

チュートリアル - 初心者用 スクリプト

Qlik Sense®

August 2022

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



1 チュートリアルへようこそ!	4
1.1 学習内容	4
1.2 対象ユーザー	4
1.3 パッケージ内容	4
1.4 このチュートリアルでのレッスン	5
1.5 追加の資料とリソース	5
2 データロードエディターでスクリプト作成	6
2.1 データロードエディター	6
2.2 スクリプトエディター	7
コマンドおよび関数の構文ヘルプへのアクセス	7
スクリプトへのコメント入力	7
コードのインデント	7
設定済みのテストスクリプトの挿入	8
2 LOAD とSELECT ステートメント	9
3 データの選択とロード	10
3 項目名の変更	19
3 データの削減	21
3 データの変換	29
3.1 Resident LOAD	29
3.2 先行するLOAD	33
3 連結	35
3.3 自動連結	35
3.4 強制連結	38
3.5 連結の防止	39
3 循環参照	41
3.6 循環参照の解決	42
3 合成キー	44
3.7 合成キーの解決	46
4 アプリでのデータ使用	48
4.1 チャートの追加	48
4.2 軸とメジャーの追加	49
軸の作成と追加	49
メジャーの作成と追加	49
4.3 お疲れ様でした!	53

1 チュートリアルへようこそ!

このチュートリアルでは、**Qlik Sense** での基礎的なスクリプト作成の方法をご紹介します。

Qlik Sense のアプリでビジュアライゼーションを作成するには、データをロードする必要があります。ロードスクリプトの使い方を理解すると、アプリにデータをロードする際に、データを準備および操作できます。

データマネージャーまたはデータロードエディターを使用して、データをロードできます。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。

1.1 学習内容

このチュートリアルを完了すると、スクリプトを使用したデータのロード、スクリプトの編集、データの変換が簡単に行えるようになります。

1.2 対象ユーザー

Qlik Sense の基本を習得している必要があります。具体的には、アプリとビジュアライゼーションの作成経験が必要です。

データロードエディターへのアクセス権があり、**Qlik Sense Enterprise on Windows** でデータをロードする必要があります。

1.3 パッケージ内容

ダウンロードした **zip** パッケージには、チュートリアルを完了するために必要な次のデータファイルが含まれています:

- *Customers.xlsx*
- *Dates.xlsx*
- *Region.txt*
- *Sales.xlsx*

また、このパッケージには、**Scripting Tutorial** (スクリプト作成チュートリアル) アプリのコピーが含まれています。ハブにアプリをアップロードできます。

学習効果を最大限に高めるため、チュートリアルの説明どおりに自分自身でアプリを構築することを推奨します。また、データロードを機能させるには、チュートリアルの説明どおりに、データファイルをアップロードしてデータファイルに接続する必要があります。

ただし、問題が発生した場合は、アプリがトラブルシューティングに役立つ場合があります。各レッスンにどのスクリプトセグメントが関連しているのかを示しました。

1.4 このチュートリアルでのレッスン

Qlik Sense の使用経験により異なりますが、このチュートリアルの所要時間は **3 ~ 4 時間** です。トピックは、順番に完了するよう構成されています。ただし、トピックを飛ばして、後で戻することもできます。幸い、テストはありません。

- データのロードについて
- LOAD と SELECT ステートメント
- データの選択とロード
- 項目名の変更
- データの削減
- データの変換
- 連結
- 循環参照
- 合成キー
- アプリでのデータ使用

1.5 追加の資料とリソース

- [Qlik](#) では、さらなる詳細情報を提供する、広範なリソースをご用意しています。
- [Qlik オンラインヘルプ](#) を利用できます。
- 無料のオンライン コースを含むトレーニングは、[Qlik Continuous Classroom](#) で利用できます。
- ディスカッション フォーラム、ブログなどは、[Qlik Community](#) にあります。

2 データロードエディターでスクリプト作成

Qlik Sense では、データロードエディター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。

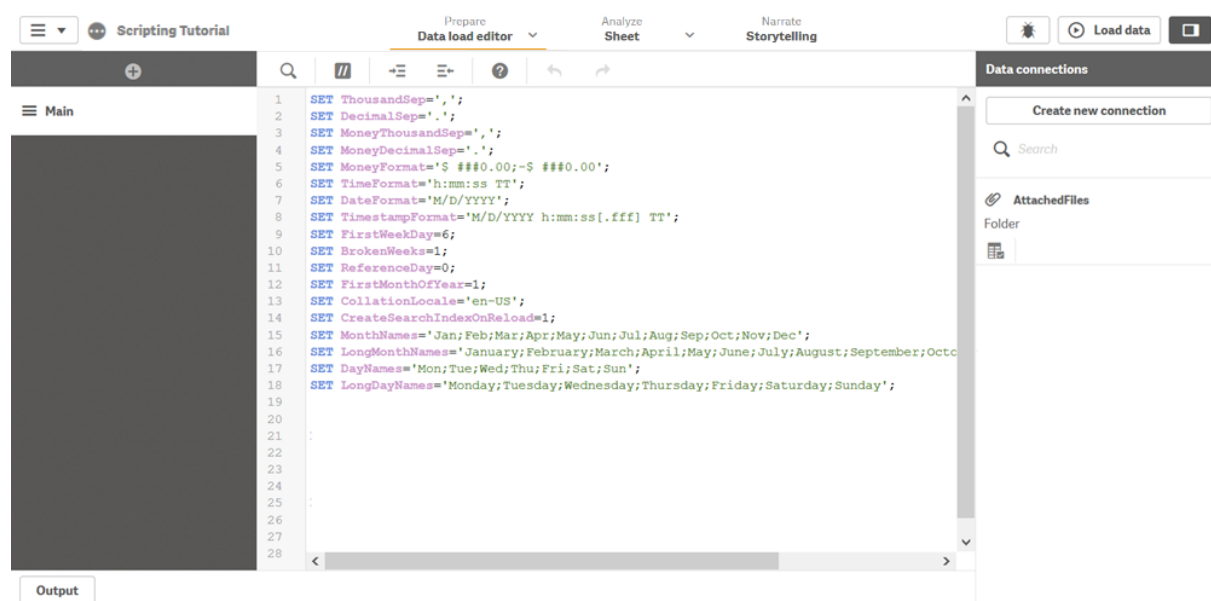
2.1 データロードエディター

データロードエディターでデータをロードするスクリプトを作成できます。エディターは **Qlik Sense** のドロップダウンメニューから使用できます。

データロードエディターを開くと、画面の中央にスクリプトエディターが表示されます。スクリプトセクションは、左メニューにタブとして表示されます。**Qlik Sense** は **Main** セクションを自動的に作成します。データ接続は右メニューに表示されます。

スクリプトは **Qlik Sense** スクリプト構文で記述する必要があります。**Qlik Sense** 構文のキーワードは青色でハイライトされます。

データロードエディター



2.2 スクリプトエディター


エディターには、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。

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。

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。コードをコメントアウトまたはコメント解除するには、 をクリックするか、**Ctrl + K** を押します。

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。 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)。

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。**Ctrl + 00** を押してテストスクリプトを挿入します。

2 LOAD とSELECT ステートメント

LOAD および SELECT ステートメントを使用すると、Qlik Sense にデータをロードできます。これらの各 ステートメントにより、内部テーブルが生成されます。LOAD はファイルからのデータのロードに使用されるのに対し、SELECT はデータベースからのデータのロードに使用されます。

このチュートリアルでは、ファイルのデータを使用するため、LOAD ステートメントを使用します。

先行する LOAD を使用して、ロードするデータのコンテンツを操作することもできます。たとえば、LOAD ステートメントで項目名を変更する必要があり、SELECT ステートメントでの項目名の変更を許可しないようにします。

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。

- 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重要ではありません。
- 項目名は大文字小文字が区別され、データテーブル間の関連付けに使用されます。このため、適切なデータモデルを作成するには、ロードスクリプトで項目名を変更する必要がある場合があります。

3 データの選択とロード

データロードエディターのデータ選択ダイアログを使用すると、**Microsoft Excel** といったサポートされているファイル形式のファイルからデータを簡単にロードすることができます。

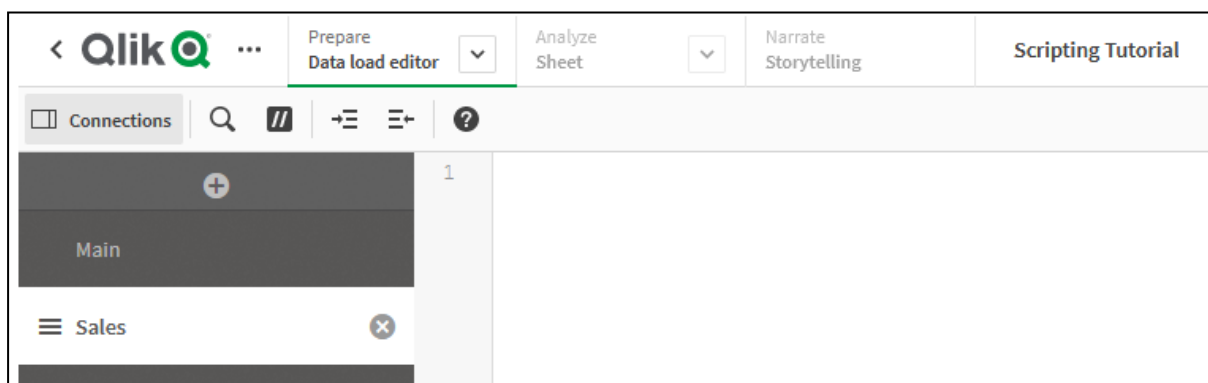
次の手順を実行します。

1. Qlik Sense を開きます。
2. 新しいアプリを作成します。
3. アプリに **Scripting Tutorial** という名前を付け、**[作成]** をクリックします。
4. アプリを開きます。



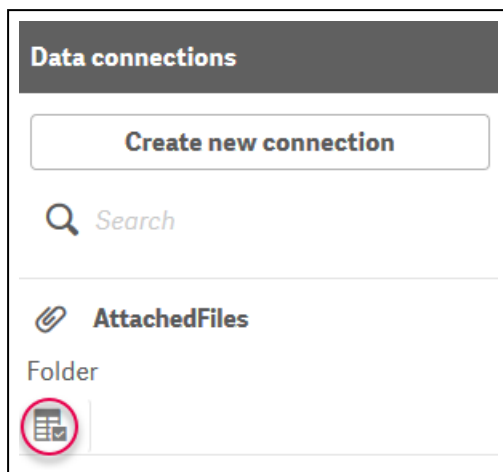
初めてアプリにデータをロードする前に **[データの追加]** を使用すれば、ファイルからのデータのロードが簡単です。ただし、このチュートリアルではスクリプトを確認できるよう、データロードエディターを使用します。

5. 上部ツールバーのドロップダウンメニューからデータロードエディターを開きます。
6. 左のメニューの **[+]** をクリックして、**Main** というセクションの下に新しいスクリプトセクションを追加します。
複数のセクションを使用することで、スクリプトをより簡単に整理できます。スクリプトセクションは、データのロード時に順番に実行されます。
7. **Sales** と入力し、セクションに名前を付けます。
Scripting Tutorial アプリ内の新しい **[Sales]** タブ



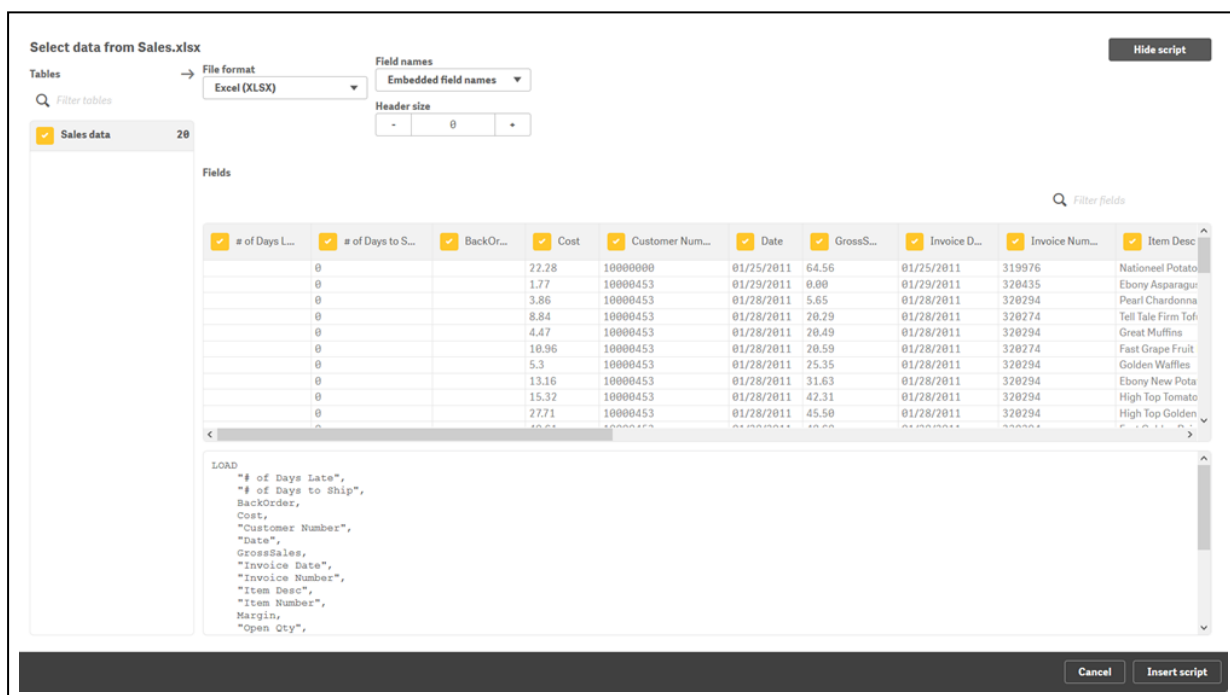
8. 右のメニューの **[AttachedFiles]** で、**[データを選択]** をクリックします。

データウィンドウの選択



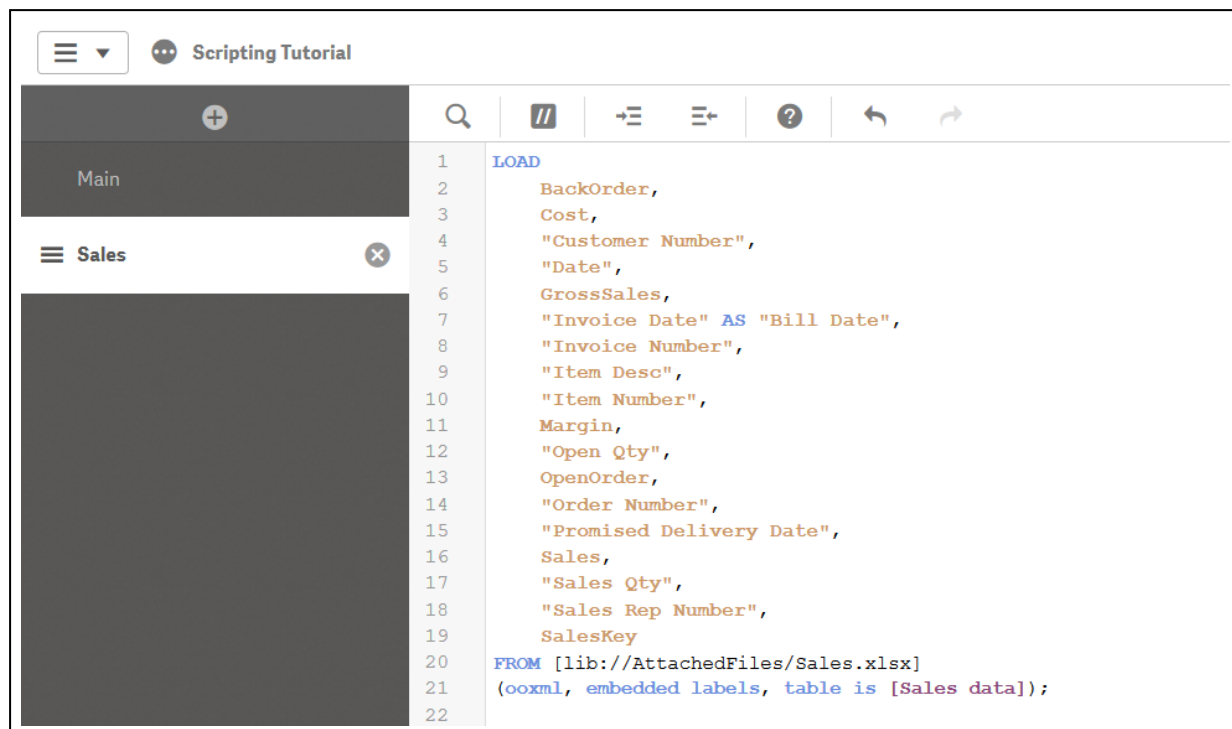
9. *Sales.xlsx* をアップロードして選択します。データプレビュー ウィンドウが開きます。

Sales データファイルのデータプレビュー ウィンドウ



10. *# of Days Late* と *# of Days to Ship* の項目の選択を解除します。項目の見出しをクリックすると、完全な項目名を確認できます。
11. [項目を絞り込む] 検索フィールドで、*date* を検索します。
12. 見出し *Invoice Date* をクリックして *Bill Date* と入力し、項目の名前を変更します。
13. [スクリプトを挿入] をクリックします。ロードスクリプトが、スクリプトエディターの *Sales* セクションに挿入されます。Qlik Sense では、スペースを含む項目名を二重引用符で囲む点に注意してください。
これで、スクリプトは次のようになります。

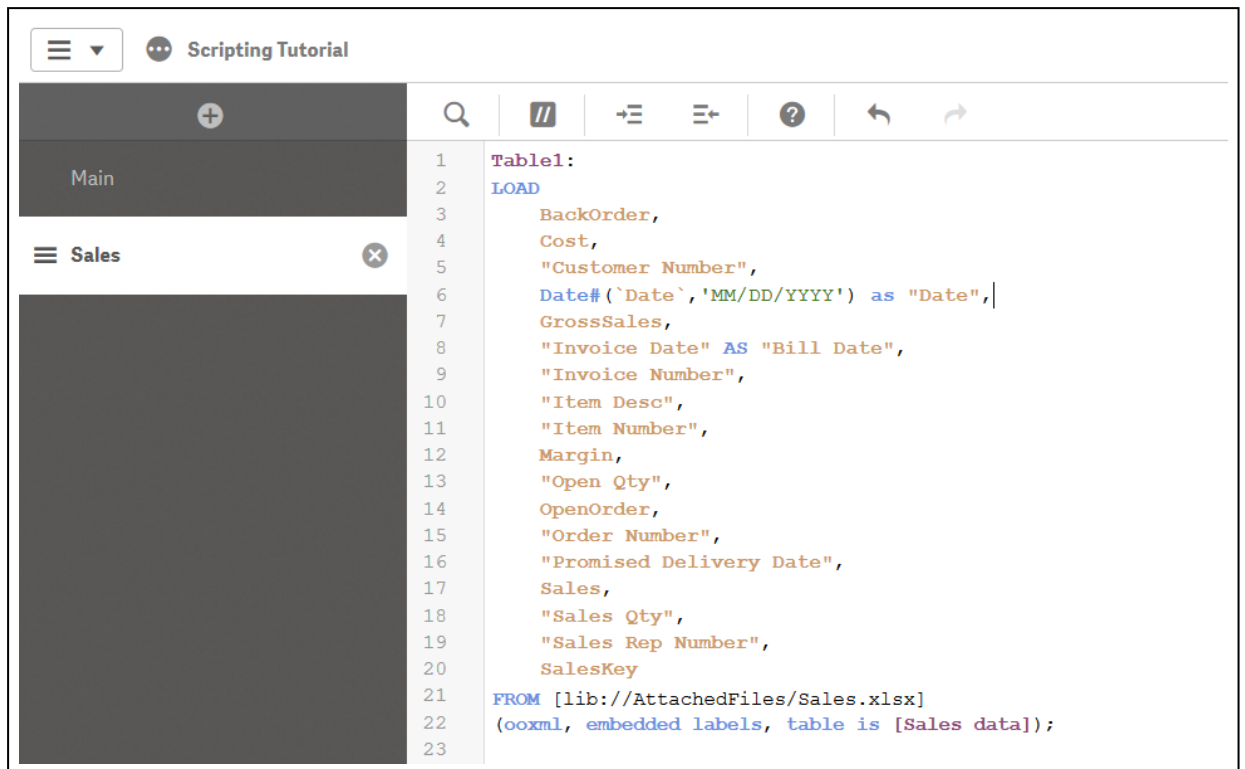
Sales タブのロード スクリプト






14. **LOAD** ステートメントの上に次の行を入力し、テーブルに **Table1** という名前を付けます。
Table1:
15. スクリプトを調整し、日付が正しく解釈されていることを確認しましょう。 **Date** 項目を次のように変更します。
Date#(`Date`, 'MM/DD/YYYY') as "Date",

これで、スクリプトは次のようになります。

Sales タブの更新されたロードスクリプト






16. 右上にある[データのロード]をクリックします。
これで、アプリにデータがロードされます。スクリプトの実行の進捗を示すウィンドウが表示されます。エラーや合成キーが発生した場合には、実行が完了した時点でそれらの問題の概要が表示されます。
17. [Close] (閉じる) をクリックします。
18. 上部 ツール バーのドロップダウン メニューからデータモデル ビューアーを開きます。 をクリックすると、データモデル ビューアーが新しいタブで開きます。
19. 上部 メニューで  と  を選択して、このチュートリアルで使用するテーブル ビューを表示します。テーブルが正しく表示されない場合は、既存のロードスクリプトを削除して、スクリプトを作り直してください。

Sales データのデータ モデル ビューアー内のテーブル ビュー

Table1
BackOrder
Cost
Customer Number
Date
GrossSales
Bill Date
Invoice Number
Item Desc
Item Number
Margin
Open Qty
OpenOrder
Order Number
Promised Delivery Date
Sales
Sales Qty
Sales Rep Number
SalesKey

では、*Dates* という別のテーブルをロードしてみましょう。テーブルをロードした後、Qlik Sense は *Date* 項目で、このテーブルを *Sales* テーブルと接続します。

20. データロードエディターを開きます。
21. をクリックして  新しいスクリプトセクションを追加します。
22. *Dates* セクションに名前を付けます。新規セクション *Dates* がまだ *Sales* の下でない場合、ポインターを  ドラッグバーの上に移動し、このセクションを *Sales* セクションの下にドラッグし、順序を変更します。
23. スクリプトの一番上の行をクリックして、 をクリックします。
// がスクリプトに追加されていることを確認します。
24. // の後に次のテキストを追加します。
Loading data from Dates.xlsx

スクリプトの先頭行は次のようになります。

// Loading data from Dates.xlsx
25. 右のメニューの [AttachedFiles] で、[データを選択] をクリックします。



[項目名] で[埋め込まれた項目名] が選択されていることを確認します。このオプションが選択されていると、データをロードする際にテーブルの項目名が含まれます。

26. *Dates.xlsx* をアップロードして選択します。データプレビュー ウィンドウが開きます。

Dates データファイルのデータプレビュー ウィンドウ

Select data from Dates.xlsx

File format: Excel (XLSX) | Field names: Embedded field names | Header size: 1 | 0 | +

Tables: Filter tables | ☒ Dates | 5

Fields: Filter fields

<input checked="" type="checkbox"/> Date	<input checked="" type="checkbox"/> Month	<input checked="" type="checkbox"/> Quarter	<input checked="" type="checkbox"/> Week	<input checked="" type="checkbox"/> Year
1/12/2011	Jan	Q1	3	2011
1/13/2011	Jan	Q1	3	2011
1/18/2011	Jan	Q1	3	2011
1/19/2011	Jan	Q1	4	2011
1/20/2011	Jan	Q1	4	2011
1/21/2011	Jan	Q1	4	2011
1/22/2011	Jan	Q1	4	2011
1/25/2011	Jan	Q1	4	2011
1/26/2011	Jan	Q1	5	2011
1/27/2011	Jan	Q1	5	2011
1/28/2011	Jan	Q1	5	2011

```

LOAD
    "Date",
    "Month",
    "Quarter",
    "Week",
    "Year"
FROM [lib://DataFiles/Dates.xlsx]
(ooxml, embedded labels, table is Dates);
    
```

Cancel | Insert script

27. [スクリプトを挿入] をクリックします。

これで、スクリプトは次のようになります。

Dates タブのロード スクリプト

Scripting Tutorial

Main | Sales | **Dates**

```

1 // Loading data from Dates.xlsx
2
3 LOAD
4     "Date",
5     "Month",
6     Quarter,
7     "Week",
8     "Year"
9
10 FROM [lib://AttachedFiles/Dates.xlsx]
11 (ooxml, embedded labels, table is Dates);
    
```

28. **LOAD** ステートメントの上の行に次のテキストを入力し、テーブルに **Table2** という名前を付けます:

Table2:

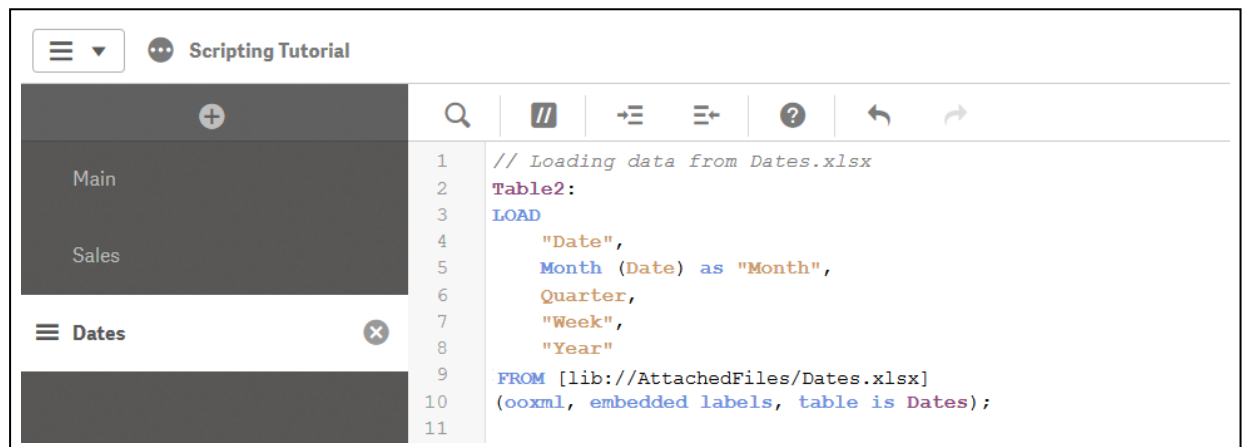
29. **Dates.xlsx** ファイルの **Month** 列が Qlik Sense で正しく解釈されるようにするには、**Month** 関数を **Date** 項目に適用する必要があります。

[月] 項目を次のように変更します。

Month (Date) as "Month",

これで、スクリプトは次のようになります。

Dates タブの更新されたロードスクリプト



これで、**Dates.xlsx** ファイルから選択されたデータをロードするスクリプトが完成しました。それでは実際にアプリにデータをロードしてみましょう。

30. 右上にある **[データのロード]** をクリックします。

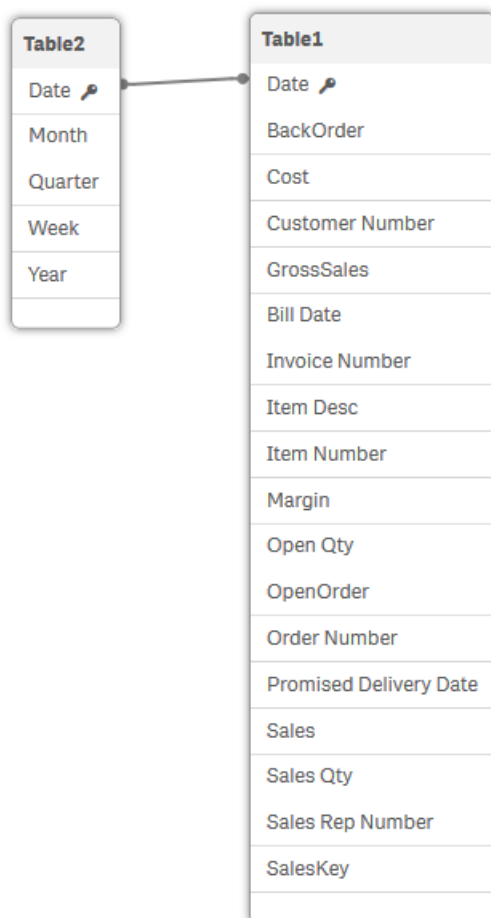
[データのロード] をクリックすると、データがアプリにロードされ、スクリプトは保存されます。

31. スクリプトの実行が完了したら、**[閉じる]** をクリックします。

32. **[データモデルビューア]** を開きます。

2つのテーブルで、名前が **Date** の2つの項目が接続されていることがわかります。

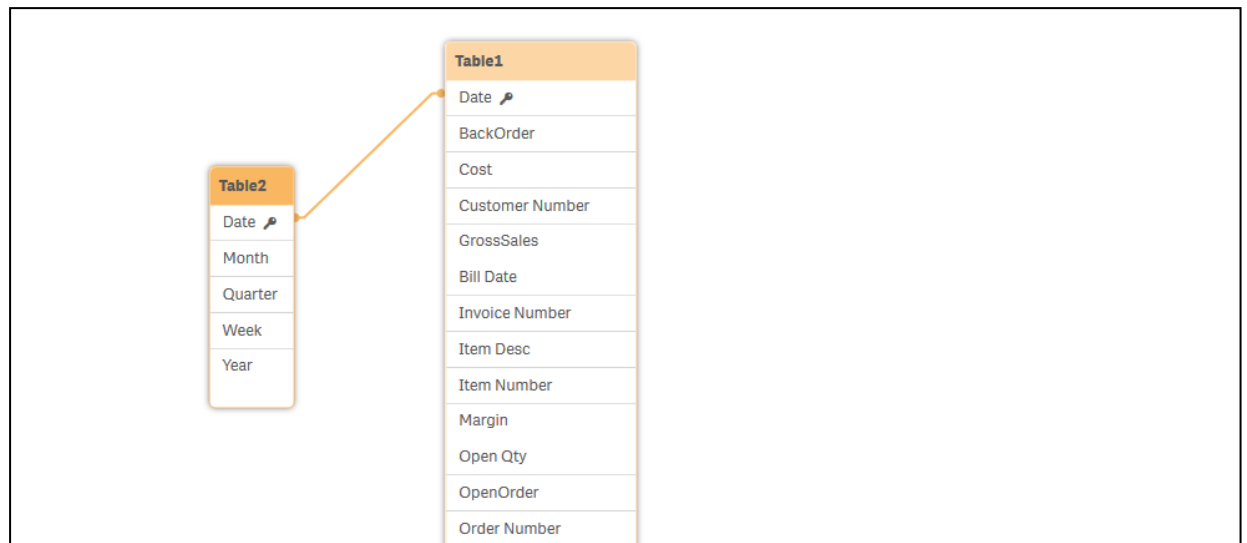
データ モデル ビューアー内のテーブル ビュー



33. 左下の【プレビュー】をクリックします。テーブル *Table2* の名前をクリックします。
- すると、このテーブルに関する情報が表示されます。【プレビュー】項目を見ると、内部テーブル *Table2* に 628 行のデータがロードされたことがわかります。テーブルの名前の代わりに項目をクリックすると、その項目に関する情報が表示されます。

3 データの選択とロード

データ モデル ビューアー内のテーブル プレビュー



▼ Preview

Table2		Preview of data				
		Date	Month	Quarter	Week	Year
Rows	628	01/12/2011	Jan	Q1	3	2011
Fields	5	01/13/2011	Jan	Q1	3	2011
Keys	1	01/18/2011	Jan	Q1	3	2011
Tags	\$key \$numeric \$integer \$timestamp \$date \$ascii \$text	01/19/2011	Jan	Q1	4	2011
		01/20/2011	Jan	Q1	4	2011
		01/21/2011	Jan	Q1	4	2011
		01/22/2011	Jan	Q1	4	2011

これで、アプリのビジュアライゼーション内でデータを使用できるようになりました。このチュートリアルの後半でその方法を説明します。

3 項目名の変更

前のトピックで、データプレビュー ウィンドウの項目名を変更する方法を説明しました。データプレビュー ウィンドウの見出しで、**Invoice Date** の名前を **Bill Date** に変更しました。ロードスクリプトを挿入すると、キーワード **AS** を使用して項目名が変更されていることを確認できました。

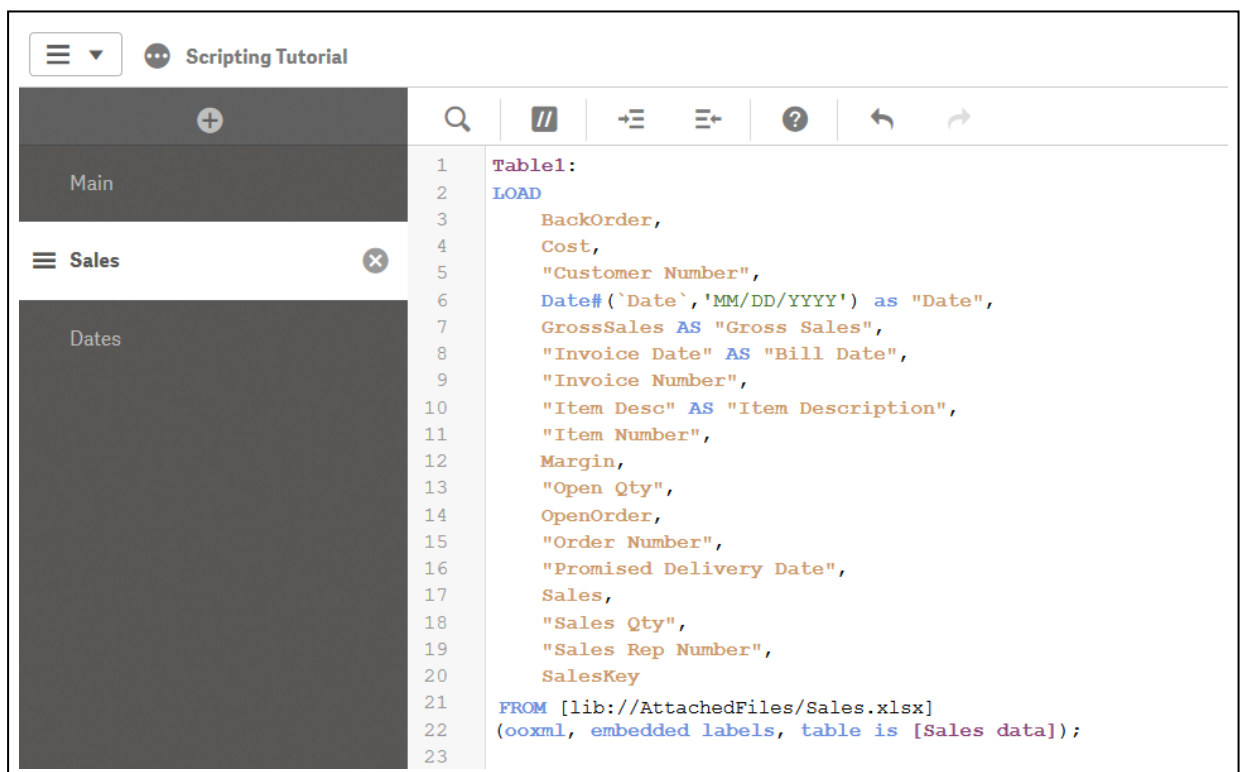
スクリプト内でこのアクションを直接実行することもできます。

次の手順を実行します。

1. *Scripting Tutorial* アプリでデータ ロード エディターを開きます。
2. **Sales** タブをクリックします。
3. ロードスクリプトを次のように変更します。スペースを含む項目は括弧で囲む必要がある点に注意してください。
 - i. **GrossSales**, を次のように変更します。
GrossSales AS "Gross Sales",
 - ii. **"Item Desc"**, を次のように変更します。
"Item Desc" AS "Item Description",

これで、スクリプトは次のようになります。

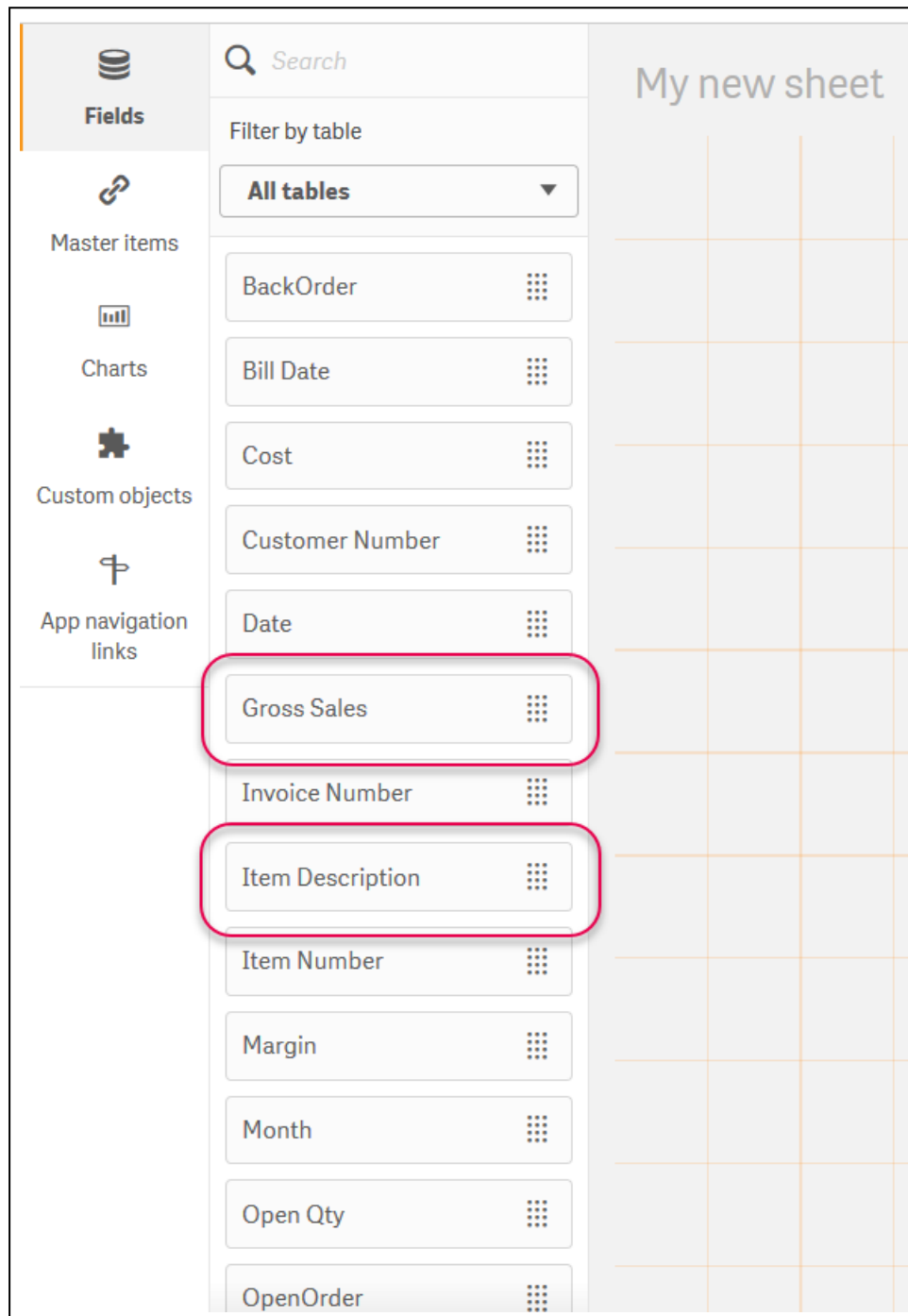
項目名を変更するスクリプトが表示されたロードスクリプトウィンドウ



4. **[データのロード]** をクリックします。データがロードされます。

5. [データモデル ビューア]を開きます。項目名が変更されていることを確認します。
6. アプリ内のすべての項目を表示することもできます。上部のツールバーで[分析/シート] タブをクリックします。シートビュー内でアプリが開きます。
7. [シートの編集]をクリックしてから、アセットパネルの[項目]をクリックします変更した項目名が表示されます。アプリ内で作成するビジュアライゼーションで、これらの項目を使用できます。

分析ビュー内の名前を変更された項目



3 データの削減

Qlik Sense では、アプリにロードするデータの量を削減するためのさまざまな方法を用意しています。たとえば、ファイルまたはデータ接続から取得するデータをフィルター処理できます。

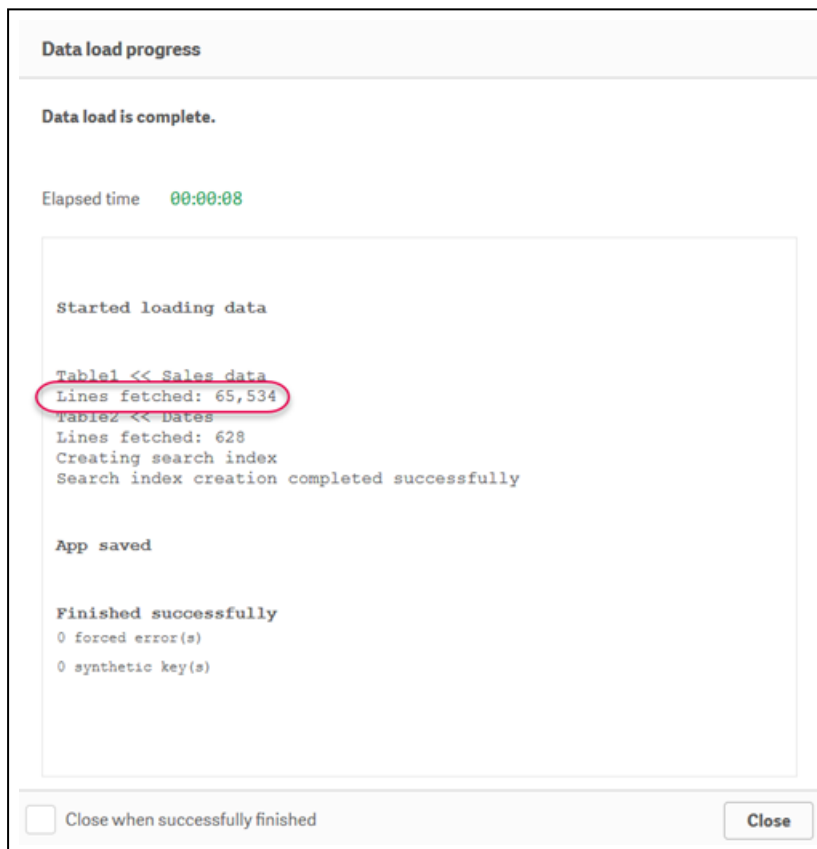
ロードスクリプトで直接データを削減することもできます。

次の手順を実行します。

1. *Scripting Tutorial* アプリでデータロードエディターを開きます。
2. [データのロード] をクリックします。

これまでに記述したロードスクリプトに基づき、Qlik Sense は 65,534 行を *Sales.xlsx* データファイルから *Table1* にロードします。ここで *Sales data* は、元の *Sales.xlsx* ファイル内の、テーブルが存在するタブの名前です。

データロード進捗状況ウィンドウ



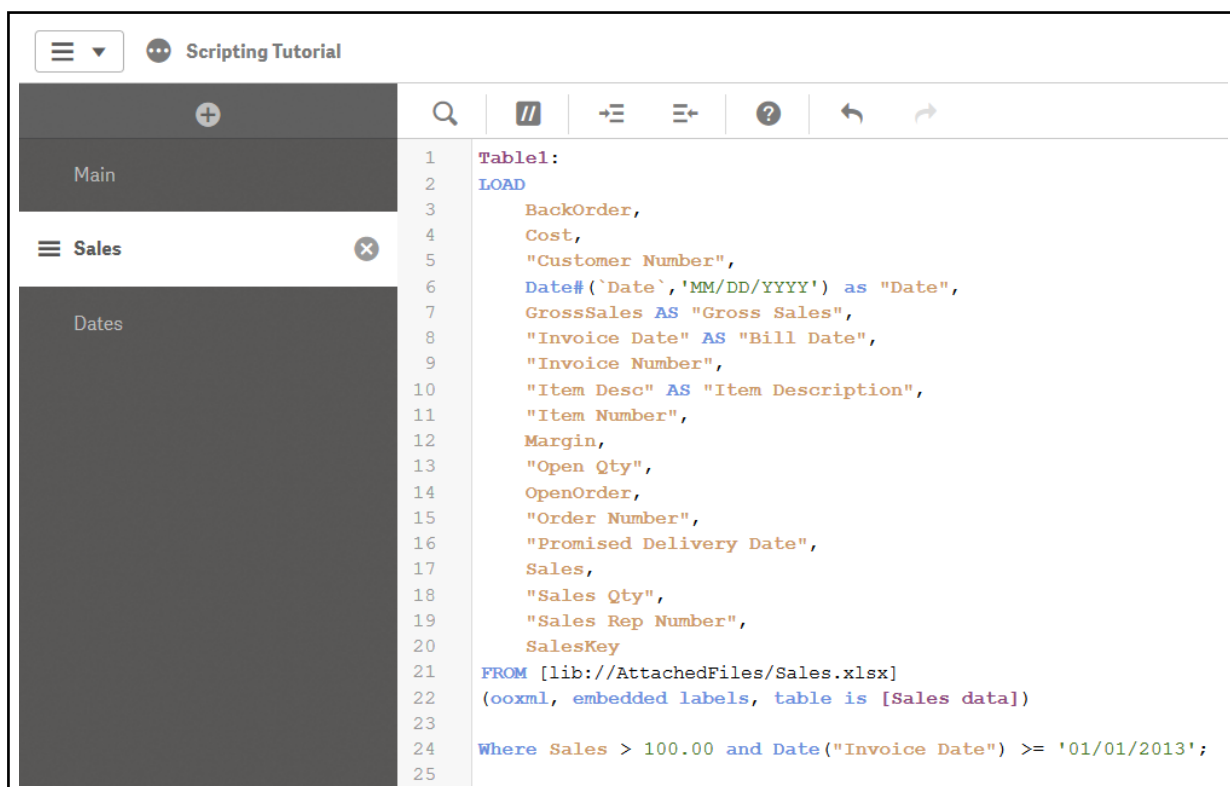
3. データロードエディターで、[Sales] タブをクリックします。
4. この行の最後にあるセミコロンを削除します。
(`ooxml, embedded labels, table is [Sales data]`);
5. 次の行をロードスクリプトの最後に追加します。

```
Where Sales > 100.00 and Date("Invoice Date") >= '01/01/2013';
```

これは、売上が\$100.00を超えているデータのみをロードするようQlik Senseに指示しています。また、Date関数も使用して、日付が2013年1月1日以降のデータのみをロードしています。

これで、スクリプトは次のようになります。

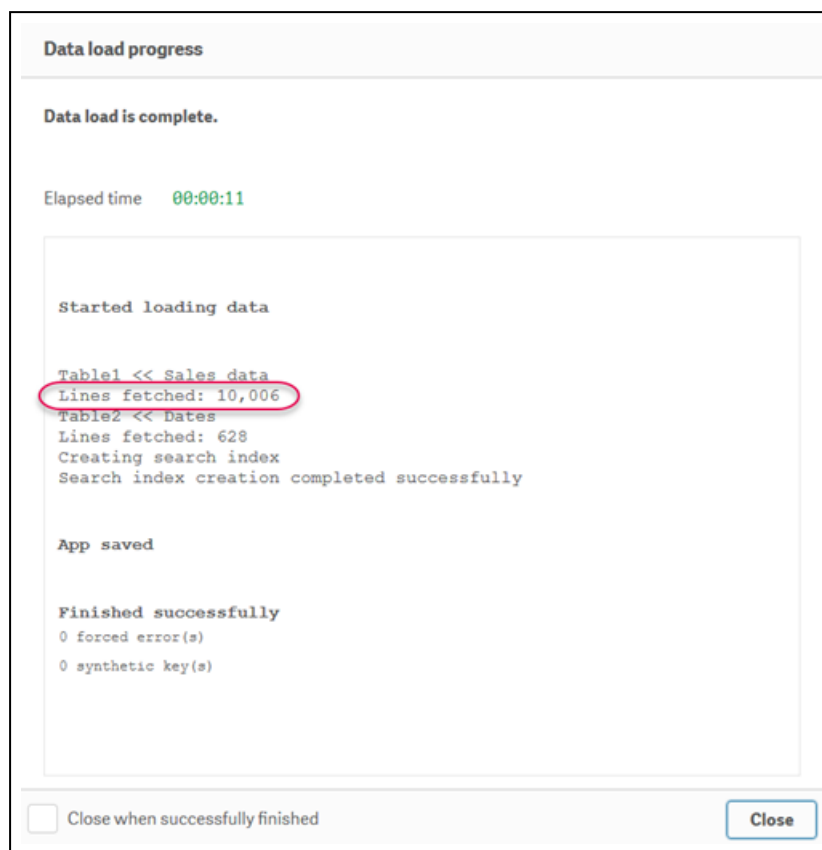
ロードするデータの量を削減するスクリプトが表示されたロードスクリプトウィンドウ



6. [データのロード] をクリックします。

更新したロードスクリプトに基づき、Qlik SenseによってSales.xlsxデータファイルからロードされる行が減少しました。

データのロード量が削減されたデータロード進捗状況ウィンドウ



7. アプリ内のテーブルにデータを追加すると、作成した条件に適合するデータのみがロードされたことを確認できます。

Bill Date および *Sales* 項目を含むテーブル

My new sheet

+ Click to add title

Bill Date	Sales
01/01/2013	100.47
01/01/2013	106.93
01/01/2013	107.72
01/01/2013	108.88
01/01/2013	115.94
01/01/2013	118.89
01/01/2013	122.13
01/01/2013	123.6
01/01/2013	123.73
01/01/2013	124.32
01/01/2013	124.72

軸として *Sales* 項目を追加した点に注意してください。これは、*Sales* 値を個別に表示するためです。*Sales* をメジャーとして追加していたら、この値は日付ごとに集計されたはずです。

通常、*Sales* はメジャーとして追加します。メジャーを使用すると、列に数値書式を適用することで、値を通貨 (ドルなど) として表示するオプションを使用できます。

Sales メジャーに適用された数値書式

Data

Sum Sales

Expression

Sum(Sales) *fx*

Label

Sum(Sales) *fx*

Number formatting

Money ▼

Format pattern

\$ ###0.00;-\$ ###0.00

Example

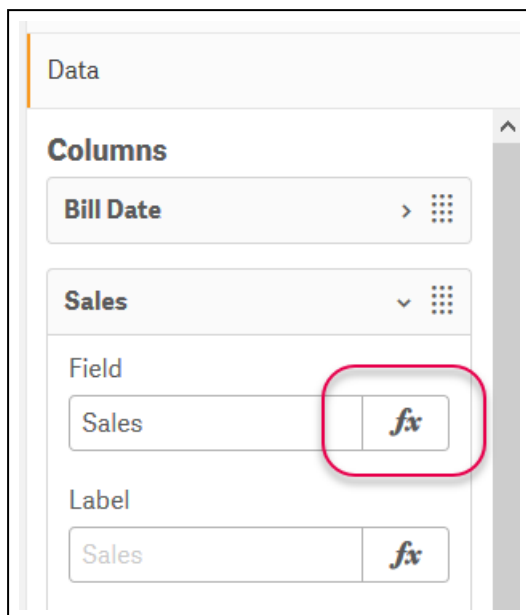
\$ 1000.12

ただし、*Sales* を軸として使用しているため、別のアプローチが必要です。この場合、チャートの数式を使用します。このチュートリアルではチャート関数について詳しく説明しませんが、よい機会なので簡単な例を紹介します。

Sales を軸として使用すると、上のテーブルのように値は数値として表示されます。

これを修正するには、**[fx]** をクリックしてチャートの数式エディタの *Sales* 項目を開き、*Money* 関数を使用します。

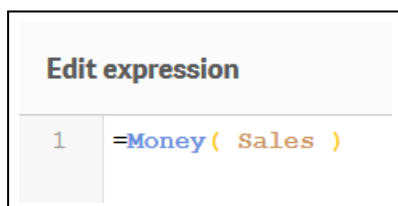
数式エディタを開く



数式エディタで次のように入力します。

=Money(Sales)

数式エディタ



項目内の値が通貨単位で表示されます。これらの単位 (この例ではドル) は、ロードスクリプトの **Main** セクションで指定されます。

Bill Date および *Sales* 項目を含むテーブル。今回は *Sales* がドル単位で表示されている

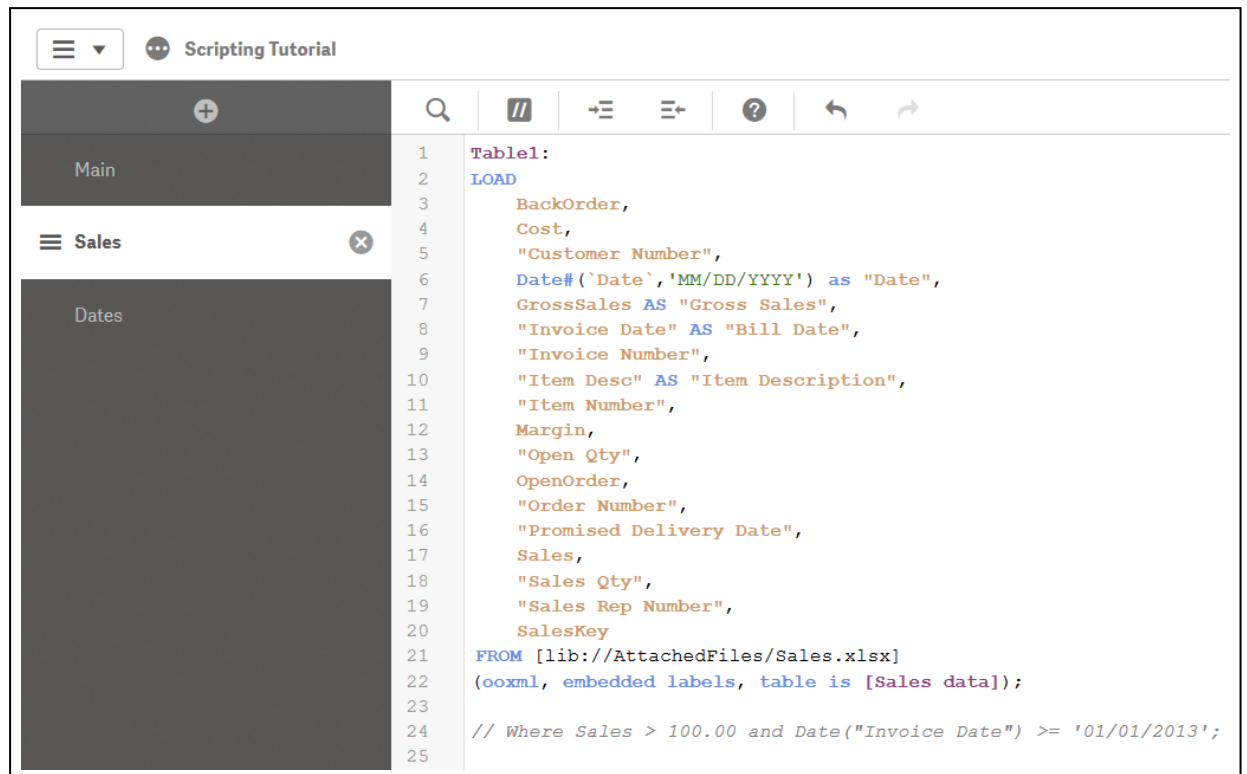
My new sheet

Click to add title

Bill Date	Money(Sales)
01/01/2013	\$ 100.47
01/01/2013	\$ 106.93
01/01/2013	\$ 107.72
01/01/2013	\$ 108.88
01/01/2013	\$ 115.94
01/01/2013	\$ 118.89
01/01/2013	\$ 122.13
01/01/2013	\$ 123.60
01/01/2013	\$ 123.73
01/01/2013	\$ 124.32
01/01/2013	\$ 124.72

8. これで、この例を完了しました。**Where** ステートメントをコメントアウトしてください。最初の **LOAD** ステートメントの最後にセミコロンを追加することを忘れないでください。
これで、スクリプトは次のようになります。

Where ステートメントをコメントアウトしたロードスクリプトウィンドウ



3 データの変換

データ ロード エディターでさまざまなテクニックを使って、データを変換および操作できます。

データ操作のメリットの1つは、ファイルからデータのサブセットのみをロードするという選択ができることです。つまり、テーブルからいくつかの列だけを選択してより効率的にデータを処理できます。また、データを複数回に渡りロードして、生データをいくつかの新しい論理テーブルに分化することもできます。さらに、複数のソースからデータをロードして、Qlik Sense にある1つのテーブルに結合することも可能です。

このトピックでは、まず **Resident LOAD** を使って、次に **Preceding LOAD** を使って、いくつかの基本的なデータ変換を実行します。

3.1 Resident LOAD

Resident ソース修飾子を **LOAD** ステートメントで使用すると、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。

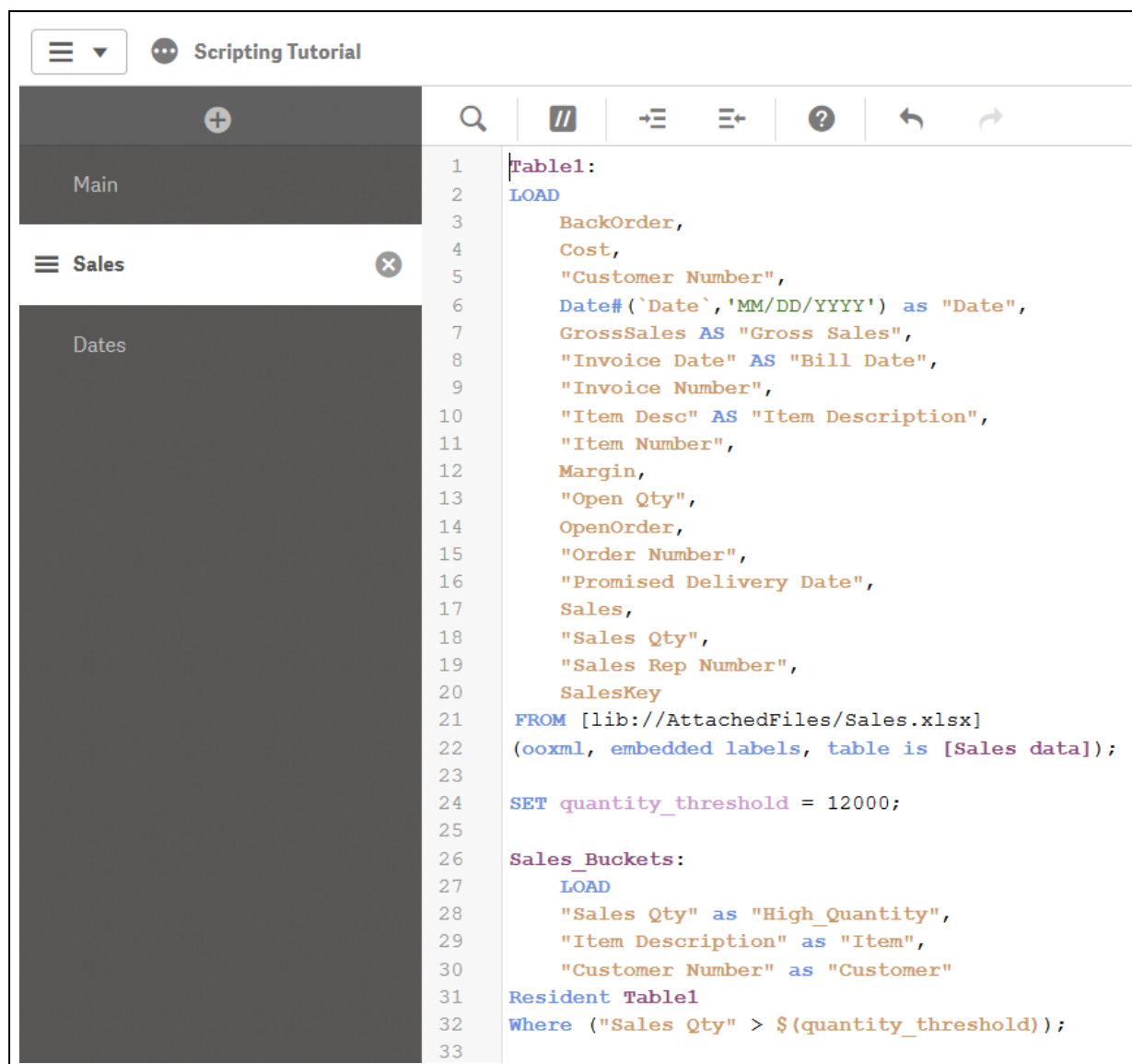
この例では、**Sales_Buckets** という新しいテーブルを作成し、**Resident LOAD** を使って **Table1** からデータをロードします。**Sales_Buckets** テーブルで、**quantity_threshold** という変数を作成し、**Where** ステートメントを使ってしきい値と一致するデータのみをロードします。

1. *Scripting Tutorial* アプリでデータ ロード エディターを開きます。
2. **Sales** タブをクリックします。
3. 次の内容をスクリプトの最後に追加します。

```
SET quantity_threshold = 12000; Sales_Buckets: LOAD "Sales Qty" as "High_Quantity", "Item Description" as "Item", "Customer Number" as "Customer" Resident Table1 where ("Sales Qty" > $(quantity_threshold));
```

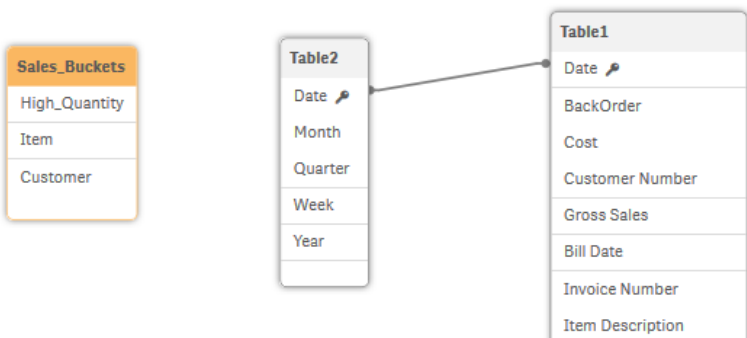
これで、スクリプトは次のようになります。

Sales_Buckets という新しいテーブルを作成するスクリプトが表示されたロードスクリプトウィンドウ



4. [データのロード] をクリックします。
5. [データモデル ビューア] を開きます。指定した項目と設定したしきい値に従ってロードされたデータを使用して *Sales_Buckets* という新しいテーブルが作成されたことを確認できます。

データ モデル ビューアー内の *Sales_Buckets* テーブル



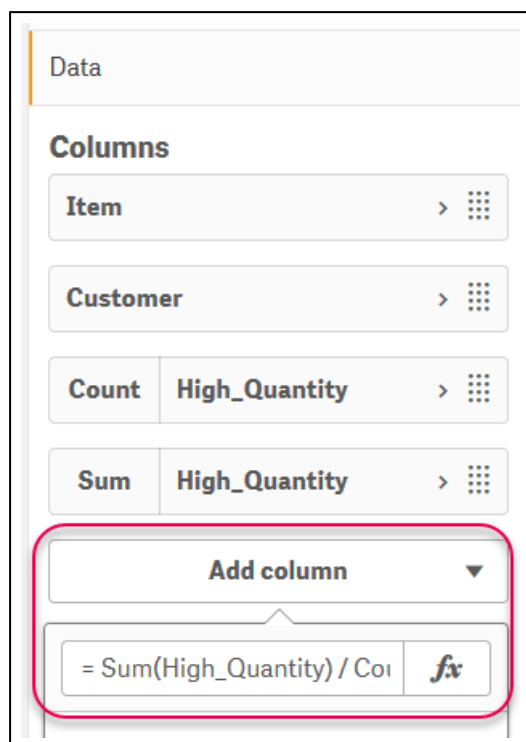
▼ Preview

Sales_Buckets		Preview of data		
		High_Quantity	Item	Customer
Rows	11			
Fields	3	16000	High Top Dried Mushrooms	10025737
Keys	0	12500	Blue Label Canned Peas	10017036
Tags	\$numeric \$integer \$ascii \$text	16000	High Top Dried Mushrooms	10025052
		13600	High Top Dried Mushrooms	10006919
		16000	High Top Dried Mushrooms	10006919
		13600	High Top Dried Mushrooms	10025737
		13600	High Top Dried Mushrooms	10025052

6. アプリのテーブルにデータを追加します。*Item* と *Customer* を軸として追加します。*Count* で集計されるメジャーとして *High-Quantity* を追加し、再度 *Sum* で集計します。続いて、次の数式を指定した新しい列をメジャーとして追加します。

= Sum(High_Quantity) / Count(High_Quantity)

数式を指定した新しいメジャー



テーブルを見ると、たとえば、*Customer 10025737* から *High Top Dried Mushrooms* の大量の注文が 4 回あり、平均数量は **14,800** であることがわかります。項目 でデータのソートを実行するには、[完了] をクリックして [編集] モードを終了します。

大量に注文した顧客を示すテーブル

Item	Customer	Count(High_Quantity)	Sum(High_Quantity)	= Sum(High_Quantity) / Count(High_Quantity)
Totals		11	158100	14372.727272727
Blue Label Canned Peas	10017036	1	12500	12500
High Top Dried Mushrooms	10006919	3	43200	14400
High Top Dried Mushrooms	10025052	3	43200	14400
High Top Dried Mushrooms	10025737	4	59200	14800

- これで、この例を完了しました。*quantity_threshold* 変数と *Sales_Buckets* テーブルのスクリプトをコメントアウトしてください。

スクリプトの最後は次のようになります。

コメントアウトされたスクリプト

```

22 (ooxml, embedded labels, table is [Sales data]);
23
24 // SET quantity_threshold = 12000;
25
26 // Sales_Buckets:
27 //     LOAD
28 //     "Sales Qty" as "High_Quantity",
29 //     "Item Description" as "Item",
30 //     "Customer Number" as "Customer"
31 // Resident Table1
32 // Where ("Sales Qty" > $(quantity_threshold));
33

```

3.2 先行するLOAD

先行するLOADを使用すると、変換を実行してフィルターを適用し、1つのパスでデータをロードできます。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。

このチュートリアルで前述したように、LOADおよびSELECTステートメントを使用して、データをQlik Senseにロードできます。各ステートメントは、1つの内部テーブルを生成します。LOADはファイルまたはインラインテーブルから、SELECTはデータベースからデータをロードする際に使用します。このチュートリアルでは、ファイルからロードしたデータを使用しました。この例では、インラインテーブルを使用します。ただし、SELECTステートメントの前に先行するLOADを使用し、データを操作できる点は注目に値します。基本は、今回説明するLOADの使用と同じです。

この例は、このチュートリアルでロードしているデータには関連していません。先行するLOADがどのように見えるかを示す目的のみに使用されています。データロードエディターで、*Transactions*というインラインテーブルを作成します。日付の変換は先行するLOADで実行され、*transaction_date*という新しい項目が作成されます。この項目は*sale_date*項目から作成されます。

1. 新しいアプリを作成し、*ReformatDate*という名前を付けます。
2. データロードエディターを開き、*TransactionData*という新しいタブを作成します。
3. 次のスクリプトを追加します。

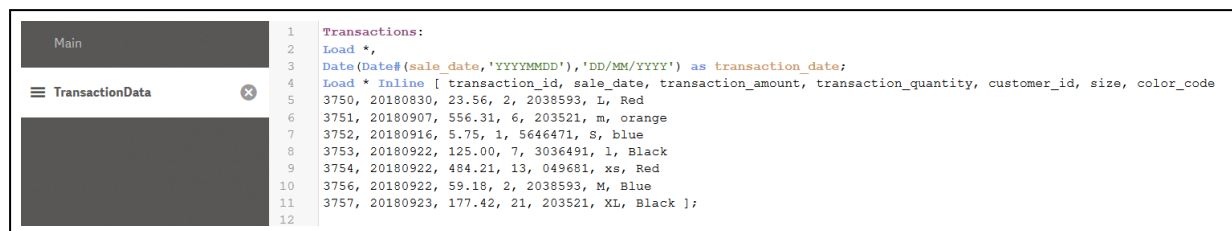
```

Transactions: Load *, Date(Date#(sale_date,'YYYYMMDD'),'DD/MM/YYYY') as transaction_
date; Load * Inline [ transaction_id, sale_date, transaction_amount, transaction_
quantity, customer_id, size, color_code 3750, 20180830, 23.56, 2, 2038593, L, Red
3751, 20180907, 556.31, 6, 203521, m, orange 3752, 20180916, 5.75, 1, 5646471, S, blue
3753, 20180922, 125.00, 7, 3036491, l, Black 3754, 20180922, 484.21, 13, 049681, xs,
Red 3756, 20180922, 59.18, 2, 2038593, M, Blue 3757, 20180923, 177.42, 21, 203521, XL,
Black ];

```

これで、スクリプトは次のようになります。

先行するLOADを記述したロードスクリプト



4. [データのロード] をクリックします。
5. [データモデル ビューア]を開きます。*Transactions* テーブルを選択して展開します。先行するLOAD ステートメントで*で指定されたとおり、すべての項目がロードされたことを確認できます。*transaction_date* という新しい項目が作成されました。この項目は、日付の書式が再設定されています。

データモデル ビューア内の *transaction_date* という新しい項目

Transactions

transaction_id

sale_date

transaction_amount

transaction_quantity

customer_id

size

color_code

transaction_date

▼ Preview

Transactions	Preview of data								
Rows	7	transaction_id	sale_date	transaction_amount	transaction_quantity	customer_id	size	color_code	transaction_date
Fields	8	3750	20180830	23.56	2	2038593	L	Red	30/08/2018
Keys	0	3751	20180907	556.31	6	203521	m	orange	07/09/2018
Tags	\$numeric \$integer \$ascii \$text \$timestamp \$date	3752	20180916	5.75	1	5646471	S	blue	16/09/2018
		3753	20180922	125.00	7	3036491	l	Black	22/09/2018
		3754	20180922	484.21	13	049681	xs	Red	22/09/2018
		3756	20180922	59.18	2	2038593	M	Blue	22/09/2018
		3757	20180923	177.42	21	203521	XL	Black	23/09/2018

3 連結

連結とは、2つのテーブルを1つにまとめる操作のことです。

2つのテーブルを連結するには、テーブルを上下に重ね合わせます。その際、各テーブルにそれぞれ固有の列名を持つ列が1つ必要となります。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。

3.3 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。

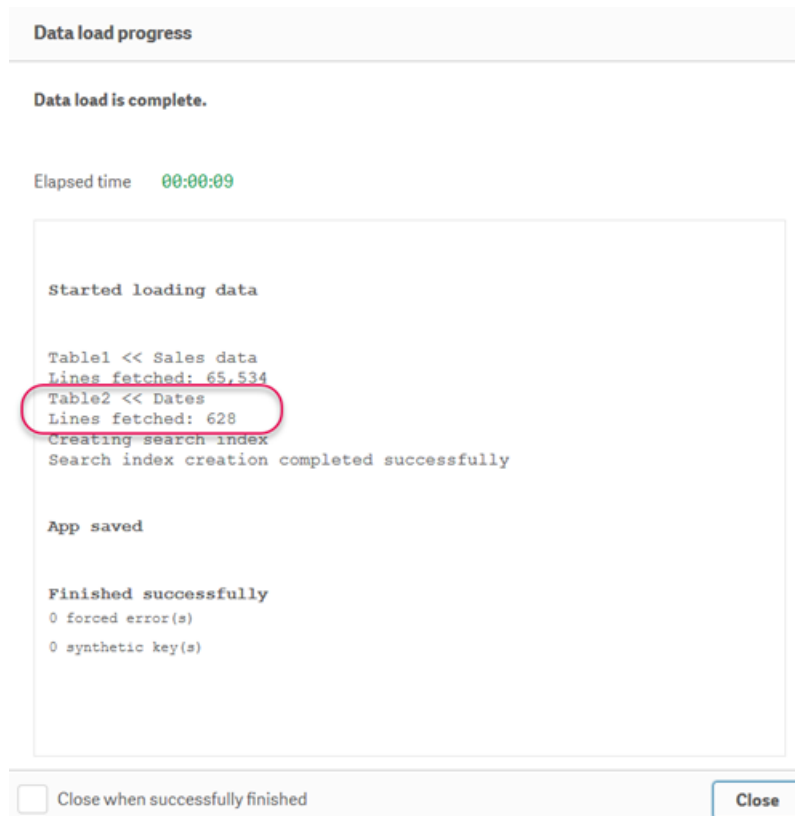
自動連結させるためには、項目の数と名前は、完全に同じでなければなりません。2つのLOAD ステートメントの順番は任意ですが、テーブルには、最初にロードされたテーブルの名前が与えられます。

次の手順を実行します。

1. *Scripting Tutorial* アプリでデータロードエディターを開きます。
2. **Dates** タブをクリックします。
3. **[データのロード]** をクリックします。

これまでに記述したロードスクリプトに基づき、Qlik Sense は 628 行を *Dates.xlsx* データファイルから *Table2* にロードします。

データロード進捗状況 ウィンドウ



4. スクリプトセクション **Dates** の新しい行で、**Table2** の LOAD ステートメントをコピーします。これによりデータが 2 回 ロードされます。2 つ目のテーブルに **Table2a** という名前をつけます。

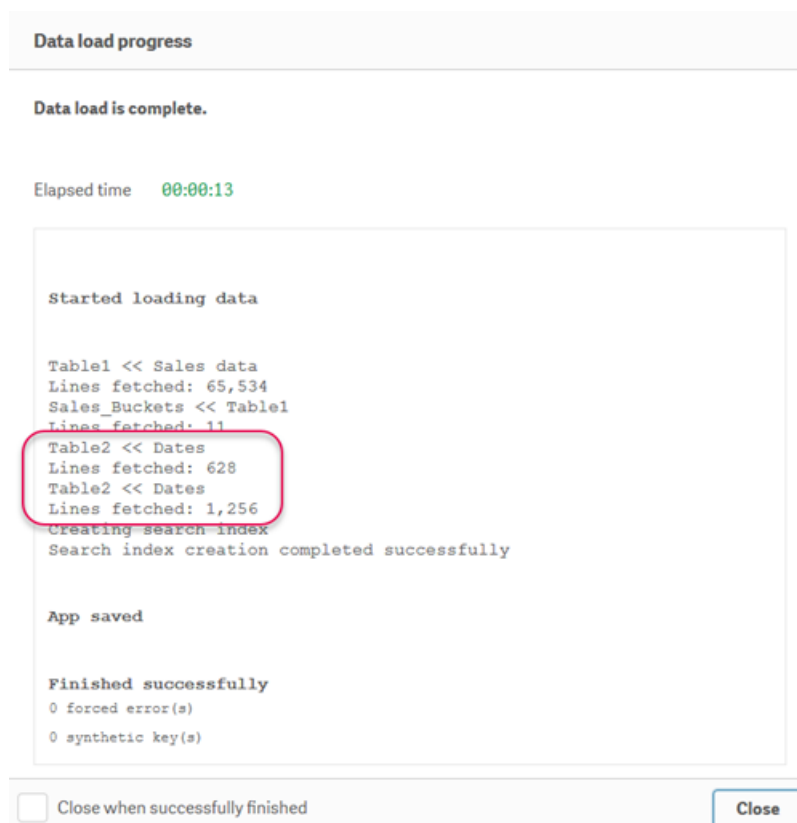
既存のスクリプトを削除して、以下のスクリプトをコピーして貼り付けることもできます。

```
Table2: LOAD      "Date",      Month (Date) as "Month",      Quarter,      "Week",
"Year" FROM [lib://AttachedFiles/Dates.xlsx] (ooxml, embedded labels, table is Dates);
Table2a: LOAD      "Date",      Month (Date) as "Month",      Quarter,      "Week",
"Year" FROM [lib://AttachedFiles/Dates.xlsx] (ooxml, embedded labels, table is Dates);
```

5. [データのロード] をクリックします。

Qlik Sense は **Table2** をロードせず、続いて **Table2a** をロードしません。その代わり、**Table2a** の項目名と項目数が **Table2** と同じであることを認識します。次に **Table2a** のデータを **Table2** に追加し、テーブル **Table2a** を削除します。その結果、**Table2** の行数が 1,256 行になります。

データロード進捗状況 ウィンドウでの連結

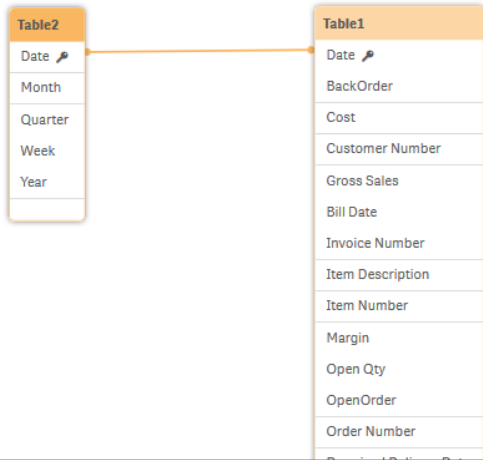


6. [データモデル ビューア]を開きます。

7. [プレビューの表示]をクリックします。

Table2 のみが作成されています。*Table2* を選択します。このテーブルには 256 行あります。

データモデル ビューアに表示された Table2



▼ Preview

Table2		Preview of data				
		Date	Month	Quarter	Week	Year
Rows	1256	01/12/2011	Jan	Q1	3	2011
Fields	5	01/13/2011	Jan	Q1	3	2011
Keys	1	01/18/2011	Jan	Q1	3	2011
Tags	\$key \$numeric \$integer \$timestamp \$date \$ascii \$text	01/19/2011	Jan	Q1	4	2011
		01/20/2011	Jan	Q1	4	2011
		01/21/2011	Jan	Q1	4	2011
		01/22/2011	Jan	Q1	4	2011

3.4 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一番最近に作成されたテーブルに連結します。

次の手順を実行します。

1. *Table2a* の LOAD ステートメントを編集し、**Concatenate** を追加して *Week* をコメントアウトします。
スクリプトは次のようになります。

```
Table2a: Concatenate LOAD      "Date",      Month (Date) as "Month",      Quarter,      //
"Year" FROM [lib://AttachedFiles/Dates.xlsx] (ooxml, embedded labels, table
is Dates);
```

Week をコメントアウトすることで、テーブルがまったく同じにならないようにします。

2. [データのロード] をクリックします。
3. [データモデル ビューア] を開きます。
ご覧のように、*Table2a* は作成されていません。

4. データモデル ビューアで **Table2** をクリックし、次に **[プレビュー]** をクリックします。
テーブルには **Date**、**Month**、**Quarter**、**Week**、および **Year** という項目があります。項目 **Week** が表示されたままになっているのは、**Table2** からロードされているためです。
5. **Table2** の **Week** をクリックします。プレビューには、この項目に **Null** でない値が **628** 個あることを示されます。しかし、他の項目をクリックすると、**Null** でない値の数は **1256** と表示されます。**Week** は **Table2** から **1** 回のみロードされています。結果として生成されるテーブルのレコード件数は、**Table2** と **Table2a** に含まれるレコードの合計となります。

3.5 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。このような動作は **NoConcatenate** ステートメントによって防ぐことができます。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。

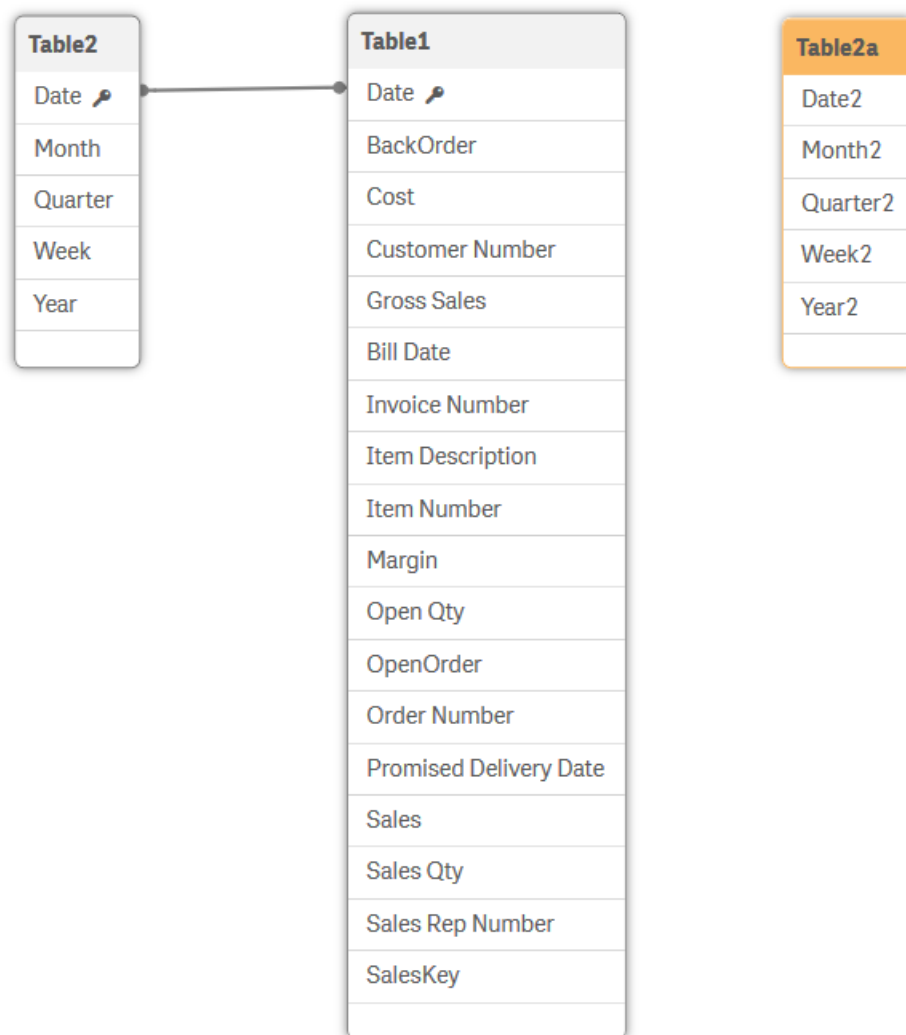
次の手順を実行します。

1. 2つのテーブルのコンテンツを完全に分離させるには、**Table2a** の **LOAD** ステートメントに **NoConcatenate** を追加し、一致する項目に基づいて **Qlik Sense** が合成キーを生成しないように、項目の名前を変更します。**Table2** の **Week** をコメント解除して2つのテーブルの項目を同じにします。
スクリプトは次のようになります。

```
Table2: LOAD      "Date",      Month (Date) as "Month",      Quarter,      "Week",
"Year" FROM [lib://AttachedFiles/Dates.xlsx] (ooxml, embedded labels, table is Dates);
Table2a: NoConcatenate LOAD      "Date" as "Date2",      Month (Date) as "Month2",
Quarter as "Quarter2",      "Week" as "Week2",      "Year" as "Year2" FROM
[lib://AttachedFiles/Dates.xlsx] (ooxml, embedded labels, table is Dates);
```

2. **[データのロード]** をクリックします。
3. **[データモデル ビューア]** を開きます。
これで2つのテーブルが完全に分離されました。

データモデル ビューアに表示された *Table2* と *Table 2a*




4. 連結の実演は以上で終了です。*Table2a* はもう必要ありません。*Table2a* の LOAD ステートメントからすべての行を削除し、**[データのロード]** をクリックします。

3 循環参照

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。

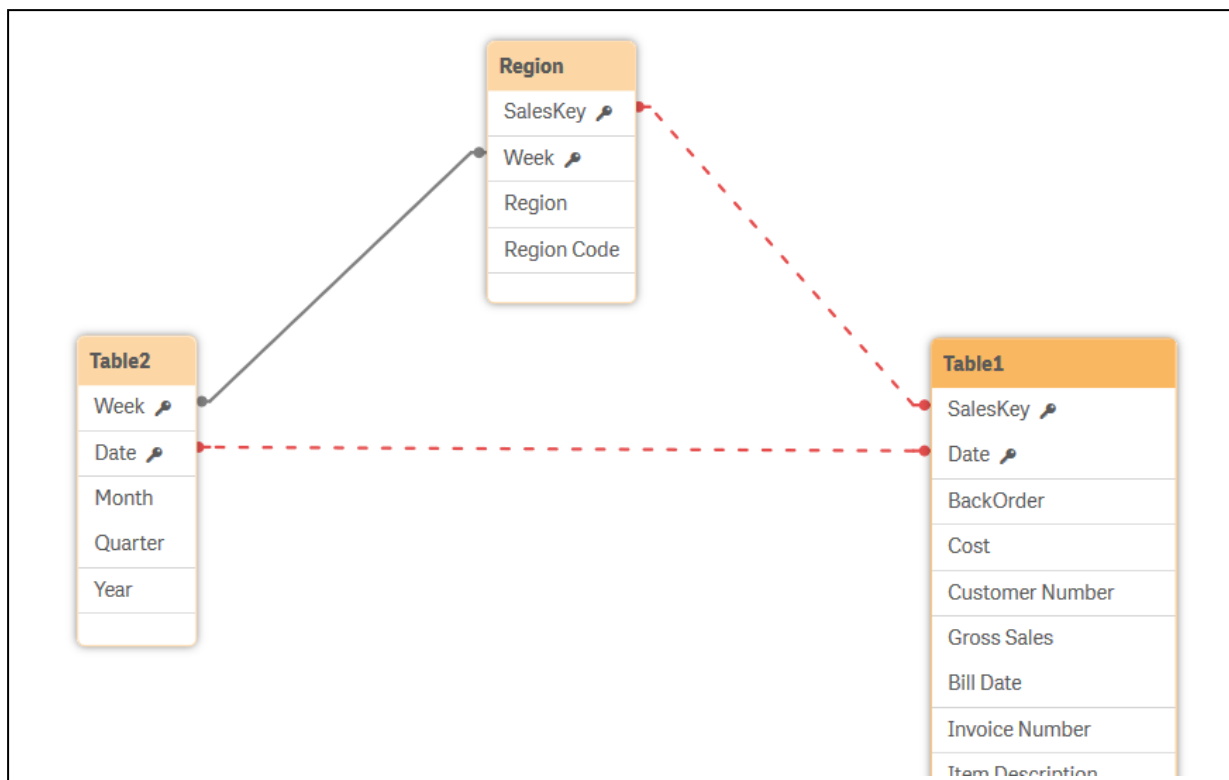
次の手順を実行します。

1. *Scripting Tutorial* アプリでデータロードエディターを開きます。
2. をクリックして  新しいスクリプトセクションを追加します。
3. *Region* セクションに名前を付けます。
4. 右のメニューの **[AttachedFiles]** で、**[データを選択]** をクリックします。
5. *Region.txt* をアップロードして選択します。データプレビュー ウィンドウが開きます。
6. すべての項目を選択し、**[ファイル名]** の下にある **[埋め込まれた項目名]** が選択されていることを確認します。このオプションが選択されていると、データをロードする際にテーブルの項目名が含まれます。
7. **[スクリプトを挿入]** をクリックします。
8. **[データのロード]** をクリックします。

すると、データのロードに問題が生じたことがわかる結果が出ます。これは、循環参照が発生したためです。**[データのロード進行状況]** ウィンドウにエラーメッセージが表示され、ロード中に循環参照が見つかったことを通知します。ロードは完了しており、アプリが保存されています。

9. **[データモデルビューア]** を開きます。
テーブル間の接続がわかりやすくなるように、テーブルをドラッグして並べ替えます。

循環参照を示すデータモデルビューア



赤の点線は、循環参照を示しています。データ解釈が曖昧になる可能性があるため、循環参照が生じないように配慮が必要です。

3.6 循環参照の解決

循環参照が生じる原因を究明するため、[データモデルビューア]にあるテーブルを検証してみましょう。

上のスクリーンショットの **Table1** と **Table2** には、両方とも **Date** という項目があります。そして、**Table1** および **Region** には、**SalesKey** という共通の項目があります。さらに、**Table2** と **Region** には、**Week** という項目があります。これは、循環参照 (ループ) が作成されていることを意味します。このような循環参照は、後のデータ分析で問題を生じさせる可能性があるため、ここで削除しておきます。

最も簡単な解決方法は、項目名を1つ変更または削除することです。この例では、アプリに不要なデータがロードされているため、このデータを削除します。

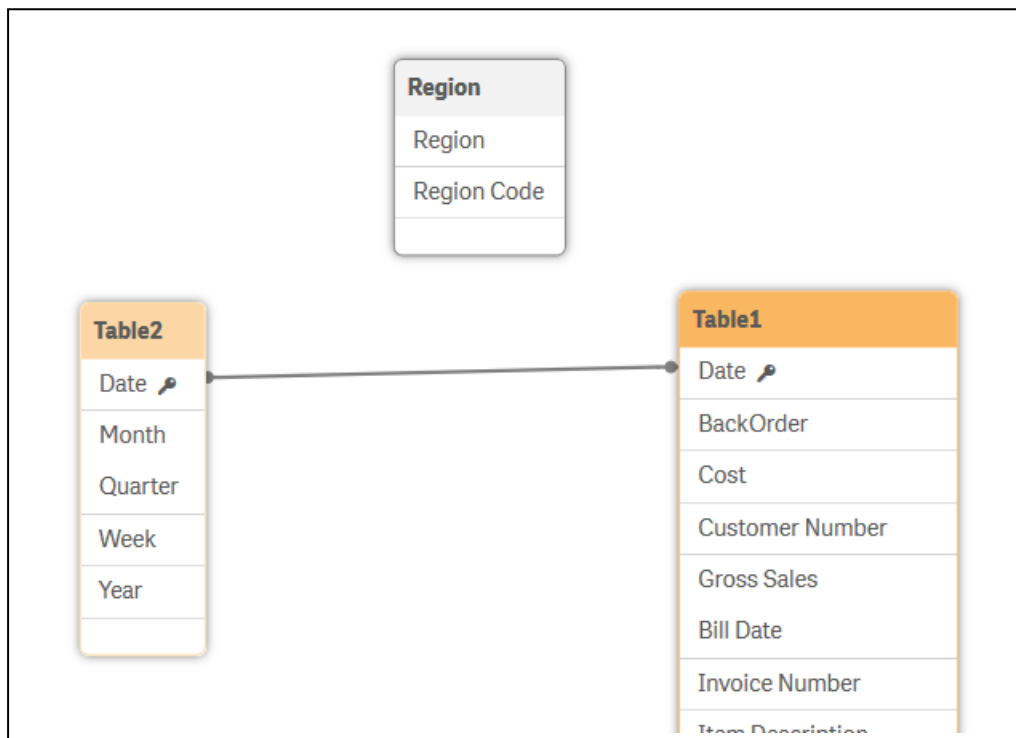
次の手順を実行します。

1. データロードエディターを開きます。
2. **Region** セクションをクリックし、**LOAD** ステートメントから次の2行を削除します。
"week",
saleskey
3. "**Region Code**" の後のコンマも忘れず削除してください。
スクリプトは次のようになります。

```
LOAD
    Region,
    "Region Code"
FROM [lib://AttachedFiles/Region.txt]
(txt, codepage is 28591, embedded labels, delimiter is '\t', msq);
```

4. [データのロード] をクリックします。
5. [データモデル ビューア] を開きます。
Region への不適切な参照が削除されました。

循環参照が削除されたデータモデル ビューア




3 合成キー

複数の内部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、合成キーを使用してこの関係进行处理します。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。

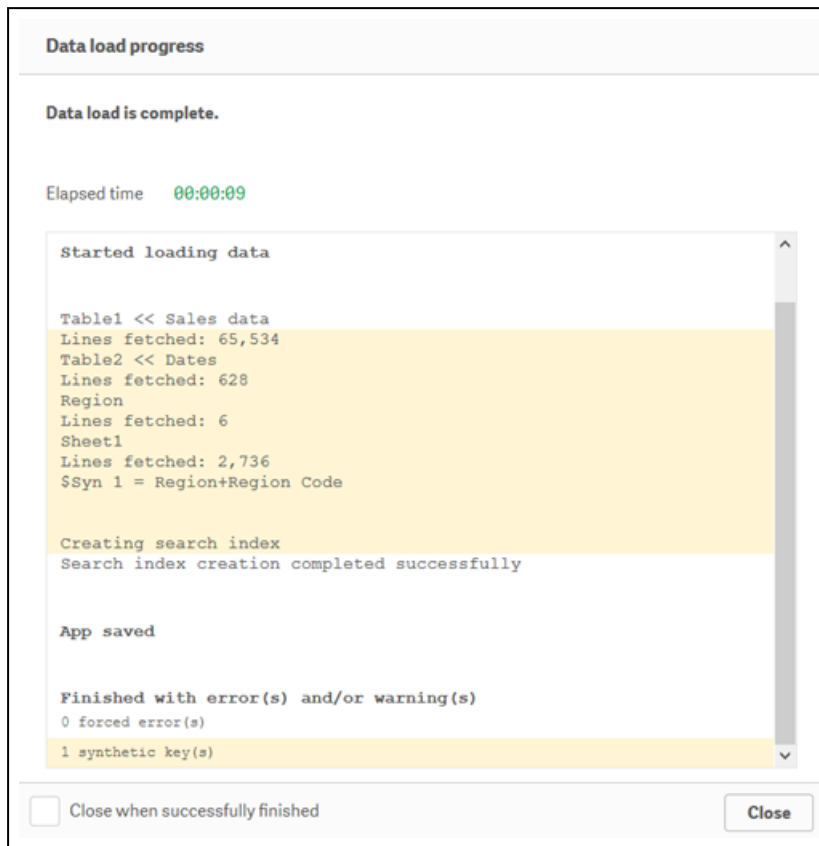
複合キーの数が増えると、メモリ使用量が増加し、パフォーマンスに影響を与える可能性があります。これは、データ量、テーブル構造、その他の要因にも依存する場合があります。そのため、相互に依存する複合キーが複数ある場合には、すべて削除することが推奨されます。

それでは、最後のデータセットをロードしましょう。

次の手順を実行します。

1. *Scripting Tutorial* アプリで**データロードエディター**を開きます。
 2. をクリックして  新しいスクリプトセクションを追加します。
 3. **Customers** セクションに名前を付けます。
 4. 右のメニューの **[AttachedFiles]** で、**[データを選択]** をクリックします。
 5. **Customers.xlsx** をアップロードして選択します。データプレビューウィンドウが開きます。
 6. **Select Sheet1**
 7. **[スクリプトを挿入]** をクリックします。
 8. **[データのロード]** をクリックします。
- 合成キーが作成されたというメッセージがデータロード進捗状況ウィンドウに表示されます。

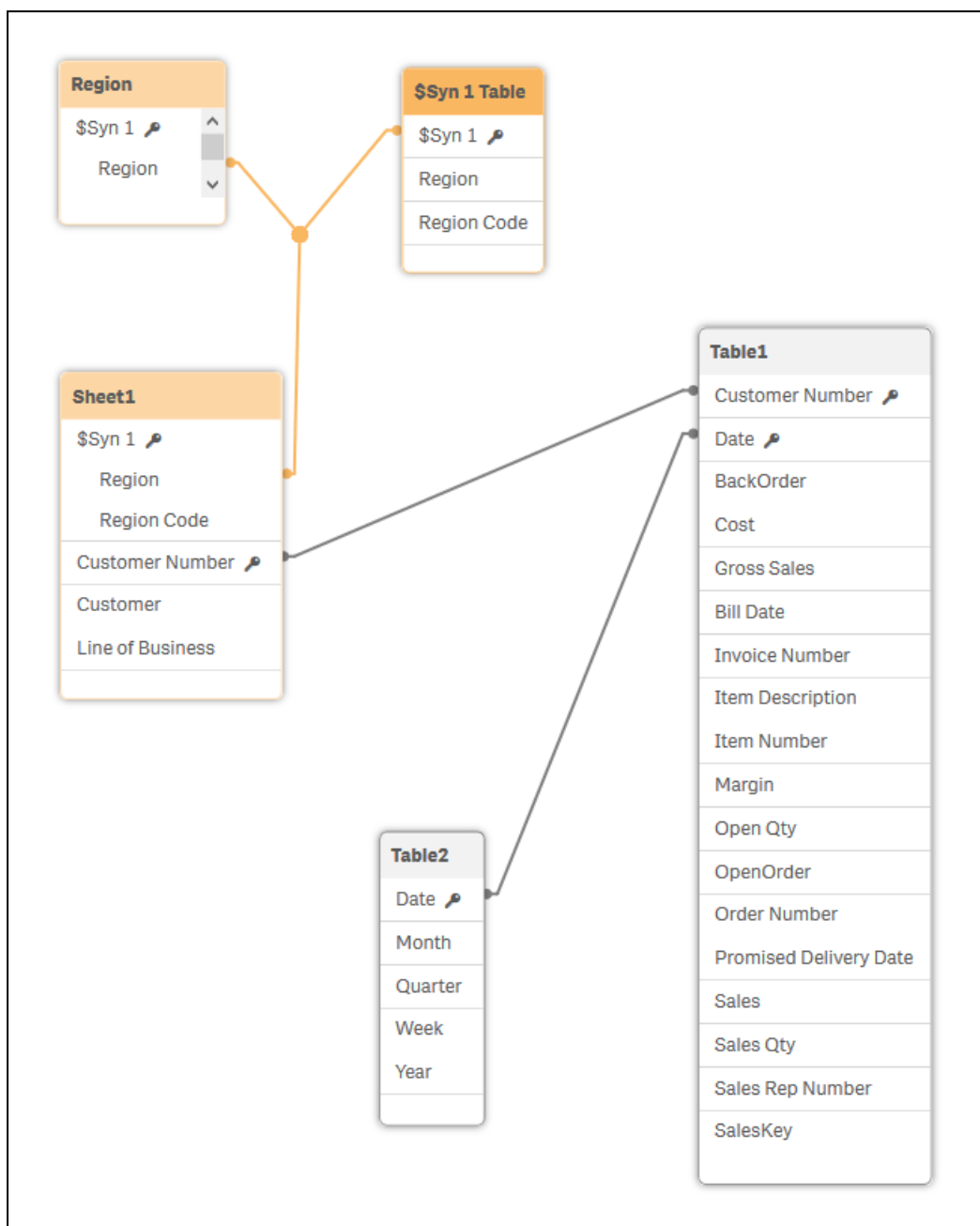
合成キーの警告が表示されたデータロード進捗状況ウィンドウ



9. [データモデルビューア]を開きます。

\$Syn 1 Table という新しいテーブルが作成されていることから、合成キーが作成されたことがわかります。このテーブルには、結合されたテーブル **Sheet1** と **Region** に共通するすべての項目、**Region** および **Region code** が含まれています。このままではテーブル間の関連性がわかりにくく、誤解を招く可能性があるため、望ましい状況とは言えません。

合成キーが表示されたデータモデルビューアー



3.7 合成キーの解決

合成キーを排除する最も簡単な方法は、テーブル内の1つまたは複数の項目の名前を変更することです。名前の変更はデータをロードする際に実行できます。それでは、合成キーを削除する方法を見ていきましょう。

次の手順を実行します。

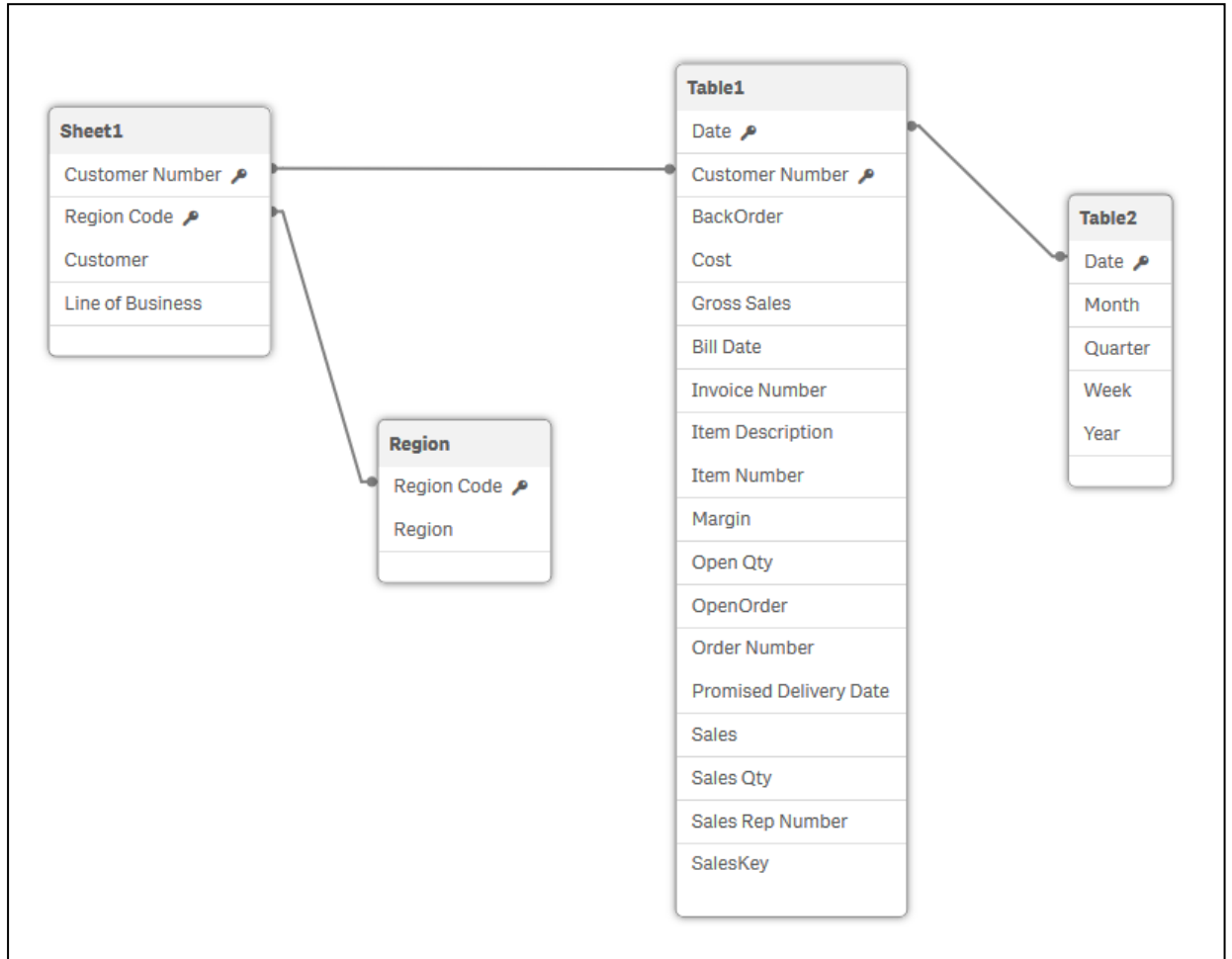
1. データロードエディターを開きます。
2. **Customers** セクションをクリックし、**LOAD** ステートメントから次の行を削除します。

Region,

3. [データのロード] をクリックします。
4. [データモデル ビューア] を開きます。

合成キーが削除されました。

合成キーが削除されたデータモデル ビューア



4 アプリでのデータ使用

このチュートリアル最終段階として、ロードしたデータをアプリのビジュアライゼーションに取り込みます。

4.1 チャートの追加

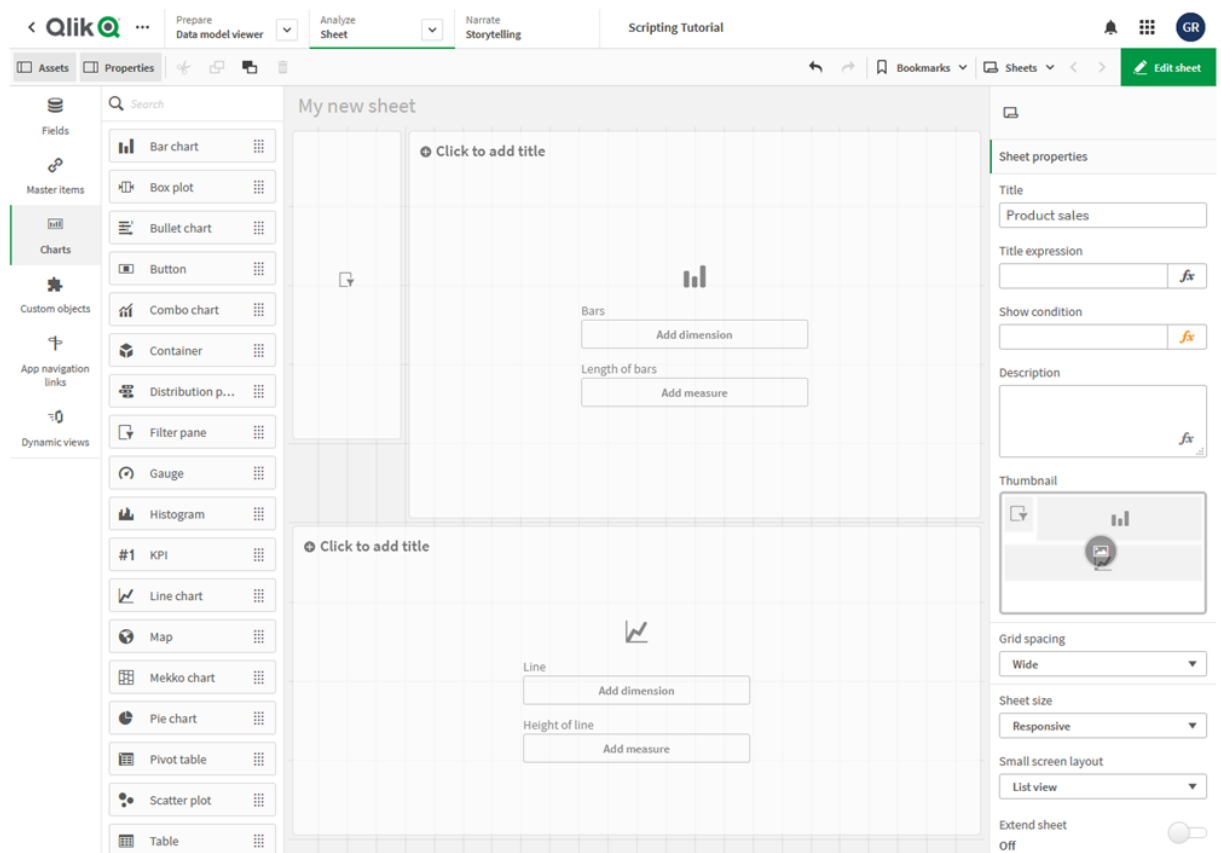
データがロードされ、データを使ってチャートを作成できるようになりました。チャートはビジュアライゼーションともいいます。必要な軸とメジャーを追加するまでチャートは完成しません。軸とメジャーの追加は、チャートを追加してから行います。

次の手順を実行します。

1. **Scripting Tutorial** アプリ内に新しいシートを作成します。
2. [チャート] タブからフィルター パネルをシートにドラッグし、ハンドルを使って、幅がセル 3 個分、高さがセル 4 個分になるようにサイズを変更します。これをシートの左上に配置します。
3. 棒グラフを右上にドラッグし、高さをセル 5 個分、幅はシートの端まで十分拡大できるだけの余裕を持たせます。
4. 折れ線グラフを残りのスペースまでドラッグします。

追加されたチャートの種類は、シート上に表示されるアイコンで確認できます。これでチャートに軸とメジャーを追加し、ビジュアライゼーションを作成できます。

空のチャートがある **Qlik Sense** シート




4.2 軸とメジャーの追加

次のステップでは、軸とメジャーを追加します。まず、時間軸を左上のフィルター パネルに追加します。フィルター パネルは、スペースの節約に有効です。**Year**、**Quarter**、**Month**、**Week** のそれぞれに個別のフィルター パネルを設ける代わりに、同じ目的で単一のフィルター パネルを使用します。

軸の作成と追加

次の手順を実行します。


1. アセット パネルの左上にある  をクリックし、**[項目]** を開きます。ここにはデータ ロード エディターにロードした全 テーブルの項目 がすべて表示 されます。
2. リストの最下部までスクロールダウンし、**Year** 項目 をクリックします。この項目 を左上にあるフィルター パネルの中央にドラッグします。
3. 同様の方法で **Quarter**、**Month**、**Week** をフィルター パネルに追加します。
Year、**Quarter**、**Month**、**Week** の 4 つの軸 があるフィルター パネルが作成 されました。

メジャーの作成と追加

通常、ビジュアライゼーションでは軸とメジャーが必要になります。メジャーとは数式の結果で、多くの場合は **Sum**、**Max**、**Min**、**Avg** (平均)、**Count** などの基本的な関数です。

棒グラフには地域ごとの売上が表示 されます。

次の手順を実行します。


1. **[項目]**  をクリックします。
2. 項目 **Region** をクリックし、棒グラフエリアの中央にドラッグします。
3. **[Add "Region"]** ("**Region**" を追加) をクリックします。
4. 項目 **Sales** をクリックし、棒グラフエリアの中央にドラッグします。
5. **[メジャーとして追加] > [Sum (Sales)]** をクリックします。
6. 右側のプロパティ パネルで、**[スタイル]** をクリックしてから **[プレゼンテーション]** をクリックします。**[水平]** を選択します。
これで棒が水平に表示 されます。
7. 右側のプロパティ パネルで **[ソート]** をクリックします。
ソート順が表示 されます。
8. **Region** の上にある **Sum([Sales])** をドラッグし、軸が **Region** (軸値、アルファベット順) ではなく、**Sum([Sales])** (メジャー値) によってソートされるようにします。
棒グラフが完了し、異なる地域の売上結果が表示 されます。これは基本的な棒グラフです。プロパティ パネル (右側) には、これを拡張するためのオプションが複数あります。1 つの例として、タイトル エリアにタイトル以外を入力してみます。
9. 次の文字列をコピーして、棒グラフのタイトル項目 に貼り付けてください。

= 'Total sales: \$' & Round(Sum(Sales)/1000000, 0.1) & 'M'

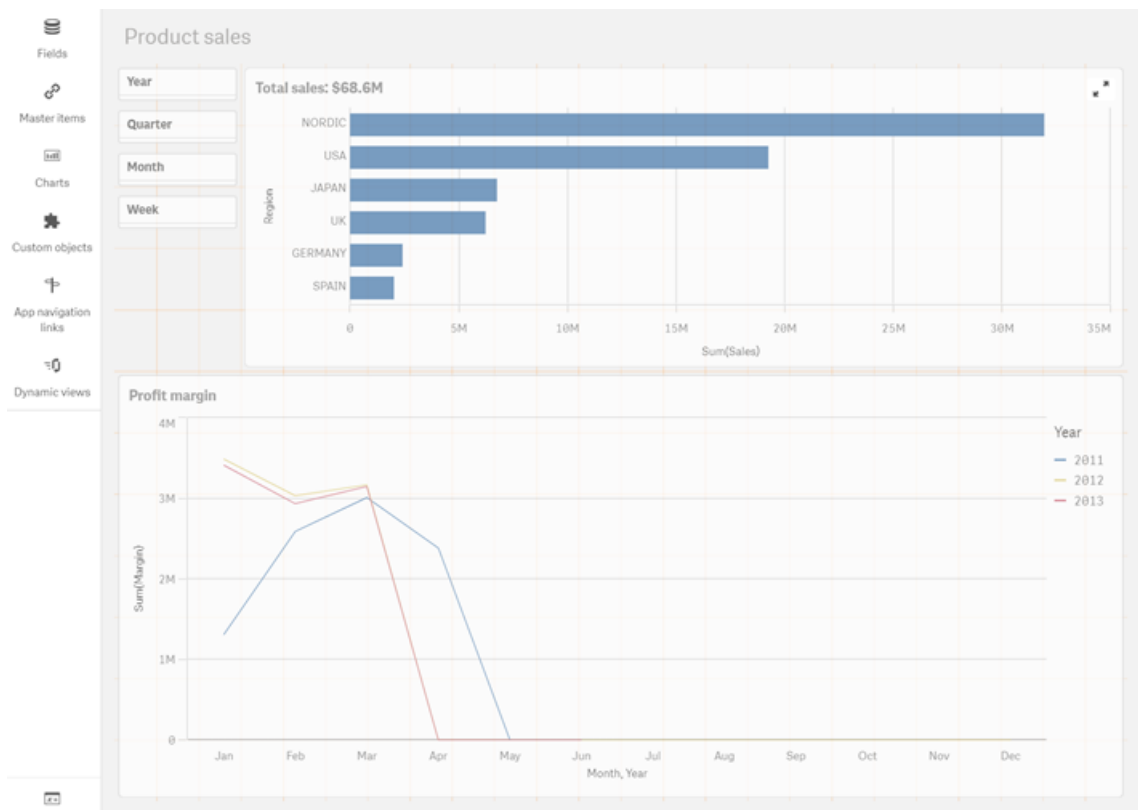
10. Enter を押します。

このシートの最終的なビジュアライゼーションは折れ線グラフです。

次の手順を実行します。

1. [項目 - 2. 項目 *Month* をクリックし、折れ線グラフエリアの中央にドラッグします。
- 3. [Add "Month"] ("Month" を追加) をクリックします。
- 4. 項目 *Year* をクリックし、折れ線グラフエリアの中央にドラッグします。
- 5. [Add "Year"] ("Year" を追加) をクリックします。
- 6. 項目 *Margin* をクリックし、折れ線グラフエリアの中央にドラッグします。
- 7. [メジャーとして追加] > [Sum (Margin)] をクリックします。
- 8. 折れ線グラフの上に「Profit margin」というタイトルを追加します。

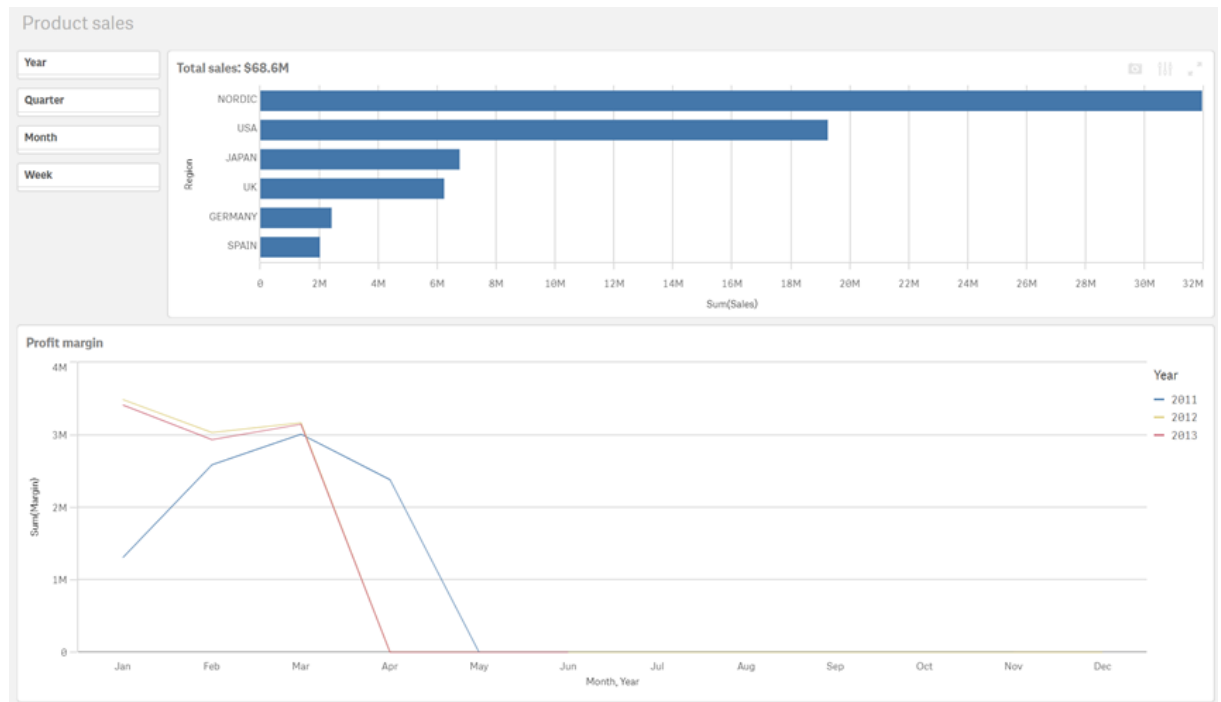
データを示すチャート



9. シートの編集を終了します。

これでシートが完成し、アイテムのクリックやコンテンツの利用が可能になりました。

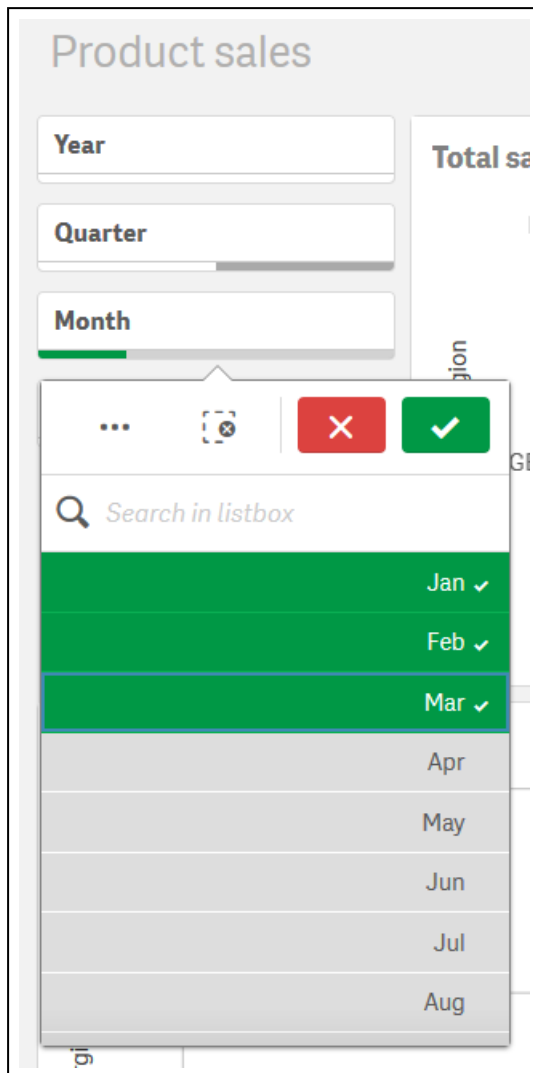
完成したシート



元の **Sales.xlsx** ファイルで月次売上データの量を制限しているため、毎年 3 月末以降はグラフに使用できるデータがほとんどありません。フィルター パネルで選択して、毎年最初の 3 か月のみを比較できます。

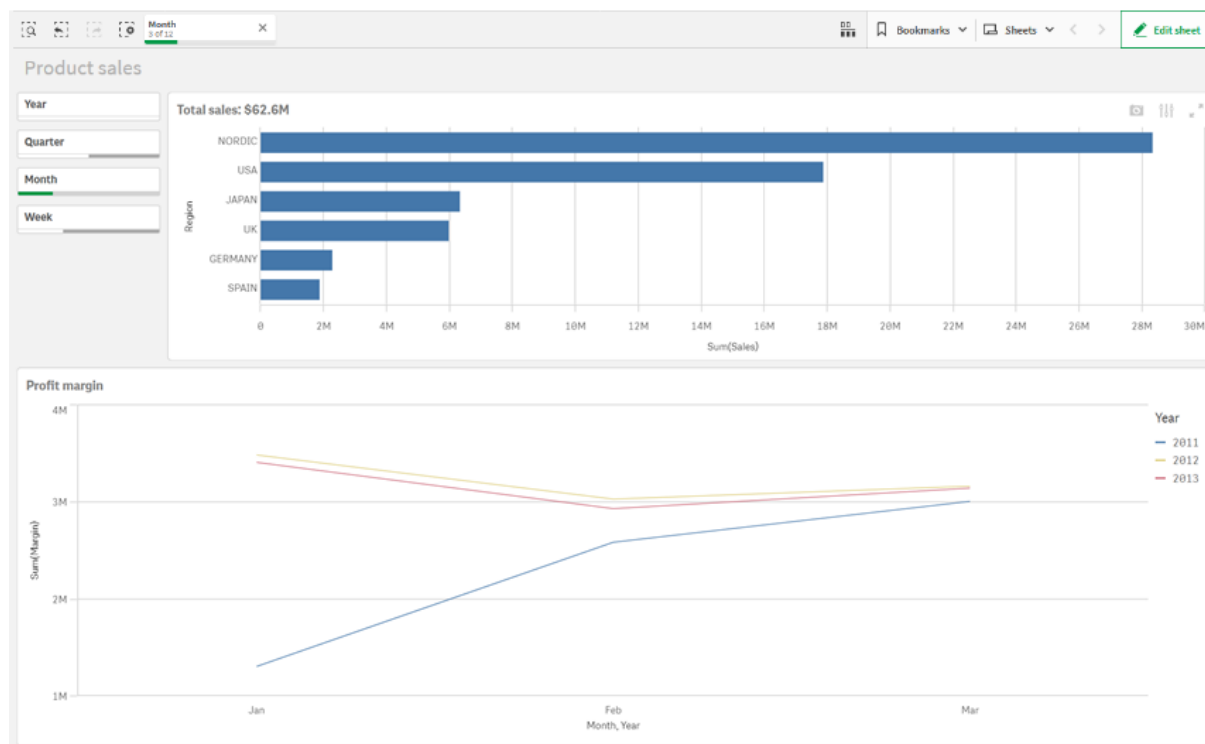
10. フィルター ボックスで **Month** 項目 をクリックし、**Jan**、**Feb**、**Mar** を選択します。

フィルター ボックス



11. フィルター パネルを閉じます。*Profit margin* チャートに、毎年最初の 3 か月のデータが表示されています。

選択に基づいて更新された *Profit margin* チャート



4.3 お疲れ様でした!

以上でチュートリアル終了です。ここでご紹介した基本情報を **Qlik Sense** でのスクリプト作成にお役立てください。**Web** サイトでは、アプリ作成のヒントやアイデアを公開しています。ぜひご利用ください。