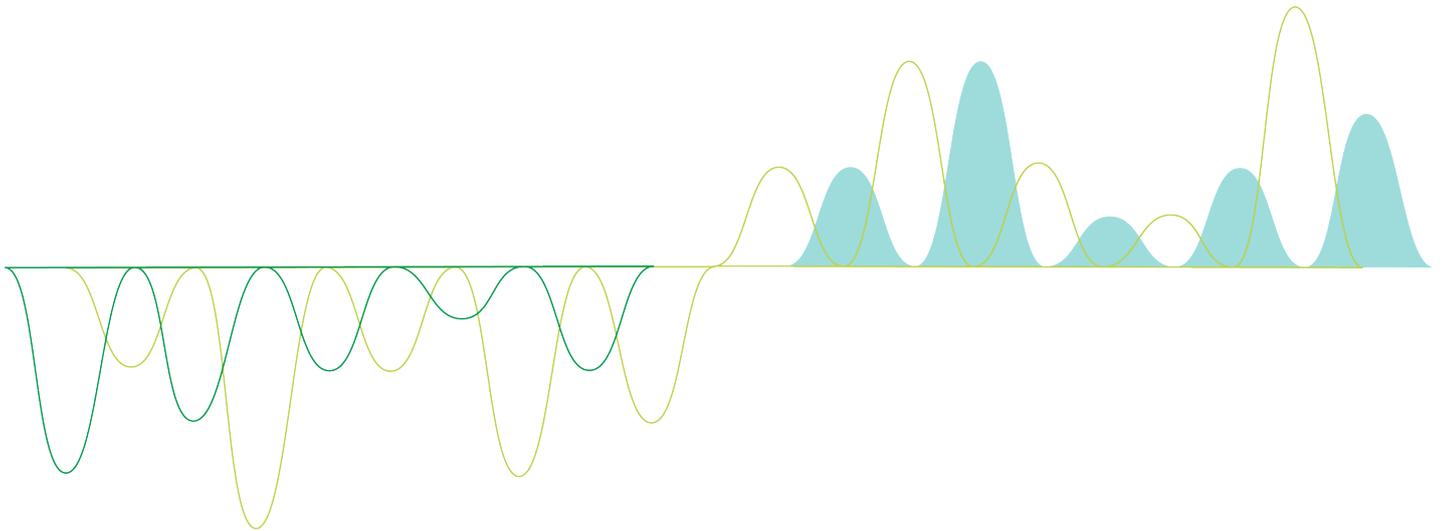


教程 - 新手脚本编写

Qlik Sense®

November 2022

版权所有 © 1993-2023 QlikTech International AB。保留所有权利。



1 欢迎学习本教程!	4
1.1 您将学到的内容	4
1.2 谁应当参加该课程	4
1.3 压缩包内容	4
1.4 本教程中的课程	4
1.5 延伸阅读和资源	5
2 在数据加载编辑器中编写脚本	6
2.1 数据加载编辑器	6
2.2 脚本编辑器	6
访问命令和函数的语法帮助	6
脚本中的注释	7
缩进代码	7
插入准备的测试脚本	7
2 LOAD 和 SELECT 语句	8
3 选择和加载数据	9
3 重命名字段	16
3 减少数据	18
3 转换数据	23
3.1 Resident LOAD	23
3.2 Preceding LOAD	26
3 多表格串联	28
3.3 自动串联	28
3.4 强制串联	31
3.5 防止串联	32
3 循环引用	34
3.6 解决循环引用	35
3 合成键	37
3.7 解决合成键	39
4 在应用程序中使用数据	41
4.1 添加图表	41
4.2 添加维度和度量	41
创建和添加维度	42
创建和添加度量	42
4.3 谢谢!	46

1 欢迎学习本教程!

欢迎学习本教程,我们将为您介绍 **Qlik Sense** 中的基本脚本编译。

在 **Qlik Sense** 中的应用程序里创建可视化效果之前,必须加载数据。了解如何使用加载脚本可以让您在将数据加载到应用程序中时准备和操作数据。

您可以使用数据管理器或数据加载编辑器加载数据。如果要创建、编辑和运行数据加载脚本,可以使用数据加载编辑器。

1.1 您将学到的内容

完成本教程后,您应该熟悉使用脚本加载数据、编辑脚本和转换数据。

1.2 谁应当参加该课程

您应当熟悉 **Qlik Sense** 的基础知识。也就是说,你已经创建了应用程序和可视化。

您需要数据加载编辑器的访问权限并且应当得到在 **Qlik Sense Enterprise on Windows** 中加载数据的许可。

1.3 压缩包内容

您下载的压缩包中含有完成本教程所需的以下数据文件:

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

此外,程序包含有脚本编写教程应用程序的副本。您可将应用程序上传至您的应用中心。

我们建议按照教程中描述的方式自行构建应用程序,以最大限度地提高学习效果。此外,您还必须按照教程中的说明上传并连接到数据文件,数据加载才能正常工作。

但是,如果遇到问题,应用程序可能会帮助您排除故障。我们已经指出了与每节课相关联的脚本段。

1.4 本教程中的课程

根据您对于 **Qlik Sense** 的经验,完成本教程需要 3-4 小时。各主题设计为按顺序完成。但您也可以暂时离开,然后随时再回来继续学习。幸运的是,不会有测验。

- 数据加载简介
- **LOAD** 和 **SELECT** 语句
- 选择和加载数据

- 重命名字段
- 减少数据
- 转换数据
- 多表格串联
- 循环引用
- 合成键
- 在应用程序中使用数据

1.5 延伸阅读和资源

- [Qlik](#) 提供了各种各样的资源帮助您进行深入学习。
- [Qlik 在线帮助](#) 可供使用。
- 培训, 包括免费的在线课程, 可在 [Qlik Continuous Classroom](#) 获取。
- 讨论论坛、博客等可见于 [Qlik Community](#)。

2 在数据加载编辑器中编写脚本

Qlik Sense 使用在数据加载编辑器中管理的数据加载脚本，连接到不同的数据源，并在其中检索数据。数据源可为数据文件，例如 Excel 文件或 .csv 文件。数据源也可谓数据库，例如 Google BigQuery 或 Salesforce 数据库。

还可使用数据管理器将数据载入，但是当您希望创建、编辑和运行数据加载脚本时，可使用数据加载编辑器。

在脚本中，指定要加载的字段和表格。脚本编译通常用于指定从数据源加载的数据。还可以使用脚本语句操作数据结构。

数据加载期间，Qlik Sense 会确定不同表格的共用字段(关键字)以关联数据。可以在数据模型查看器中监控应用程序的数据的结果数据结构。可以通过重命名字段来实现对数据结构的更改，以获得表格之间的不同关联。

在将数据加载到 Qlik Sense 后，会将其存储在应用程序中。

2.1 数据加载编辑器

您可创建脚本以在数据加载编辑器中加载数据。编辑器可通过 Qlik Sense 中的下拉菜单使用。

打开数据加载编辑器时，脚本编辑器将显示在屏幕中间。脚本段在左侧菜单中显示为标签。Qlik Sense 自动创建主段。数据连接显示在右侧菜单中。

必须使用 Qlik Sense 脚本语法编写脚本。Qlik Sense 语法关键字以蓝色突出显示。

数据加载编辑器



2.2 脚本编辑器

可以使用编辑器中的许多现有功能帮助您开发加载脚本。

访问命令和函数的语法帮助

可以通过多种方法访问 Qlik Sense 语法关键字的语法帮助。

访问帮助门户网站

可以通过两种方法访问 Qlik Sense 帮助门户网站中的详细帮助。

- 单击工具栏中的  可进入语法帮助模式。在语法帮助模式下, 可以单击语法关键字(标为蓝色且带有下划线) 以访问语法帮助。
- 将光标放置在关键字内或结尾, 然后按 **Ctrl+H** 键。



无法在语法帮助模式下编辑脚本。

使用自动完成功能

如果开始输入 **Qlik Sense** 脚本关键字, 将会显示可从中选择匹配关键字的自动完成列表。当您继续输入时, 该列表的范围会缩小, 您可以从具有建议语法和参数的模板中进行选择。工具提示显示函数的语法, 包括参数和附加语句以及指向语句或函数帮助门户说明的链接。



您也可以使用键盘快捷键 **Ctrl+空格键** 来显示关键字列表, 使用 **Ctrl+Shift+空格键** 来显示工具提示。

脚本中的注释

您可以在脚本代码中插入注释, 或者通过使用注释标记停用部分脚本代码。可以将 **//** (两个正斜杠) 右边后面的行中的所有文本看作是注释, 当脚本运行时将不会执行这些注释。

数据加载编辑器工具栏包含注释或取消注释代码的快捷键。单击  或按 **Ctrl + K** 以注释或取消注释代码。

缩进代码

您可以缩进代码以增加可读性。单击  可缩进文本(增加缩进), 单击  可减少缩进文本(减少缩进)。

插入准备的测试脚本

您可以插入准备的测试脚本以加载一组内联的数据字段。您可以使用此脚本来快速创建用于测试目的的数据集。按 **Ctrl + 00** 以插入测试脚本。

2 LOAD 和 SELECT 语句

每一语句都将生成一个内部表格。可以使用 **LOAD** 和 **SELECT** 语句将数据加载到 Qlik Sense。LOAD 用于从文件加载数据，而 **SELECT** 用于从数据库加载数据。

在本教程中，您将使用来自文件的数据，因此您将使用 **LOAD** 语句。

您还可以使用前导 **LOAD** 来操作所加载数据的内容。例如，必须在 **LOAD** 语句中重命名字段，而 **SELECT** 语句不允许更改字段名。

在将数据加载到 Qlik Sense 时应用以下规则：

- 在 Qlik Sense 中，**LOAD** 或 **SELECT** 语句生成的表格之间无任何区别。因此，在加载多个表格时，这些表格是由 **LOAD** 或 **SELECT** 语句加载，抑或由这两者共同加载都无关紧要。
- 语句或数据库中原始表格的字段顺序是由 Qlik Sense 逻辑随机排列。
- 字段名区分大小写，用于在数据表之间建立关联。因此，有时必须重命名加载脚本中的字段以获得所需的数据模型。

3 选择和加载数据

使用数据加载编辑器中的数据选择对话框, 可以轻松地从文件(如 **Microsoft Excel**) 或任何其他受支持的文件格式加载数据。

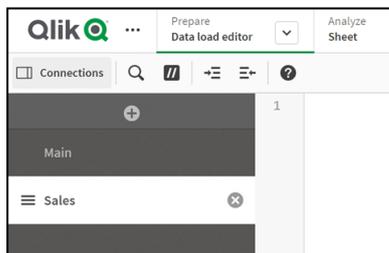
执行以下操作:

1. 打开 **Qlik Sense**。
2. 创建新应用程序。
3. 命名应用程序 *Scripting Tutorial*, 再单击**创建**。
4. 打开应用程序。



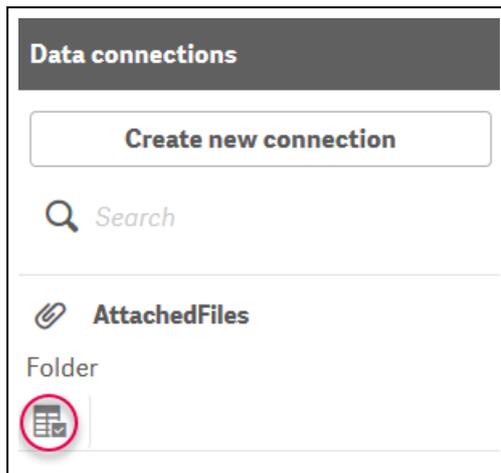
在您首次将数据载入应用程序之前, 有选项可使用**添加数据**, 从而可方便地从文件加载数据。尽管如此, 在本教程中, 我们希望看到脚本, 因此我们将使用数据加载编辑器。

5. 从顶部工具栏的下拉菜单中打开数据加载编辑器。
6. 在左侧菜单中单击 **+** 以在名为 *Main* 的部分下面添加新的脚本段。
使用多个段可以让您的脚本井井有条。加载数据时, 脚本段将按顺序执行。
7. 通过输入 *Sales*, 为段提供一个名称。
Scripting Tutorial 应用程序中新的 *Sales* 标签



8. 在右侧菜单的 **AttachedFiles** 下, 单击**选择数据**。

选择数据窗口



9. 上传然后选择 *Sales.xlsx*。数据预览窗口打开。

Sales 数据文件的数据预览窗口

# of Days Late	# of Days to Ship	BackOrder	Cost	Customer Number	Date	GrossSales	Invoice Date	Invoice Number	Item Desc	Item Number	Margin	Op
0	22.28	10000000		10000433	01/25/2011	64.56	01/25/2011	319976	Nationeel Potato Chips	10847	39.7	
0	1.77	10000433		10000433	01/29/2011	0.00	01/29/2011	320435	Ebony Asparagus	10795	-1.77	
0	3.86	10000433		10000433	01/28/2011	5.65	01/28/2011	320294	Pearl Chardonnay Wine	10895	1.56	
0	8.84	10000433		10000433	01/28/2011	29.29	01/28/2011	320274	Tell Tale Firm Tots	10505	10.64	
0	4.47	10000433		10000433	01/28/2011	20.49	01/28/2011	320294	Great Muffins	10279	15.2	
0	10.96	10000433		10000433	01/28/2011	20.59	01/28/2011	320274	Fast Grape Fruit Roll	10558	8.81	
0	5.3	10000433		10000433	01/28/2011	25.35	01/28/2011	320294	Golden Waffles	10990	19.04	
0	13.16	10000433		10000433	01/28/2011	31.63	01/28/2011	320294	Ebony New Potatoes	10797	17.2	
0	15.32	10000433		10000433	01/28/2011	42.31	01/28/2011	320294	High Top Tomatoes	10187	25.3	
0	27.71	10000433		10000433	01/28/2011	45.50	01/28/2011	320294	High Top Golden Delicious Apples	10197	15.97	
0	40.81	10000433		10000433	01/28/2011	48.88	01/28/2011	320294	Fast Golden Rainins	10561	6.11	
0	26.46	10000433		10000433	01/28/2011	48.86	01/28/2011	320294	Onover Spaghetti	10890	20.43	
0	19.55	10000433		10000433	01/28/2011	63.16	01/28/2011	320294	Bravo Beef Soup	10649	41.08	
0	23.11	10000433		10000433	01/28/2011	84.59	01/28/2011	320294	High Top Cauliflower	10186	58.1	
0	52.91	10000433		10000433	01/28/2011	104.68	01/28/2011	320263	Ebony Plums	10823	47.58	
0	35.94	10000433		10000433	01/28/2011	110.27	01/28/2011	320294	Fast Dried Apples	10504	49.92	
0	77.1	10000433		10000433	01/28/2011	156.50	01/28/2011	320265	Just Right Chicken Ramen Soup	10967	73.14	
0	85.22	10000433		10000433	01/28/2011	157.70	01/28/2011	320294	Moms Sliced Chicken	10387	66.17	
0	113.58	10000433		10000433	01/28/2011	162.74	01/28/2011	320294	High Top Golden Delicious Apples	10197	42.63	

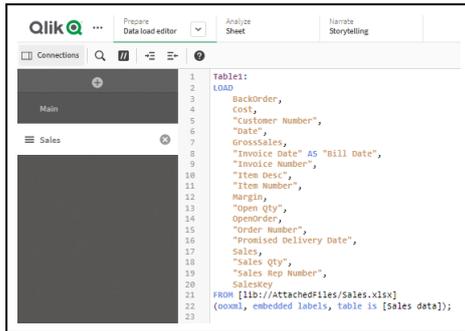
```

LOAD
    "# of Days Late",
    "# of Days to Ship",
    BackOrder,
    Cost,
    "Customer Number",
    "date",
    GrossSales,
    "Invoice Date",
    "Invoice Number",
    "Item Desc",
    "Item Number",
    Margin,

```

- 取消选择字段 *# of Days Late* 和 *# of Days to Ship*。您可能需要单击字段标题以查看完整的字段名称。
- 在筛选器字段搜索字段搜索 *date*
- 单击标题 *Invoice Date* 并键入 *Bill Date* 来重命名字段。
- 单击插入脚本。加载脚本插入脚本编辑器的 *Sales* 部分。注意 Qlik Sense 将双引号置于包含空格的字段名两旁。
您的脚本应如下所示：

在 *Sales* 标签中加载脚本



14. 在 *LOAD* 语句上方添加以下行以命名表格 *Table1*:

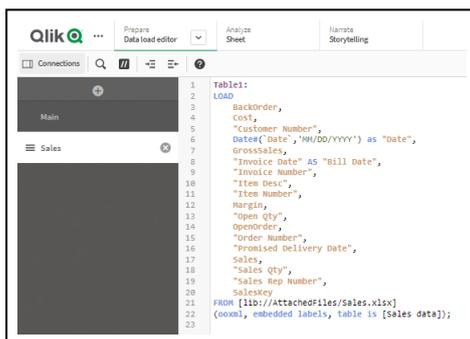
Table1:

15. 现在, 对脚本进行调整以确保日期得以正确解释。将 *Date* 字段更改为以下内容:

Date#(`Date`, 'MM/DD/YYYY') as "Date",

您的脚本应如下所示:

在 *Sales* 标签中更新的加载脚本



16. 在右上角, 单击**加载数据**。

这将把数据载入应用程序。随即显示脚本执行进度窗口。完成后, 您将看到可能错误和合成键的摘要(即使没有合成键)。

17. 单击**关闭**。

18. 从顶部工具栏的下拉菜单中打开数据模型查看器。通过单击 , 数据模型查看器将在新标签中打开。

19. 选择顶部菜单中的  和  以显示在本教程中使用的表格视图。如果您的表格未正确显示, 那么可以删除现有加载脚本并重新构建脚本。

Sales 数据的数据模型查看器中的表格视图

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

现在我们加载另一名为 *Dates* 的表格。在加载表格之后，Qlik Sense 将把它与日期字段上的 *Sales* 表格连接。

20. 打开**数据加载编辑器**。
21. 单击  以新增脚本段。
22. 命名部分为 *Dates*。如果尚未将新段 *Dates* 放置在 *Sales* 下方，可以将指针移动到  拖动条上，然后将此新段向下拖动到段 *Sales* 下方以重新排列顺序。
23. 单击脚本的顶部行，然后单击 。
确保将 // 添加至脚本。
24. 在 // 后添加以下文本：
`Loading data from Dates.xlsx`
脚本的第一行现在应该如下所示：
`// Loading data from Dates.xlsx`
25. 在右侧菜单的 **AttachedFiles** 下，单击**选择数据**。



确保选中了**字段名称**下面的**嵌入的字段名称**以包含您加载数据时表格字段的名称。

26. 上传然后选择 *Dates.xlsx*。数据预览窗口打开。

Dates 数据文件的数据预览窗口

Select data from Dates.xlsx

File format: Excel (xlsx) | Embedded field names: [v] | Header size: 1, 0

Fields:

Date	Month	Quarter	Week	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
1/29/2011	Jan	Q1	5	2011
2/1/2011	Feb	Q1	5	2011
2/2/2011	Feb	Q1	6	2011
2/3/2011	Feb	Q1	6	2011
2/4/2011	Feb	Q1	6	2011
2/5/2011	Feb	Q1	6	2011
2/8/2011	Feb	Q1	6	2011
2/9/2011	Feb	Q1	7	2011
2/10/2011	Feb	Q1	7	2011

```

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

27. 单击**插入脚本**。

您的脚本应如下所示：

在 *Dates* 标签中加载脚本

```

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

28. 在 *LOAD* 语句上方的行中添加以下内容以命名表格 *Table2*：

Table2：

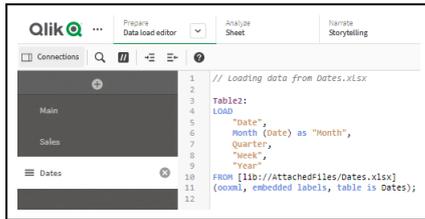
29. 要确保在 Qlik Sense 中正确地解释文件 *Dates.xlsx* 中的 *Month* 列，我们需要将 *Month* 函数应用至 *Date* 字段。

将 *Month* 字段更改为以下内容：

Month (Date) as "Month",

您的脚本应如下所示：

在 *Dates* 标签中更新的加载脚本



现在您已创建了用于从文件 *Dates.xlsx* 加载所选数据的脚本。现在需要将数据加载到应用程序：

30. 在右上角，单击**加载数据**。

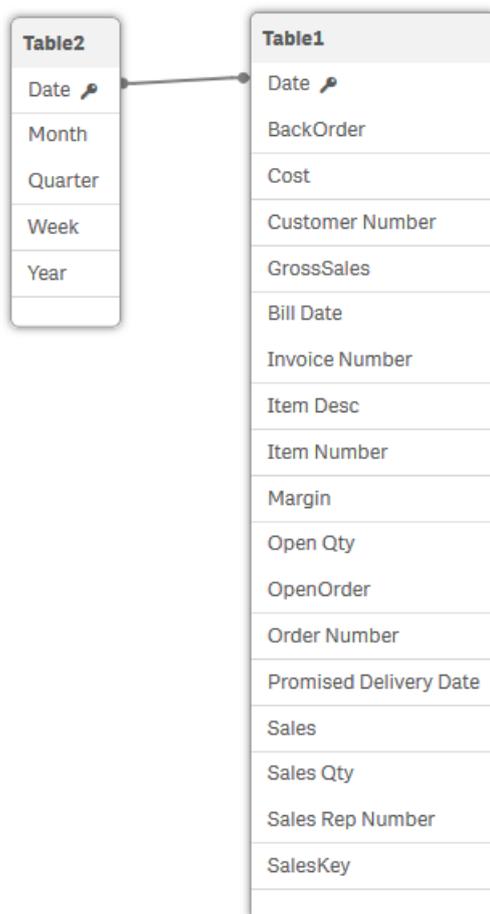
单击**加载数据**后，会将数据加载到应用程序并保存脚本。

31. 在脚本执行完成后，单击**关闭**。

32. 打开**数据模型查看器**。

现在，您可以看到已在两个表格中名为 *Date* 的两个字段之间建立了连接。

数据模型查看器中的表格视图



33. 单击左下角的**预览**。单击表格 *Table2* 的名称。

这将显示有关表格的信息。在**预览**字段中，您现在可以看到，已经有 **628** 行的数据加载到内部表格 *Table2*。如果单击表格中的某个字段，您将看到有关字段的信息。

数据模型查看器中的表格预览

The screenshot shows a data model viewer interface. On the left, there is a list of tables: **Table2** and **Table1**. **Table2** has fields: Date, Month, Quarter, Week, and Year. **Table1** has fields: Date, BackOrder, Cost, Customer Number, GrossSales, Bill Date, Invoice Number, Item Desc, Item Number, Margin, Open Qty, OpenOrder, and Order Number. Below the table list is a 'Preview' section for **Table2**.

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

数据现在可以在应用程序的可视化中使用。我们将在本教程的后面向您展示方法。

3 重命名字段

在上一主题中，我们向您展示了如何在数据预览窗口中重命名字段。在数据预览窗口的标题中，将 *Invoice Date* 重命名成了 *Bill Date*。插入加载脚本时，可以看到该字段将使用关键字 *AS* 重命名。

我们也可以直接在脚本中执行此操作。

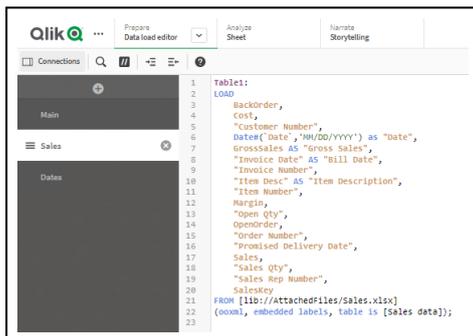
执行以下操作：

1. 在 *Scripting Tutorial* 应用程序中，打开 **数据加载编辑器**。
2. 单击 **Sales** 标签。
3. 在加载脚本中，进行以下更改：请注意，必须在包含空格的字段周围包含括号。

- i. 将 *GrossSales* 更改为：
GrossSales AS "Gross Sales",
- ii. 将 *"Item Desc"* 更改为：
"Item Desc" AS "Item Description",

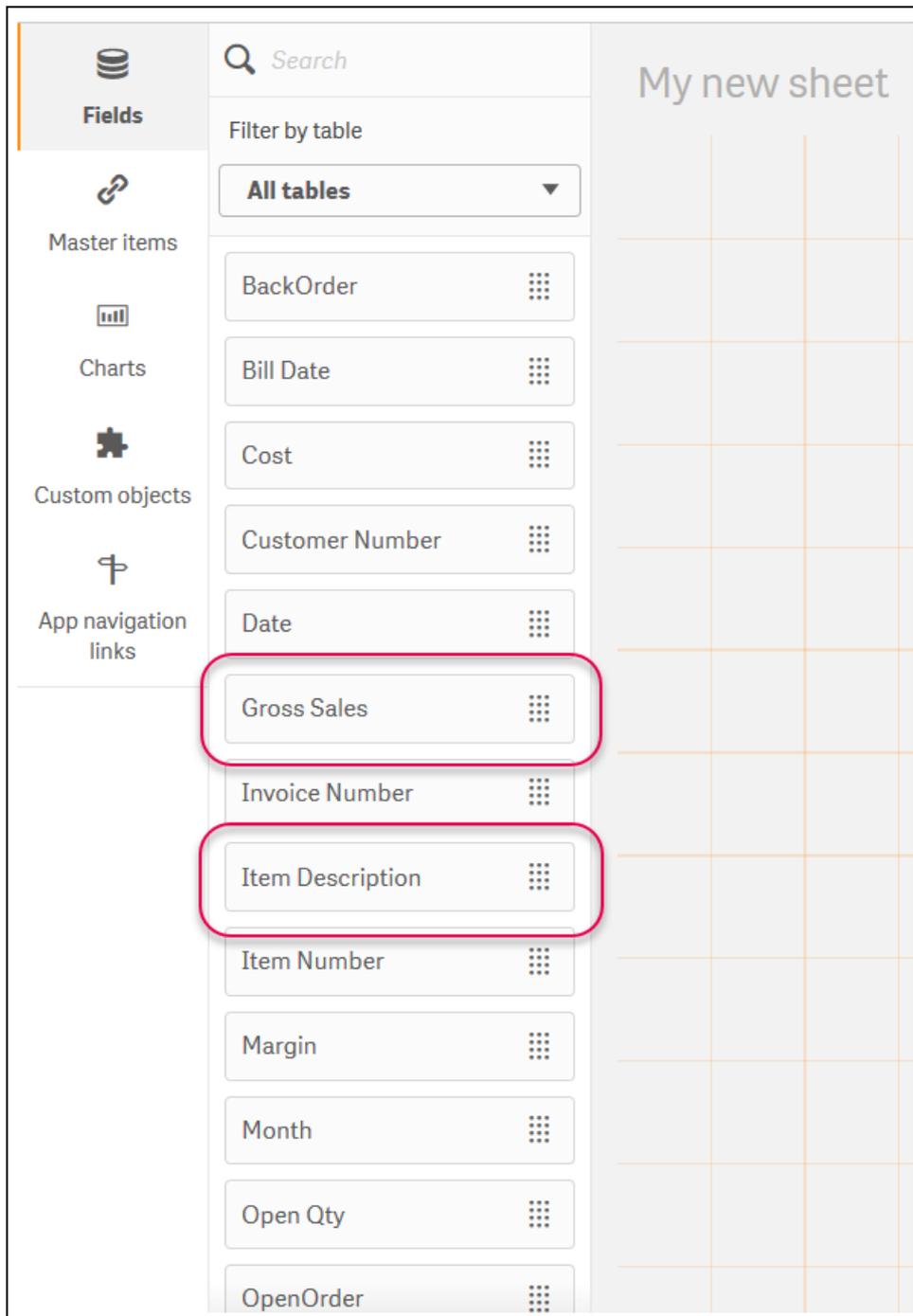
您的脚本应如下所示：

具有重命名字段的脚本的加载脚本窗口



4. 单击 **加载数据**。数据已加载。
5. 打开 **数据模型查看器**。确认已经将字段重命名。
6. 您还可以在应用程序中查看所有字段。单击顶部工具栏中的 **分析/工作表** 标签。应用程序在工作表视图中打开。
7. 单击 **编辑工作表**，然后单击资产面板中的 **字段**。您可以看到您更改的字段名。您可以在应用程序中创建的可视化中使用这些字段中的任何一个。

分析视图中重命名的字段



3 减少数据

Qlik Sense 提供几种不同的方法来减少加载到应用程序中的数据量。例如, 可以从文件或数据连接器筛选数据。

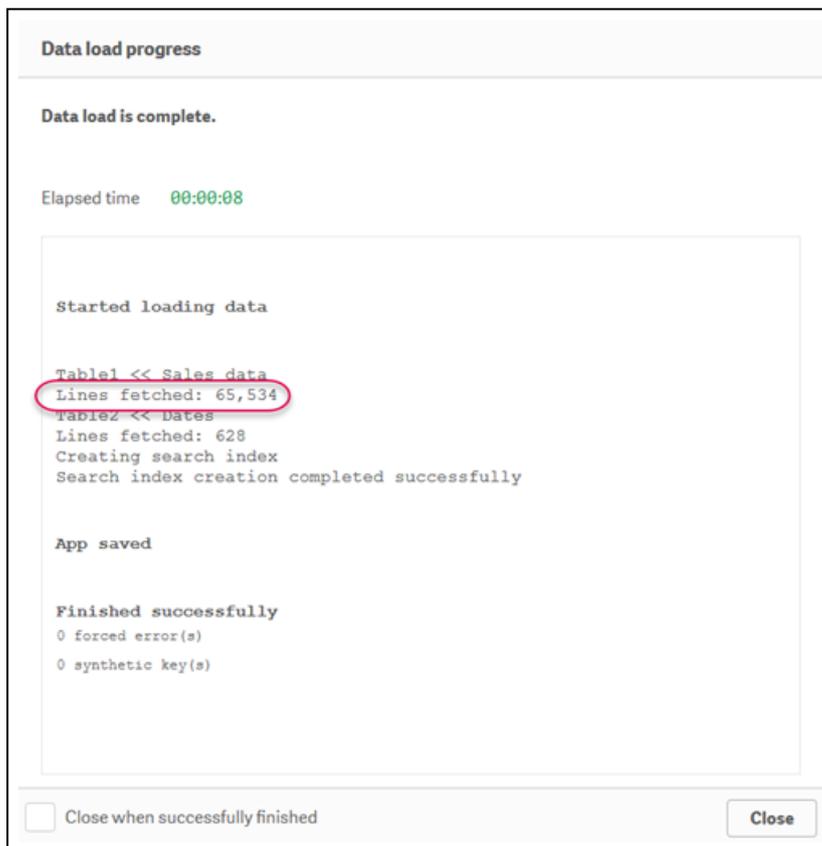
您也可以直接在加载脚本中减少数据。

执行以下操作：

1. 在 *Scripting Tutorial* 应用程序中, 打开 **数据加载编辑器**。
2. 单击 **加载数据**。

基于您目前已编写的加载脚本, Qlik Sense 从 *Sales.xlsx* 数据文件将 **65,534** 行加载到 *Table1* 中。注意, *Sales data* 是初始 *Sales.xlsx* 文件中包含您的表格的标签的名称。

数据加载进度窗口

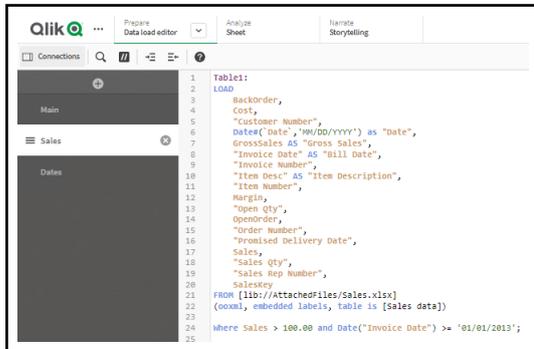


3. 在数据加载编辑器中, 单击 *Sales* 标签。
4. 删除此行末尾的分号:
`(ooxml, embedded labels, table is [Sales data]);`
5. 将以下行添加至加载脚本的末尾:
`where Sales > 100.00 and Date("Invoice Date") >= '01/01/2013';`

这让 Qlik Sense 只加载销售额大于 \$100.00 的数据。它还使用 Date 函数加载日期等于或大于 2013 年 1 月 1 日的数据。

您的脚本应如下所示：

使用脚本减少加载的数据量的加载脚本窗口



```

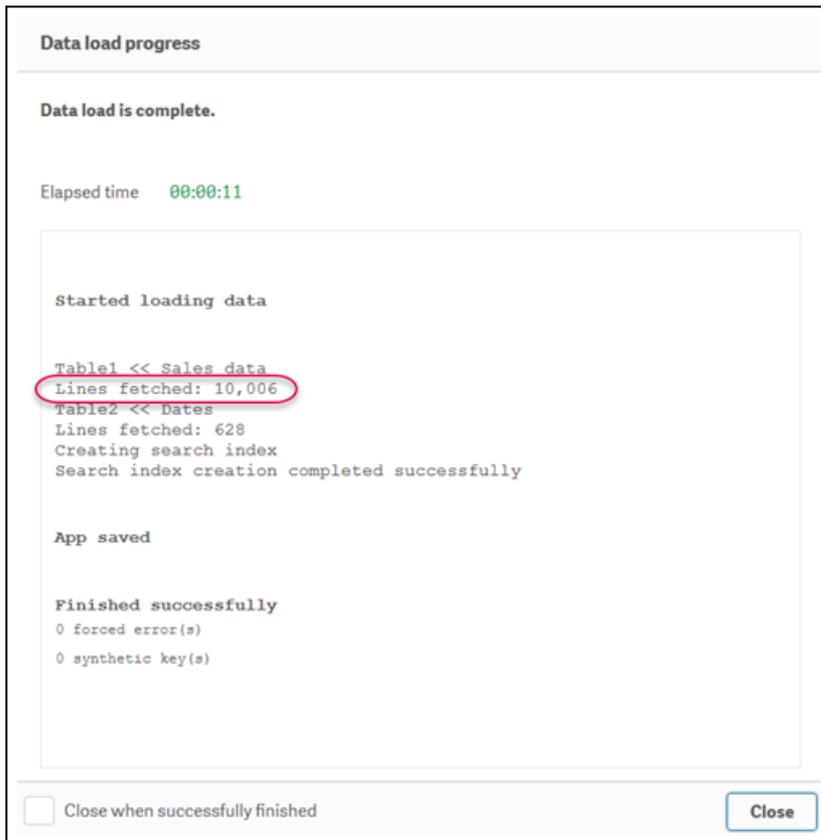
1 Table1:
2 LOAD
3   Backorder,
4   Cost,
5   "Customer Number",
6   Date(Date & '/' & '00/00/0000') as "Date",
7   GrossSales AS "Gross Sales",
8   "Invoice Date" AS "Bill Date",
9   "Invoice Number",
10  "Item Desc" AS "Item Description",
11  "Item Number",
12  Margin,
13  "Open Qty",
14  OpenOrder,
15  "Order Number",
16  "Promised Delivery Date",
17  Sales,
18  "Sales Qty",
19  "Sales Rep Number",
20 SalesKey
21 FROM [lib://AttachedFiles/Sales.xlsx]
22 (ooxml, embedded labels, table is [Sales data])
23
24 where Sales > 100.00 and Date("Invoice Date") >= '01/01/2013';
25

```

6. 单击加载数据。

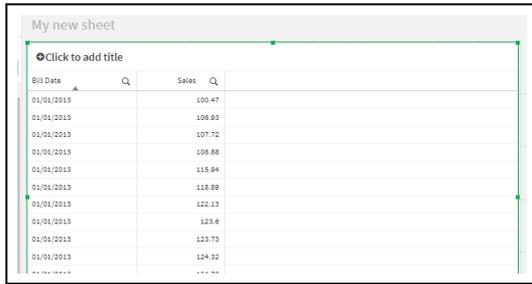
基于更新的加载脚本，Qlik Sense 现在从 *Sales.xlsx* 数据文件加载更少的行。

减少数据加载的数据加载进度窗口



7. 如果将数据添加到应用程序中的表格中，则会看到只会加载符合您创建的条件的数据。

包含 *Bill Date* 和 *Sales* 字段的表格

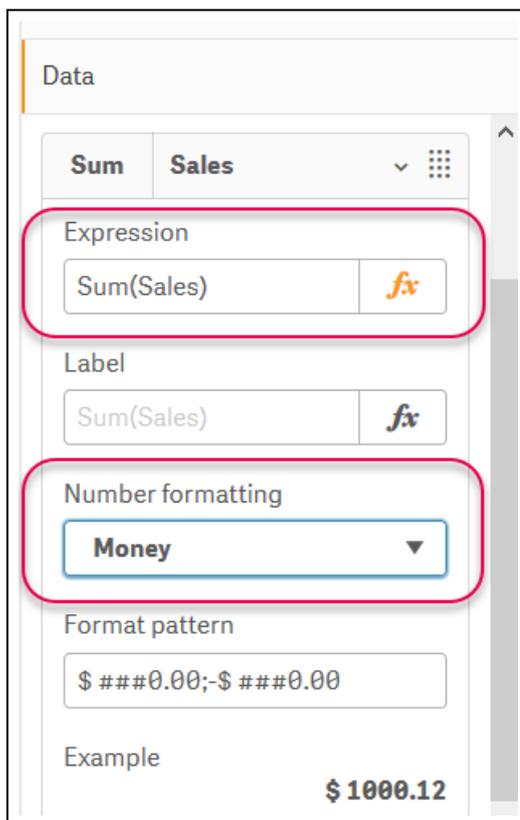


Bill Date	Sales
01/01/2015	100.47
01/01/2015	106.93
01/01/2015	107.72
01/01/2015	108.88
01/01/2015	115.94
01/01/2015	118.89
01/01/2015	122.13
01/01/2015	123.6
01/01/2015	123.73
01/01/2015	124.32

注意, 我们添加了 *Sales* 字段作为维度。这样就可以单独显示 *Sales* 值。如果我们将 *Sales* 添加为度量值, 则这些值将改为按日期聚合。

通常, 您会将 *Sales* 作为度量添加。对于度量, 可以通过对列应用数字格式将值显示为货币量(例如, 美元)。

应用至 *Sales* 度量的数字格式



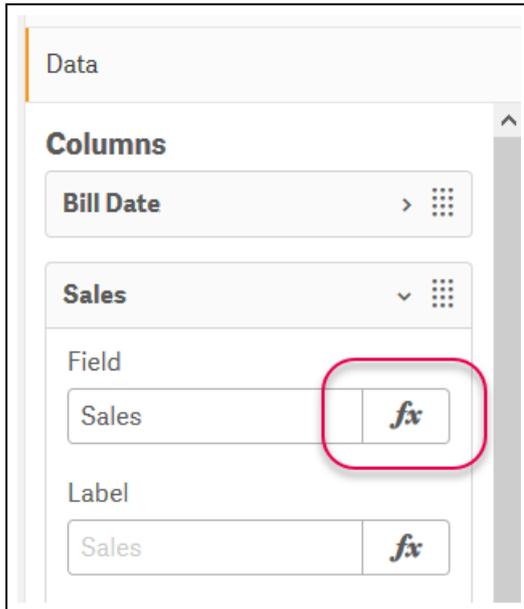
The screenshot shows the configuration panel for a measure named 'Sum(Sales)'. The 'Expression' field contains 'Sum(Sales)' and the 'Label' field also contains 'Sum(Sales)'. The 'Number formatting' dropdown is set to 'Money', which is highlighted with a red box. Below it, the 'Format pattern' is '\$ ###0.00;- \$ ###0.00'. An 'Example' shows the value '\$ 1000.12'.

然而, 当我们使用 *Sales* 作为维度时, 我们需要一个不同的方法。在该情况下, 我们将使用图表表达式。尽管在本教程中我们没有详细讨论图表函数, 但这是一个快速举出示例的好机会。

当我们使用 *Sales* 作为维度时, 值将显示为数字, 如上表所示。

要解决此问题, 可以通过单击 **fx** 然后使用 *Money* 函数打开图表表达式编辑器 *Sales* 字段。

打开表达式编辑器



在表达式编辑器中输入以下内容。

```
=Money( Sales )
```

表达式编辑器



字段中的值现在将显示为货币单位。这些单位(在本例中是美元)在加载脚本的**主要**部分中指定。

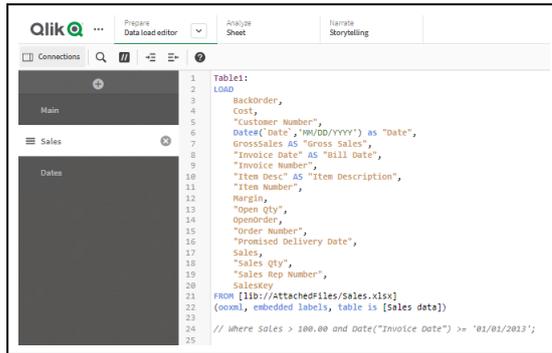
包含 *Bill Date* 和 *Sales* 字段的表格。*Sales* 现在以美元显示

Bill Date	Money(Sales)
01/01/2015	\$100.47
01/01/2015	\$100.93
01/01/2015	\$107.72
01/01/2015	\$108.89
01/01/2015	\$118.94
01/01/2015	\$118.89
01/01/2015	\$122.13
01/01/2015	\$123.60
01/01/2015	\$123.73
01/01/2015	\$124.32

- 现在我们已经完成了这个示例, 请注释掉 **Where** 语句。请勿忘记在第一 **LOAD** 语句的末尾加上分号。

您的脚本应如下所示:

注释掉 Where 语句的加载脚本窗口



The screenshot shows the Qlik Sense Data Load Editor interface. On the left, there is a navigation pane with 'Main', 'Sales', and 'Dates' sections. The main editor area displays a script with the following content:

```
1 Table1:
2 LOAD
3   BackOrder,
4   Cost,
5   "Customer Number",
6   Date#(Date, 'MM/DD/YYYY') as "Date",
7   GrossSales AS "Gross Sales",
8   "Invoice Date" AS "Bill Date",
9   "Invoice Number",
10  "Item Desc" AS "Item Description",
11  "Item Number",
12  Margin,
13  "Open Qty",
14  OpenOrder,
15  "Order Number",
16  "Promised Delivery Date",
17  Sales,
18  "Sales Qty",
19  "Sales Rep Number",
20  SalesKey
21 FROM [lib://AttachedFiles/Sales.xlsx]
22 (ooxml, embedded labels, table is [Sales data])
23 // Where Sales > 100.00 and Date("Invoice Date") >= '01/01/2013';
24
25
```

3 转换数据

您可以在数据加载编辑器中使用许多不同的技术来转换和操作数据。

数据操作的其中一个优势是,您可以选择从文件中仅加载数据的子集(如某个表格中的几个选定列),以便更高效地处理数据。还可以多次加载数据以将原始数据分隔为多个新的逻辑表格。还可以从多个源加载数据并将其合并到 Qlik Sense。

在该主题中,您将使用 **Resident load**, 然后使用 **Preceding load** 来执行一些基本数据转换。

3.1 Resident LOAD

您可以在 **LOAD** 语句中使用 **Resident** 源限定符,以便从之前加载的表格加载数据。当您想要在使用 **SELECT** 语句(没有选项使用 Qlik Sense 函数)加载的数据上执行计算时(例如日期或数值处理),这也很有用。

在该示例中,您将创建名为 *Sales_Buckets* 的新表格,然后使用 **resident load** 从 *Table1* 加载数据。在 *Sales_Buckets* 表格中,您将创建名为 *quantity_threshold* 的变量,然后使用 **Where** 语句以仅加载符合该阈值的数据。

1. 在 *Scripting Tutorial* 应用程序中,打开**数据加载编辑器**。
2. 单击 **Sales** 选项卡。
3. 将以下内容添加至您脚本的末尾:

```
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));
```

您的脚本应如下所示:

通过脚本加载脚本窗口以创建名为 *Sales_Buckets* 的新脚本

```

1 Table:
2 LOAD
3   BackOrder,
4   Cost,
5   "Customer Number",
6   Date("Date", "MM/DD/YYYY") as "Date",
7   GrossSales AS "Gross Sales",
8   "Invoice Date" as "Bill Date",
9   "Invoice Number",
10  "Item Desc" AS "Item Description",
11  "Item Number",
12  Margin,
13  "open Qty",
14  OpenOrder,
15  "Order Number",
16  "Promised Delivery Date",
17  Sales,
18  "Sales Qty",
19  "Sales Rep Number",
20  SalesQty
21 FROM [lib://AttachedFiles/Sales.xlsx]
22 (ooom, 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));
  
```

- 单击**加载数据**。
- 打开**数据模型查看器**。您可以看到自己创建了名为 *Sales_Buckets* 的新表格，具有根据您指定的字段以及您设置的阈值而加载的数据。

数据模型查看器中的 *Sales_Buckets* 表格

Sales_Buckets

Field	Value
High_Quantity	High_Quantity
Item	Item
Customer	Customer

Table2

Field	Value
Date	Date
Month	Month
Quarter	Quarter
Week	Week
Year	Year

Table1

Field	Value
Date	Date
BackOrder	BackOrder
Cost	Cost
Customer Number	Customer Number
Gross Sales	Gross Sales
Bill Date	Bill Date
Invoice Number	Invoice Number
Item Description	Item Description

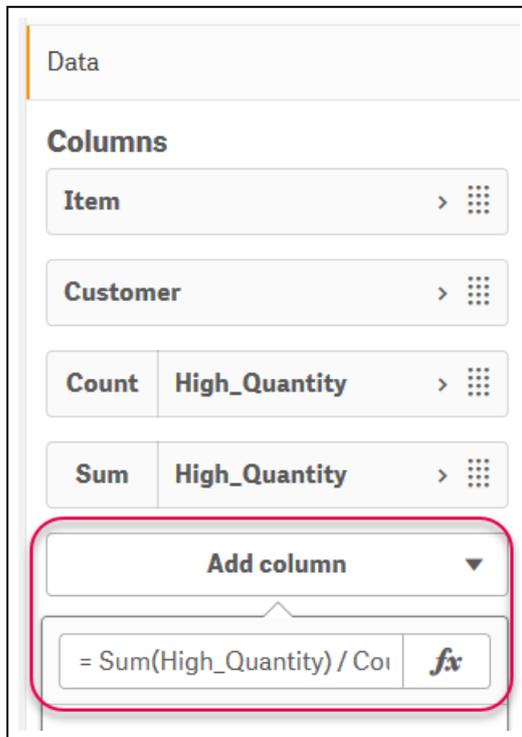
Preview

Sales_Buckets		Preview of data		
Rows	Fields	High_Quantity	Item	Customer
11	3	16000	High Top Dried Mushrooms	10025737
		12500	Blue Label Canned Peas	10017036
		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. 将数据添加至您应用程序中的表格。将 *Item* 和 *Customer* 作为维度添加。将 *High-Quantity* 作为在 *Count* 上聚合然后再次在 *Sum* 上聚合的度量添加。接下来，通过以下公式将新的列作为度量添加。

$$= \text{Sum}(\text{High_Quantity}) / \text{Count}(\text{High_Quantity})$$

使用公式的新度量



例如，您的表格显示 *Customer* 10025737 创建了高顶干蘑菇的 4 个大型订单，平均数量达到 14,800。要在字段中对数据进行排序，通过单击 **完成** 关闭 **编辑** 模式。

显示已经创建大型订单的客户的表格

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

7. 现在我们已经完成了该示例，注释掉了 *quantity_threshold* 变量和 *Sales_Buckets* 表格的脚本。脚本的末尾应当如下显示：

注释掉脚本

```

21 FROM [lib://AttachedFiles/Sales.xlsx]
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 Tables
32 // Where ("Sales Qty" > $(quantity_threshold));

```

3.2 Preceding LOAD

preceding load 可让您执行转变并应用筛选器，从而您可一次加载数据。总的来说，它是一个 **LOAD** 语句，可从下面的 **LOAD** 或 **SELECT** 语句加载，而无需指定通常使用的源限定符，如 **From** 或 **Resident**。您可以按此方法叠加任意数量的 **LOAD** 语句。首先将会对底部的语句进行求值，然后再对上面的语句进行求值，以此类推，直至求值到顶部的语句。

如本教程中之前所述，可使用 **LOAD** 和 **SELECT** 语句将数据加载到 **Qlik Sense** 中。每一语句都将生成一个内部表格。**LOAD** 用于从文件或内联表格加载数据，而 **SELECT** 用于从数据库加载数据。在本教程中您已经使用了来自文件的数据。在本例中，您将使用内联表。但是，值得注意的是，可以在 **SELECT** 语句上方使用 **preceding load** 来操作数据。基本原理与您在此所见的使用 **LOAD** 相同。

此示例与我们在本教程中加载的数据无关。此示例只是用于表明 **preceding load** 形式的示例。您将在数据加载编辑器中创建名为 **Transactions** 的内联表。日期解释将在 **preceding LOAD** 中执行，其中将创建名为 **transaction_date** 的新字段。该字段是从 **sale_date** 字段创建。

1. 新建应用程序并将其命名为 **ReformatDate**。
2. 打开数据加载编辑器，然后创建一个名为 **TransactionData** 的新标签。
3. 添加以下脚本：

```

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 ];

```

您的脚本应如下所示：

通过 *preceding load* 加载脚本

```

Main
TransactionData
1 Transactions:
2 Load *,
3 Date(Date#(sale_date,'YYYYMMDD'),'DD/MM/YYYY') as transaction_date;
4 Load * Inline [ transaction_id, sale_date, transaction_amount, transaction_quantity, customer_id, size, color_code
5 3750, 20180830, 23.56, 2, 2038593, L, Red
6 3751, 20180907, 556.31, 6, 203521, m, orange
7 3752, 20180916, 5.75, 1, 5646471, S, blue
8 3753, 20180922, 125.00, 7, 3036491, l, Black
9 3754, 20180922, 484.21, 13, 049681, xs, Red
10 3756, 20180922, 59.18, 2, 2038593, M, Blue
11 3757, 20180923, 177.42, 21, 203521, XL, Black ];
12

```

- 单击**加载数据**。
- 打开**数据模型查看器**。选择并展开 *Transactions* 表格。可以看到, 所有字段都是按照 *preceding load* 语句中 * 指定的方式加载的。创建了名为 *transaction_date* 的新字段。字段具有重新格式化的日期。

数据模型查看器中名为 *transaction_date* 的新字段

Transactions
transaction_id
sale_date
transaction_amount
transaction_quantity
customer_id
size
color_code
transaction_date

▼ Preview

Transactions	Preview of data								
	transaction_id	sale_date	transaction_amount	transaction_quantity	customer_id	size	color_code	transaction_date	
Rows	7								
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 多表格串联

串联是指获取两个表格并将两个表格组合成一个表格的操作。

两个表格中，一个表格堆叠在另一个表格上方，每个不同列名称占据一列，从而使两个表格彼此增加。数据不会更改，并且合并后的表格包含的记录数与两个原始表格的总记录数相同。此外，还可连续执行多条串联操作，以串联两个以上的表格。

3.3 自动串联

如果两个或多个已加载表格的字段名和字段数量完全相同，**Qlik Sense** 会自动串联不同语句的内容到一个表格中。

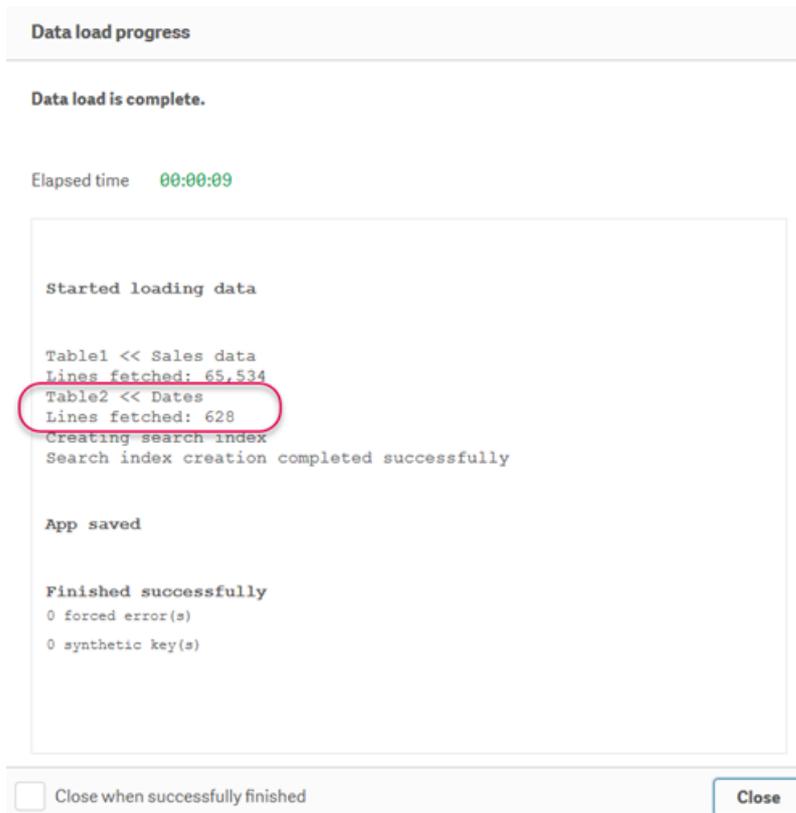
字段的数量和名称必须与要发生的自动串联完全相同。两个 **LOAD** 语句的顺序是任意的，但表格会指定首先加载的表格的名称。

执行以下操作：

1. 在 *Scripting Tutorial* 应用程序中，打开**数据加载编辑器**。
2. 单击**Dates**选项卡。
3. 单击**加载数据**。

介于您目前已编写的加载脚本，**Qlik Sense** 从 *Dates.xlsx* 数据文件将 628 行加载到 *Table2* 中。

数据加载进度窗口



- 在部分 *Dates* 中脚本内的新行上，复制并粘贴 *Table2* 的LOAD 语句。这将导致数据被加载两次。将第二个表格命名为 *Table2a*。

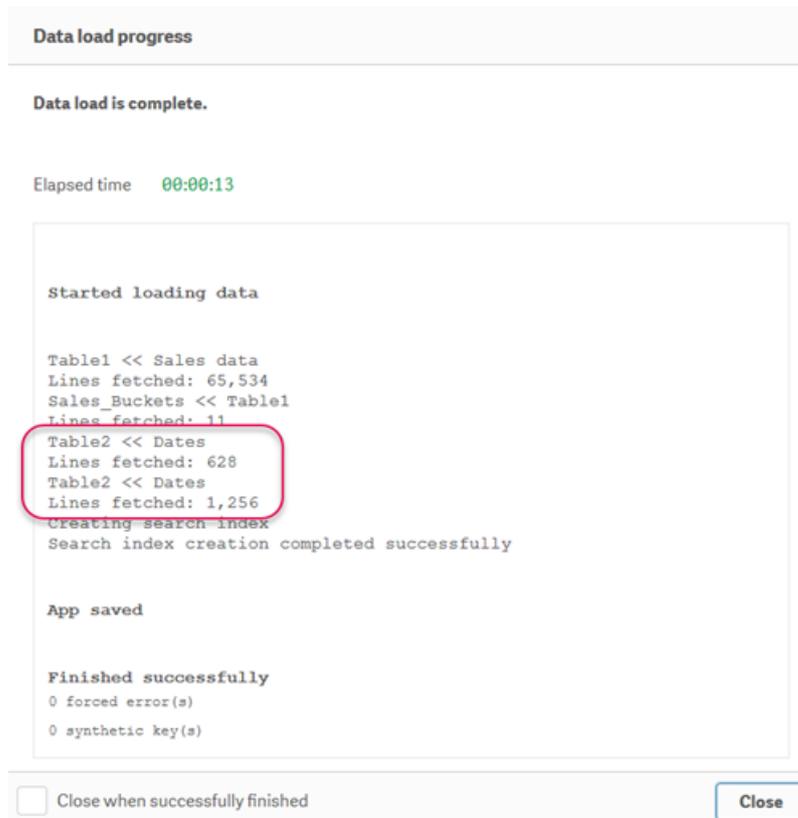
您还可以删除现有脚本，并复制并粘贴以下内容：

```
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);
```

- 单击**加载数据**。

Qlik Sense 不加载 *Table2* 然后是 *Table2a*。它将认为 *Table2a* 具有和 *Table2* 相同的字段名称和字段数目。然后它将 *Table2a* 的数据添加至 *Table2*，然后删除表格 *Table2a*。结果是 *Table2* 现在具有 1,256 行。

数据加载进度窗口中的多表格串联



6. 打开数据模型查看器。
7. 单击显示预览。
仅会创建 *Table2*。选择 *Table2*。表格具有 256 行。

显示 *Table2* 的数据模型查看器

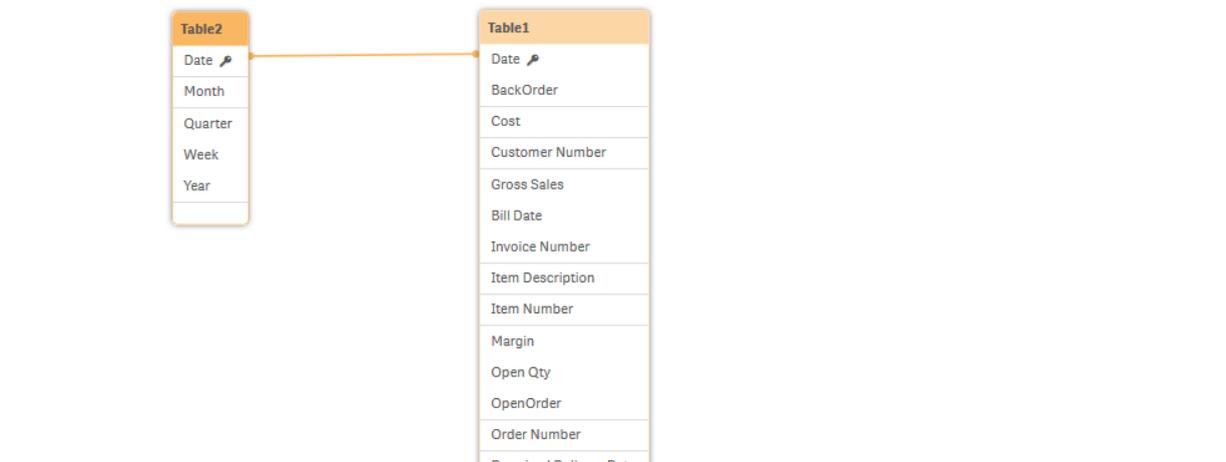


Table2		Preview of data				
Rows	1256	Date	Month	Quarter	Week	Year
Fields	5	01/12/2011	Jan	Q1	3	2011
Keys	1	01/13/2011	Jan	Q1	3	2011
Tags	\$key \$numeric \$integer \$timestamp \$date \$ascii \$text	01/18/2011	Jan	Q1	3	2011
		01/19/2011	Jan	Q1	4	2011
		01/20/2011	Jan	Q1	4	2011
		01/21/2011	Jan	Q1	4	2011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

3.4 强制串联

即使两个或多个表格的字段集并不完全相同，仍有可能强制 **Qlik Sense** 串联两个表格。这可以使用脚本的 **Concatenate** 前缀来实现，即将一个表格串联至另一个命名表格或最新创建的表格。

执行以下操作：

1. 按如下编辑 *Table2a* 的 **LOAD** 语句：添加 **Concatenate** 并注释掉 *Week*。

您的脚本现在应当如下：

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

通过注释掉 *Week*，我们可确保表格不相同。

2. 单击**加载数据**。
3. 打开**数据模型查看器**。
现在，您可以看到未创建 *Table2a*。
4. 单击数据模型查看器中的 *Table2*，然后单击**预览**。

表格具有字段 *Date*、*Month*、*Quarter*、*Week* 和 *Year*。字段 *Week* 仍会显示，这是因为该字段是从 *Table2* 中加载的。

5. 单击 *Table2* 中的 *Week*。预览显示该字段的非空值数目为 628。但是，如果单击其他任何字段，则会看到非空值的数目为 1256。*Week* 仅从 *Table2* 加载一次。值或记录的数目是 *Table2* 和 *Table2a* 中记录数的总和。

3.5 防止串联

如果两个或多个已加载表格的字段名和字段数量完全相同，Qlik Sense 会自动串联不同语句的内容到一个表格中。可以使用 `NoConcatenate` 语句来防止此行为。然后，使用相关 `LOAD` 或 `SELECT` 语句加载的表格不会与现有表格串联。

执行以下操作：

1. 为了能够完全分离两个表的内容，将 `NoConcatenate` 添加至 *Table2a* 中的 `LOAD` 语句，并重命名字段，使得 Qlik Sense 不会基于匹配字段创建合成键。取消注释 *Table2* 中的 *Week*，使得两个表格具有相同字段。

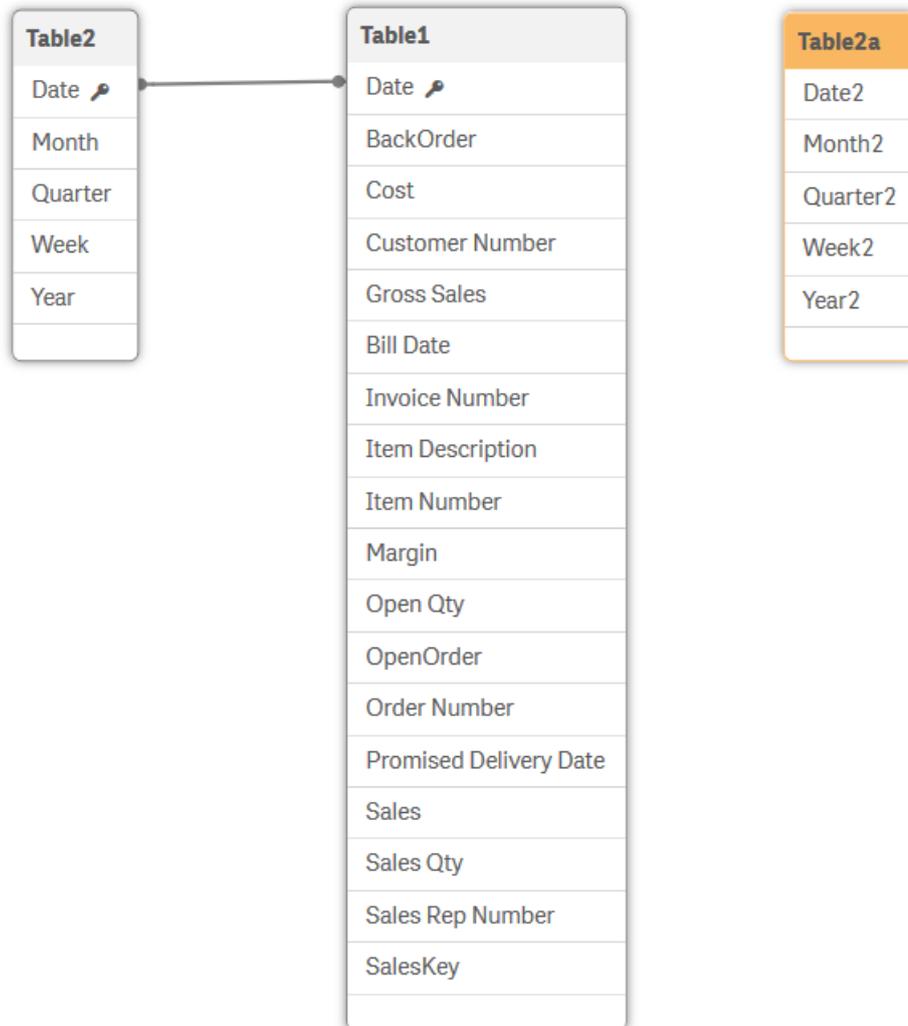
您的脚本现在应当如下：

```
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. 单击**加载数据**。
3. 打开**数据模型查看器**。

现在您可以看到，两个表格已完全分隔。

显示 *Table2* 和 *Table 2a* 的数据模型查看器



4. 现在, 我们已经完成了对串联的演示, 我们不再需要 *Table2a*。删除 *Table2a* 的 LOAD 语句中的所有行, 然后单击**加载数据**。

3 循环引用

如果数据结构中存在循环引用(“循环”),则两个表格在关联时存在不止一条关联路径。

执行以下操作:

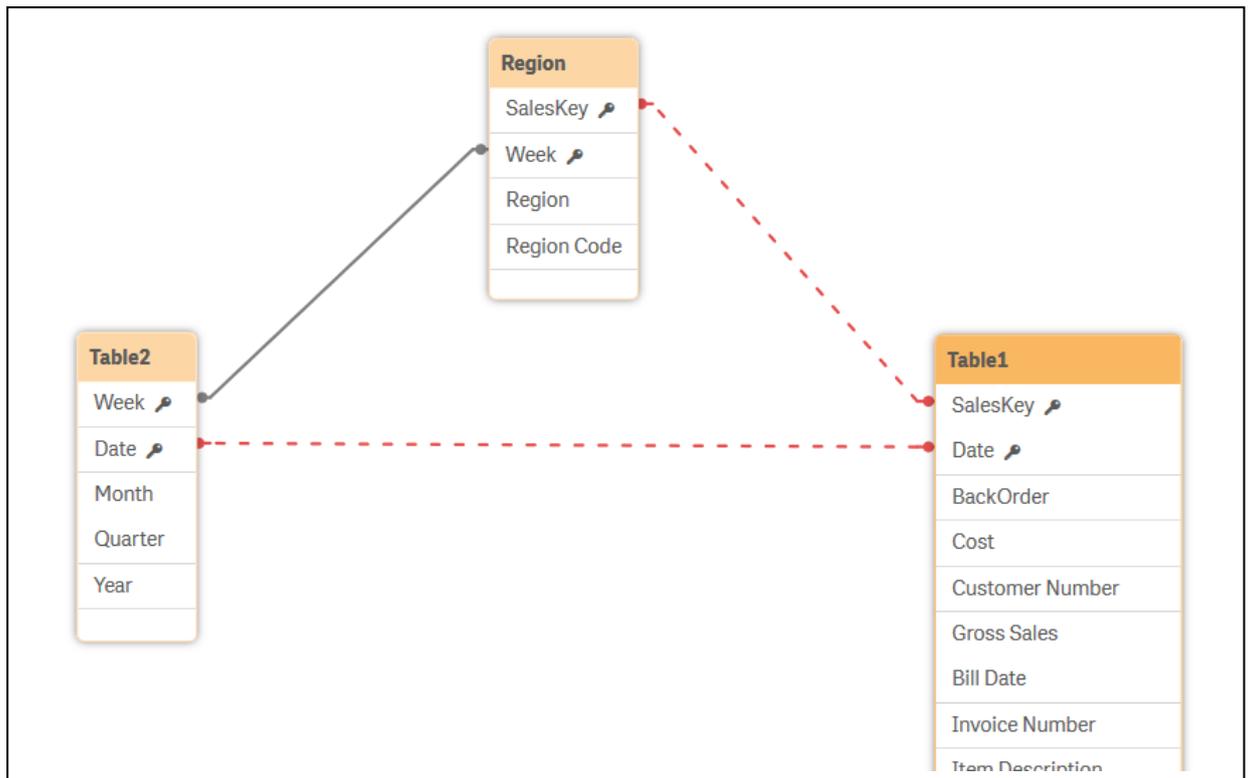
1. 在 *Scripting Tutorial* 应用程序中,打开**数据加载编辑器**。
2. 单击  以新增脚本段。
3. 命名部分为 *Region*。
4. 在右侧菜单的 **AttachedFiles** 下,单击**选择数据**。
5. 上传然后选择 *Region.txt*。数据预览窗口打开。
6. 选择所有字段并确保选中了**字段名称**下面的**嵌入的字段名称**以包含您加载数据时表格字段的名称。
7. 单击**插入脚本**。
8. 单击**加载数据**。

这次,您的数据加载似乎出现错误。已创建循环引用。**数据加载进度**窗口将显示错误信息,指出在加载时发现循环引用。但是,此加载已完成,并且此应用程序已保存。

9. 打开**数据模型查看器**。

您可以在四周拖动表格来重新安排表格,以便轻松查看表格之间的连接。

显示循环引用的数据模型查看器



红色虚线表明已创建了循环引用。应尽可能避免此类数据结构，因为这可能会导致数据解释模糊不清。

3.6 解决循环引用

为了能够了解造成循环引用的原因，让我们进一步在**数据模型查看器**查看表格。

如果查看以上截屏中的表格 *Table1* 和表格 *Table2*，您将看到它们都具有字段 *Date*。您也可以看到 *Table1* 和 *Region* 都具有字段 *SalesKey*。最后，请注意 *Region* 和 *Table2* 都具有字段 *Week*。这意味着，已经创建了一个循环，即循环引用。由于这可能导致以后的数据分析中出现问题，因此让我们来将其删除。

解决这一问题的最简单方法是重命名或删除其中一个字段。在我们的情况中，我们已加载了一些我们的应用程序所不需要的数据，并且我们可以将其删除。

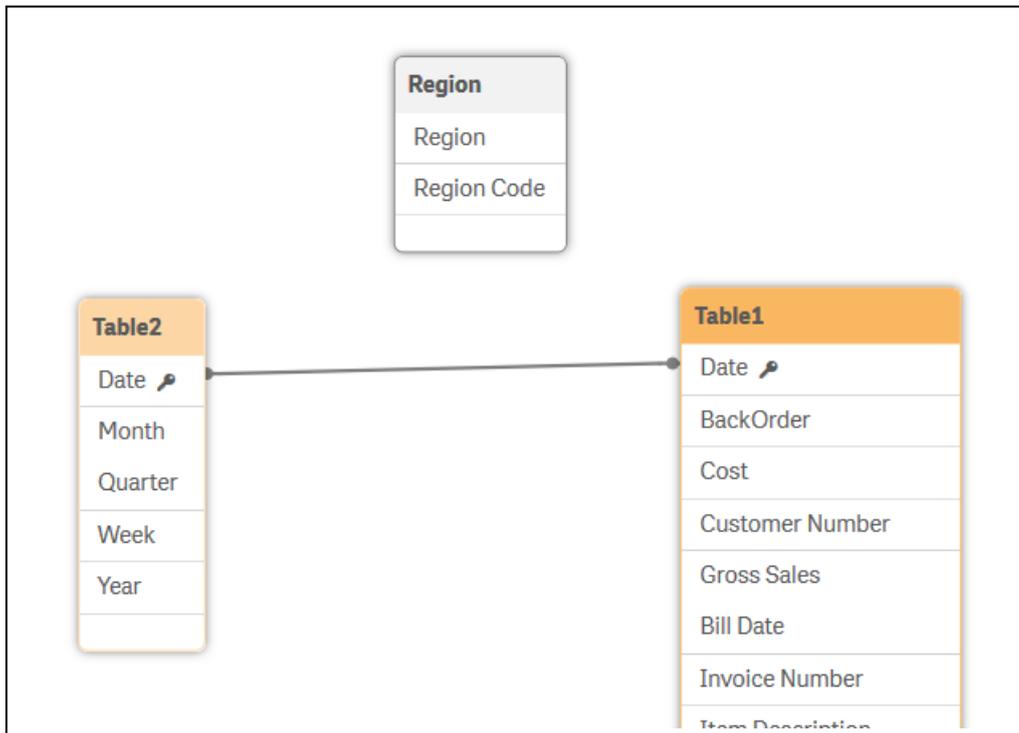
执行以下操作：

1. 打开**数据加载编辑器**。
2. 单击段 *Region* 并删除 *LOAD* 语句中的以下两行：
"week",
saleskey
3. 请务必同时删除 "*Region Code*" 后面的逗号。
您的脚本现在应当如下：

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

4. 单击**加载数据**。
5. 打开**数据模型查看器**。
现在已经删除了对 *Region* 的不必要引用。

现在已经删除了显示循环引用的数据模型查看器



3 合成键

当两个或多个内部表格拥有两个或多个共同字段时，这意味着存在复合的关键字段关系。**Qlik Sense** 会通过合成键来处理。这些关键字段是一种匿名字段，表示所有发生的复合关键字段的组合。

复合键数量的增加会导致内存使用率的提高，并会影响性能。这也可能取决于数据量、表结构和其他因素。只要有多个合成键彼此依赖，那么最佳实践是将其删除。

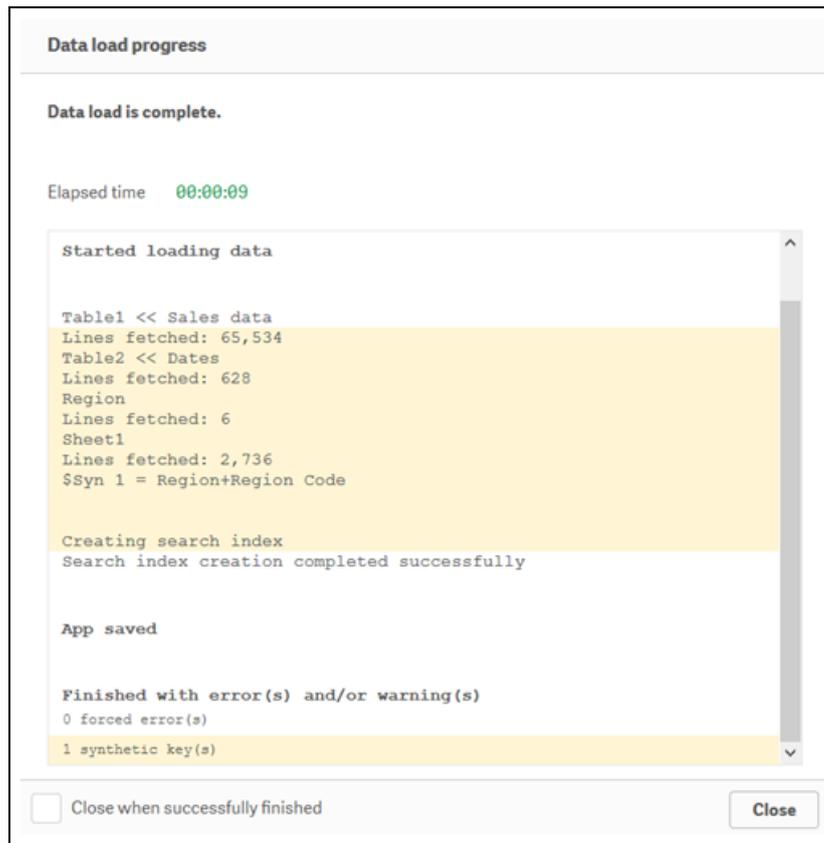
现在需要加载数据的最终集合。

执行以下操作：

1. 在 *Scripting Tutorial* 应用程序中，打开**数据加载编辑器**。
2. 单击  以新增脚本段。
3. 命名部分为 *Customers*。
4. 在右侧菜单的 **AttachedFiles** 下，单击**选择数据**。
5. 上传然后选择 *Customers.xlsx*。数据预览窗口打开。
6. 选择 *Sheet1*
7. 单击**插入脚本**。
8. 单击**加载数据**。

现在您可以在数据加载进度窗口中看到，合成键已经创建。

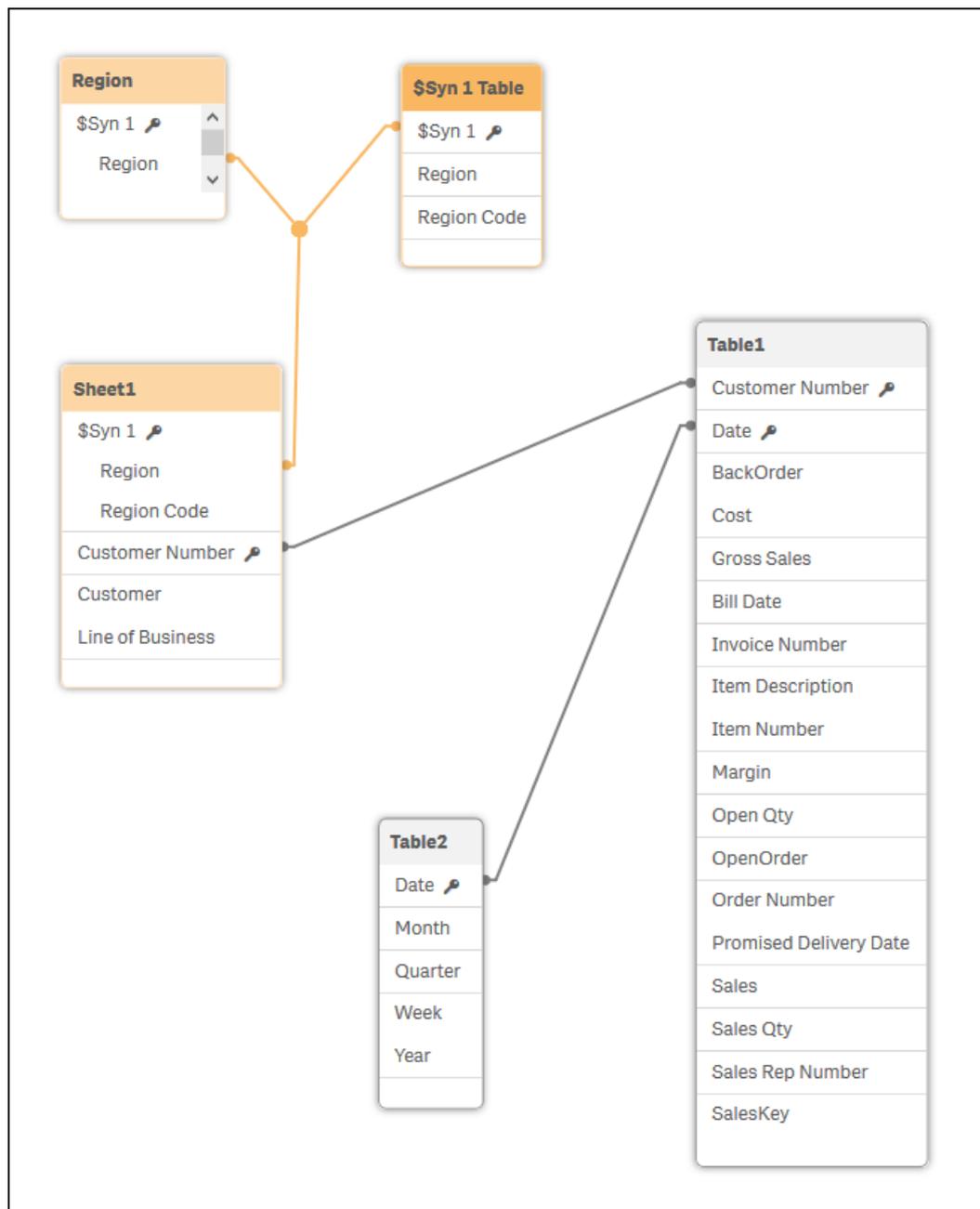
带有合成密钥警告的数据加载进度窗口



9. 打开数据模型查看器。

看到已经创建了新表格 *\$Syn 1 Table*，我们便可看到已经创建了一个合成键。其包含连接的表格 *Sheet1* 和 *Region* 共同具有的所有字段、*Region* 和 *Region code*。在此情况下，会使连接让人混淆并产生误导，因此不需要保留。

显示合成键的数据模型查看器



3.7 解决合成键

消除合成键的最简单方法是以重命名表格中的一个或多个字段。可以在加载数据时进行此操作。现在，我们将逐步进行如何删除合成键的步骤。

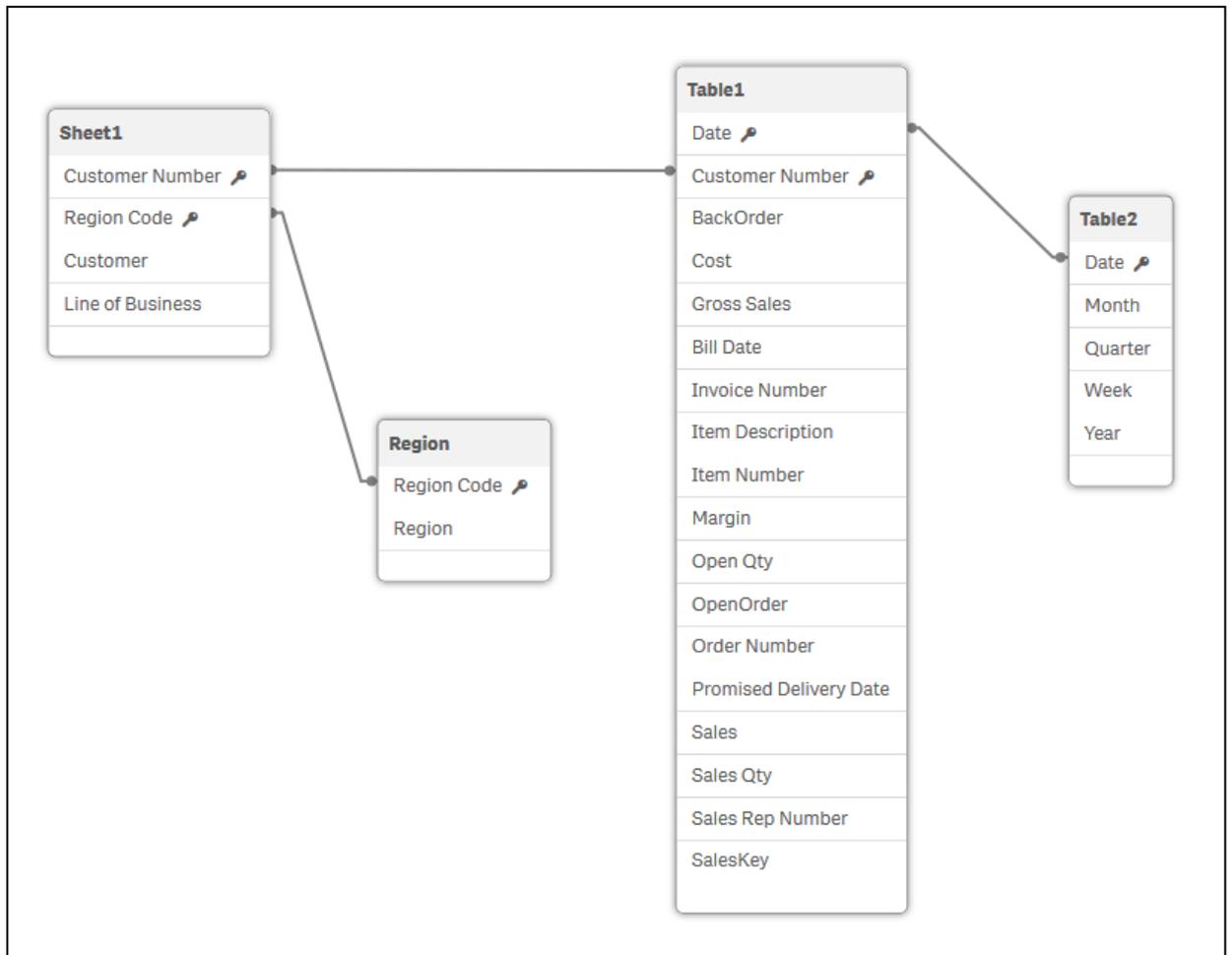
执行以下操作：

1. 打开数据加载编辑器。
2. 单击段 *Customers* 并删除 **LOAD** 语句中的以下行：

Region,

3. 单击**加载数据**。
4. 打开**数据模型查看器**。
已删除了合成键。

显示出合成键已被删除的数据模型查看器



4 在应用程序中使用数据

为完成本教程的学习，您需要将加载的数据放到应用程序的可视化中。

4.1 添加图表

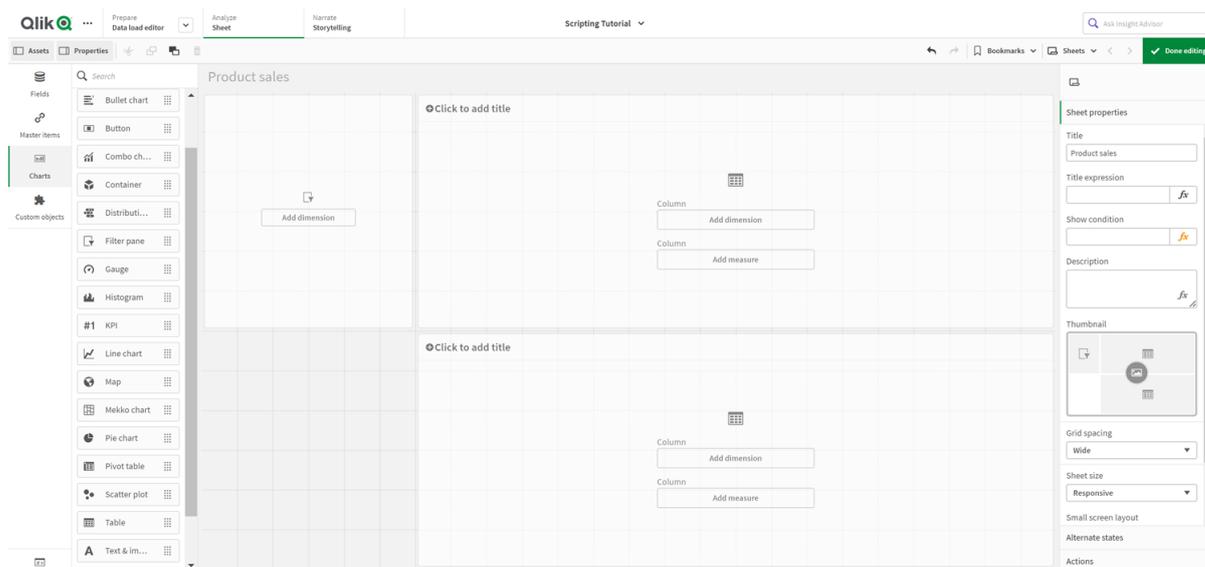
现在您的数据已经加载，您可以用这些数据创建一些图表。图表通常也称为可视化。添加必需的维度和度量后，图表即创建完成。首先添加图表，然后通过添加维度和度量继续操作。

执行以下操作：

1. 在您的 *Scripting Tutorial* 应用程序中新建工作表。
2. 将筛选器窗格从**图表**拖到表格，并使用手柄调整其大小，使其 3 格宽和 4 格高。将其放到表格的左上角。
3. 将条形图拖到右上角，使其 5 格高和 5 格宽，足以拉伸至表格侧边。
4. 将折线图拖动到剩余空间。

表格上的图标可以显示您添加的图表种类。现在可以在图表中添加维度和度量，以将其配置为可视化。

具有空图表的 *Qlik Sense* 工作表



4.2 添加维度和度量

下一步是添加维度和度量。首先将时间维度添加到左侧顶部的筛选器窗格。筛选器窗格的好处是节省空间。无需为 *Year*、*Quarter*、*Month* 和 *Week* 各自使用一个筛选器窗格，可以仅使用一个筛选器窗格来达到相同的目的。

创建和添加维度

执行以下操作：

1. 在左侧资产面板顶部，单击  以打开 **字段**。您可在此找到您在数据加载编辑器中加载的全部表格中的全部字段。
2. 向下滚动到列表底部，并单击 **Year** 字段。将其拖到左侧顶部筛选器窗格的中心。
3. 以相同的方式将 **Quarter**、**Month** 和 **Week** 添加到筛选器窗格中。
现已对四个维度创建筛选器窗格：**Year**、**Quarter**、**Month** 和 **Week**。

创建和添加度量

大多数可视化都需要维度和度量。度量是聚合表达式的结果，在许多情况下，即普通函数，例如 **Sum**、**Max**、**Min**、**Avg**(平均值) 或 **Count**。

在条形图中，您将按区域显示销售额。

执行以下操作：

1. 单击 **字段** .
2. 单击 **Region** 字段，并将其拖到条形图区域的中心。
3. 单击 **添加“Region”**。
4. 单击 **Sales** 字段，并将其拖到条形图区域的中心。
5. 单击 **添加为度量 > 总和(销售额)**。
6. 在右侧属性面板中，单击 **外观**，然后单击 **演示文稿**。选择 **水平**。
现在，这些条形显示为水平条形。
7. 在右侧属性面板中，单击 **排序**。
随即显示排序顺序。
8. 将 **Sum([Sales])** 拖到 **Region** 上，以便按 **Sum([Sales])**(度量值) 而非 **Region**(维度值，按字母顺序) 对维度排序。
此条形图创建完毕，将显示不同区域的销售额结果。此图是基本的条形图。属性面板(右侧)中有多个增强此图的选项。为展示其中一个可选项，我们使用不只是显示标题的标题区域。
9. 将以下内容粘贴到条形图的标题字段中：
`= 'Total sales: $' & Round(Sum(Sales)/1000000, 0.1) & 'M'`
10. 按回车键。

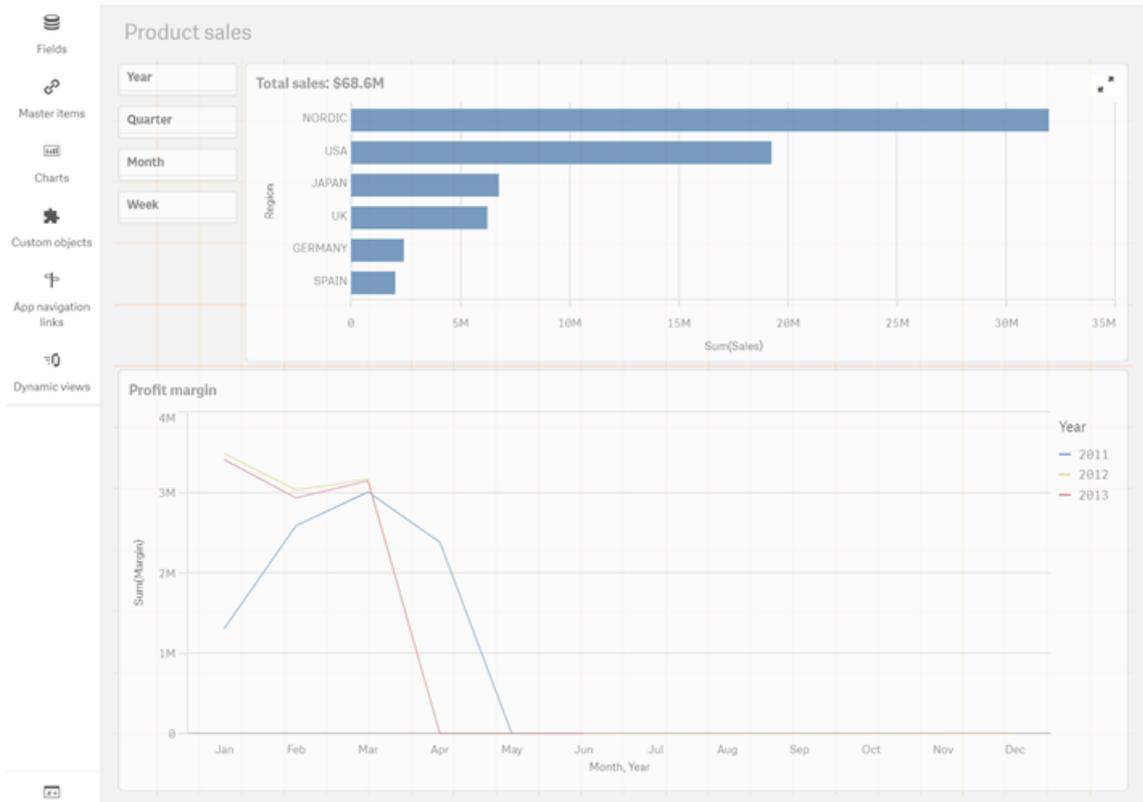
此表格上的最终可视化是折线图。

执行以下操作：

1. 单击 **字段** .
2. 单击 **Month** 字段，并将其拖到折线图区域的中心。
3. 单击 **添加“月”**。

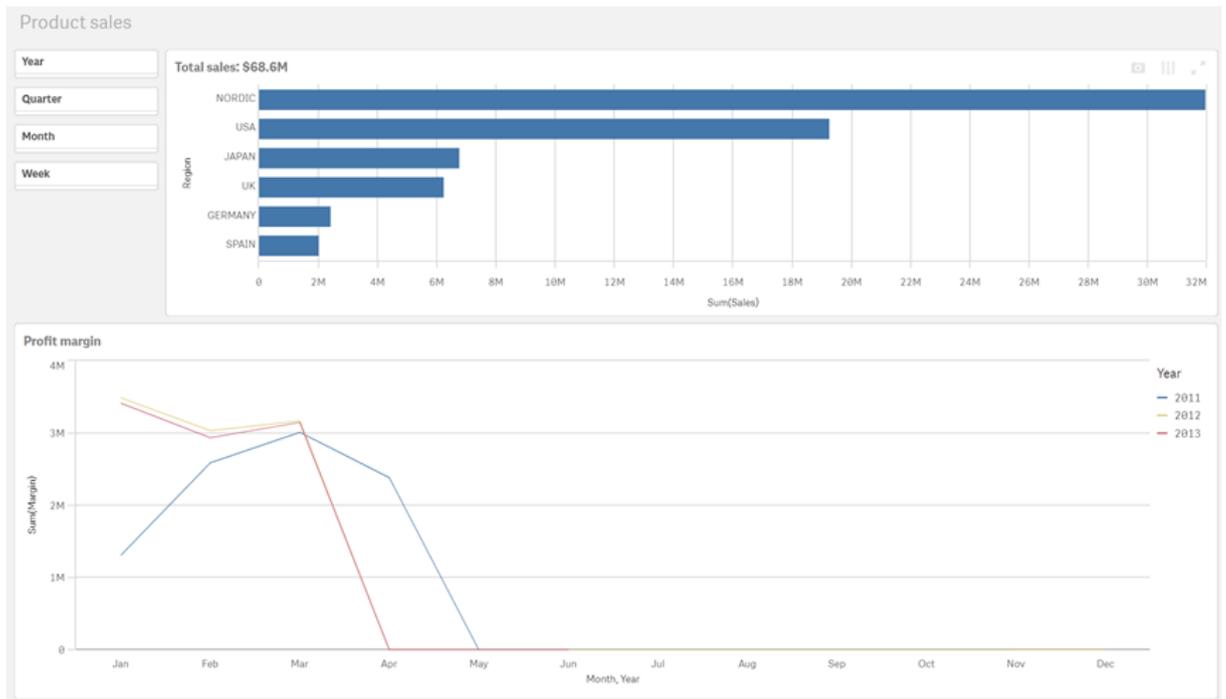
4. 单击 **Year** 字段, 并将其拖到折线图区域的中心。
5. 单击 **添加“年”**。
6. 单击 **Margin** 字段, 并将其拖到折线图区域的中心。
7. 单击 **添加为度量 > 总和(利润)**。
8. 在折线图顶部添加标题 **Profit margin**。

带数据的图表



9. 停止编辑工作表。
此表格现已完成, 您可以开始单击表格内容, 并与这些内容交互。

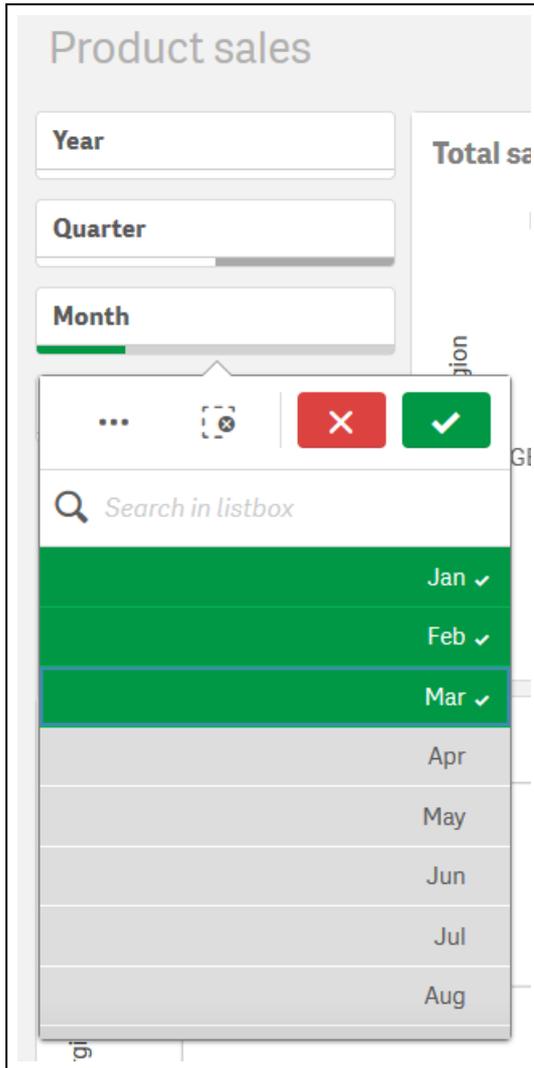
完成的工作表



由于我们在原来的 *Sales.xlsx* 文件中限制了每月销售数据的数量，因此在每年 3 月底之后，我们的图表几乎没有可用的数据。您可以在筛选器选窗格中进行选择，以便仅比较每年的前三个月。

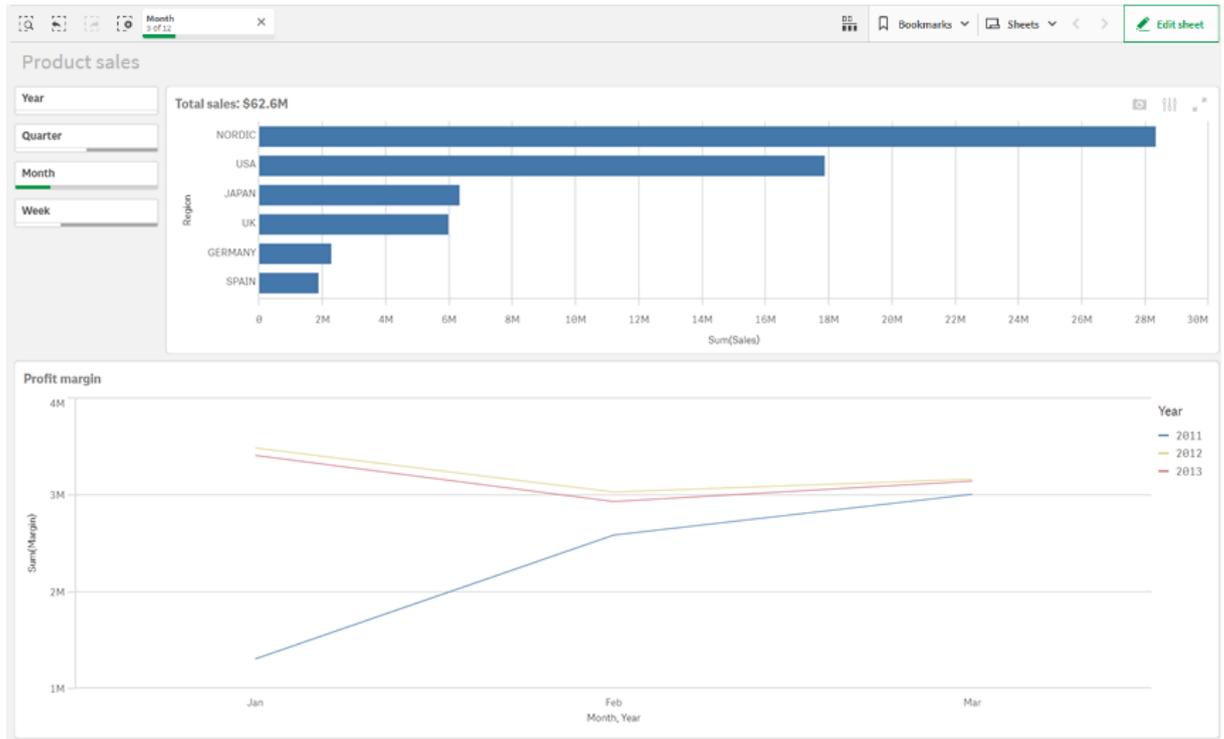
10. 单击筛选器框中的 *Month* 字段，然后选择 *Jan*、*Feb* 和 *Mar*。

筛选框



11. 关闭筛选器窗格。*Profit margin* 图表示出每年前三个月的数据。

Profit margin 图表根据选择更新。



4.3 谢谢!

现在, 您已完成本教程的学习, 希望您已获得有关 Qlik Sense 的脚本编译的一些基本知识。请访问我们的网站获取更多应用程序的灵感。