、アプリのデータ構造の概要が表示されます。



Sales	Sales	Sales Rep Number	Customer Number	%KEY	Cost	Date	GrossSales	Invoice Date	Invoice Number	Item Desc	Sales.Item Nu
Rows 96466		180	10012226	3428	-513.15	1/12/2012	-573.3835	1/12/2012	318960	Cutting Edge Sliced Ham	
Fields 15		180	10012226	3429	-105.93	1/12/2012	-204.6638	1/12/2012	318960	Washington Cranberry Juice	
Keys 2		180	10012226	3430	-88.07	1/12/2012	-165.8016	1/12/2012	318960	Moms Sliced Ham	
Tags Skey Snumeric Sinteger Stimestamp Sdate Sascil Stext		180	10012226	3431	-43.12	1/12/2012	-118.3703	1/12/2012	318960	Tip Top Lox	
		180	10012226	3432	-37.98	1/12/2012	-102.3319	1/12/2012	318960	Just Right Beef Soup	
		180	10012226	3433	-49.37	1/12/2012	-85.5786	1/12/2012	318960	Fantastic Pumpernickel Bread	
		180	10012226	3434	-45.81	1/12/2012	-68.4399	1/12/2012	318960	Gorilla 1% Milk	


ズームレベルは、[+]または[Q]をクリックするか、[Q]スライダーを使って変更できます。[Q]をクリックして、[Home]ズームレベルを1:1に復元します。

このプレビューを使うことで、テーブルや項目の内容を確認可能です。項目を選択した際に、軸とメジャーをアプリに追加することもできます。詳細については、「データモデルビューアーでのテーブルおよび項目のプレビュー (page 233)」を参照してください。




7.1 テーブルの移動

キャンバスにドラッグして、テーブルを移動できます。テーブルの位置は、アプリを保存する際に同時に保存されます。

テーブルレイアウト(位置およびサイズ)は、キャンバス右側の  をクリックするとロックできます。テーブルレイアウトのロックを解除するには、 をクリックします。

ツールバーの  の下にある次のオプションを使用して、自動的にレイアウトを調整することもできます。

テーブルを移動するオプション




UI項目	名前	説明
	[グリッドレイアウト]	グリッド内でテーブルを調整します。
	[自動レイアウト]	ウィンドウ内にテーブルが収まるように調整します。
	[レイアウトの復元]	データモデルビューアーを最後に開いたときの状態にレイアウトを復元します。

7.2 テーブルのサイズ変更

テーブルの右下隅にある矢印を使用して、テーブルの表示サイズを調整できます。表示サイズは、アプリの保存時には保存されません。

また、ツールバーで表示サイズの自動変更オプションを使用することもできます。

テーブルのサイズを変更するオプション

UI項目	名前	説明
	全て折りたたむ	テーブルを最小化して、テーブル名のみ表示します。
	リンクされている項目を表示する	すべてのテーブルサイズを縮小し、テーブル名と他のテーブルに関連付けられた全項目を表示します。
	すべてを展開	すべてのテーブルを最大化し、テーブルのすべての項目を表示します。

7.3 データモデルのパフォーマンス

これらは、データモデルのパフォーマンスに影響を与える可能性のあるインジケータです。それぞれがアプリの使いやすさを向上させるベストプラクティスです。

データモデル パフォーマンスのベストプラクティス

アクション	説明
合成キーを削除する	Qlik Sense は、2 つ以上のデータテーブルに 2 つ以上の共通の項目がある場合に合成キーを作成します。これは、スクリプトまたはデータモデルにエラーがあることを意味する場合があります。
データモデルから循環参照を削除する	2 つの項目に複数の関連付けがある場合、循環参照が発生します。 Qlik Sense は、いずれかのテーブルへの接続を変更することにより、これらを解決しようとします。ただし、循環参照の警告はすべて解決する必要があります。
データの粒度を適切にする	必要なデータのみをロードする必要があります。例えば、ユーザーのグループに必要なのは、週、月、年で分割されたデータだけです。集約データをロードするか、ロードスクリプト内でデータを集約してメモリを節約できます。ユーザーがより細かいレベルでデータを視覚化する必要がある場合は、 ODAG またはドキュメントチェーンを使用できます。
可能な場合は QVDs を使用する	QVD は、 Qlik Sense からエクスポートされたデータのテーブルを含むファイルです。このファイル形式は、スクリプトからデータを読み取る際の速度について最適化されていますが、それでも非常にコンパクトです。 QVD ファイルからのデータの読み取りは、他のデータソースから読み取る場合よりも、一般に 10~100 倍速くなっています。
QVD ファイルをロード時に最適化する	QVD ファイルは、標準 (高速) と最適化 (超高速) の 2 つのモードで読み取ることができます。使用されるモードは、スクリプトエンジンによって自動的に決定されます。最適化されたロードに関して、いくつかの制限があります。項目の名前は変更できますが、これらの操作はいずれも標準のロードとなります。 <ul style="list-style-type: none"> • ロードされた項目に対する変換。 • where 節を使用すると、Qlik Sense でレコードが解凍されます。 • ロードされた項目でのマップの使用。

アクション	説明
増分ロードを活用する	継続的に更新される、データベースからの大量のデータにアプリが接続する場合、データセット全体をリロードするには時間がかかります。代わりに、増分ロードを使用して、データベースから新しいレコードまたは変更されたレコードを取得する必要があります。
Snowflake モデルを統合する	スノーフレックデータモデルを使用している場合は、Join プレフィックスまたは他のマッピングを使用してそれらの一部を結合することにより、データテーブルの数を減らすことができる場合があります。これは、大きなファクトテーブルにとって特に重要です。経験則として、大きなテーブルは 1 つだけにすることです。
項目の数が少ないテーブルを非正規化する	項目が少ない 2 つのテーブルがある場合、それらを結合するとパフォーマンスが向上する可能性があります。
マッピングロードがあるルックアップ(リーフ)テーブルを非正規化する	1 つの項目をテーブルから別の項目に追加するだけでよい場合は、Join プレフィックスを使用しないでください。ApplyMap ルックアップ関数を使用する必要があります。
日付項目からタイムスタンプを削除または分離する	<p>文字列表現が大きく、個別の値の数が多いため、タイムスタンプが存在する場合、日付項目がスペースを埋めることがあります。分析に精度が必要ない場合は、例えば <code>Timestamp(Floor(YourTimestamp, 1/24))</code> を使用してタイムスタンプを最も近い時間に丸めるか、<code>Date(Floor(YourTimestamp))</code> を使用して時間コンポーネントを完全に削除できます。</p> <p>タイムスタンプが必要な場合は、日付自体から切り離すことができます。同じ <code>Floor()</code> 関数を使用し、<code>Time(Frac(YourTimestamp))</code> などを使用して、抽出された時間で新しい項目を作成できます。</p>
データモデルから不要な項目を削除する	<p>データモデルに必要な項目のみをロードする必要があります。Load * と SELECT の使用は避けてください。以下を保持してください。</p> <ul style="list-style-type: none"> • 分析に必要な項目。 • アプリで実際に使用されている項目。

アクション	説明
大量のデータを処理するとき にリンクテーブルを回避する	可能な場合はリンクテーブルを使用してください。ただし、大量のデータを処理している場合、連結されたテーブルはリンクテーブルよりもパフォーマンスが高くなる可能性があります。
連結された軸を新しい項目に 分割する	連結された軸を別々の項目に分割する必要があります。これにより、項目内の値の一意の出現回数が減少します。これは、タイムスタンプを最適化する方法に似ています。
可能な場合は AutoNumber を使用する	最適化されたロードを作成するには、 QVD ファイルのデータをロードしてから、 AutoNumber ステートメントを使用して値をシンボル キーに変換します。
データアイランドを回避する	データアイランドは便利ですが、通常はパフォーマンスに影響します。選択値のアイランドを作成する場合は、変数を使用します。
QVD は増分時間枠に基づいて 保存する	QVD は毎月などのセグメントに格納する必要があります。これらの小さな月次 QVD は、すべてのデータを必要としない可能性がある多くの異なるアプリをサポートできます。


その他のベストプラクティスについては、データモデリングのベストプラクティス (page 236) を参照してください。

7.4 データモデル ビューアでのテーブルおよび項目のプレビュー

データモデル ビューアでは、画面下部のパネルに任意のデータテーブルのプレビューを表示できます。このプレビューを使うことで、テーブルや項目の内容を素早く確認可能です。項目を選択した際に、軸とメジャーをアプリに追加することもできます。

また、選択したテーブルや項目のメタデータがプレビュー パネルに表示されます。

プレビュー パネルは、次の 2 通りの方法で表示/非表示を切り替えられます。

- ツールバーで  をクリックします。
- [プレビュー] ヘッダーをクリックします。



Direct Discovery データは、プレビュー  では表示されません。

テーブルのプレビューを表示する

次の手順を実行します。

- データモデル ビューアのテーブル ヘッダーをクリックします。

プレビュー パネルには、選択されたテーブルの項目と値が表示されます。

Item master		Preview of data				
Rows	827	Item Number	Product Group	Product Line	Product Sub Group	Product Type
Fields	5	10001	Beverages	Drink	Juice	Pure Juice Beverages
Keys	1	10002	Beverages	Drink	Flavored Drinks	Drinks
Tags	\$key, \$numeric, \$integer\$ascii, \$text	10003	Beverages	Drink	Flavored Drinks	Drinks
		10004	Beverages	Drink	Soda	Carbonated Beverages
		10005	Beverages	Drink	Soda	Carbonated Beverages

項目のプレビューを表示する

次の手順を実行します。

- データモデル ビューアのテーブル項目をクリックします。

プレビュー パネルは、選択した項目およびその値、項目のメタデータとともに表示されます。項目をマスター軸またはマスター メジャーとして追加することもできます。

Product Group		Preview of data			
Density	100%	Item Number	Product Group	Product Line	Product Sub Group
Subset ratio	100%	10001	Beverages	Drink	Juice
Has duplicates	true	10002	Beverages	Drink	Flavored Drinks
Total distinct values	15	10003	Beverages	Drink	Flavored Drinks
Present distinct values	15	10004	Beverages	Drink	Soda
Non-null values	827	10005	Beverages	Drink	Soda
Tags	\$ascii, \$text				

- [密度]** は、テーブルのレコード総数に対する、その項目内で NULL 以外の値を持つレコードの数の割合です。
- [サブセット割合]** は、データモデル内の他のテーブルのその項目のユニーク値の合計数に対する、そのテーブル内で検出されたその項目のユニーク値の数の割合です。この項目が該当するのは、キー項目のみです。
- 項目に **[パーフェクトキー]** と記載されている場合、すべての行に固有のキーの値が含まれます。

7.5 データモデルビューアからのマスター軸の作成

未公開のアプリで作業する際には、マスター軸を作成して再利用することができます。公開されたアプリのユーザーは、マスター軸にアクセスできますが、これを変更することはできません。データモデルビューアは、公開済みのアプリでは利用できません。

次の手順を実行します。


1. データモデルビューアでは、項目を選択してから**[プレビュー]**パネルを開きます。
2. **[軸として追加]**をクリックします。
[軸の新規作成]ダイアログが開き、選択した項目が表示されます。選択した項目の名前が軸のデフォルト名としても使用されます。
3. 必要に応じて名前を変更し、オプションで説明、色、タグを追加します。
4. **[追加]**をクリックします。
5. **[完了]**をクリックしてダイアログを閉じます。

これでアセットパネルのマスターアイテムタブに軸が保存されました。



それぞれの軸を追加してから、**[軸を追加]**をクリックすると、複数の軸をマスターアイテムとして追加できます。終了したら、**[完了]**をクリックしてください。



Direct Discovery のテーブルは  データモデルビューアでは表示されません。

7.6 データモデルビューアからのマスターメジャーの作成

未公開のアプリで作業する際には、マスターメジャーを作成して再利用することができます。公開されたアプリのユーザーは、マスターメジャーにアクセスできますが、これを変更することはできません。

次の手順を実行します。

1. データモデルビューアでは、項目を選択してから**[プレビュー]**パネルを開きます。
2. **[メジャーとして追加]**をクリックします。
[メジャーの新規作成]ダイアログが開き、選択された項目が表示されます。選択された項目の名前は、メジャーのデフォルトの名前としても使用されます。
3. メジャーの数式を入力します。
4. 必要に応じて名前を変更し、オプションで説明、色、タグを追加します。
5. **[追加]**をクリックします。

これでアセットパネルのマスターアイテムタブにメジャーが保存されました。

8 データモデリングのベストプラクティス

Qlik Sense アプリへのデータロード方法は、データ構造や希望するデータモデルによって異なります。このセクションでは、こうしたデータロード方法について説明します。

8.1 データ列を行に変える

次のようなデータがあり、売上高を別個の項目にする場合:

元のデータテーブル

年	Q1	Q2	Q3	Q4
2013	34	54	53	52
2014	47	56	65	67
2015	57	56	63	71

提案されたアクション

テーブルをロードするときに、**Crosstable** プレフィックスを使用します。

結果は次のようになります。

Crosstable プレフィックスを適用した後のテーブル

Year	Quarter	Sales
2013	Q1	34
2013	Q2	54
2013	Q3	53
2013	Q4	52
2014	Q1	47
...

8.2 データ行を項目に変える

次のような3つの項目を持つ汎用テーブルがあり、各属性を個別のテーブルにする場合:

3つの項目を含む Generic テーブル

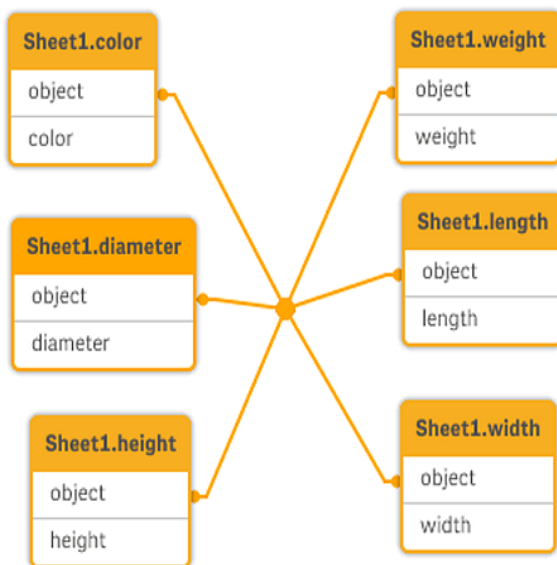
Object	Attribute	Value
ball	color	red
ball	diameter	25
ball	weight	3

Object	Attribute	Value
box	color	56
box	height	30
box	length	20
box	width	25

提案されたアクション

[Generic] ロードプレフィックスを使用して汎用データモデルを作成します。

次のようなデータモデルを取得します。



8.3 階層レベルに整理されたデータ(組織図など)をロードする

次のような隣接するノードテーブルにデータが格納されている場合:

隣接するノードテーブル

NodeID	ParentNodeID	Title
1	-	General manager
2	1	Country manager
3	2	Region manager

提案されたアクション

Hierarchy プレフィックスを使用してデータをロードし、展開ノードテーブルを作成します。

展開されたノードテーブル

NodeID	ParentNodeID	Title	Level1	Level2	Level3
1	-	General manager	General manager	-	-
2	1	Country manager	General manager	Country manager	-
3	2	Region manager	General manager	Country manager	Region manager

8.4 大きなデータベースから新しいレコードや更新されたレコードのみをロードする

多数のレコードを含むデータベースがあり、アプリでデータを更新するためにデータベース全体はリロードしたくありません。新しい、または更新されたレコードのみをロードし、データベースから削除されたレコードを除去したいと思っています。

提案されたアクション

QVD ファイルを使用して、増分ロードソリューションを実装します。

8.5 共通項目を持つ2つのテーブルのデータを結合する

Qlik Sense は自動的に共通項目をテーブルに関連付けるが、テーブルの結合方法を制御する必要がある場合、

提案されたアクション: Join / Keep

Join または **Keep** プレフィックスを使用して2つのテーブルを1つの内部テーブルに結合できます。

提案されたアクション: マッピング

テーブルを結合する代わりにマッピングを使用して、マッピングテーブルに関連付けられた値の検索を自動化します。これにより、ロードするデータ量を減らすことができます。

8.6 不連続値を間隔に一致させる

1つの不連続数値のテーブル (Event) があり、これを1つ以上の間隔 (Start とEnd) に一致させる場合、

離散数値のテーブル (Event)

Time	Event (イベント)	コメント
00:00	0	シフト1の開始
01:18	1	ラインの停止
02:23	2	ラインの50%の再開
04:15	3	ライン速度 100%

Time	Event (イベント)	コメント
08:00	4	シフト2の開始
11:43	5	稼働の終了

間隔が含まれているテーブル (Start および End)

開始	End	順序
01:00	03:35	A
02:30	07:58	B
03:04	10:27	C
07:23	11:43	D

提案されたアクション

IntervalMatch プレフィックスを使用して、Time 項目を Start と End で定義される間隔にリンクします。

間隔の開始と終了が明示的に定義されていない場合は、次の表のようにタイムスタンプのみを変更して間隔テーブルを作成する必要があります。

変更の日付と時刻を含むテーブル

通貨	データの変更	Rate
EUR	-	8.59
EUR	28/01/2013	8.69
EUR	15/02/2013	8.45
USD	-	6.50
USD	10/01/2013	6.56
USD	03/02/2013	6.30

8.7 不一致項目値を処理する

データに、異なるテーブル間で名前が一致しない項目値が含まれる場合例えば、あるテーブルには国の値 US が、別のテーブルには United States が含まれます。関連付けが阻止されます。

テーブル 1

国	地域
米国	メリーランド
米国	アイダホ
米国	ニューヨーク
米国	カリフォルニア

Table 2

Country	Population
United States	304
Japan	128
Brazil	192
China	1333

提案されたアクション

マッピングテーブルを使用してデータクレンジングを実行することで、項目値を比較して正しく関連付けることができます。

8.8 不一致の項目値の大文字の処理

自分のデータに、テーブル間で書式設定が整合していない項目値が含まれているとします。例えば、あるテーブルのTypeにはsingleという値が含まれており、別のテーブルの同じ項目にはSingleという値が含まれているとします。この状況では、Type項目にsingleとSingleの値の両方が含まれており、大文字と小文字の違いがあるため、関連付けが妨げられます。

テーブル 1

タイプ	価格
単色	23
2色	39

テーブル 2

タイプ	色
単色	赤色
単色	青
2色	白色
2色	ブラック

提案されたアクション

[データの追加] でデータをロードした場合は、この問題をデータマネージャーで修正できます。

次の手順を実行します。

1. データマネージャーのテーブルエディタで、Table2を開きます。
2. Type項目の名前をTable2.Typeに変更します。
データプロファイリングを有効にして、[データの追加] で単にテーブルを追加した場合は、自動関連付けが行われないようにするため、その項目にはTable2.Typeという名前がすでに付けられています。この場合は、次の手順によって2つのテーブルが関連付けられます。

3. 数式 `Lower(Table2.Type)` を使用して計算項目を作成し、**Type** という名前を付けます。
4. **[データのロード]** をクリックします。

これで、**Table1** と **Table2** が項目 **Type** によって関連付けられます。この項目には、**single** や **double** のような小文字の値のみが含まれています。

大文字の部分が異なる値を使用する場合は、同様の手順で関連付けを行うことができますが、テーブルでは、同じ名前の項目を使用して関連付けが行われることに注意してください。

- **Single** のようにすべての値を大文字にする場合は、代わりに **Table1** で計算された **Type** 項目を作成し、数式 `Capitalize(Table1.Type)` を使用します。
- **SINGLE** のようにすべての値を大文字にする場合は、両方のテーブルで計算された **Type** 項目を作成し、それぞれのテーブルで数式 `Upper(Table1.Type)` と `Upper(Table2.Type)` を使用します。

8.9 地理空間データをロードしてマップのデータを可視化する

マップを使用して可視化するデータ(国別または店舗別の売上データなど)がある場合、マップビジュアライゼーションを使用するには、エリアデータまたはポイントデータをロードする必要があります。

提案されたアクション

KML ファイルまたは Excel ファイルから、データ値の場所と一致するエリアデータまたはポイントデータをロードできます。また、実際のマップの背景もロードする必要があります。

8.10 増分ロードにより新規および更新された記録をロード

継続的に更新される、データベースからの大量のデータがアプリに含まれる場合、データセット全体をリロードするには時間がかかります。この場合、データベースから新しいレコードまたは変更されたレコードをロードします。その他のデータはアプリで利用できるはずですが、増分ロードでは QVD ファイルを使用し、それを可能にします。

基本手順を以下に説明します。

1. データベースのソーステーブルから、新規または更新されたデータをロードします。
このプロセスには時間がかかりますが、限定されたレコード数のみをロードします。
2. QVD ファイルから、アプリですでに利用可能なデータをロードします。
多数のレコードがロードされますが、このプロセスに時間はかかりません。
3. 新しい QVD ファイルを作成します。
これが、次回増分ロードで使用されるファイルとなります。
4. ロードされるテーブルごとに手順を繰り返します。

以下の例には、増分ロードを使用する場合は示されていますが、ソースデータベースの構造や操作モードによっては、より複雑なソリューションが必要な場合もあります。

- 追加のみ (通常はログファイルに使用)
- 挿入のみ (更新または削除なし)

- 挿入および更新 (削除なし)
- 挿入、更新、および削除

QVD ファイルは、最適化と標準のどちらかのモードで読み取ることができます(使用されるモードは、オペレーションの複雑さによって **Qlik Sense** エンジンが自動的に選択します)。最適化モードは、標準モードの約 10 倍、普通の方法でのデータベースのロードの約 100 倍の速度です。

追加のみ

最も単純なケースはログ ファイルです。このようなファイルではレコードが追加されるだけで、削除されることはありません。次の条件が適用されます。

- データベースは、テキストファイルに含まれているログ ファイル (またはレコードが挿入または削除ではなく追加される他のファイル) でなくてはなりません (ODBC、OLE DB、またはその他のデータベースはサポートされていません)。
- **Qlik Sense** は、以前に読み取られたレコード数を常に記録し、ファイルの末尾に追加されたレコードだけをロードします。

```
Buffer (Incremental) Load * From LogFile.txt (ansi, txt, delimiter is '\t', embedded labels);
```

挿入のみ (更新または削除なし)

単純なログ ファイル以外のデータベースにデータが存在する場合は、追加の方法は機能しません。ただし、少しの作業を追加するだけで問題を解決できます。次の条件が適用されます。

- データソースには、任意のデータベースを使用できます。
- **Qlik Sense** は、最後のスクリプト実行後にデータベースに挿入されたレコードをロードします。
- **ModificationTime** 項目 (または同様の項目) は、どのレコードが新しいかを **Qlik Sense** が認識するために必要です。

```
QV_Table:
```

```
SQL SELECT PrimaryKey, X, Y FROM DB_TABLE
```

```
WHERE ModificationTime >= #$(LastExecTime)#
```

```
AND ModificationTime < #$(BeginningThisExecTime)#;
```

```
Concatenate LOAD PrimaryKey, X, Y FROM File.QVD;
```

```
STORE QV_Table INTO File.QVD;
```

SQL WHERE 句のハッシュ記号は日付の開始と終了を定義します。ご使用中のデータベースの適正な日付構文については、データベースマニュアルを確認してください。

挿入および更新 (削除なし)

次のケースは、以前にロードされたレコード内のデータがスクリプト実行時に変更されている場合に適用可能です。次の条件が適用されます。

- データソースには、任意のデータベースを使用できます。
- **Qlik Sense** は、最後のスクリプト実行後にデータベースに挿入されたレコード、またはデータベースで更新されたレコードをロードします。
- **ModificationTime** 項目 (または同様の項目) は、どのレコードが新しいかを **Qlik Sense** が認識するために必要です。
- **Qlik Sense** が QVD ファイルから更新されたレコードをソートするためには、プライマリーキー項目が必要です。
- このソリューションでは、QVD ファイルの読み取りは (最適化モードではなく) 標準モードで行われます。標準モードはデータベース全体をロードするよりも、かなり高速です。

QV_Table:

```
SQL SELECT PrimaryKey, X, Y FROM DB_TABLE
```

```
WHERE ModificationTime >= #$(LastExecTime)#;
```

```
Concatenate LOAD PrimaryKey, X, Y FROM File.QVD
```

```
WHERE NOT Exists(PrimaryKey);
```

```
STORE QV_Table INTO File.QVD;
```

挿入、更新、および削除

最も処理が難しいケースは、スクリプト実行時にソースデータベースからレコードが実際に削除されている場合です。次の条件が適用されます。

- データソースには、任意のデータベースを使用できます。
- **Qlik Sense** は、最後のスクリプト実行後にデータベースに挿入されたレコード、またはデータベースで更新されたレコードをロードします。
- **Qlik Sense** は、最後のスクリプト実行後にデータベースから削除されたレコードを削除します。
- **ModificationTime** 項目 (または同様の項目) は、どのレコードが新しいかを **Qlik Sense** が認識するために必要です。

- Qlik Sense が QVD ファイルから更新されたレコードをソートするためには、プライマリー項目が必要です。
- このソリューションでは、QVD ファイルの読み取りは (最適化モードではなく) 標準モードで行われます。標準モードはデータベース全体をロードするよりも、かなり高速です。

```
Let ThisExecTime = Now( );
```

```
QV_Table:
```

```
SQL SELECT PrimaryKey, X, Y FROM DB_TABLE
```

```
WHERE ModificationTime >= #$(LastExecTime)#
```

```
AND ModificationTime < #$(ThisExecTime)#;
```

```
Concatenate LOAD PrimaryKey, X, Y FROM File.QVD
```

```
WHERE NOT EXISTS(PrimaryKey);
```

```
Inner Join SQL SELECT PrimaryKey FROM DB_TABLE;
```

```
If ScriptErrorCount = 0 then
```

```
STORE QV_Table INTO File.QVD;
```

```
Let LastExecTime = ThisExecTime;
```

```
End If
```

8.11 Join と Keep を使用したテーブルの結合

結合とは 2 つのテーブルを 1 つにまとめる操作です。結合により生成されるテーブルのレコードは元のテーブルのレコードを組み合わせたもので、通常、生成されたテーブル内の特定の組み合わせに関する 2 つのレコードが 1 つ以上の共通項目に共通の値を持っています。そのため、これを「自然結合」と呼びます。Qlik Sense では、結合はスクリプトで実行され、論理テーブルが作成されます。

スクリプト内で事前にテーブルを結合することができます。その場合、Qlik Sense ロジックは、個々のテーブルではなく、結合の結果である単一の内部テーブルを処理するようになります。この処理が必要な場合もありますが、次のような欠点があります。

- ロードされたテーブルのサイズが大きくなり、**Qlik Sense** の動作が遅くなることがあります。
- 元のテーブル内のレコード数を取得できなくなり、情報の一部が失われる可能性があります。

Keep 機能は、**Qlik Sense** にテーブルを保存する前に、2 つのテーブルの片方または両方をテーブルデータの共通部分に縮小させる効果を持ち、明示的な参加を使用する必要がある場合の数を減らすために設計されています。



このマニュアルでは、結合という用語は、内部テーブルが作成される前に行われる結合という意味で使用されています。しかし、内部テーブルが作成された後で行われる関連付けも、基本的には結合です。

SQL SELECT ステートメント内で結合

ODBC ドライバを使用して、**SELECT** ステートメント内で結合を作成できます。これは、**Join** プレフィックスを使用した結合とほぼ同じです。

ただし、ほとんどの ODBC ドライバは、完全な (双方向) 外部結合を作成できません。これらのドライバで作成できるのは、左または右外部結合に限られます。左 (右) 外部結合には、左 (右) テーブルに結合キーが存在する組み合わせだけが含まれますが、完全な外部結合にはすべての組み合わせが含まれます。**Qlik Sense** は、自動的に完全な外部結合を作成します。

また、**SELECT** ステートメントによる結合は、**Qlik Sense** で作成する結合よりもはるかに複雑です。

```
SELECT DISTINCTROW
[Order Details].ProductID, [Order Details].
UnitPrice, Orders.OrderID, Orders.OrderDate, Orders.CustomerID
FROM Orders
RIGHT JOIN [Order Details] ON Orders.OrderID = [Order Details].OrderID;
```

この **SELECT** ステートメントは、架空の会社への注文 (**Orders**) を含むテーブルを、注文詳細 (**Order Details**) を含むテーブルに結合します。これは右外部結合で、**OrderDetails** のすべてのレコードに加え、**Orders** テーブルに存在しない **OrderID** を持つレコードも含まれています。ただし、**Orders** に存在しても、**OrderDetails** に存在しない注文は含まれていません。

Join

結合を作成する最も簡単な方法は、スクリプトで **Join** プレフィックスを使用することです。これは、内部テーブルを別の名前のテーブルまたは最後に作成されたテーブルに結合します。この結合は外部結合になり、2 つのテーブルからのすべての可能な値の組み合わせを作成します。

```
LOAD a, b, c from table1.csv;
join LOAD a, d from table2.csv;
```

その結果の内部テーブルには、項目 **a**、**b**、**c**、**d** が含まれています。レコード数は、2 つのテーブルの項目値によって異なります。



結合される項目の名前は、完全に同じである必要があります。結合される項目数は任意です。通常、テーブルには1つまたは複数の共通する項目があります。共通する項目が1つもない場合、テーブルのデカルト積は生成されません。すべての項目が共通することも考えられますが、通常これは意味がありません。**Join** ステートメントで、先にロードされたテーブルの名前が指定されていない限り、**Join** プレフィックスは最後に作成されたテーブルを使用します。そのため、2つのステートメントの順序を任意に指定することはできません。

Keep

データロードスクリプトにおける明示的な **Join** プレフィックスは、2つのテーブルの完全な結合を実行します。その結果、1つのテーブルが生成されます。多くの場合、このような結合からは、かなり大きなテーブルが作成されます。**Qlik Sense** の主な機能の1つは、テーブルを結合する代わりにテーブル間に関連付けを作成できることです。これにより、メモリの使用量が削減され、速度が向上し、柔軟性が大幅に高まります。**keep** の機能は、明示的な結合の使用回数を減らす目的で設計されています。

2つの **LOAD** または **SELECT** ステートメント間の **Keep** プレフィックスは、**Qlik Sense** にテーブルが保存される前に、2つのテーブルの片方または両方をテーブルデータの共通部分に縮小させます。**Keep** プレフィックスの前には、**Inner**、**Left**、または **Right** のいずれかのキーワードを置く必要があります。テーブルからのレコードの選択は、対応する結合と同じ方法で行われます。ただし、2つのテーブルは結合されず、個別のテーブルとしてそれぞれ異なる名前でも **Qlik Sense** に保存されます。

Inner

データロードスクリプトの **Join** および **Keep** プレフィックスの前に、プレフィックス **Inner** を置くことができます。

Join の前に使用すると、2つのテーブル間の結合が内部結合であると指定されます。結果のテーブルには、両方のテーブルのすべてのデータセットのうち、2つのテーブル間で共通するものだけが含まれます。

Keep の前に使用すると、**Qlik Sense** に保存される前に、2つのテーブルは互いの共通部分に縮小されます。

これらの例では、Table1 と Table2 のソーステーブルを使用しています。

Table 1

A	B
1	aa
2	cc
3	ee

Table2

A	C
1	xx
4	yy

Inner Join

まず、これらのテーブルで **Inner Join** を実行すると、VTable が作成されます。このテーブルには 2 つのテーブルのデータを結合した結果、1 行だけ、両方のテーブルに存在するレコードだけが表示されます。

VTable:

```
SELECT * from Table1;
inner join SELECT * from Table2;
```

VTable

A	B	C
1	aa	xx

Inner Keep

それに対して、**Inner Keep** を実行すると、2 つのテーブルが残ります。これらの 2 つのテーブルは、共通項目 A を介して関連付けられます。

VTab1:

```
SELECT * from Table1;
```

VTab2:

```
inner keep SELECT * from Table2;
```

VTab1

A	B
1	aa

VTab2

A	C
1	xx

Left

データロードスクリプトの **Join** および **Keep** プレフィックスの前に、プレフィックス **left** を置くことができます。

Join の前に使用すると、2 つのテーブル間の結合が左結合であると指定されます。結果のテーブルには、1 つ目のテーブルのすべてのデータセットと 2 つのテーブルの共通するものだけが含まれます。

Keep の前に使用すると、Qlik Sense に保存される前に、2 つ目のテーブルが 1 つ目のテーブルとの共通部分に縮小されます。

これらの例では、Table1 と Table2 のソーステーブルを使用しています。

Table1

A	B
1	aa

A	B
2	cc
3	ee

Table2

A	C
1	xx
4	yy

まず、これらのテーブルで **Left Join** を実行すると、VTable が生成されます。このテーブルには、Table1 のすべての行に加え、Table2 の一致行の項目が含まれています。

VTable:

```
SELECT * from Table1;
left join SELECT * from Table2;
```

VTable

A	B	C
1	aa	xx
2	cc	-
3	ee	-

それに対して、**Left Keep** を実行すると、2つのテーブルが残ります。これらの2つのテーブルは、共通項目 A を介して関連付けられます。

VTab1:

```
SELECT * from Table1;
```

VTab2:

```
left keep SELECT * from Table2;
```

VTab1

A	B
1	aa
2	cc
3	ee

VTab2

A	C
1	xx

Right

データロードスクリプトの **Join** および **Keep** プレフィックスの前に、プレフィックス **right** を置くことができます。

Join の前に使用すると、2つのテーブル間の結合が右結合であると指定されます。結果のテーブルには、2つ目のテーブルのすべてのデータと2つのテーブルで共通するものだけが含まれます。

Keep の前に使用すると、Qlik Sense に保存される前に、1つ目のテーブルが2つ目のテーブルとの共通部分に縮小されます。

これらの例では、Table1 と Table2 のソーステーブルを使用しています。

Table1

A	B
1	aa
2	cc
3	ee

Table2

A	C
1	xx
4	yy

まず、これらのテーブルで **Right Join** を実行すると、VTable が生成されます。このテーブルには、Table2 のすべての行に加え、Table1 の一致行の項目が含まれています。

VTable:

```
SELECT * from Table1;  
right join SELECT * from Table2;
```

VTable

A	B	C
1	aa	xx
4	-	yy

それに対して、**Right Keep** を実行すると、2つのテーブルが残ります。これらの2つのテーブルは、共通項目 A を介して関連付けられます。

VTab1:

```
SELECT * from Table1;
```

VTab2:

```
right keep SELECT * from Table2;
```

VTab1

A	B
1	aa

VTab2

A	C
1	xx
4	yy

8.12 結合の代替としてのマッピング

Qlik Sense の **Join** プレフィックスは、データモデルで複数のデータテーブルを組み合わせる強力な方法です。

ただし、組み合わせられたテーブルが大きくなり、パフォーマンスに問題が生じる可能性があるという欠点もあります。そこで、別のテーブルから単一の値を探す必要がある場合など、**Join** の代わりにマッピングを使用することができます。結合ではテーブルのレコード数が変わるため、計算のスピードダウンとエラーの原因になる可能性がある不要なデータのロードを回避できます。

マッピング テーブルは、比較項目 (入力) とマッピング値項目 (出力) という2つの列で構成されます。

この例では注文テーブル (**Orders**)、を使用しており、顧客の国を把握する必要があります。この情報は、顧客テーブル (**Customers**) に保管されています。

Orders データテーブル

OrderID	OrderDate	ShipperID	Freight	CustomerID
12987	2007-12-01	1	27	3
12988	2007-12-01	1	65	4
12989	2007-12-02	2	32	2
12990	2007-12-03	1	76	3

Customers データテーブル

CustomerID	名前	国	...
1	DataSales	Spain	...
2	BusinessCorp	Italy	...
3	TechCo	Germany	...
4	Mobecho	France	...

顧客の国 (Country) を見つけるには、以下のようなマッピング テーブルが必要です。

マッピング テーブル

CustomerID	国
1	Spain
2	Italy

CustomerID	国
3	Germany
4	France

MapCustomerIDtoCountry という名前を付けたマッピング テーブルは、以下のようにスクリプトで定義されます。

```
MapCustomerIDtoCountry:
Mapping LOAD CustomerID, Country From Customers ;
```

次に、注文テーブルをロードする際に、**ApplyMap** 関数を使用してマッピングを適用します。

```
Orders:
LOAD *,
ApplyMap('MapCustomerIDtoCountry', CustomerID, null()) as Country
From Orders ;
```

ApplyMap 関数の 3 番目のパラメータは、マッピング テーブルで値が見つからない場合に何を返すのか定義するために使われます。この例では、**Null()** です。

この結果、テーブルは次のようになります。

結果テーブル

OrderID	OrderDate	ShipperID	Freight	CustomerID	国
12987	2007-12-01	1	27	3	Germany
12988	2007-12-01	1	65	4	France
12989	2007-12-02	2	32	2	Italy
12990	2007-12-03	1	76	3	Germany

8.13 データロードスクリプトでのクロス集計の使用

クロスタブは、ヘッダーデータが直交する 2 つのリストに値のマトリックスを持つ一般的なテーブルの種類です。他のデータテーブルにデータを関連付けたい場合は、必ずしも最適なデータ形式ではありません。

このトピックではクロスタブのピボットを無効にする方法、つまり、データロードスクリプト内の **crosstable** ステートメントに対する **LOAD** プレフィックスを使用して、タブの一部を行と入れ替える方法を説明します。

1 つの行見出し列により、クロスタブのピボットを無効にする

クロスタブでは、いくつか行見出しとなる列を表示することがあります。これらの列は、直接的な方法で読み取ります。この場合、行見出しとなる列 **Year** が 1 つあり、1 か月当たりの販売データのマトリックスがあります。

1 つの行見出し列を含むクロスタブ

Year	Jan	Feb	Mar	Apr	May	Jun
2008	45	65	78	12	78	22

Year	Jan	Feb	Mar	Apr	May	Jun
2009	11	23	22	22	45	85
2010	65	56	22	79	12	56
2011	45	24	32	78	55	15
2012	45	56	35	78	68	82

このテーブルを単純に Qlik Sense にロードすると、**Year**(年)に1つ、そして各月に1つずつ項目が生成されます。ただし、通常はこのようなテーブルではデータを適切に管理できません。3つの項目を生成したいはずです。

- 行見出しとなる列、この場合では **Year** には、上のテーブルで緑色の印が付いています。
- この場合では月の名前 **Jan - Jun** で表される属性項目には、黄色の印が付いています。この項目には、**Month** と名前を付けることができます。
- データマトリクスの値には、青色の印が付いています。この場合、値は販売データを表すので、**Sales** と名前を付けることができます。

それには **LOAD** または **SELECT** ステートメントに **crosstable** プレフィックスを追加します。

```
crosstable (Month, Sales) LOAD * from ex1.xlsx;
```

これにより、次のテーブルが Qlik Sense で作成されます。

LOAD または SELECT ステートメントに **crosstable** プレフィックスが追加されているテーブル

Year	Month	Sales
2008	Jan	45
2008	Feb	65
2008	Mar	78
2008	Apr	12
2008	May	78
2008	Jun	22
2009	Jan	11
2009	Feb	23
...

2つの行見出し列により、クロスタブのピボットを無効にする

この場合は、左側に2つの行見出し列があり、その後にはマトリクス列が続きます。

2つの行見出し列を含むクロスタブ

Salesman	Year	Jan	Feb	Mar	Apr	May	Jun
A	2008	45	65	78	12	78	22

Salesman	Year	Jan	Feb	Mar	Apr	May	Jun
A	2009	11	23	22	22	45	85
A	2010	65	56	22	79	12	56
A	2011	45	24	32	78	55	15
A	2012	45	56	35	78	68	82
B	2008	57	77	90	24	90	34
B	2009	23	35	34	34	57	97
B	2010	77	68	34	91	24	68
B	2011	57	36	44	90	67	27
B	2012	57	68	47	90	80	94

行見出し列の数は、次のように **crosstable** プレフィックスの 3 番目のパラメータとして指定できます。

```
crosstable (Month, Sales, 2) LOAD * from ex2.xlsx;
```

その結果、Qlik Sense で読み込んだ際に次のようになります

crosstable プレフィックスの 3 番目のパラメータとして指定された行見出し列を含むテーブル

Salesman	Year	Month	Sales
A	2008	Jan	45
A	2008	Feb	65
A	2008	Mar	78
A	2008	Apr	12
A	2008	May	78
A	2008	Jun	22
A	2009	Jan	11
A	2009	Feb	23
...

8.14 汎用データベース

汎用データベースは、項目名が項目値として 1 列に、項目値が 2 番目の列に保存されるテーブルです。通常、汎用データベースは、さまざまなオブジェクトの属性に対して使用されます。

以下の **GenericTable** の例をご覧ください。これは、**ball** と **box** の 2 つのオブジェクトを含む汎用データベースです。**color** および **weight** など、いくつかの属性は両方のオブジェクトに共通していますが、**diameter**、**height**、**length**、**width** は共通ではありません。

GenericTable

object	attribute	値
ボール	色	赤色
ボール	diameter	10 cm
ボール	weight	100 g
ボックス	色	ブラック
ボックス	高さ	16 cm
ボックス	length	20 cm
ボックス	weight	500 g
ボックス	幅	10 cm

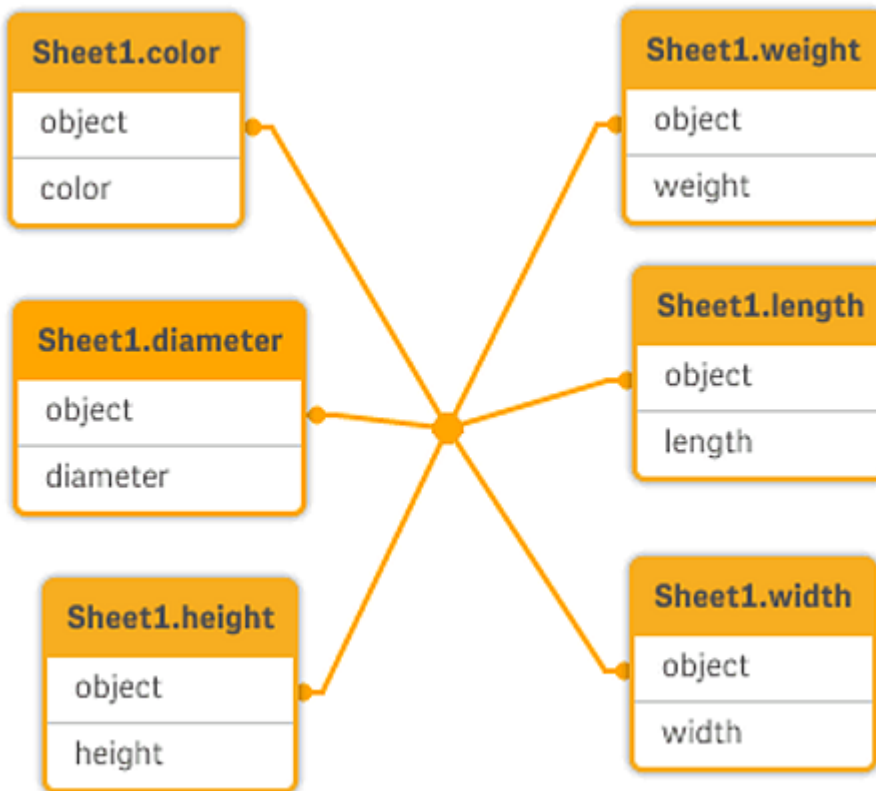
属性の多くは特定のオブジェクトに関連していないため、それぞれの属性が固有の列を持つような方法でデータを保存することは不適切と考えられます。

一方、length と color、weight を混在させて表示すると、混乱を招く可能性があります。

標準的な方法でこのデータベースを Qlik Sense にロードし、データがテーブルに表示される場合、以下のようになります。

object ▲	attribute	value
ball	color	red
ball	diameter	10 cm
ball	weight	100 g
box	color	black
box	height	16 cm
box	length	20 cm
box	weight	500 g
box	width	10 cm

ただし、このテーブルが汎用データベースとしてロードされた場合、2列目と3列目が個別のテーブルに分割されます(2列目の一意の値ごとにテーブルが1つ作成されます)。



次の例に示すように、これを行うための構文は単純です。

例

ロード スクリプト

アプリにスクリプト例を追加して実行します。結果を表示するには、結果列に含まれている項目をアプリのシートに追加します。

```
Sheet1: Generic Load * inline [ object, attribute, value ball, color, red ball, diameter, 10
cm ball, weight, 100 g box, color, black box, height, 16 cm box, length, 20 cm box, weight,
500 g box, width, 10 cm ];
```

結果

結果のテーブル

object	色	diameter	length	高さ	幅	weight
ball	red	10 cm	-	-	-	100 g
box	black	-	20 cm	16 cm	10 cm	500 g

8.15 不連続データと間隔の一致

intervalmatch プレフィックスを **LOAD** または **SELECT** ステートメントで使用すると、不連続の数値が1つ以上の数値間隔にリンクされます。これはとても有用な機能で、実稼働環境などで使用できます。

Intervalmatch の例

次の2つのテーブルを見てください。最初のテーブルは、さまざまな注文 (**Order**) の製造の開始時刻 (**Start**) と終了時刻 (**End**) を示しています。2つ目のテーブルには個々のイベントが含まれています。どの注文がイベントの影響を受けたか、どの注文がどのシフトで処理されたかを知るには、どのように離散したイベントを注文に関連付ければよいでしょうか。

テーブル OrderLog

開始	End	順序
01:00	03:35	A
02:30	07:58	B
03:04	10:27	C
07:23	11:43	D

テーブル EventLog

Time	Event (イベント)	コメント
00:00	0	シフト1の開始
01:18	1	ラインの停止
02:23	2	ラインの50%の再開
04:15	3	ライン速度100%
08:00	4	シフト2の開始
11:43	5	稼働の終了

まず、通常どおりに2つのテーブルをロードします。次に、**Time** 項目を **Start** 項目と **End** 項目で定義される間隔にリンクします。

```
SELECT * from OrderLog;
SELECT * from EventLog;
Intervalmatch (Time) SELECT Start,End from OrderLog;
```

これで、以下のように Qlik Sense でテーブルを作成できます。

[開始]と[終了]で定義された間隔にリンクされた[時刻]項目を含むテーブル

Time	Event (イベント)	コメント	順序	開始	End
0:00	0	シフト1の開始	-	-	-
1:18	1	ラインの停止	A	1:00	3:35
2:23	2	ラインの50%の再開	A	1:00	3:35
4:15	3	ライン速度 100%	B	2:30	7:58
4:15	3	ライン速度 100%	C	3:04	10:....
8:00	4	シフト2の開始	C	3:04	10:....
8:00	4	シフト2の開始	D	7:23	11:....
11:43	5	稼働の終了	E	7:23	11:.....

主に注文 A がラインストップ (Line stop) の影響を受けた一方で、ラインスピード (Line speed) の減少は注文 B と C にも影響を及ぼしたことが分かります。注文 C と D だけは、一部が **Shift 2** で処理されています。

intervalmatch を使用する場合は、次の点に注意してください。

- **intervalmatch** ステートメントの前に、不連続データポイントを含む項目 (上の例では **Time**) が **Qlik Sense** に読み取られている必要があります。**intervalmatch** ステートメントは、データベース テーブルからこの項目を読み取りません。
- **intervalmatch LOAD** または **SELECT** ステートメントで読み取られたテーブルは、厳密に 2 つの項目 (上記の例では、**Start** と **End**) を含んでいる必要があります。他の項目へのリンクを確立するには、間隔項目と共に、個別の **LOAD** または **SELECT** ステートメント (上記の例では最初の **SELECT** ステートメント) で他の項目を読み取る必要があります。
- 間隔は常に閉じています。つまり、終端は間隔に含まれます。**NULL** の範囲は間隔を無限に拡張しますが (無制限)、数値以外の範囲は間隔を無視します (未定義)。
- 間隔は重なる場合があります、不連続値は一致する間隔すべてにリンクされます。

拡張された **intervalmatch** 構文を使用した、穏やかに変化する軸 (SCD) 問題の解決

拡張された **intervalmatch** 構文を使用して、ソースデータ内で軸の変化が遅いという周知の問題に対処することができます。

サンプル スクリプト:

```
SET NullInterpret='';
```

```
IntervalTable:
```

```
LOAD Key, ValidFrom, Team
```

```
FROM 'lib://dataq/intervalmatch.xlsx' (ooxml, embedded labels, table is IntervalTable);
```

```
Key:
```

```
LOAD
```

```
Key,
```

```
ValidFrom as FirstDate,
```

```
date(if(Key=previous(Key),
```

```
previous(ValidFrom) - 1)) as LastDate,
```

```
Team
RESIDENT IntervalTable order by Key, ValidFrom desc;
```

```
drop table IntervalTable;
```

```
Transact:
```

```
LOAD Key, Name, Date, Sales
FROM 'lib://dataqv/intervalmatch.xlsx' (ooxml, embedded labels, table is Transact);
```

```
INNER JOIN intervalmatch (Date,Key) LOAD FirstDate, LastDate, Key RESIDENT Key;
```

欠損値は NULL 値ではなく空の文字列として定義されるため、**nullinterpret** ステートメントはテーブル ファイルからデータを読み取る場合にのみ必要となります。

IntervalTable からデータをロードすると、次のテーブルが生成されます。

Table with data loaded from IntervalTable

Key	FirstDate	Team
000110	2011-01-21	Southwest
000110	-	Northwest
000120	-	Northwest
000120	2013-03-05	Southwest
000120	2013-03-05	Northwest
000120	2013-03-05	Southwest

nullasvalue ステートメントでは、リストされている項目に NULL 値をマップできます。

previous および **order by** を使用して、*Key*、*FirstDate*、*LastDate* (属性項目) を作成します。その後、このキーテーブルで置き換えられた *IntervalTable* がドロップされます。

Transact からデータをロードすると、次のテーブルが生成されます。

Transact からロードされたデータを含むテーブル

キー	名前	日付	売上
000110	Spengler Aaron	2009-08-18	100
000110	Spengler Aaron	2009-12-25	200
000110	Spengler Aaron	2011-02-03	300
000110	Spengler Aaron	2011-05-05	400
000120	Ballard John	2011-06-04	500
000120	Ballard John	2013-01-20	600
000120	Ballard John	2013-03-10	700
000120	Ballard John	2013-03-13	800
000120	Ballard John	2013-09-21	900

inner join に続く **intervalmatch** ステートメントは、*Transact* に接続する合成キーで上記のキーを置き換えます。その結果、次のテーブルが生成されます。

inner join に続く **intervalmatch** ステートメントを含むテーブル

キー	チーム	名前	FirstDate	LastDate	日付	売上
000110	Northwest	Spengler Aaron	-	2011-01-20	2009-08-18	100
000110	Northwest	Spengler Aaron	-	2011-01-20	2009-12-25	200
000110	Southwest	Spengler Aaron	2011-01-21		2011-02-03	300
000110	Southwest	Spengler Aaron	2011-01-21		2011-05-05	400
000120	Northwest	Ballard John		2013-01-05	2011-06-04	500
000120	Southwest	Ballard John	2013-01-06	2013-03-04	2013-01-20	600
000120	Southwest	Ballard John	2013-03-05		2013-03-10	700
000120	Southwest	Ballard John	2013-03-05		2013-03-13	800
000120	Southwest	Ballard John	2013-03-05		2013-09-21	900

8.16 単一日付からの日付間隔の作成

時間の間隔が明示的に開始および終了を含むことなく保存される場合があります。代わりに、変更タイムスタンプという1つの項目により暗黙のうちに定義されています。

複数の通貨の現在のレートを記載した下記のテーブルがある場合、各通貨レートの変化は新しい変換レートとともにそれぞれの行に表示されています。また、テーブルには最初に変換が実施される前の、当初の変換レートに対応する日付が空の行も含まれています。

Currency rates (通貨レート)

通貨	Change Date (変更日)	Rate
EUR	-	8.59
EUR	28/01/2013	8.69
EUR	15/02/2013	8.45
USD	-	6.50
USD	10/01/2013	6.56
USD	03/02/2013	6.30

このテーブルは、重複しない間隔のセットを定義し、開始日は **Change Date** と呼ばれ終了日は次の間隔の開始点によって定義されています。しかし、終了日が専用の列に明示的に保存されていないため、そのような列を作成し、新しいテーブルが間隔のリストとなるようにする必要があります。

次の手順を実行します。

1. 新しいアプリを作成し、名前を付けます。
2. **データ ロード エディター**で新しいスクリプト セクションを追加します。
3. 次のインライン テーブルを追加します。**Change Date** 列の日付がローカル日付書式と同じ書式になっていることを確認します。

```
In_Rates:
LOAD * Inline [
Currency,Change Date,Rate
EUR,,8.59
EUR,28/01/2013,8.69
EUR,15/02/2013,8.45
USD,,6.50
USD,10/01/2013,6.56
USD,03/02/2013,6.30
];
```

4. 参照する時間幅を決定します。その時間幅の開始点は、日付の最初の日の前であり、終了点は最終日の後でなければなりません。

スクリプトの一番上に以下を追加します。

```
Let vBeginTime = Num('1/1/2013');
Let vEndTime = Num('1/3/2013');
Let vEpsilon = Pow(2,-27);
```

5. ソースデータをロードし、空の日付を前の手順で定義した時間幅の開始点に変更します。変更した日付は、**"From Date"** としてロードする必要があります。

まず **Currency** に応じてテーブルをソートしますが、最新の日付が一番上に来るように **"From Date"** を降順でソートします。

次の **In_Rates** テーブルを追加します。

```
Tmp_Rates:
LOAD Currency, Rate,
Date(If(IsNum([Change Date]), [Change Date], $(#vBeginTime))) as FromDate
Resident In_Rates;
```

6. **"To Date"** を計算する2度目のデータパススルーを実行します。現在のレコードに前のレコードとは異なる通貨がある場合、それは新しい通貨の最初のレコードとなる(だが間隔は最後)ため、手順 1 で定義した時間幅の終了日を使用する必要があります。通貨が同一の場合は、前のレコードの **"From Date"** を取得して時間の小さい単位を差し引き、この値を現在のレコードの **"To Date"** として使用します。

次の **Tmp_Rates** テーブルを追加します。

```
Rates:
LOAD Currency, Rate, FromDate,
Date(If( Currency=Peek('Currency'),
Peek('FromDate') - $(#vEpsilon),
$(#vEndTime)
)) as ToDate
Resident Tmp_Rates
Order By Currency, FromDate Desc;
```

```
Drop Table Tmp_Rates;
```

これで、スクリプトは次のようになります。

```

Let vBeginTime = Num('1/1/2013');
Let vEndTime = Num('1/3/2013');
Let vEpsilon = Pow(2,-27);

In_Rates:
LOAD * Inline [
Currency,Change Date,Rate
EUR,,8.59
EUR,28/01/2013,8.69
EUR,15/02/2013,8.45
USD,,6.50
USD,10/01/2013,6.56
USD,03/02/2013,6.30
];

Tmp_Rates:
LOAD Currency, Rate,
    Date(If(IsNum([Change Date]), [Change Date], $(#vBeginTime))) as FromDate
Resident In_Rates;

Rates:
LOAD Currency, Rate, FromDate,
    Date(If( Currency=Peek('Currency'),
        Peek('FromDate') - $(#vEpsilon),
        $(#vEndTime)
    )) as ToDate
Resident Tmp_Rates
Order By Currency, FromDate Desc;

Drop Table Tmp_Rates;

```

スクリプトは、次の方法でソーステーブルを更新します。

更新されたソーステーブル

通貨	Rate	FromDate	ToDate
EUR	8.45	15/02/2013	vEndTime
EUR	8.69	28/01/2013	14/02/2013 23:59:59
EUR	8.59	vBeginTime	28/01/2013 23:59:99
USD	6.30	03/02/2013	vEndTime
USD	6.56	10/01/2013	2/02/2013 23:59:59
USD	6.50	vBeginTime	9/01/2013 23:59:59

アプリに、テーブルが次のように表示されます。

データのプレビュー

通貨	Rate	FromDate	ToDate
EUR	8.45	15/02/2013	01/03/2013

通貨	Rate	FromDate	ToDate
EUR	8.69	28/01/2013	14/02/2013
EUR	8.59	01/01/2013	28/01/2013
USD	6.30	03/02/2013	01/03/2013
USD	6.56	10/01/2013	2/02/2013
USD	6.50	01/01/2013	9/01/2013

`Intervalmatch` メソッドを使用することにより、既存の日付との比較にこのテーブルを使用することができます。

8.17 階層データのロード

階層の深さが一定でない n レベルの階層は、地理的もしくは組織的次元をデータで表す場合などによく使用されます。

これらの種類の階層は、通常隣接するノードテーブルに保存されます。つまり、テーブルでは、それぞれのレコードはノードと一致し、親ノードへの参照を含む項目を持っています。

ノードテーブル

NodeID	ParentNodeID	タイトル
1	-	General manager
2	1	Region manager
3	2	Branch manager
4	3	Department manager

このようなテーブルでは、ノードは 1 つのレコードにのみ保存されますが、子供をいくつでも持つことができます。当然のことながら、テーブルには、ノードの属性が記述された追加項目が含まれている可能性があります。

隣接するノードテーブルは、メンテナンスには役立ちますが、毎日の業務には適していません。そこでクエリと分析では、他の表現を使用します。展開されたノードテーブルは一般的な表現の 1 つで、階層の各レベルは別個の項目に保存されます。展開されたノードテーブルのレベルは、ツリー構造などで容易に使用できます。

`hierarchy` のキーワードをデータロードスクリプトで使用すると、隣接するノードテーブルを展開ノードテーブルに変換できます。

```
Hierarchy (NodeID, ParentNodeID, Title, 'Manager') LOAD
  NodeID,
  ParentNodeID,
  Title
FROM 'lib://data/hierarchy.txt' (txt, codepage is 1252, embedded labels, delimiter is ',',
msq);
```

展開されたノードテーブル

NodeID	ParentNodeID	タイトル	Title1	Title2	Title4	Title4
1	-	General manager	General manager	-	-	-
2	1	Region manager	General manager	Region manager	-	-
3	2	Branch manager	General manager	Region manager	Branch manager	-
4	3	Department manager	General manager	Region manager	Branch manager	Department manager

展開ノードテーブルの問題は、検索や選択を行う際に対象となるレベルを知っておく必要があり、レベル項目を簡単に使用できないという点にあります。この問題を解決する1つの表現が先祖テーブルで、ブリッジテーブルとも呼ばれます。

先祖テーブルには、データで見つかった子供と先祖の関係をすべて記録するレコードが1つあり、子供と先祖のキーと名前を含みます。つまり、すべてのレコードは特定のノードがどのノードに属するかを示します。

hierarchybelongsto キーワードをデータロードスクリプトで使用すると、隣接するノードテーブルを先祖テーブルに変換できます。

8.18 ユーザー独自のマップデータのロード

マップビジュアライゼーションを作成するには、アプリのデータに接続している地理情報にアクセスする必要があります。

Qlik Sense では次のものを使用できます。

- マップレイヤーに場所を配置するための、項目内の名前データ。
- WGS-84 のジオポイントが含まれた項目 (緯度と経度)。
- KML ファイルなど、地理データソースのジオポイント、ポリゴン、またはラインが含まれた項目。
- GeoJSON、LineString、または MultiLineString フォーマットのジオデータが含まれた項目。
- WGS-84 以外の座標系が含まれた項目 (カスタム マップをベース マップとして使用する場合)。

データプロファイリングを有効にした状態で [データマネージャー] でマップデータをロードすると、データプロファイリングサービスによって国名、都市名、および緯度と経度の項目が特定されて、対応するジオメトリが新しい項目にロードされます。[データロードエディター] を使用すると、必要に応じて、座標の項目を1つの項目に統合でき、便利です。

マップビジュアライゼーションの項目で対応している名前データ

マップビジュアライゼーションでは、項目内の名前データを使用してマップレイヤーに場所が配置されます。以下の場所のタイプを使用できます。

- 大陸の名前
- 国名

- ISO alpha 2 国コード
- ISO alpha 3 国コード
- 州名などの第一レベルの行政区画名
- 第二レベルの行政区画名
- 第三レベルの行政区画名
- 第四レベルの行政区画名
- 都市、村や人口の多い場所の名前
- 郵便番号
- IATA 空港コード
- ICAO 空港コード



使用可能な場所は、国によって異なる場合があります。名前付きの場所が使用できない場合、座標またはエリアデータを場所として使用します。

Qlik Sense マッピング内の境界線の配置と国の命名において、認められた手法とベストプラクティスを使用する、認識された項目リーダーから取得したマッピングデータと場所データを使用します。

Qlik Sense ユーザーは、自身の独立した背景マッピングを統合できるようになります。標準マッピングが適さない場合、**Qlik Sense** では、顧客が提供する背景マッピング、境界線およびエリアをロードすることもできます。

KML ファイルからのポイントデータおよびエリアデータのロード

[**データマネージャー**] および [**データロードエディター**] を使用して、KML ファイルのデータをマップに追加できます。データ選択ダイアログでは、データが含まれていない場合でも、すべての項目が選択されています。例えば、KML ファイルにはエリアデータが含まれますが、ポイントデータは含まれません。空のポイント項目またはエリア項目が含まれた KML ファイルのデータを **Qlik Sense** に追加する場合、空の項目を除外できるため、データの無いマップ軸を作成するというリスクを避けることができます。

KML 項目の項目をマップレイヤーに追加するときに、名前項目に意味のある名前データが含まれている場合は、その項目をレイヤーの軸として追加する必要があります。その後、エリア項目またはポイント項目を [**場所項目**] として追加する必要があります。レイヤー内でデータを視覚化する仕組みには違いがなく、名前項目内のテキストがツールヒントとして表示されます。



KML ファイルにポイントデータ、ラインデータ、またはエリアデータが含まれていない場合は、そのファイルからデータをロードすることはできません。KML ファイルが破損している場合はエラーメッセージが表示され、データをロードすることができません。

データプロファイリングを使用したマップデータのロード

データプロファイリングを有効にした状態で [**データマネージャー**] で [**データの追加**] を使用して地理データをロードすると、**Qlik Sense** はデータに次のものが含まれているかどうかを認識するように試みます。

- ユーザーのデータからの国と都市の名前
- 都市のような 1 つの場所の地理情報 (経度、緯度)

- 地域や国を表すエリア情報 (ジオポイントのポリゴン)

認識すると、地理情報を含んだ新しい項目が自動的に作成されます。



[データの追加]を使用する場合は、データプロファイリングを有効化する必要があります。これはデフォルトの選択です。データプロファイリングを無効化すると、地理データが検出されず、地理情報を含む新しい項目が作成されません。

データの準備中に都市が認識されると、新しい項目にはジオポイントが含まれ、国が認識されると、新しい項目にはエリアの多角形データが含まれます。この項目は **<data field>_GeoInfo** と命名されます。例えば、データに都市名を含む **Office** という名前の項目が含まれると、**Office_GeoInfo** という名前のジオポイントを含む項目が作成されます。



Qlik Sense はデータのサブセットを分析して都市や国を含む項目を認識します。マッチングが **75 %** 未満の場合は、地理情報を持つ項目は作成されません。項目が地理データとして認識されない場合は、項目の種類を手動で地理データに変更できます。

詳細については、「項目の種類を変更する (page 53)」を参照してください。

地理情報が含まれる項目は、**[関連付け]** プレビュー パネルまたは **[テーブル]** ビューでジオポイントまたは多角形データを表示しません。代わりに、これらのデータは総称的に **[GEO DATA]** として示されます。これにより、**[関連付け]** および **[テーブル]** ビューの表示速度が向上します。ただし、**[シート]** ビューでビジュアライゼーションを作成する場合は、これらのデータを使用できます。

ポイントデータのロードと書式設定

ポイントデータ(座標)を使用して、マップを作成できます。次の2つの形式がサポートされています。

- ポイントデータは2つの項目(緯度と経度)に保存されます。ポイントレイヤーの**[緯度]**項目と**[経度]**項目で、ポイントレイヤーに項目を追加できます。必要に応じて、これらの項目を1つの項目に統合できます。1つの項目に統合するには、次のようにします。
 - **[データの追加]** でデータプロファイリングを有効にしてテーブルをロードした場合は、緯度と経度の項目が認識され、自動的にジオポイント項目が作成されます。
 - データロードスクリプトを使用してデータをロードした場合は、関数 `GeoMakePoint()` を使用して、**[x, y]**形式のポイントデータを含む1つの項目を作成する必要があります。詳細については、「例: データロードスクリプトを使用した、個別の緯度列と経度列からのポイントデータのロード (page 266)」を参照してください。
- ポイントデータは1つの項目に保存されます。各ポイントは、**x**と**y**座標: **[x, y]**配列として指定されます。地理的座標の場合は**[経度, 緯度]**に該当します。この形式を使用して**[データロードエディター]**でデータをロードする場合、ポイントデータ項目に**\$geopoint;**のタグをつけることをお勧めします。詳しくは、次を参照してください。例: データロードスクリプトを使用した、単一の列からのポイントデータのロード (page 266)。

下記の例では、複数のファイルにある企業のオフィスの所在地に関する同じデータが2つの異なる形式で含まれていると想定します。

例: データロードスクリプトを使用した、個別の緯度列と経度列からのポイントデータのロード
Excel ファイルには、各オフィスに関する次のコンテンツが含まれています。

- オフィス
- 緯度
- 経度
- 従業員数

ロードスクリプトは次のようになります。

```
LOAD
Office,
Latitude,
Longitude,
Employees
FROM 'lib://Maps/Offices.xls'
(biff, embedded labels, table is (Sheet1$));
```

項目 Latitude と Longitude のデータを組み合わせて、そのポイントに新しい項目を定義します。

スクリプトを実行し、マップビジュアライゼーションを作成します。ポイント軸をマップに追加します。

LOAD コマンドの上の次の文字列を追加して、スクリプトに軸 Location を作成できます。

```
LOAD *, GeoMakePoint(Latitude, Longitude) as Location;
```

関数 GeoMakePoint() は、緯度データと経度データを結合します。

項目 Office に \$geoname というタグを付け、その項目がジオポイントの名前として認識されるようにすることをお勧めします。**LOAD** コマンドの最後の文字列の後に次の行を追加します。

```
TAG FIELDS Office WITH $geoname;
```

完全なスクリプトは次のようになります。

```
LOAD *, GeoMakePoint(Latitude, Longitude) as Location;
LOAD
Office,
Latitude,
Longitude,
Employees
FROM 'lib://Maps/Offices.xls'
(biff, embedded labels, table is (Sheet1$));
```

```
TAG FIELDS Office WITH $geoname;
```

スクリプトを実行し、マップビジュアライゼーションを作成します。ポイント軸をマップに追加します。

例: データロードスクリプトを使用した、単一の列からのポイントデータのロード
Excel ファイルには、各オフィスに関する次のコンテンツが含まれています。

- オフィス
- 場所
- 従業員数

ロードスクリプトは次のようになります。

```
LOAD
Office,
Location,
Employees
FROM 'lib://Maps/Offices.xls'
(biff, embedded labels, table is (Sheet1$));
```

項目 **Location** にはポイントデータが含まれているため、**\$geopoint** のタグを付けてポイントデータ項目として認識させることをお勧めします。項目 **Office** に **\$geoname** というタグを付け、その項目がジオポイントの名前として認識されるようにすることをお勧めします。**LOAD** コマンドの最後の文字列の後に次の行を追加します。

```
    TAG FIELDS Location WITH $geopoint;
    TAG FIELDS Office WITH $geoname;
```

完全なスクリプトは次のようになります。

```
LOAD
Office,
Location,
Employees
FROM 'lib://Maps/Offices.xls'
(biff, embedded labels, table is (Sheet1$));
    TAG FIELDS Location WITH $geopoint;
    TAG FIELDS Office WITH $geoname;
```

スクリプトを実行し、マップ ビジュアライゼーションを作成します。ポイント軸をマップに追加します。

8.19 データクレンジング

さまざまなテーブルからデータをロードする際には、同じ情報を示す項目値に必ずしも同じ名前が付けられているわけではないという点に注意してください。この一貫性の欠如は、余計な手間がかかるだけでなく、関連性を見落とす原因にもなるため、対処が必要です。この問題は、項目値を比較するマッピング テーブルを作成することで解決できます。

マッピング テーブル

mapping load または **mapping select** からロードされたテーブルは、他のテーブルとは異なる方法で処理されます。これらのテーブルはメモリの別の領域に保存され、スクリプトの実行中にマッピング テーブルとしてのみ使用されます。また、スクリプトの実行後は、自動的に削除されます。

ルール:

- マッピング テーブルは、1 列目に比較値、2 列目にマッピング値を含む 2 列構成でなければなりません。
- 2 つの列に名前を付ける必要がありますが、名前と列に含まれる値に関連性はありません。これらの列の名前は、通常の内部テーブルの項目名とは一切関係がありません。

マッピング テーブルの使用

国をリストした複数のテーブルをロードすると、1 つの国の名前が複数の異なる名前で表示されることがあります。この例では、米国が **US**、**U.S.**、**United States** としてリストされています。

連結されたテーブル内で、米国を示す3つの異なるレコードが存在しないようにするために、表示されたものに類似したテーブルを作成し、それをマッピング テーブルとしてロードします。

スクリプト全体は、次のようになります。

```
CountryMap:
Mapping LOAD x,y from MappingTable.txt
(ansi, txt, delimiter is ',', embedded
labels);
Map Country using CountryMap;
LOAD Country, City from CountryA.txt
(ansi, txt, delimiter is ',', embedded labels);
LOAD Country, City from CountryB.txt
(ansi, txt, delimiter is ',', embedded labels);
mapping ステートメントは、MappingTable.txt ファイルを CountryMap という名前のマッピング テーブルとしてロードします。
```

map ステートメントは、ロード済みの *CountryMap* マッピング テーブルを使用して *Country* 項目のマッピングを有効にします。

LOAD ステートメントは、テーブル *CountryA* および *CountryB* をロードします。これらのテーブルは、*Country* を含め、同じ項目のセットを持っているため連結され、それぞれ項目値がマッピング テーブルの1列目の項目値と比較されます。項目値 *US*、*U.S.*、*United States* が検出され、マッピング テーブルの2列目の値 (*USA*) に置き換えられます。

自動マッピングは、項目が *Qlik Sense* テーブルに保存される一連のイベントの最後に実行されます。典型的な **LOAD** または **SELECT** ステートメントでは、イベントの順序は概ね次のようになります。

1. 数式の評価
2. **as** による項目の名前変更
3. **alias** による項目の名前変更
4. テーブル名の修飾 (該当する場合)
5. 項目名が一致する場合はデータのマッピング

つまり、マッピングは数式の一部として項目名が出現するたびに行われるのではなく、*Qlik Sense* テーブルの項目名に値が保存される際に実行されます。

マッピングを無効にするには、**unmap** を使用します。

数式レベルのマッピングの場合は、**applymap** 関数を使用します。

サブstring レベルのマッピングの場合は、**mapsubstring** 関数を使用します。

8 Insight Advisor の論理 モデルのカスタマイズ

ビジネス ロジックを使用すると、クエリから洞察を生成するときに Insight Advisor がアプリデータを解釈する方法をカスタマイズできます。

ビジネスロジックは、Insight Advisor がデータを解釈する方法を定義し、データモデルの値の代替用語を次のように処理します。

- Insight Advisor 検索 ベースの分析
- Insight Advisor Chat
- Associative Insights

Insight Advisor はデータモデル内の項目の関係と使用を理解するために、Qlik cognitive engine と学習された参照元に依存しています。オプションで、Insight Advisor がアプリに使用する論理モデルをカスタマイズできます。ビジネス ロジックに語彙を追加して、ユーザーが Insight Advisor にクエリを実行するときに使用する可能性のある代替用語を Insight Advisor が処理できるようにすることができます。

Qlik Sense では、ビジネス ロジックの次の領域をカスタマイズできます。

- **論理 モデル:** モデルで使用される項目とグループをカスタマイズし、パッケージを作成し、項目階層を定義し、動作を設定します。
ビジネス ロジックを使用した *Insight Advisor* の論理 モデルの構築 (page 269)
- **語彙:** 用語を追加し、それらをデータの項目と値に関連付けて、ユーザーが Insight Advisor にクエリを実行するときに代替用語を使用できるようにします。
Insight Advisor の語彙の作成 (page 285)

ビジネス ロジック オプションは、**[準備]** タブで使用できます。

8.20 ビジネス ロジックを使用した Insight Advisor の論理 モデルの構築

Insight Advisor は、学習された参照元に基づいた論理モデルを使用して、どの項目が洞察の対象となるかを判断します。ビジネス ロジックで、アプリに独自の論理モデルを定義できます。

Insight Advisor はデータモデル内の項目の関係と使用を理解するために、Qlik cognitive engine と学習された参照元に依存しています。必要に応じて、論理モデルをカスタマイズして、Insight Advisor の結果を改善できます。**[準備]** タブの**[ビジネスロジック]**で、**論理モデル**をカスタマイズできます。



アプリでビジネス ロジックが有効になっている場合、参照元機械学習はアプリで無効になります。

論理 モデルの理解

アプリの論理モデルは、Insight Advisor がビジュアライゼーションを生成するときに使用する概念モデルです。アプリの基礎となるデータモデルから構築されます。各アプリは単一の論理モデルを持っています。項目とマスターアイテムは、論理モデルのコアコンポーネントです。それらはグループに編成されています。グループは、項目または

8 Insight Advisor の論理 モデルのカスタマイズ

マスター アイテム間の概念的な関連付けまたは関係を示します。論理モデルには、グループ間の関係についての情報も含まれています。

論理モデルは、Insight Advisor の機能に直接影響します。たとえば、ユーザーがトレンド分析を表示する項目を選択すると、Insight Advisor はプライマリカレンダーグループの一部である日付項目を見つけようとします。項目が[売上高]の場合、Insight Advisor は[注文日]などの項目を[従業員の生年月日]項目よりも優先しません。

ビジネスロジックは、システムが自然言語クエリの項目を選択する方法にも影響します。例えば、[製品名]と[製品コード]の項目は1つのグループとしてグループ化されます。「製品別売上高を表示」がクエリで使用されている場合、[製品名]はグループに適しているため、「製品」に使用されます。

ビジネスロジックに使用されるデフォルトの論理モデルは、スタースキーマです。ビジネスロジックを使用すると、スタースキーマが最適でない場合に、アプリのさまざまなモデリングを構築できます。ビジネスロジックは、準加法メジャーまたは非常に大きなメジャーテーブルを含む論理モデルでの集計の制約にも役立ちます。これにより、Insight Advisor でのアプリデータの探索が改善されます。

論理モデルのカスタマイズ

アプリの論理モデルをカスタマイズするために、**論理モデル**は次のセクションに分かれています。

- **概要:**概要には、ビジネスロジックの概要が表示されます。**[項目とグループ]**、**[パッケージ]**、**[階層]**、または**[動作]**のカードをクリックすると、対応するセクションが開きます。
- **項目とグループ:** **[項目とグループ]**を使用すると、論理モデルで項目とマスターアイテムが属するグループを定義できます。
- **パッケージ:** **パッケージ**を使用すると、関連するグループのコレクションを作成できます。これにより、同じパッケージ内にはないグループが一緒に使用されるのを防ぐことができます。
- **階層:****階層**を使用すると、グループ間のドリルダウン関係を定義できます。
- **動作:****動作**を使用すると、項目間の優先関係または拒否関係を指定できます。動作は、必要な選択を強制することもできます。
- **カレンダー期間:****カレンダー期間**を使用すると、Insight Advisor の既定の分析期間を作成できます。

論理モデルをカスタマイズするには、以下を実施します。

1. アプリのビジネスロジックのカスタマイズを有効にします。
2. 項目とグループを定義します。
項目とグループの定義 (page 271)
3. 任意でグループをパッケージに追加します。
パッケージを使用した論理モデル範囲の設定 (page 275)
4. 任意でグループ間の階層を定義します
階層を使用したドリルダウン分析の作成 (page 276)
5. 任意で動作を適用します。
論理モデルへの動作の適用 (page 276)
6. オプションで、カレンダー期間を作成します。
カレンダー期間を使用した分析期間の定義 (page 279)

論理モデルを既定値にリセットできます。一時的にビジネスロジックを無効にすることもできます。

カスタム ビジネス ロジックの有効化

次の手順を実行します。

1. アプリで、[準備] をクリックし、[論理 モデル] を選択します。
2. [続行] をクリックします。

これで、カスタム ビジネス ロジックがアプリで有効になります。参照元機械学習は無効になりました。

ビジネス ロジックのリセット

論理 モデルをデフォルトのモデルにリセットできます。リセットすると、カスタム ビジネス ロジックが無効になり、Insight Advisor で参照元機械学習が可能になります。

次の手順を実行します。

1. [論理 モデル] で、[既定値にリセット] をクリックします。
2. [確認] をクリックします。

ビジネス ロジックの無効化

カスタム論理 モデルを無効化できます。ビジネス ロジックのリセットとは異なり、カスタム ビジネス ロジックは後で再び有効にすることができます。ビジネス ロジックが無効になっている間は、アプリの既定のビジネス ロジックが使用されます。

次の手順を実行します。

1. [論理 モデル] で、[ロジックを無効化] をクリックします。
2. [確認] をクリックします。

項目 とグループの定義

項目 とグループは、ビジネス ロジックにおける論理 モデルの主要なコンポーネントです。項目 とマスター アイテムが属 するグループを定義 することで、Insight Advisor がそれらを使用 する方法を定義 できます。

ビジネス ロジックがどのように項目 とグループを処理 するかは、**論理 モデルの項目 とグループ**セクションで定義 します。項目 とマスター アイテムをグループ化 して、Insight Advisor 分析の関係を示 することができます。グループ内の 各アイテムについて、分析で Insight Advisor がどのように処理 するかを定義 できます。

グループを使用 すると、項目 とマスター アイテムを関連 する概念的なグループに編成 できます。たとえば、顧客に 関連 するすべての項目を、ロード元のデータモデルのテーブルに 関係なく、1 つのグループにグループ化 できます。Insight Advisor はこの情報を使用 して、ビジュアライゼーションで一緒に表示 する項目を決定 します。グループを使用 してパッケージを作成 することもできます。これにより、関連 するグループのみを一緒に使用するよう に Insights の範囲が制限 されます。次の 3 種類のグループがあります。

- **軸**: 軸グループは通常、軸として分類 される項目で構成 されます。軸グループには、メジャーまたは日付として分類 された項目を含める ことができます。


- **メジャー**: メジャーグループは、関連するメジャー項目で構成されます。メジャーのみがメジャーグループに属することができます。
- **カレンダー**: カレンダーグループには、論理モデル内の時間軸が含まれます。カレンダーグループには軸のみを含めることができ、少なくとも1つの時間項目(日付、タイムスタンプ、年など)が必要です。年、月、日など、カレンダーを定義する個別の項目がある場合、カレンダーグループは便利です。また、会計四半期や会計年度など、他のデータ関連の項目もグループ化できる場合は、グループ化します。

既定では、項目とマスターアイテムはグループ順に並べられています。これを無効にすると、**[グループ内]**という追加項目がテーブルに追加されます。

グループの定義に加えて、グループに属する個々の項目とマスターアイテムのプロパティを定義することもできます。

グループの作成


グループを作成した後は、グループの種類を変更できません。グループの名前を変更し、グループに項目を追加または削除できます。グループは削除することもできます。グループを削除すると、グループ内のすべてのアイテムがグループ解除されます。

既存のグループを編集するには、グループ行の中、または**[グループ内]**列のグループ名の後にあるをクリックします。

次の手順を実行します。

1. **[グループを作成]** をクリックします。
2. グループ名を入力します。
3. グループの種類を選択します。
4. **[利用可能な項目]** からグループに項目を追加します。
5. **[作成]** をクリックします。

項目とマスターアイテムの定義

[項目とグループ] は、アプリの項目とマスターアイテム、およびそれらが属するグループを含むテーブルで構成されます。項目とマスターアイテムのプロパティを編集するには、そのアイテムの行にある列の値を調整します。追加のオプションについては、行のをクリックします。次の操作が可能です。


- **アイテムを移動**: アイテムを別のグループに移動します。
- **動作を作成**: 現在のアイテムが属するグループの動作を作成します。
- **グループ解除**: 現在のグループからアイテムを削除します。グループ解除されたアイテムは Insight Advisor から除外されます。

行のチェックボックスを使用して複数の行を選択し、複数のアイテムに同じ変更を加えることもできます。この方法で選択すると、プロパティ設定とオプションがテーブルの上に表示されます。

項目とマスターアイテムのプロパティ

テーブルは次の列に分かれています。

- 名前
- グループ
- 可視性
- 分類
- データ値のルックアップ
- 既定の集計
- 好調なトレンド
- お気に入り
- 全体的な集計
- 既定期間の粒度

[名前] は項目名をリストします。**[グループ]** には、項目が属するグループ名がリストされます。次のセクションでは、他の項目のさまざまな値と設定について概説します。 をクリックすると、テーブルに表示する列を設定できます。

可視性

可視性は、Insight Advisor でアイテムを使用できるかどうかを制御します。可能な値は 2 つあります。

- **表示**: このアイテムは Insight Advisor で使用できます。
- **非表示**: このアイテムは Insight Advisor では使用できません。非表示の項目は、データ値のルックアップに対して有効にしないでください。

[項目とグループ] で非表示のアイテムをすべて非表示にするには、**[表示のみを表示]** を選択します。

分類

分類は、属性が分析で果たすことができる既定のロールを定義します。次のタイプを使用して、項目とグループを分類できます。

- **軸**: 軸としてのみ使用される項目
- **メジャー**: メジャーとしてのみ使用される項目。
- **ブール値**: バイナリ軸。
- **日付**: 日付を含む一時的な軸。
- **タイムスタンプ**: タイムスタンプを含む一時的な項目。
- **年**: 年のデータを含む一時的な軸。
- **週**: 週のデータを含む一時的な軸。
- **四半期**: 四半期のデータを含む一時的な軸。
- **月**: 月のデータを含む一時的な軸。
- **平日**: 曜日のデータを含む一時的な軸。短い形式(月、火)、長い形式(月曜日、火曜日)、または 1~7 の数値。
- **月日**: 月の日を示す 1~31 の数値を含む一時的な軸。
- **年の日数**: 年間の日数 (1~366) を含む一時的な軸。
- **時間**: 時間のデータを含む一時的な軸。
- **メール**: メールアドレスを含む軸。
- **住所**: 住所を含む軸。

- **国**: 国名を含む軸。
- **州/地方**: 州や地方などの第 1 レベルの行政区域を表す軸。
- **都市**: 都市を表す軸。
- **ジオポイント**: 地理データを含む軸。
- **地理ポリゴン**: 地理ポリゴンデータを含む軸
- **地理的**: 国や地域などの地理的な場所を表す軸。
- **郵便番号**: 郵便番号を含む軸。
- **経度**: 経度データを含む軸。
- **緯度**: 緯度データを含む軸。
- **パーセンテージ**: 雇用率やインフレなどのパーセント値を表すメジャー項目。
- **財政**: 収入、費用、給与などの財政 メジャー。
- **序数**: 値に固有の順序がある軸。
- **一時的**: 時間に関連する軸。

あるクエリでは軸と見なされ、別のクエリではメジャーと見なされる項目がある場合があります。ベストプラクティスとして、項目の別のユース ケース用に 2 番目の項目またはマスター アイテムを作成することをお勧めします。

データ値のルックアップ

[データ値のルックアップ] は、自然言語 クエリを実行するときに Insight Advisor が項目から値をルックアップできるかどうかを制御します。

データ値のルックアップが有効になっている項目の数を減らすと、誤検知の結果を回避し、クエリ時間を短縮できます。たとえば、データモデルに 3 つの項目名があるとします。名、姓、フルネーム。3 つの項目すべてで**データ値のルックアップ**が有効になっている場合、ユーザーが「Drew」を検索すると、3 つの項目すべてからの結果で混乱させられる可能性があります。



データ値のルックアップはメジャーと非表示項目では無効にしておく必要があります。

既定の集計

既定の集計は、Insight Advisor のメジャーの標準集計を設定します。次の集計を使用できます。

- **sum**
- **avg**
- **min**
- **max**
- **Count**
- **countDistinct**

項目に既定の集計がある場合、Insight Advisor はメジャーとして使用するとき常にその集計を適用します。ユーザーは Insights チャートを編集して、Insight Advisor で集計を別のタイプに変更できます。

既定の集計をマスター アイテムに割り当てることはできません。

好調なトレンド

好調なトレンドはメジャーの望ましいトレンドが増加または減少するかを設定します。既定では、メジャーの好調なトレンドは増加します。次のトレンドを使用できます。

- 上
- 下

好調なトレンドは、結果をランク付けするために Insight Advisor によって作成されたビジュアライゼーションで使用されます。

お気に入り

お気に入りは、メジャーが Insight Advisor にとって特に興味深いものであることを示します。Insight Advisor は、**[生成]**を使用する場合など、ユーザーのクエリや選択なしでビジュアライゼーションを生成するときに、このメジャーをより頻繁に使用します。

全体的な集計

全体的な集計は、マスター メジャーを含むクエリのナラティブで使用する集計を Insight Advisor が自分で決定できない場合に、どの集計を使用する必要があるかを Insight Advisor に示します。一部のマスター メジャーは複雑な数式を使用するため、Insight Advisor は、ストーリーでマスター メジャーの寄与を計算するときに使用する集計を認識していない場合があります。全体的な集計を設定すると、Insight Advisor はこれらの複雑なマスター メジャーのストーリーを計算できます。

マスター メジャーの全体的な集計のみを設定できます。

既定期間の粒度

Insight Advisor で分析を生成するときにカレンダーグループの日付項目で使用する、既定の期間の粒度、例えば月や年などを設定できます。粒度は、自動カレンダーから取得するか、同じカレンダーグループから粒度を含む項目値を選択できます。

既定のカレンダー期間の動作に、既定のカレンダー設定を使用できます。既定の

パッケージを使用した論理 モデル範囲の設定

パッケージを使用すると、論理モデルの特定の関心領域の洞察の範囲を定義および制限できます。

パッケージは関連するグループのコレクションです。Insight Advisor は結果を生成するときに、同じパッケージ内のグループのみを一緒に使用します。グループは複数のパッケージに属することができます。論理モデル パッケージは、**パッケージ**で作成および管理されます。

例えば、データモデルに個別のビジネス範囲のテーブルが含まれている場合があります。テーブル間に接続がある場合、売上高の日付項目が、サポートの日付項目を持つ Insights チャートで使用される可能性があります。売上高グループとサポートグループを別々のパッケージに入れることにより、それらのビジネス範囲の洞察が、関連するパッケージの項目のみを使用するようにすることができます。

パッケージの使用はオプションです。パッケージを定義しない場合、すべての項目とグループは分析のために同じパッケージにあると見なされます。

*** をクリックして **[編集]** または **[削除]** を選択すると、パッケージを編集または削除できます。

パッケージの作成

次の手順を実行します。

1. [パッケージを作成] をクリックします。
2. パッケージ名を入力します。
3. 利用可能なグループからパッケージにグループを追加します



パッケージには、少なくとも1つのメジャーグループが含まれている必要があります。

4. [作成] をクリックします。

階層を使用したドリルダウン分析の作成

論理モデル内のどのグループが階層的な関係を持っているかを定義することで、Insight Advisor は指定された階層に沿ったブレイクダウン分析を提供し、グループ間の依存関係をより良く検出することができます。

論理モデル階層は、**階層**で作成および管理されます。階層はビジネスロジックではオプションです。

階層を使用すると、推奨事項 (製品カテゴリ別売上高、製品サブカテゴリ、製品など) の内訳をドリルダウンできます。階層は、Insight Advisor が推奨事項に地理的アイテムを混在させないようにするのも役立ちます。例えば、サプライヤの郡をサプライヤの都市と一緒に、および顧客の国と顧客の都市と一緒に階層を作成すれば、サプライヤの国と顧客の都市などの誤った組み合わせでそれらが使用されるのを防ぐことができます。

階層は、選択を行うときに依存関係を検出するための連想洞察でも使用されます。例えば、Insights チャートでスウェーデンを選択すると、[都市]項目を使用する別のチャートでスウェーデンの都市の値だけが選択されます。

⋮ をクリックして [編集] または [削除] を選択すると、階層を編集または削除できます。

階層の作成

次の手順を実行します。

1. [階層を作成] をクリックします
2. 階層名を入力します。
3. 階層の最下位レベルから最上位レベルの順序でグループを階層に追加します。
4. [作成] をクリックします。

論理モデルへの動作の適用

動作を使用すると、メジャーグループと他のグループ間の優先関係または拒否関係を設定できます。動作を使用して、Insight Advisor で値の選択を強制することもできます。

8 Insight Advisor の論理 モデルのカスタマイズ

分析で常に一緒に使用するか、分析で一緒に使用しないでください。また、グループの項目が **Insight Advisor** によって使用されているときに、常に選択された状態で含まれるようにする項目値がある場合もあります。動作を使用すると、これらの設定をグループごとに設定できます。論理モデル動作は**動作**で作成および管理されます。

論理モデルでは次の動作を使用できます。

- 必要な選択
- 関係を優先
- 関係を拒否
- 既定のカレンダー期間

*** をクリックして **[編集]** または **[削除]** を選択すると、動作を編集または削除できます。

必要な選択

必要な選択動作により、グループの項目を使用するときに含める必要がある項目値を指定できます。動作には、複数の必要な選択を含めることができます。

例えば、**国**、**人口**、**年**などの項目があるとします。**Insight Advisor** で推奨事項を生成するときに、**国**と**人口**を使用するチャートを取得し、すべての年のデータの合計をチャートに含める場合があります。必要な選択動作を使用して、**人口**または**国**の洞察を生成するときに、代わりに現在の年を使用するように洞察の結果を制限できます。

関係の優先

関係行動を優先することは、洞察を生成するときに、より頻繁に一緒に使用する必要があるグループを選択するように **Insight Advisor** を導きます。優先関係を指定すると、**Insight Advisor** は結果の生成時に優先グループを使用します。優先関係は、あいまいさがある場合に役立ちます。

優先関係は、グループが他のグループで使用されるのを妨げません。分析ですべてのグループの選択肢が等しい場合にのみ、優先グループを選択します。他のグループがより適切な分析では、代わりに非優先グループを使用できます。

例えば、**4** つのグループがあります。

- 売上
- 顧客
- 製品
- 営業担当者

売上は顧客と優先関係があります。売上を使用した内訳分析の場合、顧客は他のグループよりも優先的に選択されます。売上のトレンド分析では、代わりに製品が分析に使用される場合があります。

関係の拒否

関係の動作を拒否すると、洞察を生成するときに、**Insight Advisor** が選択したグループを一緒に使用できなくなります。これは、同じパッケージ内の一部のグループが分析で一緒に使用するのに役立たない場合に役立ちます。拒否関係を使用して、アプリのパフォーマンスに影響を与える可能性があるグループが一緒に使用される

のをブロックすることもできます。スター スキーマデータモデルでは、分析を複雑にする 1 対多および多対 1 の関係を持つグループが存在する場合があります。拒否関係は、これらのグループ間の関連付けを防ぐことができます。

拒否関係を持つグループからの洞察をユーザーが明示的に要求すると、拒否関係は無効になります。たとえば、売上高とサプライヤは拒否関係を持ちます。誰かがクエリー '*show me sales*' を実行した場合、売上高とサプライヤを含む分析は生成されません。誰かがクエリー '*show me sales by supplier*' を実行した場合、売上高とサプライヤを含む結果が生成されます。

既定のカレンダー期間

既定のカレンダー期間の動作では、選択したグループのビジュアライゼーションで既定の期間として使用されるカレンダー期間が割り当てられます。既定のカレンダー期間は、Insight Advisor がそのグループの項目のビジュアライゼーションを作成するたびに適用されます。グループには単一の既定のカレンダー期間を設定できます。既定のカレンダー期間の動作は、日付項目に設定された既定値の期間の粒度を無効にします。

カレンダー期間は通常、次の方法で分析に適用されます。

- ファクト分析とランク分析では、カレンダー期間から現在または最初に選択された期間が使用されます。カレンダー期間から選択された前または 2 番目の期間が、比較のために分析に追加されます。
- トレンド分析と同様の分析では、カレンダー期間の集計された粒度のみが使用されます。
- 他の分析では、カレンダー期間から現在または最初に選択された期間が使用されます。

例えば、クライアント満足度データを含むグループクライアント満足度があるとします。既定では、アプリのユーザーの多くは、今月のデータの表示にのみ関心があります。今月のカレンダー期間を作成し、それを既定のカレンダー期間にすることにより、クライアント満足度の Insight Advisor ビジュアライゼーションには、現在の月のデータのみが表示されます。

選択した分析タイプの既定のカレンダー期間を無視する

既定のカレンダー期間の一部またはすべてを無視するように分析タイプを構成できます。これは、既定のカレンダー期間の適用方法をより具体的に制御するのに役立ちます。例えば、ファクト分析で比較を行わずに、現在の期間のみを表示したい場合があります。

分析タイプを選択して、期間 1 (前または 2 番目に選択された期間)、期間 2 (現在または最初に選択された期間)、またはカレンダー期間の粒度を無視できます。期間 1 を無視すると、2 つの期間を比較する分析で前の期間の使用を停止するので、期間 2 のみが表示されます。期間 2 または粒度を無視すると、その分析タイプの既定で、このカレンダー期間の使用が停止します。

動作の作成

次の手順を実行します。

1. **動作を作成** をクリックします。
2. **[適用先]** でグループを選択します。
3. 動作の種類を選択します

4. 以下のいずれかを行います。

- **優先関係**または**拒否関係**を構成する場合は、その関係を適用するグループを選択します。
- **必要な選択**を構成する場合は、これが1つの値かどうかを選択します。**必要**から、項目とその必要な値を選択します。
さらに**追加**をクリックすると、必要な選択をさらに追加できます。
- **既定のカレンダー期間**を設定する場合は、カレンダーグループとそのグループからカレンダー期間を選択します。オプションで、**[期間 1 を無視する]**、**[期間 2 を無視する]**、**[粒度を無視する]**の下にある既定のカレンダー期間から、例外にする必要がある分析タイプを指定します。

5. **[作成]**をクリックします。

カレンダー期間を使用した分析期間の定義

カレンダーグループのカレンダー期間を定義して、Insight Advisor がビジュアライゼーションを作成するときに使用する時間枠を制御します。

カレンダー期間は、論理モデルの分析で使用する Insight Advisor の対象期間を定義します。例えば、すべての時間の総売上高を示す KPI は、当月の総売上高を示す KPI よりも有用性が低い場合があります。今月のカレンダー期間を作成し、それを総売上高で使用する既定期間として設定できます。論理モデルのカレンダー期間は、**カレンダー期間**で作成および管理されます。

カレンダー期間は、単一の期間または相対的な期間の比較を定義します。例えば、次のカレンダー期間を作成できます。

- 今月の値のみを表示する
- 今月と先月を比較する
- 現在の四半期と昨年現在の四半期を比較する

Insight Advisor は、チャートを作成するときにカレンダー期間を使用します。カレンダー期間により、追加の分析タイプが可能になります。**Insights** でチャートを編集するときに、カレンダー期間を選択して適用できます。

Insight Advisor がそのグループの項目とマスターアイテムのチャートを作成するときに、グループで使用される既定のカレンダー期間を設定できます。既定のカレンダー期間は、**動作**で設定されます。

カレンダー期間の作成

項目とグループで定義されたカレンダーグループからカレンダー期間を作成します。データロードスクリプトで autoCalendar 関数を使用する項目の有無にかかわらず、カレンダー期間を作成できます。



autoCalendar と同じ宣言済み項目を使用する場合は、カスタム カレンダーで **[自動カレンダーを使用]** オプションを使用できます。

自動カレンダーを使用したカレンダー期間の作成

自動カレンダーを使用してカレンダー期間を作成するときに、分析する粒度を選択します。粒度は、例えば、年の月または年の四半期です。次に、最新の値を使用するか、比較を作成するかを選択できます。

次の手順を実行します。

1. **[カレンダー期間を作成]** をクリックします。
2. カレンダーグループを選択してから、**[自動 カレンダーを使用]** を選択します。
3. カレンダー期間の名前を入力します。
4. カレンダー期間の粒度を選択します。
5. 以下のいずれかを行います。
 - 期間粒度項目の最新の値を使用するには、**[最後にソートされた値を使用]** を選択します。
 - 比較 カレンダー期間を作成するには、次のいずれかを実行します。
 - **[比較]** で、比較する期間を選択します。Insight Advisor が現在の期間ではなく最後の完了期間を使用する必要がある場合は、**[最後の完了期間]** を選択します。
 - **[カスタム]** で、**[オフセット]** で分析する期間を選択し、**[オフセットの比較]** で比較する期間を選択します。
[オフセット] と**[オフセットの比較]** は数値を使用し、0 が現在の期間です。
例えば、年の四半期を期間の粒度として使用した場合、0 は現在の期間になり、3 は 3 四半期前になります。
6. **[作成]** をクリックします。

カスタム カレンダーを使用したカレンダー期間の作成

ロードスクリプトで `autoCalendar` を使用せずに、カレンダー期間を作成できます。例えば、カスタム カレンダーがある場合や、`autoCalendar` に存在しないカスタム カレンダー フラグを使用したい場合があります。

カスタム カレンダーを使用してカレンダー期間を作成するには、カレンダーグループを選択し、集計する時間項目を選択します。次に、最新の値を使用するか、比較を作成するかを選択します。2 つの比較方法がサポートされています。

- **相対**: 前の相対期間数を含む項目を使用する相対比較。相対期間項目には、集計項目の各値の現在の日付からの相対期間を定義する数値が含まれている必要があります。次に、分析用のオフセットと比較オフセットを選択できます。
例えば、今月と先月を比較するには、集計項目として**[日付]**を選択します。次に、相対期間項目として**MonthsAgo**を選択します。**MonthsAgo** は数式を使用して、**Date** の各値が現在の月から何か月前かを評価します。現在の月が 7 月の場合、7 月の値は 0、6 月の値は 1、5 月の値は 2 というようになります。次に、オフセットに 0 を選択し、比較オフセットに 1 を選択できます。
- **フラグ**: フラグの比較では、2 つの個別のフラグ付き期間を定義する 2 つの項目を使用します。
例えば、第 1 四半期と第 2 四半期は、フラグの比較に使用される 2 つの期間である可能性があります。比較項目には、フラグが立てられた期間にある集計項目値を示すバイナリ値が含まれている必要があります。
第 1 四半期と第 2 四半期を比較するには、集計項目として**[Date]**を選択します。フラグ項目として、**InQuarter1** と**InQuarter2**を選択できます。これらの項目は、数式を使用して、値が第 1 四半期または第 2 四半期にあるかどうかを評価します。

次の手順を実行します。

1. **[カレンダー期間を作成]** をクリックします。
2. カレンダーグループを選択し、**[自動 カレンダーを使用]** チェックボックスをオフにします。

3. カレンダー期間の名前を入力します。
4. 使用する集計日項目を選択します。
5. 以下のいずれかを行います。
 - 期間粒度項目の最新の値を使用するには、**[最後にソートされた値を使用]**を選択します。
 - 比較カレンダー期間を作成するには、次のいずれかを実行します。
 - **[相対]**で、**[相対期間前]**には、**[集計日]**で選択した項目の相対期間データを含むロードスクリプトで定義された項目を選択します。分析の期間を**[オフセット]**で設定し、次に比較の期間を**[オフセットの比較]**で設定します。
[オフセット]と**[オフセットの比較]**は数値を使用し、**0**が現在の期間です。
 - **[フラグ]**で、現在の期間のフラグを含む項目を選択してから、比較期間のフラグを含む項目を選択します。
6. **[作成]**をクリックします。

制限事項

ビジネスロジックのカレンダー期間には、次の制限があります。

- 既定のカレンダー期間は、数式に特定の期間が含まれる時間ベースのマスターメジャーには適用されません。

カスタム カレンダーを使用したカレンダー期間の作成の段階的手順

この段階的手順では、カスタム カレンダー項目とフラグを使用してカレンダー期間を作成する方法を説明します。

カレンダー期間は、自動カレンダーを使用するか、ご使用のデータから個別の日付/時間項目を使用して作成できます。比較分析のために、バイナリデータを含む項目を使用して時間の枠をフラグすることもできます。

カスタム カレンダーデータで作成できるカレンダー期間には、次の3種類があります。

- 最後の並べ替え値:最後の並べ替え値のカレンダー期間には、選択した集計項目にある最近の期間が表示されます。ランク分析などの比較をサポートする Insight Advisor 分析では、最後の並べ替え値に前回の期間も表示されます。
- 相対比較:相対カレンダー期間は、現在の日付からのデータの相対期間を含む項目を使用します。これは、現在または前の期間とさらに前の期間との比較を提供します。
- フラグ比較:フラグでは、比較分析のために、バイナリデータを含むブール値として分類される項目を使用して2つの時間の枠をフラグします。



相対的な期間の比較を示すために、このアプリのデータソースには将来の日付のデータが含まれています。ロードスクリプトは、データソースから現在の日付までのデータをロードします。画像に表示される期間が変更されるため、この画像にある結果は、ユーザーの結果とは異なる場合があります。

はじめに

サンプル パッケージをダウンロードして解凍します。

[カレンダー期間の例のアプリ](#)

QVF ファイルには、次のデータファイルが含まれています。

- **TutorialCustomCalendarData.xlsx**
カレンダー期間を通じて可能な相対比較を示すために、*TutorialCustomCalendarData.xlsx* には将来の日付のデータが含まれています。アプリロードスクリプトは、ロード時に現在の日付のアプリ内データを更新します。

QVF ファイルを Qlik Sense にインポートし、XLSX ファイルをアプリに添付します。アプリをインポートしてデータファイルをアプリに添付したら、**データロードエディター**でアプリデータをロードします。

データの例

この例で使用されるデータは、次のロードスクリプトでロードされています。

```
Sales: LOAD City, Country, Customer, OrderDate, ProductID, Quantity, Sales, "Q4-2018", "Q1-2019", "Q2-2019", "Q3-2019", "Q4-2019", "Q1-2020", "Q2-2020", "Q3-2020", "Q4-2020", "Q1-2021", "Q2-2021", "Q3-2021", Month([OrderDate]) AS [Month], Year([OrderDate]) AS [Year], Day([OrderDate]) AS [Day], Dual(Year(OrderDate)&'- '&Month(OrderDate), monthstart(OrderDate)) AS [YearMonth], 12*Year(Today())+Month(Today())-12*Year(OrderDate)-Month(OrderDate) AS [MonthsAgo]
```

```
FROM [lib://AttachedFiles/TutorialCustomCalendarData.xlsx]
```

```
(ooxml, embedded labels, table is Sales) Where OrderDate <= Today(1);
```

ロードスクリプトは、年、月、日用に別々の項目を作成します。これら3つの項目は、次の計算された項目を作成するために使用されます。

- **[YearMonth]** には、年と月の情報が含まれます。これは、この例における集計のためのプライマリ項目です。
- **[MonthAgo]** は、現在の日付に対して特定の月からの月であるかどうかを計算します。

データには、データが網羅する異なる四半期の複数の項目が含まれます。これらの項目には、**[Sales]** テーブルの各値がどの会計四半期に属するかを示すバイナリデータが含まれています。

タスク

ここでは、3つの異なるカレンダー期間の作成方法を説明します。

- 自動カレンダーなしでカスタム カレンダー期間を作成する
- 相対カレンダー期間を作成する
- フラグ比較カレンダー期間の作成

最後の並べ替え値を使用したカレンダー期間の作成

最初のカレンダー期間では、最後の並べ替え値を使用して **[YearMonth]** のカレンダー期間を作成します。

次の手順を実行します。

1. 例のアプリで、**[準備]** をクリックします。
2. **[ビジネス ロジック]** で、**[論理 モデル]** を選択します。
3. **[カレンダー期間を作成]** をクリックします。
4. **[OrderDate]** を選択します。

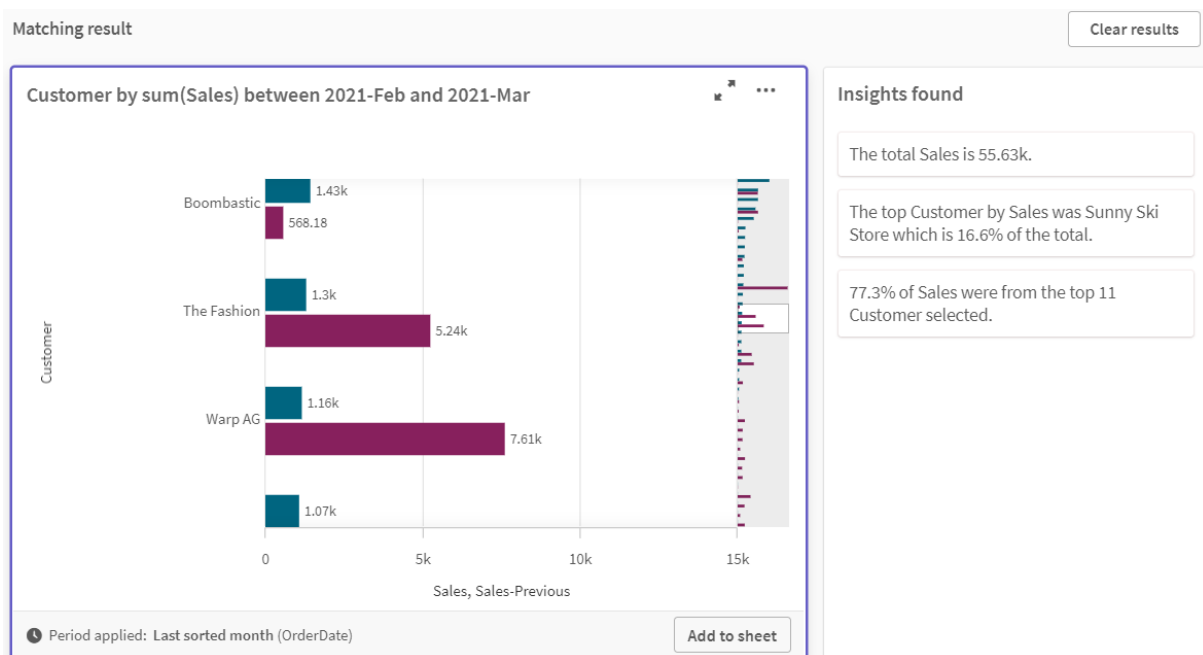
5. [カレンダー期間名] に、[最後の並べ替え月] を入力します。
6. [集計日] に、[YearMonth] を選択します。
7. [最後にソートされた値を使用] を選択します。
8. [作成] をクリックします。

結果

[Insights] に移動し、[顧客別の売上を表示] を検索します。

チャート [顧客別の sum(Sales)] を選択し、分析期間 [最後の並べ替え月] を適用します。チャートは更新され、今月と前月の比較を表示します。

[最後の並べ替え月] を [2021 年 2 月と 2021 年 3 月の間の sum(Sales) による顧客] に適用



相対比較 カレンダー期間の作成

次に、相対カレンダー期間を作成します。相対カレンダー期間には次が必要です。

- 時間の枠を含む集計項目 (年、月、四半期など)。
- 項目からの日付から今日までの日付の相対的位置を含む項目。

これらの項目から、オフセットを定義します。オフセットは相対的の差異で、選択された時間の枠にあり、2つの期間の比較における現在の日付からのものです。現在または前回の期間 ([オフセット] で 0 または 1 に設定) を最大 12 期間までの古い期間 ([オフセットを比較] で数字を 1 から 12 に設定) と比較できます。

このカレンダー期間では、集計項目として [YearMonth]、相対期間項目として [MonthsAgo] を使用します。今月と今年の同じ月を比較したいと思います。

次の手順を実行します。

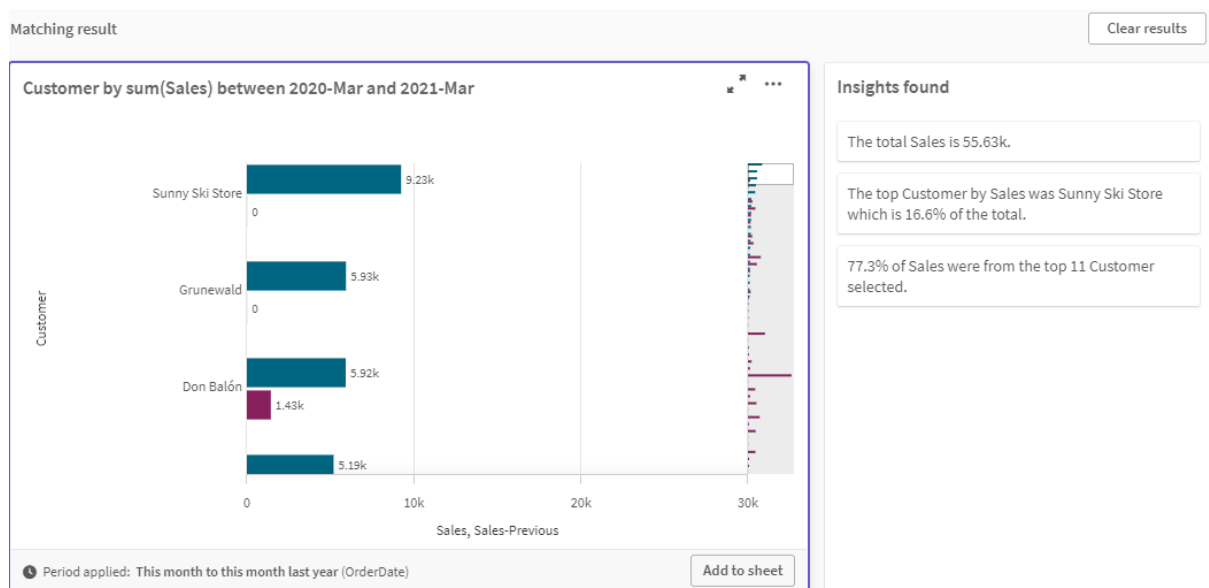
1. [カレンダー期間を作成] をクリックします。
2. [OrderDate] を選択します。
3. [カレンダー期間名] に、[今月から前年同月] を入力します。
4. [集計日] に、[YearMonth] を選択します。
5. [相対期間前] で、[MonthsAgo] を選択します。
6. [オフセット] で、0 を選択します。
7. [オフセットを比較] で、12 を選択します。
8. [作成] をクリックします。

結果

[Insights] に移動し、[顧客別の売上を表示] を検索します。

チャート[顧客別の $\text{sum}(\text{Sales})$] を選択し、分析期間 [今月から前年同月] を適用します。チャートは更新され、今月と前年同月の比較を表示します。

2020年3月と2021年3月の間の $\text{sum}(\text{Sales})$ による顧客



フラグ比較 カレンダー期間の作成

フラグ比較 カレンダー期間では、2つの項目を使用して、分析のために集計日項目から2つの区切られた期間をフラグします。

例のアプリデータには、異なる会計四半期のために区切られた項目があります。各項目にはバイナリデータがあり、対応する日付が四半期内にあるかどうかを表示します。これらと[YourMonth]を使用して、フラグ比較 カレンダー期間を作成します。

次の手順を実行します。

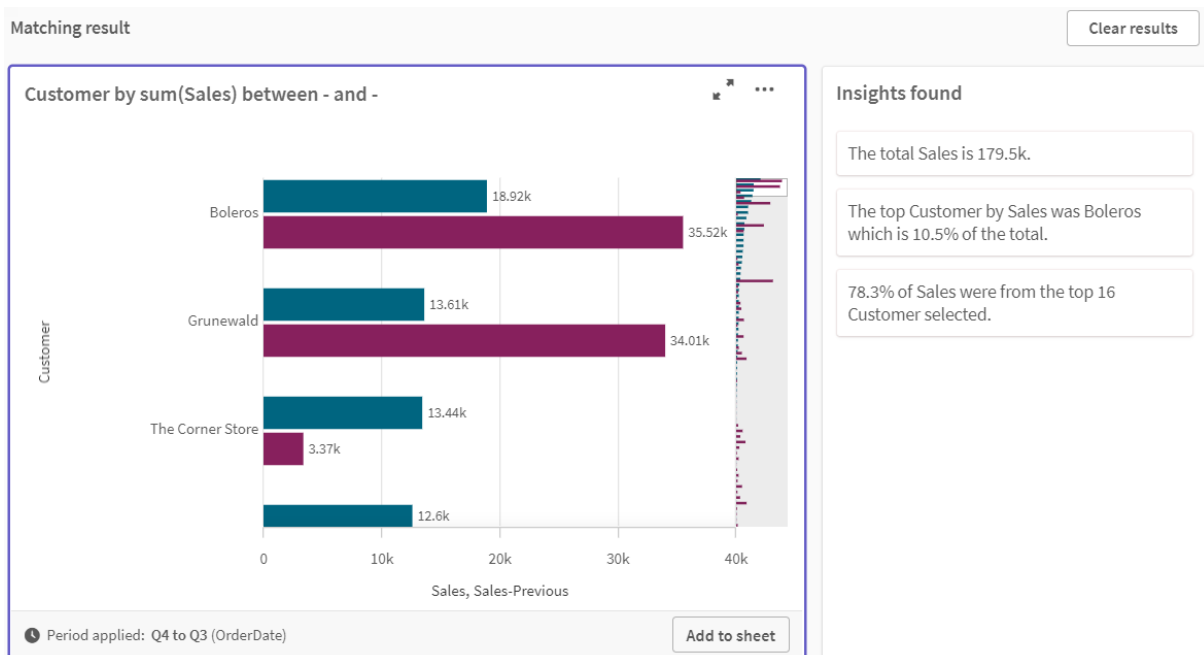
1. [カレンダー期間を作成] をクリックします。
2. [OrderDate] を選択します。
3. [カレンダー期間名] に [第 4 四半期から第 3 四半期] と入力します。
4. [集計日] に、[YearMonth] を選択します。
5. [フラグ] をクリックします。
6. [現在の期間フラグ] で、[2020 年 第 4 四半期] を選択します。
7. [比較期間フラグ] で、[2020 年 第 3 四半期] を選択します。
8. [作成] をクリックします。

結果

[Insights] に移動し、[顧客別の売上を表示] を検索します。

チャート [顧客別の $\text{sum}(\text{Sales})$] を選択し、分析期間 [第 4 四半期から第 3 四半期] を適用します。チャートは更新され、2020 年 第 4 会計四半期と2020 年 第 3 会計四半期の比較を表示します。

第 4 四半期と第 3 四半期の間の $\text{sum}(\text{Sales})$ による顧客



8.21 Insight Advisor の語彙の作成

Insight Advisor の語彙を作成できます。これによって、Insight Advisor のシノニムを定義し、特定の用語で使用する分析を定義できます。

語彙は、[準備] タブにある [ビジネスロジック] の [語彙] で作成します。語彙を使用すると、自然言語クエリの精度が向上します。たとえば、以下の目的で語彙を使用できます：

- 項目、マスターアイテム、および値の代替名の追加。
- コード化された値の名前の定義
- 特定の用語またはクエリーで使用する Insight Advisor 分析タイプを定義します。

語彙は、次の 2 種類の語彙をサポートします。

- **同義語**: 同義語を使用して、別の用語を追加できます。
詳細については、「*Insight Advisor* への同義語の追加 (page 286)」を参照してください。
- **カスタム分析**: カスタム解析では、選択した用語でどの分析を使用するかを定義することができます。
詳細については、「*Insight Advisor* へのカスタム分析の追加 (page 287)」を参照してください。

制限事項

語彙には次の制限があります。

- 語彙は、**Associative Insights** ではサポートされていません。
- 語彙は、日付として分類された項目の用語の作成をサポートしていません。
- 以下の語句は、**Qlik Sense** で使用される予約語であり、語彙として使用できません。
 - アプリ
 - アプリケーション
 - 軸
 - Dim
 - メジャー
 - Msr
- **Insight Advisor** では、語彙の用語の語彙化または語幹化は行われません。たとえば、用語が「**open**」である場合、「**opened**」または「**opening**」を使用したクエリーは、「**open**」として入力されたものであるとは取り扱われません。同様に、用語が「**opened**」である場合、「**open**」および「**opening**」はその用語として認識されません。
- 語彙の用語が項目値としても存在し、クエリーにその値と共有用語を含む項目が含まれている場合、インサイトアドバイザーは、その語彙の用語を使用せず、クエリーをその項目を含むクエリーとして処理して、その項目値を使用します。

Insight Advisor への同義語の追加

Insight Advisor の語彙を作成できます。これにより、データモデルに存在しない用語や値を定義してクエリーで使用できます。

語彙は、**[準備]** タブにある **[ビジネス ロジック]** の **[語彙]** で作成できます。語彙を使用すると、自然言語クエリーの精度が向上します。たとえば、語彙を使用して、

- 項目、マスターアイテム、および値に代替名や同義語を定義できます。
たとえば、**[利益]** 項目の **[売上]**、**[収益]**、および **[収入]**。
- コード化された値の名前
たとえば、医療分類コードなど。
- 項目からの値のグループ名

たとえば、年齢が入力された項目に年齢範囲を定義し、それに名前を付けるなど。

- データにない一般的な頭字語と略語。
たとえば、年 の場合は *yr*、数値の場合は *num* です。

語彙を作成するときは、その語彙に関連する用語を定義します。その後で、用語をアプリ内の項目やマスターアイテムに関連付けます。必要であれば、用語を特定の項目やマスターアイテムの個別の値に関連付けることもできます。項目の値に用語を定義する際に、以下の条件を使用できます。

- 超
- 以上
- 未満
- 以下
- In
- 次には含まれていません
- 範囲内



項目の条件の可用性は、項目の値によって異なります。

語彙の作成はアプリの言語に従います。語彙は言語によって分けられます。



Qlik Sense は英語、フランス語、ロシア語、スペイン語の語彙に対応しています。

次の手順を実行します。

1. アプリで、[準備] をクリックし、[語彙] を選択します。
2. 言語 ドロップダウンから言語を選択します。
3. [用語を作成] をクリックします。
4. 語彙の用語を入力します。
5. 語彙が適用される項目またはマスターアイテムを選択します。
6. 必要に応じて、条件と条件が適用される値を選択します。
7. [作成] をクリックします。

制限事項

同義語には以下の制限があります。

- マスターアイテムおよび時間項目では、条件はサポートされません。

Insight Advisor へのカスタム分析の追加

特定の用語に使用する特定の分析を定義できます。これにより、ユーザーが Insight Advisor で受け取る分析をコントロールしたり、好みの分析を作成することができます。

8 Insight Advisor の論理 モデルのカスタマイズ

カスタム分析によって、特定の条件に対する応答を定義し、Insight Advisorが返す分析を制御できます。カスタム分析には複数の用語を定義できます。また、カスタム分析では、異なる言語で異なる用語を定義することもできます。



Qlik Sense は英語、フランス語、ロシア語、スペイン語の語彙に対応しています。

カスタム分析

Create terms ×

Language Terms: ⓘ

En

+ Add language

Analysis type: Rank ▼ Variations: 1 Dimension, 1 Measure ▼

Dimensions & measures Constraints Options

You can set defaults for dimensions and measures used with the selected analysis type. Setting defaults is optional.

Dimension: ▼

Measure: ▼

Cancel Create

カスタム分析を作成するとき、用語を分析に追加して分析タイプを選択します。その後、デフォルトの軸またはメジャーを指定できます。デフォルト値が設定されていない場合、Insight Advisor はクエリに基づいた分析に使用する適切な軸またはメジャーを決定します。

オプションで、分析タイプに制約を設定して、スコープを制限することができます。例えば、ある特定の測定値を超える値のみを分析に含めることができます。



項目の条件の可用性は、項目の値によって異なります。

分析タイプに応じて、分析タイプで使用可能なグラフタイプから、分析で使用するチャートの種類を任意で選択することができます。チャートの種類を選択した場合、選択したデータに別のチャートがより適切であると判断されない限り、カスタム分析ではそのチャートの種類が常に使用されます。クエリに対してより適切なチャートが存在する場合でも、そのチャートタイプが使用されるようにチャートを適用することを選択できます。たとえば、円グ

8 Insight Advisor の論理 モデルのカスタマイズ

ラフを適用できます。200個の個別の値を含む次元を検索に使用した場合、そのデータに適していなくても、円グラフが使用されます。また、分析の種類に応じて、グラフの並べ替えオプションを設定したり、表示される値の数に制限を適用することもできます。

分析タイプ

次の表は、カスタム解析を行う際に使用できる解析タイプの概要です。一部の分析では、選択できる異なるバリエーションがあります。これらの決定した項目値が、分析で使用されます。

分析タイプ

分析タイプ	使用	バリエーション
異常	履歴データのパターンを分析することにより、スパイクの異常を識別します。	メジャー1つ、時間軸1つ
除外済み	複数の軸によるメジャーの分類。	軸2~3つ、メジャー1つ
クラスター (K平均法)	1つの軸で2つのメジャーを比較するには、 KMeans2D 関数を使用して結果をクラスタリングします。	軸1つ、メジャー2つ
比較	1つまたは2つの軸で2~4つのメジャーを比較します。2つのバリエーションに対応しています。 <ul style="list-style-type: none"> 1つの軸、2~4つのメジャー 2つの軸、2~3つのメジャー 	1つの軸、2~4つのメジャー 2つの軸、2~3つのメジャー
寄与度	累積寄与度のあるランクを作成します (パレートチャート)。	軸1つ、メジャー1つ
相関	最大2つのオプション軸における2つのメジャー間の相対的な相関関係を示します。	軸0~2つ、メジャー2つ
ファクト	集計結果を表示します。	1メジャー
マップ分布	複数の地理的な軸によるメジャーの分類。	メジャー0~2つ、地理軸1~2つ
相互情報	ターゲットと選択したアイテム間の統計依存を表示します。 依存インジケータの範囲は、0% (依存関係なし) から100% (強い依存関係) です。 項目1つ (メジャーまたは軸) をターゲットとして、そして1から10の軸またはメジャーをドライバーとして選択することができます。 相互情報はランダムに選択されたデータのサンプルを使用します。同じ項目または選択に対するこの分析タイプの結果は、結果として異なる場合があります。	2~11の項目値

8 Insight Advisor の論理 モデルのカスタマイズ

分析タイプ	使用	バリエーション
概要	複数の軸によるメジャーの概要を表示します。	軸 1~2つ、メジャー1つ
前の期間 対 現在の 期間	現在の期間と前の期間のメジャーを比較します。	1 メジャー
プロセス制御 (平均)	計算された2つのコントロール制限間における時間のメジャーのパフォーマンスの指標を表示します。	メジャー1つ、時間軸1つ
プロセス制御 (移動平均)	過去7期間にわたって、計算された2つのコントロール制限間における時間のメジャーのパフォーマンスの指標を表示します。	メジャー1つ、時間軸1つ
ランク	オプションのグループ化を使用した、メジャーによる軸の値のランキング。	軸1つ、メジャー1つ
ランク付けされたグループ	時間経過のメジャーに対する1つ以上の軸のネストされたランキングを表示します。	軸3つ、メジャー1つ
動向	時間の経過に伴うメジャーのパフォーマンスを表示します。必要に応じて、カーディナリティの低い軸別に分割します。 2つのバリエーションに対応しています。 <ul style="list-style-type: none"> 0~1つの軸、1つのメジャー、1つの日 / 時間の軸 2~3つのメジャー、1つの日 / 時間の軸 	0~1つの軸、メジャー1つ、時間軸1つ メジャー2~3つ、時間軸1つ
年度累計	軸によるメジャー(今年と昨年の結果)の内訳。	軸1つ、メジャー1つ、時間軸1つ

カスタム分析の作成。

次の手順を実行します。

1. アプリで、**[準備]** をクリックし、**[語彙]** を選択します。
2. **[カスタム分析]** をクリックします。
3. **[用語を作成]** をクリックします。
4. 言語 ドロップダウンから言語を選択し、カスタム分析の用語を入力します。
5. 任意で、別の言語を追加してその言語の用語を追加します。
6. **分析の種類** から、分析の種類を選択します。
7. 利用可能な場合は、使用するバリエーションを選択します。
8. 分析用の軸とメジャーを指定する場合は、**データ**で、使用する既定の項目を選択します。
9. 分析に制約を取り入れる場合は、**[Constraints (制約)]** で項目、条件、値を選択します。
10. **[チャート]** の **[チャートの種類]** で、オプションでカスタム分析で使用するプライマリチャートを選択します。

他のチャートの方が適切な場合でもこのチャートタイプを使用するには、[チャートの選択の強制] を選択します。

11. 分析の種類に利用可能な場合、チャートの並べ替えと制限を設定します。
12. [作成] をクリックします。
13. アプリが Insight Advisor Chat で利用できる場合は、ナビゲーションバーでアプリ名をクリックし、[Insight Advisor を使用] で [ハブでは] をオフにしてからオンにします。

制限事項

カスタム分析には次の制限があります。

- 次のシナリオでは、[チャートの選択の強制] の設定は実行されません。
 - When an unsupported number of fields are added to the custom analysis than the selected chart type supports.
例えば、一部のチャートは 2 つの軸をサポートしていますが、3 つの軸はサポートしていません。2 つの軸の内訳分析を作成し、mekko チャートを適用すると、予想どおりに機能します。3 つの軸の内訳分析を作成して mekko チャートを選択した場合、mekko チャートは 3 つの軸をサポートしていないため、使用されません。
 - ユーザーが軸の制限を超える原因となる軸に制約を追加した場合。
例えば、軸カテゴリと製品を使用して内訳分析を作成し、mekko チャートを選択して適用します。これは予想どおりに機能します。軸の国に制約を追加すると、チャートで 3 つの軸が使用されるようになったため、mekko チャートは適用されません。
 - 分散ウォーターフォールグラフが選択されているが、カレンダー期間が適用されていない場合。
- Insight Advisor Chat では、フォローアップの質問はカスタム分析では機能しません。例えば、「売上概要」、カスタム分析をクエリし、「利益について」などのフォローアップの質問をすると、Insight Advisor Chat はそれを新しい質問として扱われます。
- ロードスクリプトで \$hidden とタグ付けされた項目は、論理モデルで表示するように設定されている場合でも、カスタム分析では利用できません。

9 トラブルシューティング - データのロード

このセクションでは、Qlik Sense でのデータのロードおよびモデリング時に発生する可能性のある問題について説明します。

9.1 [データの追加] にドロップしてもファイルが添付されない

Qlik Sense で、[データの追加/ファイルを添付] ダイアログにファイルをドラッグアンドドロップして添付しようとしても、ファイルはアップロードされません。

考えられる原因

ファイルが ZIP アーカイブに保存されています。Qlik Sense では ZIP アーカイブから個別にファイルを添付することはできません。

提案されたアクション

ZIP アーカイブからファイルを抽出してから添付してください。

9.2 ANSI 以外でエンコードされたデータファイルでの文字セットの問題

ODBC データ接続を使用する場合、ANSI 以外でエンコードされたデータファイルでの文字エンコードで問題が発生する場合があります。

考えられる原因

ODBC データ接続は、文字セットのエンコードに対応する完全な機能を備えていません。

提案されたアクション

次の手順を実行します。

- 可能な場合、文字コードの処理のために多くのオプションをサポートするフォルダデータ接続を使用してデータファイルをインポートします。Microsoft Excel スプレッドシートまたはテキストデータファイルをロードする場合、おそらくこれが最良の選択肢です。

9.3 データロード時の循環参照についての警告

考えられる原因

複数のテーブルをロードした場合、2つの項目間の関連付けに1つ以上のパスが存在するようにテーブルを関連付けて、データ構造でループを行えるようにすることができます。

提案されたアクション

9.4 固定長レコードファイルからデータを選択するとき 列が予想通りに並ばない

考えられる原因

ファイルは、タブ記号を使用して列を並べます。一般的には、選択ダイアログで [項目の区切り] を選択すると、項目の見出しは予想したデータと一致しません。

この場合、タブ記号は、通常、文字数と同じです

提案されたアクション

次の手順を実行します。

1. [項目名] の [項目名が存在しません] を選択します。
2. [項目の区切り] を選択します。
3. ヘッダーと一致する列が表示されるまで、[タブサイズ] の設定を増やします。
4. 列の適切な位置をクリックして、項目の区切りを挿入します。
5. [データプレビュー] を選択します。
6. [項目名] の [埋め込まれた項目名] を選択します。

列が正しく並び、各項目には正しい項目名が表示されます。

9.5 コネクタが動作しない

データロードエディターで別個にインストールされたコネクタへのデータ接続を作成しようとすると、接続に失敗するか、または既存の接続に不明としてのラベルが表示されます。

コネクタが正常にインストールされない

考えられる原因

コネクタがインストール手順に従って正常にインストールされません。アプリがマルチノードサイトでコネクタを使用する場合は、すべてのノードにコネクタをインストールする必要があります。

提案されたアクション

次の手順を実行します。

- コネクタがインストール手順に従って、サイトのすべてのノードにインストールされていることを確認します。

コネクタがQlik Sense対応しない

考えられる原因

データを選択できるようにする場合、QlikView コネクタは Qlik Sense に対応する必要があります

提案されたアクション (QVX SDK を使用して、自分自身でコネクタを開発した場合)。

次の手順を実行します。

- データを選択するインターフェイスを使用して、コネクタを Qlik Sense に対応させる必要があります。

提案されたアクション (コネクタが提供された場合)

次の手順を実行します。

- コネクタのブライヤに問い合わせ、Qlik Sense 対応コネクタを入手します。

9.6 SQL Server の再起動後にデータ接続が機能しなくなる

考えられる原因

SQL Server へのデータ接続を作成し、SQL Server を再起動するとデータ接続が機能しなくなる場合があります。データを選択できなくなります。Qlik Sense から SQL Server への接続は切断され、再接続できなくなります。

提案されたアクション

Qlik Sense:

次の手順を実行します。

- アプリを閉じ、ハブからもう一度開いてください。

Qlik Sense Desktop:

次の手順を実行します。

1. すべてのアプリを閉じてください。
2. Qlik Sense Desktop を再起動します。

9.7 データロードエディターがスクリプトを表示しない

データロードエディターを開くと、エディタの内容が空白となりスクリプトを編集できません。

考えられる原因

例えば、膨大な量のネストされた if ステートメントなど、スクリプトは非常に複雑な構造を含んでいます。

提案されたアクション

URL に `/debug/dle_safe_mode` を追加して、データロードエディターをセーフモードで開きます。構文の強調表示やオートコンプリート機能が解除されますが、スクリプトを編集し保存することができるようになります。



スクリプトの複雑な部分を別のテキストファイルに移動し、**include** 変数を使用して、実行時にテキストファイルをスクリプトに挿入することを検討してください。

9.8 データロードスクリプトはエラーなしで実行されますが、データがロードされません

スクリプトは構文またはロードエラーなしで実行されますが、データが予期したとおりにロードされません。一般的な推奨方法は、スクリプトを通じて実行するデバッグをアクティブにし、実行した結果を調べます。ただし、ここでは一部のエラー原因を確認できます。

ステートメントがセミコロンで終了していない

考えられる原因

セミコロンでステートメントが終了することを忘れていました。

提案されたアクション

次の手順を実行します。

- すべてのステートメントがセミコロンで終了します。

文字列内の単一引用符

考えられる原因

SET 変数 ステートメントなどの文字列に単一引用符が含まれています。

提案されたアクション

次の手順を実行します。

- 文字列に単一引用符が含まれている場合、追加引用符を使用してエスケープする必要があります。

9.9 データマネージャーがデータを含むアプリでテーブルを表示しない

バージョン 3.0 より前の Qlik Sense で作成されたアプリを開くと、データマネージャーはテーブルを表示せずに、アプリにデータが含まれていない、というメッセージを表示します。

考えられる原因

バージョン 3.0 以後の Qlik Sense の改良されたデータモデルは、データのプロファイリングと準備を完了するためデータのリロードが必要です。

提案されたアクション

データマネージャーで、[データのロード] をクリックします。これにはアプリで使用されているデータソースにアプリがアクセス出来る必要があります。

9.10 サーバーでアプリを作成しているすべてのユーザーのデータマネージャーワークフローが壊れている

ユーザーが [データの追加] や [データマネージャー] で [データのロード] を使用する際、またはブラウザでアプリを更新する際にエラーが生じます。

考えられる原因

データマネージャーは QVD ファイルを使用して、ロードされたデータをキャッシュに格納します。これらのファイルは使用されなくなると自動的に削除されますが、大量に蓄積されていたり破損されていたりする場合、エラーを引き起こすことがあります。

提案されたアクション

QVD ファイルを含むフォルダーへのパス。Qlik Sense サーバーでは、キャッシュは次の場所に置かれます。

<Qlik Sense 共有フォルダー>\Apps\DataPrepAppCache

Qlik Sense Desktop では、キャッシュは次の場所に置かれます。

C:\ユーザー\<ユーザー名>\Documents\Qlik\Sense\Apps\DataPrepAppCache

9.11 OLE DB データソースでのデータ選択の問題

考えられる原因


OLE DB データ接続からデータを選択できない場合、接続がどのように設定されているかを確認する必要があります。

提案されたアクション

次の手順を実行します。

1. 接続文字列が正しく設計されていることを確認します。
2. 適切なログオン資格情報を使用していることを確認します。

9.12 シート表示でデータ項目が日付項目として認識されない

日付またはタイムスタンプデータを含む項目がありますが、シート表示で日付項目として認識されず、アセットパネルとその他の項目リストで  が一緒に表示されません。

テーブルを追加したときに、データプロファイルが解除された

考えられる原因

テーブルを追加したときに、[データの追加] ボタンの横にある  からデータプロファイルを無効にしました。

このオプションを選択すると、認識されている日付項目とタイムスタンプ項目は正常に機能しますが、アセットパネルとその他の項目リストにはそれらの項目に **12** が一緒に表示されず、拡張されたプロパティ項目を使用できません。

提案されたアクション

[データマネージャー] を開き、[データのロード] をクリックします。

これで、シート表示のアセットパネルのすべての日付項目とタイムスタンプ項目に、**12** が一緒に表示されます。それでも **12** が一緒に表示されない場合は、項目のデータで日付として認識されない形式が使用されている可能性があります。

日付形式が認識されない

考えられる原因

テーブルをロードしたときに、日付項目の入力形式が認識されませんでした。通常、Qlik Sense では、ローカル設定と一般的な日付形式に基づいて日付項目を自動的に認識しますが、場合によっては入力形式を指定する必要があります。

提案されたアクション

[データマネージャー] を開き、日付として認識されなかった項目を含むテーブルを編集します。この項目には、高い確率で、一般項目であることを示す **12** が一緒に表示されています。項目タイプを [日付] または [タイムスタンプ] に変更し、入力形式を項目データに適合する形式にします。

9.13 ファイル添付時に [無効なパス] というエラーメッセージが表示される

考えられる原因

ファイル名が長すぎます。Qlik Sense は、最長 171 文字のファイル名しかサポートしていません。

提案されたアクション

ファイル名を、171 文字以下の名前に変更してください。

9.14 QlikView ドキュメントから変換したアプリをロードする際にエラーが発生する

QlikView ドキュメントから変換したアプリをリロードする際に、製品間の違いが原因でエラーが発生することがあります。

絶対ファイルパス参照がスクリプト内で使用されている

考えられる原因

ロードスクリプトが絶対パスを使用するファイルを参照していますが、これは Qlik Sense の標準モードではサポートされていません。エラーメッセージの例は "Invalid Path" および "LOAD statement only works with lib:// paths in this script mode" です。

提案されたアクション

次の手順を実行します。

- すべてのファイル参照を **lib://** 参照 (Qlik Sense 内でのデータ接続) に置換してください。

サポートされていない関数/ステートメントがスクリプトで使用されている

考えられる原因

データロードエディターでスクリプトを実行するときに構文エラーが発生した場合、Qlik Sense でサポートされていない QlikView スクリプトステートメントまたは関数の使用に関連している場合があります。

提案されたアクション

次の手順を実行します。

- 無効なステートメントを削除して、有効なステートメントと置き換えてください。

9.15 Microsoft Excel: データ マネージャーまたはデータロードエディターでのファイルからのデータのロードが失敗する

考えられる原因

Excel スプレッドシートでは、[ペインのフリーズ] または 画面の [分割] が有効になっており、テーブルに空のセルがあります。

提案されたアクション

[ペインのフリーズ] または画面の [分割] を無効にするか、スプレッドシートをクリーンアップしてから、データをリロードします。

9.16 Microsoft Excel: ファイルへの接続時、およびファイルからのデータのロード時に問題が発生する (ODBC 経由)

考えられる原因

Microsoft Excel ファイルへの ODBC データ接続の設定時、または ODBC データ接続経由での Microsoft Excel ファイルからのデータのロード時に問題が発生する場合があります。これは一般に、Windows における ODBCDSN 構成に関する問題か、または関連付けられている ODBC ドライバーに関する問題に起因します。

提案されたアクション

Qlik Sense は、Microsoft Excel ファイルのロードをネイティブでサポートしています。可能であれば、ODBC データ接続を、Microsoft Excel ファイルを含むフォルダーへのフォルダーデータ接続に置き換えてください。

9.17 ディスクスペースが不足している

システムのディスクスペースが不足する理由はいくつかあり、QVD ファイルにロードされたデータをキャッシュに格納するデータマネージャーの方法が原因の1つである可能性があります。

提案されたアクション

QVD ファイルを含むフォルダーへのパス。Qlik Sense サーバーでは、キャッシュは次の場所に置かれます。

```
<Qlik Sense 共有 フォルダ>\Apps\DataPrepAppCache
```

Qlik Sense Desktop では、キャッシュは次の場所に置かれます。

```
C:\ユーザー\<ユーザー名>\Documents\Qlik\Sense\Apps\DataPrepAppCache
```

9.18 データロード時の合成キーについての警告

複数のファイルをロードした場合、データをロード後合成キーが作成されたことを示す警告が表示される場合があります。

考えられる原因

2つのテーブルに共通項目が1つ以上存在する場合、Qlik Sense は合成キーを作成して関連付けを解決しようとしています。

提案されたアクション

関連付けに意味がある場合は、合成キーについて何もする必要はありませんが、データモデルビューアーを使用してデータ構造を確認することをお勧めします。

9.19 共通項目を含むテーブルが項目名によって自動的に関連付けられない

[データの追加] を使用して複数のテーブルを追加しました。テーブルには、1つの共通項目名を含む複数の項目が含まれていますが、自動的に関連付けられません。

考えられる原因

テーブルを追加したときに、[データの追加] ダイアログ ボックスでデフォルトのオプションを選択して、データプロファイルを有効化しました。このオプションにより、テーブル間で共通のすべての項目名が自動記述されます。例えば、このオプションを使用して共通の項目 F1 を持つテーブル A およびテーブル B を追加する場合、この項目はテーブル A では F1、テーブル B では B.F1 という名前になります。このため、テーブルは自動的に関連付けられません。

提案されたアクション

[データマネージャー] を開き、[関連付け] ビューを選択します。これで、データプロファイルの提案に従って、両方のテーブルを関連付けることができます。

テーブルを関連付けたら、[データのロード] をクリックします。