



# 数据加载和建模

---

Qlik® Sense

1.0.1

Copyright © 1993-2014 QlikTech International AB. 保留所有权利。



版权所有 © 1993-2014 QlikTech International AB。保留所有权利。

Qlik®、QlikTech®、Qlik® Sense、QlikView®、Sense™ 和 Qlik 徽标是在多个国家/地区注册的商标，或另供 QlikTech International AB 用作商标。本文引用的其他商标是其各自所有者的商标。

---

<b>1 关于本文档</b> .....	<b>7</b>
<b>2 快速数据加载</b> .....	<b>8</b>
2.1 故障排除 .....	8
<b>3 使用数据加载编辑器</b> .....	<b>9</b>
3.1 快速入门 .....	10
3.2 打开数据加载编辑器 .....	10
3.3 连接到数据源 .....	10
创建新的数据连接 .....	10
删除数据连接 .....	11
编辑数据连接 .....	11
插入连接字符串 .....	11
从数据连接选择数据 .....	12
引用脚本中的数据连接 .....	12
数据连接存储在什么地方? .....	13
ODBC 数据连接 .....	13
OLE DB 数据连接 .....	14
文件夹数据连接 .....	15
Web 文件数据连接 .....	16
从文件中加载数据 .....	17
从数据库中加载数据 .....	18
3.4 选择要加载的数据 .....	20
从数据库选择数据 .....	20
从带分隔符的表格文件选择数据 .....	22
从 Microsoft Excel 文件选择数据 .....	24
从固定记录表格文件选择数据 .....	25
从 QVD 或 QVX 文件选择数据 .....	27
从 HTML 文件选择数据 .....	28
从 XML 文件选择数据 .....	30
从 KML 文件选择数据 .....	30
3.5 编辑数据加载脚本 .....	31
组织脚本代码 .....	33
脚本中的注释 .....	34
在加载脚本中搜索 .....	35
保存加载脚本 .....	35
3.6 调试数据加载脚本 .....	36
调试工具栏 .....	36
输出 .....	36
变量 .....	37
中断点 .....	37
3.7 运行脚本以加载数据 .....	38
3.8 数据加载编辑器工具栏 .....	38
主工具栏 .....	38
编辑器工具栏 .....	39
<b>4 查看数据模型</b> .....	<b>40</b>

4.1 视图 .....	40
4.2 在数据模型查看器中移动表格和调整表格大小 .....	40
移动表格 .....	40
调整表格大小 .....	41
4.3 数据模型查看器中的表格和字段预览 .....	41
显示表格预览 .....	41
显示字段预览 .....	42
4.4 使用数据模型查看器创建主维度 .....	42
4.5 使用数据模型查看器创建主度量 .....	43
4.6 数据模型查看器工具栏 .....	43
<b>5 使用区域权限管理安全性 .....</b>	<b>45</b>
5.1 脚本中的区域 .....	45
区域权限系统字段 .....	45
5.2 动态数据缩减 .....	46
5.3 继承访问限制 .....	47
<b>6 使用 Direct Discovery 访问大数据集 .....</b>	<b>48</b>
6.1 Direct Discovery 数据与内存中数据之间的差别 .....	48
内存中模型 .....	48
Direct Discovery .....	49
内存中字段与 Direct Discovery 字段之间的性能差别 .....	51
内存中数据与数据库数据之间的差别 .....	52
缓存和 Direct Discovery .....	53
6.2 Direct Discovery 字段类型 .....	53
DIMENSION 字段 .....	54
MEASURE 字段 .....	54
DETAIL 字段 .....	54
6.3 Direct Discovery 支持的数据源 .....	54
SAP .....	54
Google Big Query .....	55
MySQL 和 Microsoft Access .....	55
6.4 使用 Direct Discovery 时的限制 .....	55
支持的数据类型 .....	55
安全性 .....	55
不支持 Qlik Sense 功能 .....	56
6.5 Direct Discovery 中的多表格支持 .....	56
6.6 记录 Direct Discovery 访问 .....	57
<b>7 数据建模简介 .....</b>	<b>59</b>
7.1 了解数据结构 .....	59
数据加载语句 .....	59
执行脚本 .....	59
字段 .....	60
逻辑表格 .....	61
表格名 .....	62
表格标签 .....	62

逻辑表格之间的关联 .....	63
合成键 .....	64
Qlik Sense 中的数据类型 .....	65
7.2 了解循环引用 .....	69
解决循环引用 .....	70
7.3 重新命名字段 .....	71
7.4 串联表格 .....	71
自动串联 .....	71
强制串联 .....	72
防止串联 .....	72
7.5 从之前加载的表格中加载数据 .....	72
Resident 或前面的 LOAD? .....	73
前置 LOAD .....	73
<b>8 数据建模的最佳实践 .....</b>	<b>75</b>
8.1 数据和字段的指导原则 .....	76
加载数据数量的指导原则 .....	76
数据表格和字段的上限 .....	76
加载脚本段的建议限制 .....	76
数字格式和时间格式的约定 .....	76
8.2 使用 QVD 文件 .....	79
QVD 文件的目的 .....	79
创建 QVD 文件 .....	80
从 QVD 文件读取数据 .....	80
8.3 使用 Join 和 Keep 合并表格 .....	81
SQL SELECT 语句中的联接 .....	81
Join .....	81
Keep .....	82
Inner .....	82
Left .....	83
Right .....	84
8.4 将映射用作联接的替代 .....	85
8.5 使用交叉表 .....	87
8.6 通用数据库 .....	88
8.7 离散数据的匹配间隔 .....	90
使用扩展的 intervalmatch 语法解决维度变化缓慢问题 .....	92
8.8 层次结构 .....	93
8.9 货币符号扩展 .....	94
使用变量进行货币符号扩展 .....	94
使用参数进行货币符号扩展 .....	95
使用表达式进行货币符号扩展 .....	96
8.10 在脚本中使用引号 .....	96
在 LOAD 语句内 .....	97
在 SELECT 语句中 .....	97
在 LOAD 语句外 .....	97

---

上下文外字段引用和表格引用 .....	97
名称和文字之间的差异 .....	97
数值和字符串文字之间的差异 .....	98
8.11 数据清理 .....	98
映射表 .....	98
使用映射表 .....	98
8.12 数据中的通配符 .....	99
星号 .....	99
OtherSymbol .....	100
8.13 NULL 值处理 .....	101
从 ODBC 关联/选择空值 .....	101
从文本文件创建空值 .....	101
表达式中 NULL 值的传送 .....	102

# 1 关于本文档

在创建 Qlik Sense 应用程序后，第一步是将数据加载到应用程序以便于研究和分析。本文档介绍如何使用数据加载编辑器创建数据加载脚本，以及如何在数据模型查看器中查看生成的数据模型。此外，您还将了解如何使用段访问功能来管理安全性，以及如何使用 **Direct Discovery** 访问大数据集。

请务必参阅 Qlik Sense 中的概念 指南，了解有关显示的不同主题的相关基本概念的信息。

此外，Qlik Sense 中还提供了数据建模介绍和最佳实践。



有关脚本函数和图表函数的详细参考，请参阅 **Qlik Sense 在线帮助**。

本文档派生自 Qlik Sense 的在线帮助。它非常适合于想要轻松阅读部分在线帮助或打印页面的用户，并且与在线帮助相比不包括任何其他信息。

请使用在线帮助或其他文档了解更多信息。

以下文档可用：

- Qlik Sense 中的概念
- 使用应用程序
- 创建可视化
- 发现和分析
- 用数据讲故事
- 脚本语法和图表函数指南

如需这些文档和更多信息，请访问 [help.qlik.com](http://help.qlik.com)。

## 2 快速数据加载



快速数据加载仅在 *Qlik Sense Desktop* 中可用。

通过在 菜单中单击**快速数据加载**，您可以在应用程序中快速添加其他数据文件，如 Microsoft Excel 电子表格或逗号定界的文件文件。

**另请参阅：**如何准备 *Excel* 文件以使用 *Qlik Sense* 加载(第 18 页)



还可以将数据文件拖放到 *Qlik Sense Desktop* 窗口中，以启动快速数据加载。

执行以下操作：

1. 选择如何加载数据文件。  
**添加数据** - 这将在您已加载的数据中添加其他数据文件。  
**替换数据** - 这将删除您先前在应用程序中加载的所有数据。



如果您选择**替换数据**，那么这可能影响您已创建的可视化。如果新数据不包含相同名称的字段，那么可视化可能需要更新。

2. 选择要加载的数据文件。  
选择文件后，随即将打开**选择数据源**对话框。
3. 选择要加载的表格和字段。
4. 在选择字段完成后，单击**加载数据**。  
随即将关闭**选择数据源**对话框，并且数据导入。数据加载进度对话框提供数据加载的结果。如果加载数据时出现警告，请参见以下“故障诊断”一节以了解更多信息。

如果应用程序不包含工作表，那么将创建新工作表，您可以开始编辑新工作表以添加数据可视化。

### 2.1 故障排除

如果已加载了多个文件，那么您可能在加载数据后收到一个警告。两个最常见的警告是：

合成键	如果两个表格包含多个公共字段，那么 <i>Qlik Sense</i> 将创建一个合成键以解决链接问题。
循环引用	如果已加载了两个以上的表格，那么共同字段可能导致数据结构中出现循环。

这两个问题都可以在数据加载编辑器中解决。

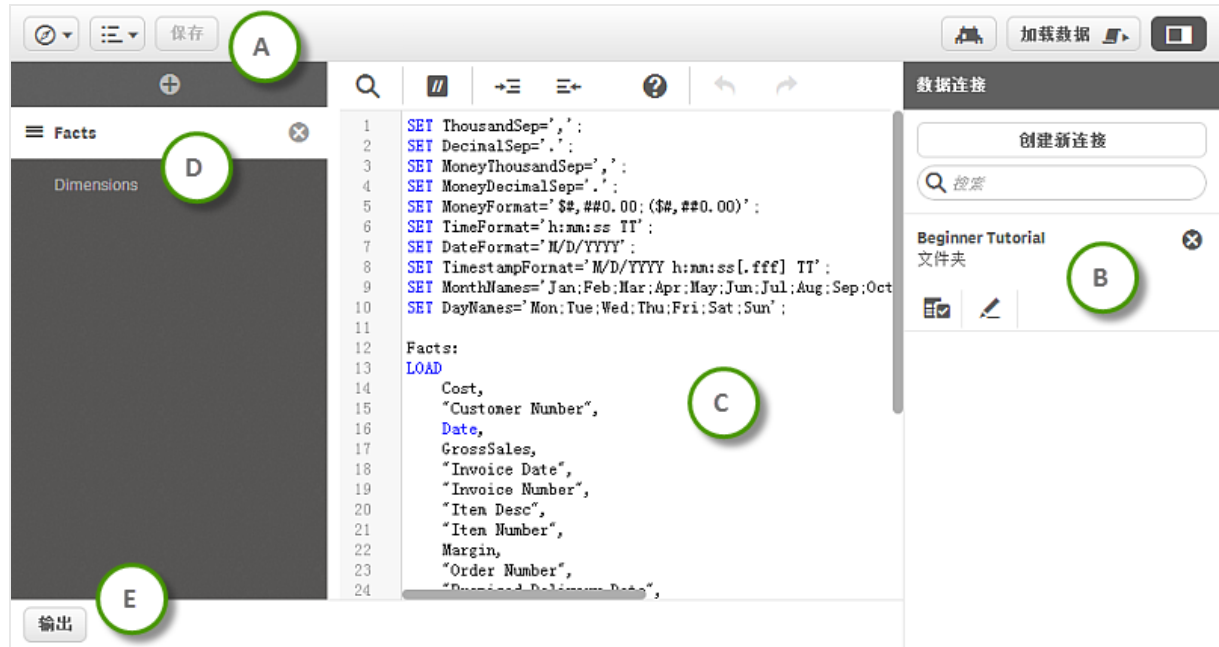


## 3 使用数据加载编辑器

本节介绍如何使用数据加载编辑器来创建或编辑可用于将数据模型加载到应用程序的数据加载脚本。

数据加载脚本将应用程序连接到数据源，并从数据源将数据加载到应用程序。在加载数据后，可用于对应用程序进行分析。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。


脚本可以手动键入，也可以自动生成。复杂的脚本语句必须，或至少部分由手动输入。



A	工具栏包含数据加载编辑器最常用的命令：导航菜单、全局菜单、保存、调试和加载数据。此外，该工具栏还显示应用程序的保存和数据加载状态。	数据加载编辑器工具栏 第 38 页
B	在数据连接下，可以将快捷方式保存到常用的数据源(数据库或远程文件)。这也是您开始选择要加载的数据的位置。	连接到数据源 第 10 页
C	可以在文本编辑器中编写和编辑脚本代码。每个脚本行均已编号，并且脚本已按照语法成分进行颜色编码。文本编辑器工具栏包含用于执行搜索并替换、帮助模式、撤消和重做的命令。原始脚本已经包含一部分通常不需要编辑的预定义区域变量设置，如 SET ThousandSep=。	编辑数据加载脚本 第 31 页
D	可以将脚本分成脚本段，使得更易于阅读和维护。脚本段自上而下执行。	组织脚本代码 第 33 页
E	输出用于显示在脚本期间所生成的所有消息。	

### 3.1 快速入门

如果您想要从数据库加载文件或表格，则需要先在**数据连接**中完成以下步骤：


1. 将**创建新连接**新连接到数据源 如果数据连接已不存在。
2.  从连接中选择数据。

使用**插入脚本**对话框完成选择后，可以选择**加载数据**将数据模型加载到应用程序。

### 3.2 打开数据加载编辑器

您可以从应用概述、工作表视图或数据模型查看器打开数据加载编辑器。

执行以下操作：

- 单击工具栏中的 ，然后选择**数据加载编辑器**。

随即打开数据加载编辑器。

### 3.3 连接到数据源

数据加载编辑器中的数据连接提供了一种用来将快捷方式保存到常用数据源 数据库、本地文件或远程文件 的方法。**数据连接**列出了已按字母顺序保存的连接。您可以使用搜索/筛选框通过特定名称或类型缩小连接列表的范围。

存在以下类型的连接：

- 标准连接器：
  - **ODBC** 数据库连接。
  - **OLE DB** 数据库连接。
  - **文件夹目录**连接用于定义本地文件或网络文件夹的路径。
  - **Web 文件**连接用于从位于 Web URL 的文件选择数据。
- 自定义连接器：

**Qlik Sense** 不直接支持为数据源自定义开发的连接器。自定义连接器是使用 **QVX SDK** 开发，或者由 **Qlik** 或第三方开发商所提供。在标准 **Qlik Sense** 安装中，无需安装任何可用的自定义连接器。



您只能查看您拥有的数据，或者您有权读取或更新的数据。如果需要，请联系您的 **Qlik Sense** 系统管理员以获取访问权。

### 创建新的数据连接

执行以下操作：

1. 单击**创建新连接**
2. 从下拉列表中选择要创建的数据源类型。

随即打开选定数据源类型的设置对话框。

3. 输入数据源设置，然后单击**保存**以创建数据连接。

数据连接现在已创建，并以您作为默认所有者。如果您希望其他用户能够使用服务器安装中的连接，那么您需要在 **Qlik Management Console** 中编辑连接的访问权限。




您的用户名和域将附加连接名称，以确保其唯一性。



如果未显示**创建新连接**，则意味着您没有添加数据连接的访问权限。如果需要，请联系您的 **Qlik Sense** 系统管理员以获取访问权。


### 删除数据连接

执行以下操作：

1. 单击要删除的数据连接的 。
2. 确认您要删除连接。


现在，已删除数据连接。



如果未显示 ，则意味着您没有删除数据连接的访问权限。如果需要，请联系您的 **Qlik Sense** 系统管理员以获取访问权。

### 编辑数据连接

执行以下操作：


1. 单击要编辑的数据连接的 。
2. 编辑数据连接的详细信息。连接详细信息特定于连接类型。
3. 单击**保存**。

现在，已更新数据连接。



如果编辑数据连接的名称，若希望继续引用相同连接，那么还需要在脚本中编辑对连接的所有现有引用 (`lib://`)。



如果未显示 ，则意味着您没有更新数据连接的访问权限。请联系您的 **Qlik Sense** 系统管理员。

### 插入连接字符串

**ODBC**、**OLE DB** 和自定义连接都需要连接字符串。

执行以下操作：

单击要插入连接字符串的连接图标。


已在数据加载编辑器的当前位置插入选定数据连接的连接字符串。



此外，还可以通过拖动数据连接，然后将其放置到脚本中要插入的位置插入连接字符串。

### 从数据连接选择数据

如果要选择在应用程序中加载的数据，则需要执行本节所述的以下步骤：

1. 将**创建新连接**新连接到数据源 如果数据连接已不存在。
2.  从连接中选择数据。

### 引用脚本中的数据连接

可以使用数据连接来引用语句中的数据源和脚本中的函数，通常，您希望在这两个位置中使用路径来引用文件名。

引用文件的语法是 `'lib://(connection_name)/(file_name_including_path)'`

#### 示例 1: 通过文件夹数据连接来加载文件

此示例从 MyData 数据连接中定义的位置加载文件 `orders.csv`。

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

#### 示例 2: 通过子文件夹来加载文件

此示例从 DataSource 数据连接文件夹加载文件 `Customers/cust.txt`。Customers 是 MyData 数据连接中定义的位置中的子文件夹。

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

#### 示例 3: 从 Web 文件加载

此示例从 PublicData Web 文件数据连接加载表格，其包含指向实际 URL 的链接。

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

#### 示例 4: 从数据库加载

此示例从 MyDataSource 数据连接文件夹加载表格 `Sales_data`。

```
LIB CONNECT TO 'MyDataSource';  
LOAD *;  
SQL SELECT * FROM `Sales_data`;
```

### 数据连接存储在什么地方？

可以使用 Qlik Sense Repository Service 存储连接。可以使用 Qlik Sense 服务器部署中的 Qlik Management Console 管理数据连接。Qlik Management Console 可让您删除数据连接，设置访问权限和执行其他系统管理任务。



在 Qlik Sense Desktop 中，所有连接都保存在应用程序中，且没有加密。这可能包括有关在创建连接时所输入的用户名、密码和文件路径的详细信息。这意味着如果您与其他用户共享应用程序，将能够以纯文本格式提供所有这些详细信息。在设计共享的应用程序时，您需要考虑到这种情况。

### ODBC 数据连接

您可以创建数据连接用于从在 Windows 控制面板的 **ODBC 数据源管理员** 对话框中创建和配置的 ODBC 数据源中选择数据。



如果在使用 ODBC 数据连接时，非 ANSI 编码数据文件出现问题，那么您可以尝试导入包含文件夹连接的数据文件，这支持更多用于处理字符代码的选项。

### 创建新的 ODBC 数据连接

执行以下操作：

1. 单击**创建新连接**，然后从下拉列表中选择 **ODBC**。  
随即打开**创建新连接 (ODBC)** 对话框。
2. 从可用的数据源列表用**用户 DSN** 或**系统 DSN** 选择要使用的数据源。  
可以根据 **32 位** 或 **64 位** 来过滤**系统 DSN** 连接。  
对于**用户 DSN** 源，您需要通过**使用 32 位连接**来指定是否使用 32 位驱动程序。
3. 添加**用户名**和**密码** 如果数据源需要。
4. 如果要使用与默认 DSN 名称不同的名称，请编辑**名称**。
5. 单击**保存**。

现在，已将该连接添加到**数据连接**，且可以连接到数据源和从已连接的数据源选择数据。

### 编辑 ODBC 数据连接

执行以下操作：

1. 单击要编辑的 **ODBC** 数据连接的 。  
随即打开**编辑连接 (ODBC)** 对话框。
2. 您可以编辑以下属性：  
从可用的数据源列表用**用户 DSN** 或**系统 DSN** 选择要使用的数据源。  
**用户名**  
**密码**

名称

3. 单击**保存**。

现在，已更新连接。



如果数据源已更新，则将不会自动更新所创建的连接设置。这意味着需要特别注意存储用户名和密码，尤其是在 **DSN** 中更改 **Windows** 集成安全性和数据库登录之间的设置。

### 使用 ODBC 数据连接移动应用程序

如果在 Qlik Sense 站点/Qlik Sense Desktop 安装之间移动应用程序，则包括数据连接。如果应用程序包含 ODBC 数据连接，则需要确保新部署中还存在相关 ODBC 数据源。ODBC 数据源的命名和配置需要一致，并且需要指向相同的数据库或文件。

### OLE DB 数据连接

您可以创建数据连接用于从 OLE DB 数据源选择数据。

#### 创建新的 OLE DB 数据连接

执行以下操作：

1. 单击**创建新连接**，然后从下拉列表中选择 **OLE DB**。
2. 从现有提供者的列表中选择**提供者**。
3. 输入要连接的**数据源**的名称。这可以是服务器名称，或在某些情况下可以是数据库文件的路径。这取决于所使用的 **OLE DB** 提供者。

**示例：**

如果选择了 Microsoft Office 12.0 Access Database Engine OLE DB Provider，请输入 Access 数据库文件的文件名，包括完整的文件路径：

`C:\Users\{user}\Documents\Qlik\Sense\Apps\Tutorial source files\Sales.accdb`



如果无法连接到数据源，将显示一条警告消息。

4. 选择要使用的凭据的类型 如果需要 。
  - **Windows 集成安全性**: 使用该选项，可以使用现有的 Windows 凭据。
  - **指定用户名和密码**: 使用该选项，需要输入用户名和密码。


如果数据源不需要凭证，请将**用户名**和**密码**留为空白。

5. 如果要测试连接，请单击 **LOAD**，然后单击**选择数据库...** 以用于建立数据连接。




当从数据连接中选择数据时，您仍然可以使用数据源的所有其他可用数据库。

6. 如果要使用与默认提供者名称不同的名称，请编辑**名称**。

 不得在连接名称中使用以下字符：\ / : \* ? " ' < > |


7. 单击**保存**。

 **保存**按钮只有在所输入的连接详细信息正确且自动连接测试成功时才可用。

现在，已将该连接添加到**数据连接**，且可以连接到数据源和从已连接 OLE DB 数据源选择数据。如果所输入的连接字符串正确。

### 编辑 OLE DB 数据连接

执行以下操作：

1. 单击要编辑的 **OLE DB** 数据连接的 。  
随即打开**编辑连接 (OLE DB)**对话框。
2. 您可以编辑以下属性：
  - **连接字符串** 其包含对**提供者**和**数据源**的引用
  - **用户名**
  - **密码**
  - **名称**
3. 单击**保存**。

现在，已更新连接。

### 故障排除

如果无法从数据连接中选择数据，请验证所设计的连接字符串是否正确，以及是否使用合适的凭据登录。

### 文件夹数据连接

您可以创建数据连接用于从文件夹（物理驱动器或共享网络驱动器）中的文件选择数据。

在 Qlik Sense 服务器安装上，文件夹需要从运行 Qlik Sense 引擎的系统访问。如果使用其他计算机或触摸设备连接到此系统，则无法连接到设备上的文件夹，除非可以从运行 Qlik Sense 引擎的系统访问该目录。

### 创建新的文件夹数据连接

执行以下操作：

1. 单击**创建新连接**，然后从下拉列表中选择**文件夹**。  
随即打开**创建新的数据连接(文件夹)**对话框。安装 Qlik Sense 后，会创建一个名为 `C:\Users\{user}\Documents\Qlik\Sense\Apps` 的工作目录。该目录是在对话框中选择的默认目录。



如果**文件夹**选项不可用，那么您没有添加文件夹连接的访问权限。请联系您的 Qlik Sense 系统管理员。

2. 输入指向包含数据文件的文件夹的**路径**。您可以执行以下操作之一：
  - 选择文件夹
  - 键入有效的本地路径 如：`C:\data\MyData\`
  - 键入 UNC 路径 如：`\\myserver\filedir\`。
3. 输入要创建的数据连接的**名称**
4. 单击**保存**。

现在，已将该连接添加到**数据连接**，且可以连接到数据源和从已连接的文件夹中的文件选择数据。

### 编辑文件夹数据连接

执行以下操作：

1. 单击要编辑的文件夹数据连接的 。  
随即打开**编辑连接 文件夹**对话框。



如果 已禁用，那么您没有编辑文件夹连接的访问权限。请联系您的 Qlik Sense 系统管理员。

2. 您可以编辑以下属性：
  - 路径**
  - 名称**
3. 单击**保存**。

现在，已更新连接。

### Web 文件数据连接

您可以创建数据连接用于从驻留在 Web 服务器、通过 URL 地址访问以及通常为 HTML 或 XML 格式的文件中选择数据。

#### 创建新的 Web 文件数据连接

执行以下操作：


1. 单击**创建新连接**，然后从下拉列表中选择 **Web 文件**。  
随即打开**创建新的数据连接 Web 文件**对话框。
2. 输入指向 Web 文件的 **URL**。
3. 输入要创建的数据连接的**名称**。
4. 单击**保存**。

现在，已将该连接添加到**数据连接**，并且您可以连接到 Web 文件和从 Web 文件选择数据。



### 编辑 Web 文件数据连接

执行以下操作：

1. 单击要编辑的 Web 文件数据连接的 。  
随即打开 **编辑连接 Web 文件** 对话框。
2. 您可以编辑以下属性：  
**URL**  
**名称**
3. 单击 **保存**。

现在，已更新连接。

### 从文件中加载数据

Qlik Sense 可以从各种格式的文件中读取数据：

- 文本文件，字段数据在其中以逗号、制表符或分号等分隔符分隔 以逗号分隔的变量 (CSV) 文件。
- dif 文件 Data Interchange Format
- fix 文件 固定记录长度
- HTML 表格
- Excel 文件
- xml 文件
- Qlik 本地 QVD 和 QVX 文件。

大多数情况下，表格的第一行表示字段名。

使用 **LOAD** 语句在脚本中加载文件。**LOAD** 语句可以包含一组完整的脚本表达式。


要从其他 Qlik Sense 应用程序读取数据，可以使用 **Binary** 语句。

**示例：**

```
directory c:\databases\common;  
LOAD * from TABLE1.CSV (ansi, txt, delimiter is ',', embedded labels);  
LOAD fieldx, fieldy from TABLE2.CSV (ansi, txt, delimiter is ',', embedded labels);
```

### 在数据加载编辑器中从文件加载数据

您可以使用 **选择数据** 对话框选择要加载的数据，而不需要在数据加载编辑器中手动键入语句。执行以下操作：

1. 打开数据加载编辑器。
2. 创建 **文件夹** 数据连接 如果还没有。数据连接应指向包含要加载的数据文件的目录。
3. 点击数据连接中的 ，以打开数据选择对话框。

现在可以从文件选择数据，以及插入加载数据所需的脚本代码。



还可以使用 ODBC 接口将 Microsoft Excel 文件加载为数据源。在此情况下，需要创建 ODBC 数据连接，而非文件夹数据连接。

### 如何准备 Excel 文件以使用 Qlik Sense 加载

如果要将 Microsoft Excel 文件加载到 Qlik Sense，那么有很多函数可供您用于变换和清理您的数据加载脚本中的数据，但是直接在 Microsoft Excel 电子表格文件中准备源数据会更加方便。本节提供了一些提示，旨在帮助您以最少必需脚本代码编写来准备要装入到 Qlik Sense 的电子表格。

#### 使用列标题

如果在 Excel 中使用列标题，并且在 Qlik Sense 中选择数据时选择了**嵌入的字段名称**，那么列标题将自动用作字段名称。还建议您避免在标签中使用换行符，并且将标题作为工作表的第一行。

#### 设置数据的格式

如果内容在表格中排列为原始数据，则可以更加轻松地将 Excel 文件加载到 Qlik Sense 中。最好避免以下情况：

- 聚合，如求和或计数。可以在 Qlik Sense 中定义和计算聚合。
- 重复标题。
- 不属于数据的多余信息，如注释。最佳方法是创建一个注释列，以便您在 Qlik Sense 中加载文件时可以轻松跳过。
- 交叉表数据布局。例如，如果您具有一个按月的列，那么您应具有一个名为“Month”的列，并且在 12 行中写入相同数据，每月占一行。然后，您可以在 Qlik Sense 中以交叉表格式查看。
- 中间标题，例如，某一行的内容是“Department A”且后跟与 A 部门相关的行，您应改为创建名为“Department”的列，并使用相应的部门名称来填充此列。
- 合并单元格。改为在每个单元格中列出单元格值。
- 空白单元格，其中的值由前一个值来暗示。您需要填充具有重复值的空白，以使每个单元格都包含数据值。

#### 使用命名区域

如果只需要读取工作表的一部分，那么您可以选择列和行的某个区域，并将其定义为 Excel 中的命名区域。Qlik Sense 可以从命名区域以及工作表中加载数据。

通常，您可以将原始数据定义为命名区域，并在命名区域外部保持所有额外的注释和图例。这将能够更加轻松地将数据加载到 Qlik Sense。

### 从数据库中加载数据


可以通过以下连接器将商用数据库系统中的数据加载到 Qlik Sense：

- 使用 Microsoft ODBC 接口或 OLE DB 的**标准连接器**。要使用 ODBC，必须安装驱动程序来支持 DBMS，并且必须在 Windows 控制面板的**ODBC 数据源管理器**中将数据库配置为 ODBC 数据源。
- **自定义连接器**，具体而言，经开发可将 DBMS 中的数据加载到 Qlik Sense。

### 从 ODBC 数据库中加载数据

从诸如 Microsoft Access 之类的数据库或可通过 ODBC 数据源访问的任何其他数据库开始加载数据的最简单方式是，使用数据加载编辑器中的数据选择对话框。

为此，请执行以下操作：

1. 需要为要访问的数据库提供 ODBC 数据源。此设置在 Windows 控制面板的 **ODBC 数据源管理器** 中配置。如果尚无任何数据库，则需要添加和配置一个，例如指向 Microsoft Access 数据库。
2. 打开数据加载编辑器。
3. 创建 **ODBC** 数据连接，并指向步骤 1 中提及的 ODBC 连接。
4. 点击数据连接中的 ，以打开数据选择对话框。

现在可以从数据库选择数据，以及插入加载数据所需的脚本代码。

### ODBC

要通过 ODBC 访问 DBMS DataBase Management System，必须已为相关 DBMS 安装 ODBC 驱动程序。替代方法是从数据库导出数据至 Qlik Sense 可读的文件。

通常需要为操作系统安装一些 ODBC 驱动程序。更多驱动程序可以从软件零售商购买，可以在互联网上找到，或者由 (DBMS) 制造商提供。一些驱动程序可以免费重新分配。

此处所述的 ODBC 接口是指客户端计算机上的接口。如果计划使用 ODBC 访问网络服务器上的多用户关系型数据库，可能需要安装更多允许客户端访问服务器上数据库的 DBMS 软件。联系 DBMS 供应商，了解所需软件的更多信息。

### 添加 ODBC 驱动程序

必须安装 ODBCDBMS 的 (DataBase Management System) 驱动程序，Qlik Sense 才能访问数据库。有关详细信息，请参阅所使用的 DBMS 的文档。

### 64 位和 32 位版本的 ODBC 配置

64 位版本的 Microsoft Windows 操作系统包含以下版本的 Microsoft Open DataBase Connectivity (ODBC)Data Source Administrator 工具 (*Odbcad32.exe*):

- 32 位版本的 *Odbcad32.exe* 文件位于 `%systemdrive%\Windows\SysWOW64` 文件夹下。
- 64 位版本的 *Odbcad32.exe* 文件位于 `%systemdrive%\Windows\System32` 文件夹下。

### 创建 ODBC 数据源

必须为要访问的数据库创建 ODBC 数据源。此操作可在 ODBC 安装期间或之后阶段完成。



开始创建数据源之前，必须先确定数据源是**用户 DSN**还是**系统 DSN**建议。只能使用正确的用户凭据访问用户数据源。在服务器安装中，通常需要创建系统数据源才能与其他用户共享数据源。

执行以下操作：

1. 打开 `Odbcad32.exe`。
2. 转到**系统 DSN** 标签，以创建系统数据源。
3. 点击**添加**。  
随即出现 **新建数据源** 对话框，并显示已安装的 ODBC 驱动程序列表。
4. 如果列出正确的 ODBC 驱动程序，请选择它，然后单击**完成**。  
随即出现特定于所选数据库驱动程序的对话框。
5. 命名数据源并设置必要的参数。
6. 单击**确定**。

### OLE DB

Qlik Sense 支持用于连接外部数据源的 OLE DB(Object Linking and Embedding, Database)接口。大量外部数据库可以通过 OLE DB 访问。

### 数据库中的逻辑

来自数据库应用程序的几个表格可以同时包括在 Qlik Sense 逻辑中。当某字段存在于多个表格中时，这些表格可通过此关键字段进行逻辑连接。

在选择一个值后，与此选择项相容的全部值都将作为可选项显示。而所有其他值作为排除项显示。


如果选择几个字段中的值，将假设逻辑与。

如果选择同一个字段的几个值，将假设逻辑或。


在某些情况下，字段内的选择项可以设置为逻辑与。

## 3.4 选择要加载的数据

您可以通过使用互动的**选择数据**对话框，选择要从文件或数据库表格以及数据源视图加载的字段。该对话框显示了不同的选择内容或转换选项，具体取决于用作数据源的文件或数据库的类型。并且，在选择字段时，还可以在该对话框中重新命名字段。在选择字段完成后，可以将生成的脚本代码插入脚本。

可以通过单击数据加载编辑器中数据连接的  打开**选择数据**。

### 从数据库选择数据

要从数据库选择数据，可以单击数据加载编辑器中  或 ODBC 数据连接的 **OLE DB** 开始选择。您可以在这里选择要从数据库表或数据源视图加载的字段。您可以在一次会话中从多个数据库、表格和视图中选择字段。

### 选择数据库

执行以下操作：

1. 从下拉列表中选择**数据库**。
2. 选择数据库的**所有者**。

**表格**列表已使用选定数据库中的可用视图和表格填充。

### 选择表格和视图

表格列表将包括数据库的表格、视图、同义词、系统表格和别名。

如果要选择表格的所有字段，请执行以下操作：

- 选中表格名称旁边的复选框。

如果要选择**表格**的特定字段，请执行以下操作：


- 单击表格名称 而非复选框 。
- 现有表格内容将会更新**字段**，并且可以通过单击字段继续执行。

包含所有选定字段的表格在复选框中都标有选中标记，而包含一部分选定字段的表格在复选框和右边选定字段数中都标有蓝色方块。

### 选择字段

**字段**列出了选定**表格**中的所有可用字段。可以通过在文本框中输入文件名的一部分来**筛选**列表中的字段。

可以在以下任一视图中显示字段：

- **数据预览**用于显示水平列布局中的数据和加载的预览数据。主键字段使用  表示。
- **元数据**用于显示垂直列表布局中的数据和字段元数据，如显示的主键和数据类型。

执行以下操作：

- 使用每个字段名称旁边的复选框选择要包括的字段。

在从选定表格选择要包括的字段后，可以继续从同一数据库的其他表格中选择字段，或选择另一个数据库，然后从该数据库中选择表格和字段。

**选择汇总**用于提供所选择的数据库的概述，以及选定表格和字段的数量。

### 重新命名字段

您可以重新命名字段。这在以下两种情况下特别有用：

- 如果加载包含同名字段的两个文件，则默认情况下这两个字段将会关联，并在 **Qlik Sense** 中被视为一个字段。如果加载它们作为单独的字段，请重新命名字段，以便使它们变成不同。
- 如果加载包含应理解为一个字段的字段的两个文件，但字段在各自的文件中拥有不同的名称，则可以重命名字段 在任一文件中 以使名称相同，从而作为一个字段进行加载。



在通过搜索筛选字段时，不能同时在数据选择向导中重命名字段。首先必须在文本框中删除搜索字段。



不能重命名同一表格中的两个字段，使它们具有相同的名称。

执行以下操作：

- 单击要重新命名的字段标题，输入新的名称，然后按 **Enter** 键。

该字段已重新命名，且脚本预览也已更新。



在 **as** 语句中，使用字段定义中的 **LOAD** 重新命名相应的字段。



如果重新命名表格中的字段，将会自动插入 **LOAD** 语句，而不考虑**包括 LOAD 语句** 设置。

### 预览脚本

插入的语句显示在脚本预览中，可以通过单击**预览脚本**选择进行隐藏。

### 包括 LOAD 语句

如果选择**包括 LOAD 语句**，可使用 **SELECT** 语句在 **LOAD** 语句前面生成 **SELECT** 语句作为输入。



如果重新命名表格中的字段，无论此设置怎样，都将会自动插入 **LOAD** 语句。

### 在脚本中插入

在选择字段完成后且想要在脚本中生成 **LOAD/SELECT** 语句时，请执行以下操作：


- 单击**插入脚本**。

数据选择窗口关闭，且已根据选择内容在脚本中插入 **LOAD/SELECT** 语句。

### 从带分隔符的表格文件选择数据

您可以从以下格式的带分隔符的表格文件中加载数据，例如：**CSV, TXT, TAB, PRN, MEM** 或 **SKV**，或带自定义分隔符号的文件。在选择字段后，可以插入用来将字段加载到脚本所需的脚本代码。

要开始选择数据，请执行以下操作：

1. 单击数据加载编辑器中文件夹数据连接的 。
2. 从文件夹的可用文件下拉列表中选择文件，然后单击**选择**。

随即显示选择对话框，以及使用预览数据更新的**字段**。

### 设置文件选项

可以使用从文件导出的设置来设置预览数据的格式，但可能根据您的目的调整文件选项：

**文件格式**      设置为**分隔**或**固定记录**。

在选择后，选择数据对话框将会根据所选的文件格式进行调整。

<b>字段名称</b>	设置以指定表格是包含 <b>嵌入的字段名称</b> 还是 <b>无字段名称</b> 。
<b>分隔符</b>	设置在表格文件中使用的 <b>分隔符</b> 字符。
<b>引用</b>	设置以指定处理引号的方式：  <b>无</b> = 不接受引号字符  <b>标准</b> = 标准引用 引号可以用作字段值的第一个和最后一个字符  <b>MSQ</b> = 新样式引用 允许字段包括多行内容 。
<b>标题大小</b>	将要省略的行数设置为表格标题。
<b>字符集</b>	设置在表格文件中使用的字符集。
<b>注释</b>	数据文件可以包含记录之间的注释，以开头为包含一个或多个特殊字符的行表示，如 //。  指定一个或多个字符来表示注释行。 <b>Qlik Sense</b> 无法加载以此处指定的字符为开头的行。
<b>忽略 EOF</b>	如果数据包含 <b>EOF</b> 字符作为字段值的一部分，可以选择 <b>忽略 EOF</b> 。

根据设置的选项来设置预览数据的格式。

### 选择字段

**字段** 列出了可供选择的所有字段。您可以执行下列操作之一：

- 使用每个字段名称旁边的复选框选择要包括的字段。
- 使用**选择所有字段**复选框可选择所有字段。

### 重新命名字段

您可以重新命名字段。这在以下两种情况下特别有用：

- 如果加载包含同名字段的两个文件，则默认情况下这两个字段将会关联，并在 **Qlik Sense** 中被视为一个字段。如果加载它们作为单独的字段，请重新命名字段，以便使它们变成不同。
- 如果加载包含应理解为一个字段的字段的两个文件，但字段在各自的文件中拥有不同的名称，则可以重命名字段 在任一文件中 以使名称相同，从而作为一个字段进行加载。

执行以下操作：

- 单击要重新命名的字段标题，输入新的名称，然后按 **Enter** 键。

该字段已重新命名，且脚本预览也已更新。



在 **LOAD** 语句中，使用字段定义中的 **as** 重新命名相应的字段。

### 预览脚本

插入的语句显示在脚本预览中，可以通过单击**预览脚本**选择进行隐藏。

### 插入脚本

在选择字段完成后且想要在脚本中生成 **LOAD/SELECT** 语句时，请执行以下操作：


- 单击**插入脚本**。

**选择数据源** 窗口关闭，且已根据选择内容在脚本中插入 **LOAD** 语句。

### 从 Microsoft Excel 文件选择数据

您可以从选定的 **Microsoft Excel** 文件加载数据。在选择字段后，可以插入用来将字段加载到脚本所需的脚本代码。

要开始选择数据，请执行以下操作：

1. 单击数据加载编辑器中**文件夹**数据连接器的 。
2. 从文件夹的可用文件下拉列表中选择文件，然后单击**选择**。

随即显示**选择数据自**对话框，并且**表格**列表已使用选定 **Microsoft Excel** 文件中的可用工作表和命名区域填充。



还可以使用 **ODBC** 接口将 **Microsoft Excel** 文件用作数据源。在这种情况下，需要使用 **ODBC** 数据连接 而非**文件夹**数据连接 。

### 选择表格

**表格**列表包含选定 **Microsoft Excel** 文件的所有工作表和命名区域。

如果要选择表格中的所有字段，请执行以下操作：

- 选中表格名称旁边的复选框。

如果要选择表格的特定字段，请执行以下操作：

- 单击表格名称 而非复选框 。
- 将会使用现有表格内容更新**字段**，并且可以选择所需的字段。

包含所有选定列的表格在复选框中都标有选中标记，而包含一部分选定字段的表格在复选框和表格名称右边选定字段数中都标有方块。

### 选择字段

**字段**列出了选定表格中的所有可用字段。您可能需要指定以下设置：

**字段名称：**

- **嵌入的字段名称**，如果字段名称 标题 存储在数据的第一行中。
- **无字段名称**，如果没有任何字段名称。

**标题大小：**



- 设置为在加载数据时要忽略的行数。

执行以下操作：

- 使用每个字段名称旁边的复选框选择要包括的字段。

在从选定表格选择要包括的字段后，可以继续从同一文件的其他表格中选择字段。

### 重新命名字段

您可以重新命名字段。这在以下两种情况下特别有用：

- 如果加载包含同名字段的两个文件，则默认情况下这两个字段将会关联，并在 **Qlik Sense** 中被视为一个字段。如果加载它们作为单独的字段，请重新命名字段，以便使它们变成不同。
- 如果加载包含应理解为一个字段的字段的两个文件，但字段在各自的文件中拥有不同的名称，则可以重命名字段 在任一文件中 以使名称相同，从而作为一个字段进行加载。

执行以下操作：

- 单击要重新命名的字段标题，输入新的名称，然后按 **Enter** 键。

该字段已重新命名，且脚本预览也已更新 如果已选择该字段 。



在 **LOAD** 语句中，使用字段定义中的 **as** 重新命名相应的字段。

### 预览脚本

插入的语句显示在脚本预览中，可以通过单击**预览脚本**选择进行隐藏。

### 插入脚本

在选择字段完成后且想要在脚本中生成 **LOAD/SELECT** 语句，请执行以下操作：

- 单击**插入脚本**。

**选择数据源**窗口关闭，且已根据选择内容在脚本中插入 **LOAD** 语句。


### 从固定记录表格文件选择数据

您可以从固定记录数据文件中选择数据，其中这些文件包含多列和固定字段大小的每条记录 数据行 ，通常用空格或制表符进行填充。

#### 示例：固定记录表格文件

Item	Id	Price
watch	001	2.75
Ball	002	3.25

要从固定记录文件选择数据，请执行以下操作：

1. 单击数据加载编辑器中**文件夹**数据连接的 .
2. 从文件目录的可用文件下拉列表中选择文件，然后单击**选择**。  
随即显示**选择数据源**对话框，以及使用预览数据更新的**字段**。
3. 将**文件格式**设置为**固定记录**。  
预览数据将适应为固定记录格式。

### 设置文件选项

可以使用从文件导出的设置来设置预览数据的格式，但可能需要根据您的目的调整文件选项：

<b>字段名称</b>	设置以指定表格是包含 <b>嵌入的字段名称</b> 还是 <b>无字段名称</b> 。
<b>标题大小</b>	将 <b>标题大小</b> 设置为要省略的行数作为表格标题。
<b>字符集</b>	设置在表格文件中使用的字符集。
<b>制表符大小</b>	设置一个制表符在表格文件中表示的空格数。
<b>记录行大小</b>	设置一条记录在表格文件中涉及的行数。默认值为 1。

根据设置的选项来设置预览数据的格式。

### 设置字段断开位置

您可以通过两种不同方式设置字段断开位置：

- 可以在**字段断开位置**中手动输入以逗号分隔的字段断开位置。每个位置都标志着字段的开始。

**示例：1,12,24**

- 启用**字段断开**能够以互动方式在字段数据预览中编辑字段断开位置。**字段断开位置**已使用选定位置更新。您可以：
  - 单击字段数据预览可插入字段断开。
  - 单击字段断开可将其删除。
  - 拖动字段断开可将其移动。

### 选择字段

**字段**列出了可供选择的所有字段。您可以执行下列操作之一：

- 使用每个字段名称旁边的复选框选择要包括的字段。
- **选择所有字段**

### 重新命名字段

您可以重新命名字段。这在以下两种情况下特别有用：

- 如果加载包含同名字段的两个文件，则默认情况下这两个字段将会关联，并在 **Qlik Sense** 中被视为一个字段。如果加载它们作为单独的字段，请重新命名字段，以便使它们变成不同。

- 如果加载包含应理解为一个字段的字段的两个文件，但字段在各自的文件中拥有不同的名称，则可以重命名字段 在任一文件中 以使名称相同，从而作为一个字段进行加载。

执行以下操作：

- 单击要重新命名的字段标题，输入新的名称，然后按 **Enter** 键。

该字段已重新命名，且脚本预览也已更新。



在 **LOADS** 语句中，使用字段定义中的 **as** 重新命名相应的字段。

### 预览脚本

插入的语句显示在脚本预览中，可以通过单击**预览脚本**选择进行隐藏。

### 插入脚本

在选择字段完成后且想要在脚本中生成 **LOAD/SELECT** 语句时，请执行以下操作：

- 单击**插入脚本**。

**选择数据源**窗口关闭，且已根据选择内容在脚本中插入 **LOAD** 语句。

## 从 QVD 或 QVX 文件选择数据

您可以从 **Qlik Sense** 创建的 **QVD** 文件和自定义连接器创建的 **QVX** 文件加载数据。在选择要加载的字段后，可以插入将字段加载到脚本所需的脚本代码。

要选择数据，请执行以下操作：

1. 单击数据加载编辑器中**文件夹**数据连接的
2. 从文件夹中可用文件的下拉列表中选择 **QVD** 或 **QVX** 文件，然后单击**选择**。

随即显示**选择数据源**对话框，以及使用预览数据更新的**字段**。

### 选择字段

**字段**列出了可供选择的所有字段。您可以执行以下操作之一：

- 通过选中每个字段名称旁边的复选框选择要包括的字段。
- 单击**选择所有字段**

### 重新命名字段

您可以重新命名字段。这在以下两种情况下特别有用：

- 如果加载包含同名字段的两个文件，则默认情况下这两个字段将会关联，并在 **Qlik Sense** 中被视为一个字段。如果加载它们作为单独的字段，请重新命名字段，以便使它们变成不同。
- 如果加载包含应理解为一个字段的字段的两个文件，但字段在各自的文件中拥有不同的名称，则可以重命名字段 在任一文件中 以使名称相同，从而作为一个字段进行加载。

执行以下操作：

- 单击要重新命名的字段标题，输入新的名称，然后按 **Enter** 键。

该字段已重新命名，且脚本预览也已更新。



在 **LOAD** 语句中，使用字段定义中的 **as** 重新命名相应的字段。

### 预览脚本

插入的语句显示在脚本预览中，可以通过单击**预览脚本**选择进行隐藏。

### 插入脚本


在选择字段完成后且想要在脚本中生成 **LOAD** 语句，请执行以下操作：

- 单击**插入脚本**。

**选择数据源**窗口关闭，且已根据选择内容在脚本中插入 **LOAD** 语句。

## 从 HTML 文件选择数据

您可以从 HTML 文件中的表格加载数据，即使用 **<TABLE>** 元素编码的表格。

单击数据加载编辑器中**文件夹**或**Web 文件**连接的  后，随即打开**选择数据源**对话框。您可以在这里选择要从 HTML 文件的表格加载的字段。您可以从多个表格中选择字段。

### 选择表格

**表格**列出了 HTML 文件中的所有表格。

如果要选择表格的所有字段，请执行以下操作：

- 选中表格名称旁边的复选框。

如果要选择表格的特定字段，请执行以下操作：

- 单击表格名称 而非复选框 。  
**字段**已使用现有表格内容更新，且可以继续选择字段。

包含所有选定字段的表格在复选框中都标有选中标记，而包含一部分选定字段的表格在复选框和右边选定字段数中都标有方块。

### 选择字段

**字段**列出了选定**表格**中的所有可用字段。可以通过在文本框中输入文件名的一部分来**筛选**列表中的字段。



在通过搜索筛选字段时，不能同时在数据选择向导中重命名字段。首先必须在文本框中删除搜索字段。



不能重命名同一表格中的两个字段，使它们具有相同的名称。

您可能需要指定以下设置：

### 字段名称：

- **嵌入的字段名称**，如果字段名称 字段标题 存储在数据的第一行中。
- **无字段名称**，如果没有任何字段名称。

### 字符集：

执行以下操作：

- 通过选中每个字段名称旁边的复选框选择要包括的字段。

在从选定表格选择要包括的字段后，可以继续从同一文件的其他表格中选择字段。

## 重新命名字段

您可以重新命名字段。这在以下两种情况下特别有用：

- 如果加载包含同名字段的两个文件，则默认情况下这两个字段将会关联，并在 **Qlik Sense** 中被视为一个字段。如果加载它们作为单独的字段，请重新命名字段，以便使它们变成不同。
- 如果加载包含应理解为一个字段的字段的两个文件，但字段在各自的文件中拥有不同的名称，则可以重命名字段 在任一文件中 以使名称相同，从而作为一个字段进行加载。



在通过搜索筛选字段时，不能同时在数据选择向导中重命名字段。首先必须在文本框中删除搜索字段。



不能重命名同一表格中的两个字段，使它们具有相同的名称。

执行以下操作：

- 单击要重新命名的字段标题，输入新的名称，然后按 **Enter** 键。

该字段已重新命名，且脚本预览也已更新。



在 **LOAD** 语句中，使用字段定义中的 **as** 重新命名相应的字段。

## 预览脚本

插入的语句显示在脚本预览中，可以通过单击**预览脚本**选择进行隐藏。

## 插入脚本

在选择字段完成后且想要在脚本中生成 **LOAD** 语句，请执行以下操作：


- 单击**插入脚本**。

**选择数据源**窗口关闭，且已根据选择内容在脚本中插入 **LOAD** 语句。

### 从 XML 文件选择数据

您可以加载以 XML 格式存储的数据。

执行以下操作：

1. 单击数据加载编辑器中**文件夹**或**Web 文件**连接的 。
2. 选择要从中加载数据的 XML 文件。

随即显示**选择数据源**对话框，以及使用预览数据更新的**字段**。您可以在这里选择要从文件的表格加载的字段。您可以从多个表格中选择字段。

### 选择表格和字段

**表格**列出了可供选择的所有表格。选择表格后，**字段**已使用可用字段更新。默认情况下，选择所有数据。

- 如果要排除某个表格或字段，选中该表格或字段名称旁边的复选框。

### 重新命名字段

您可以重新命名字段。这在以下两种情况下特别有用：

- 如果加载包含同名字段的两个文件，则默认情况下这两个字段将会关联，并在 **Qlik Sense** 中被视为一个字段。如果加载它们作为单独的字段，请重新命名字段，以便使它们变成不同。
- 如果加载包含应理解为一个字段的字段的两个文件，但字段在各自的文件中拥有不同的名称，则可以重命名字段 在任一文件中 以使名称相同，从而作为一个字段进行加载。

执行以下操作：

- 单击要重新命名的字段标题，输入新的名称，然后按 **Enter** 键。

该字段已重新命名，且在选择其他内容后也会更新脚本预览。



在 **LOAD** 语句中，使用字段定义中的 **as** 重新命名相应的字段。

### 预览脚本

插入的语句显示在脚本预览中，可以通过单击**预览脚本**选择进行隐藏。

### 插入脚本

在选择字段完成后且想要在脚本中生成 **LOAD** 语句，请执行以下操作：


- 单击**插入脚本**。

**选择数据源**窗口关闭，且已根据选择内容在脚本中插入 **LOAD** 语句。

### 从 KML 文件选择数据

您可以加载以 KML 格式存储的地理数据，以在创建地图可视化时使用。

执行以下操作：

1. 单击数据加载编辑器中**文件夹**或**Web 文件**连接的 。
2. 选择要从中加载数据的 KML 文件。

随即显示**选择数据源**对话框，以及使用预览数据更新的**字段**。您可以在这里选择要从文件的表格加载的字段。您可以从多个表格中选择字段。

### 选择表格和字段

**表格**列出了可供选择的所有表格。选择表格后，**字段**已使用可用字段更新。默认情况下，选择所有数据。

- 如果要排除某个表格或字段，选中该表格或字段名称旁边的复选框。

### 重新命名字段

您可以重新命名字段。这在以下两种情况下特别有用：

- 如果加载包含同名字段的两个文件，则默认情况下这两个字段将会关联，并在 **Qlik Sense** 中被视为一个字段。如果想要加载它们作为单独的字段，请使用不同的名称重新命名字段。
- 如果加载包含应理解为一个字段的字段的两个文件，但字段在各自的文件中拥有不同的名称，则可以重新命名字段以便使名称变成相同，从而加载作为一个字段。

执行以下操作：

- 单击要重命名的字段标题，输入新的名称，然后按 **Enter** 键。

该字段已重新命名，且在选择其他内容后也会更新脚本预览。



在 **Load** 语句中，使用字段定义中的 **as** 重新命名相应的字段。

### 预览脚本

插入的语句显示在脚本预览中，可以通过单击**预览脚本**选择进行隐藏。

### 插入脚本

在选择字段完成后且想要在脚本中生成 **LOAD** 语句，请执行以下操作：

- 单击**插入脚本**。

**选择数据源**窗口关闭，且已根据选择内容在脚本中插入 **LOAD** 语句。

## 3.5 编辑数据加载脚本

您可以在数据加载编辑器的文本编辑器中编写脚本。您可以在这里手动更改使用数据选择弹出窗口生成的 **LOAD/SELECT** 语句，并输入新的脚本。

脚本 必须使用 **Qlik Sense** 脚本语法编写 以颜色编码，以便于区分不同元素。注释高亮显示为绿色，而 **Qlik Sense** 语法关键字高亮显示为蓝色。每个脚本行都有编号。

可以使用编辑器中的许多现有功能帮助您开发加载脚本：

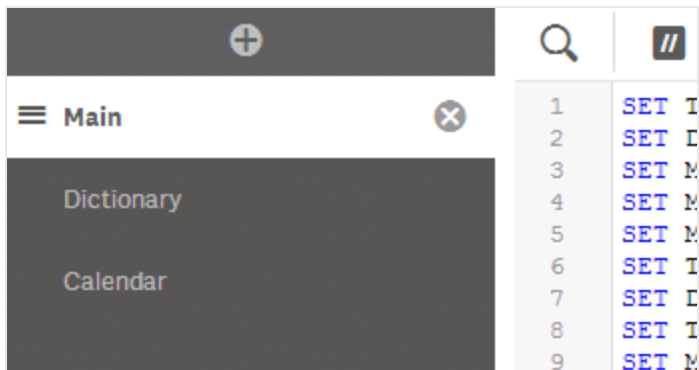
<p><b>详细的语法帮助</b></p>	<p>可以通过两种方法访问 Qlik Sense 语法关键字的详细的语法帮助。</p> <ul style="list-style-type: none"> <li>• 点击工具栏中的  可进入语法帮助模式。在语法帮助模式下，可以单击语法关键字 标为蓝色且带有下划线 以访问语法帮助。</li> </ul> <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;">  无法在语法帮助模式下编辑脚本。         </div> <ul style="list-style-type: none"> <li>• 将光标放置在关键字内或结尾，然后按 <b>Ctrl+H</b> 键。</li> </ul>
<p><b>自动完成</b></p>	<p>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。当继续输入时，该列表的范围会缩小。例如，输入 <i>month</i>。</p> <p>执行以下操作：</p> <ul style="list-style-type: none"> <li>• 在列表中选择一个关键字，方法是单击此关键字或者按 <b>Enter</b> 键。</li> </ul>
<p><b>工具提示</b></p>	<p>在 Qlik Sense 脚本函数后面输入左括号后，工具提示会显示该函数的语法，包括参数、返回值类型和其他说明。</p>
<p><b>准备的测试脚本</b></p>	<p>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。</p> <p>执行以下操作：</p> <ul style="list-style-type: none"> <li>• 按 <b>Ctrl+00</b> 键</li> </ul> <p>已将测试脚本代码插入脚本。</p>
<p><b>缩进代码</b></p>	<p>您可以缩进代码以增加可读性。</p> <p>执行以下操作：</p> <ol style="list-style-type: none"> <li>1. 选择要更改缩进的一行或多行。</li> <li>2. 单击  可缩进文本 增加缩进 ，单击  可减少缩进文本 减少缩进 。</li> </ol> <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;">  还可以使用键盘快捷键：  <b>Tab</b> 缩进  <b>Shift+Tab</b> 减少缩进         </div>
<p><b>搜索和替换</b></p>	<p>您可以在整个脚本段中搜索和替换文本。</p> <p>点击工具栏中的  可打开<b>搜索和替换</b>面板。</p>
<p><b>选择所有代码</b></p>	<p>您可以选择当前脚本段中的所有代码。</p>



	<p>执行以下操作：</p> <ul style="list-style-type: none"><li>按 <b>Ctrl+A</b></li></ul> <p>已选择当前脚本段中的所有脚本代码。</p>
--	---

### 组织脚本代码

您可以将脚本分成段来组织结构。按照脚本段自上而下的顺序执行脚本。脚本至少必须包含一个脚本段。



### 创建新的脚本段

您可以插入新的脚本段来组织代码。

执行以下操作：

- 单击 **+**。

将新分段插入当前选定分段的后面。

### 删除脚本段

您可以删除脚本段，包括脚本段中的所有代码。

 删除脚本段后无法恢复。
---

执行以下操作：

- 单击分段选项卡旁边的 **✕** 可将其删除。您需要确认删除。  
现在，已删除分段。

### 重新命名脚本段

您可以重新命名脚本段。

执行以下操作：


1. 单击分段的名称和类型来编辑名称。
2. 编辑完成后，按 **Enter** 键或单击分段以外的区域。

现在，已重新命名分段。

### 重新排列脚本段

您可以重新排列脚本段的顺序来更改脚本的执行顺序。

执行以下操作：

1. 选择要移动的脚本段。
2. 将光标放置在  拖拽条上，然后拖动脚本段重新排列顺序。

已重新排列脚本段。


### 脚本中的注释

您可以在脚本代码中插入注释和备注，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。可以将该功能用作一个切换键，如果没有对选定代码进行注释，则该功能将会对其进行注释，反之亦然。

#### 注释


执行以下操作：

1. 选择一行或多行没有注释的代码，或者将光标放置在一行的开头。
2. 单击 ，或按 **Ctrl+K** 键。

现在，已对选定代码进行注释。

#### 取消注释

执行以下操作：

1. 选择一行或多行已经注释的代码，或者将光标放置在注释行的开头。
2. 单击 ，或按 **Ctrl+K** 键。

现在，已取消选定代码的注释。



可以通过多种方法在脚本代码中插入注释：

- 使用 **Rem** 语句
- 使用 **/\*** 和 **\*/** 将代码段括起来。

#### 示例：

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

```
/* This is a comment  
   that spans two lines */
```




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

### 在加载脚本中搜索

您可以在整个脚本段中搜索和替换文本。

#### 搜索

打开数据加载编辑器。执行以下操作：

1. 点击工具栏中的 。  
随即显示搜索下拉对话框。
2. 在搜索框中，输入您要查找的文本。需要至少输入两个字符。  
搜索结果会在脚本代码的当前段中突出显示。另外，将会在分段标签旁边指明找到的文本实例的数量。
3. 可以通过单击  和  浏览结果。




点击工具栏中的  可关闭搜索对话框。



此外，选择**搜索分段中所有匹配项**可在所有脚本段中进行搜索。将会在每个分段标签旁边指明找到的文本实例的数量。搜索函数不区分大小写。

#### 替换

执行以下操作：

1. 点击工具栏中的 。  
随即显示搜索下拉对话框。
2. 在“搜索”文本框中输入要查找的文本。
3. 在“替换”文本框中输入替换文本，然后单击**替换**。
4. 单击  以查找搜索文本的下一个实例，然后执行下列操作之一：
  - 单击**替换**以替换文本。
  - 单击  以查找下一个实例。

点击工具栏中的  可关闭搜索对话框。




也可以单击**替换分段中所有匹配项**以替换当前脚本段中搜索文本的所有实例。替换功能区分大小写，并且替换的文本与替换字段中给出的大小写相同。随即显示一条消息，包含有关替换的实例数量的信息。


### 保存加载脚本

在保存脚本后会保存整个应用程序，但数据不会自动重新加载。

执行以下操作：

- 单击数据加载编辑器工具栏中的**保存**以保存脚本。

现在已保存该脚本，但应用程序仍然包含之前重新加载的旧数据，这在工具栏中有表示。如果要更新包含新数据的应用程序，单击脚本编辑器工具栏中的**加载数据** 。

当保存脚本后，系统会自动检查语法错误。语法错误在代码中高亮显示，并且使用脚本段旁边的  表示包含语法错误的所有脚本段。




当加载数据时，脚本将自动保存到应用程序。

## 3.6 调试数据加载脚本

您可以使用数据加载编辑器中的调试实用程序通过使用中断点单步调试脚本的执行，这样能够让您检查脚本执行的变量值和输出。您可以指定是查看任何还是所有**输出**、**变量**和**中断点**。

要显示调试面板，请执行以下操作：





- 在数据加载编辑器工具栏中单击 。
- 随即在数据加载编辑器的底部打开调试面板。




在调试模式下运行的同时 即从开始调试执行直到脚本执行或执行结束，都无法创建连接、编辑连接、选择数据、保存脚本或加载数据


## 调试工具栏

数据加载编辑器调试面板包含一个工具栏，该工具栏包含用于控制调试执行的下列选项：

<p><b>限制加载</b></p>	<p>启用该选项可限制要从每个数据源加载的数据的行数。如果数据源很大，该选项可用于减少执行时间。</p> <p>输入要加载的行数。</p> <div data-bbox="446 1444 1385 1585" style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> 该选项仅适用于物理数据源。例如，自动生成和不限内联加载。</p> </div>
<p></p>	<p>在调试模式下开始或继续执行，直到下一个中断点。</p>
<p></p>	<p>步进到代码的下一行。</p>
<p></p>	<p>在此结束执行。如果在执行所有代码之前结束，则生成的数据模型仅包含截止执行结束时所在的代码行的数据。</p>

## 输出

**输出**用于显示在调试执行期间所生成的所有消息。您可以在显示所有消息时通过单击  选择锁定滚动输出。

此外，输出菜单 () 包含下列选项：



清除	单击该选项可删除所有输出消息。
选择所有文本	单击该选项可选择所有输出消息。
滚动到底部	单击该选项可滚动到最后一条输出消息。

## 变量


变量列出了在脚本中定义的所有保留变量、系统变量和变量，并在脚本执行期间显示当前值。

### 将变量设置为常用变量

如果要在执行期间检查特定变量，可以将它们设置为常用变量。常用变量显示在变量列表的顶部，标有黄色星形。要将变量设置为常用变量，请执行以下操作：

- 单击变量旁边的 。
- 现在， 显示为黄色，然后将变量移动到变量列表的顶部。

### 筛选变量

可以应用筛选器使用变量菜单 () 中的以下选项只显示选定类型变量：


显示所有变量	单击该选项可显示所有类型变量。
显示系统变量	单击该选项可显示系统变量。 系统变量由 <b>Qlik Sense</b> 定义，但可以在脚本中更改变量值。
显示保留变量	单击该选项可显示保留变量。 保留变量由 <b>Qlik Sense</b> 定义且无法更改其值。
显示用户定义的变量	单击该选项可显示用户定义的变量。 用户定义的变量是您在脚本中定义的变量。

## 中断点

您可以将中断点添加到脚本以停止调试执行某些代码行，并在此时检查变量值和输出消息。当到达中断点后，可以选择停止执行，继续执行直到下一个中断点或步进到下一行代码。已在脚本中列出了所有中断点，引用代码段和行号。

### 添加中断点

要在代码行中添加中断点，请执行下列操作之一：



- 在脚本中，直接单击要添加中断点的行号右边的区域。  
行号旁边的  表示在此代码行中有中断点。




您甚至可以在关闭调试面板后添加中断点。


### 删除中断点


您可以通过执行下列操作之一删除中断点：

- 在脚本中，单击行号旁边的 。
- 在中断点列表中，单击中断点旁边的 .

还可以单击 ，然后选择**全部删除**从脚本中删除所有中断点。


### 启用和禁用中断点

创建中断点时，中断点默认处于启用状态，在中断点列表中通过中断点旁边的  表示。可以通过在中断点列表中选择和取消选择个别中断点以启用和禁用中断点。

还可以使用中断点菜单 () 中的以下选项：

- 全部启用
- 全部禁用

## 3.7 运行脚本以加载数据


点击工具栏中的  运行脚本，并在应用程序中重新加载数据。在加载数据之前，应用程序自动保存。

随即显示**数据加载进度**对话框，并且可以**中止**加载。数据加载完成后，对话框状态已更新 **已成功完成**或**数据加载失败**，且可能出现错误和警告 如针对合成键 的概要。如果要在对话框关闭后查看，概要还会显示在**输出**中。



如果希望**数据加载进度**对话框始终在成功执行后自动关闭，请选择**成功完成后关闭**。


## 3.8 数据加载编辑器工具栏


工具栏可让您在数据加载脚本中执行全局操作，例如撤消/重做、调试和搜索/替换。还可以单击**加载数据**  以在您的应用程序中重新加载数据。


### 主工具栏




导航菜单包含以下选项：

 **应用概述**

 **数据模型查看器**

 **打开应用中心**



可以通过单击  在新选项卡中打开 **应用概述** 和 **数据模型查看器**。



菜单包含以下选项：


**快速数据加载** - 将其他数据文件加载到应用程序中- 仅在 Qlik Sense Desktop 中可用。

 **帮助**

 **关于**



调试脚本

**加载数据** 

执行脚本并重新加载数据。在重新加载数据之前，应用程序自动保存。



切换 **数据连接** 视图。

### 编辑器工具栏



在脚本中搜索和替换文本。



注释/取消注释



缩进



减少缩进



激活语法帮助模式。在帮助模式下，可以在编辑器中单击语法关键字 标为蓝色 访问详细的语法帮助。



无法在帮助模式下编辑脚本。



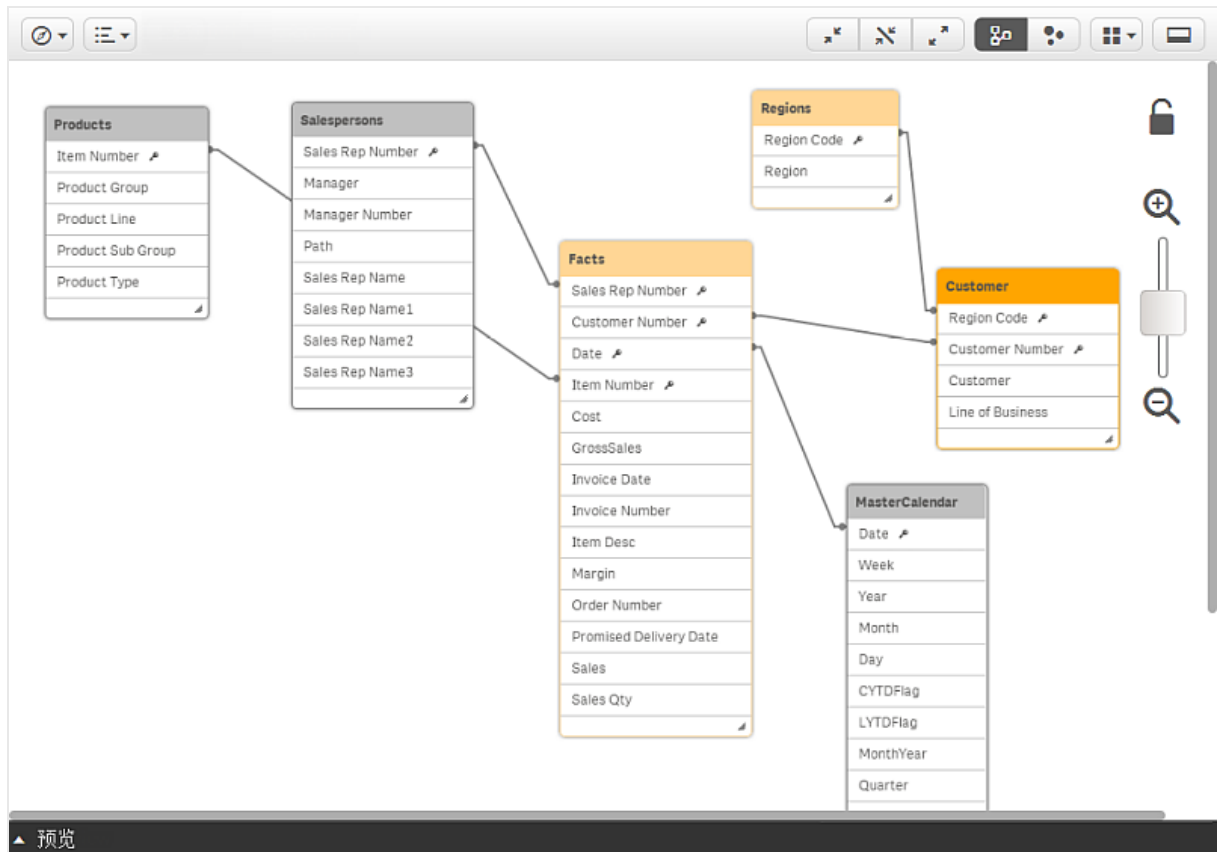
撤消上一次更改 可以撤消多个步骤 。这等同于按 **Ctrl+Z** 键。



重做最新的 **撤消**。这等同于按 **Ctrl+Y** 键。

## 4 查看数据模型

数据模型查看器可为您提供应用程序数据结构的概述。可以预览数据模型查看器中的表格和字段的数据。还可以动态创建维度和度量。



在数据模型查看器中，用方框表示每个数据表格，将表格名称用作标题并包含表格中列出的所有字段。用线表示表格关联，并用虚线表示循环引用。当选择一个表格或字段时，突出显示的关联立即能够让您明白字段和表格的关联方式。

您可以更改缩放级别，只需点击 、 或使用滑块即可。

### 4.1 视图

可以选择查看：



- 内部表视图 - Qlik Sense 数据模型，包括组合字段
- 源表视图 - 源数据表格的数据模型

### 4.2 在数据模型查看器中移动表格和调整表格大小




#### 移动表格

可以移动表格，只需在背景上拖动它们即可。当保存应用程序时，将保存表格位置。



您可以锁定表格位置，只需点击背景左侧的  即可。要对表格位置解锁，请点击 。




还可以使用工具栏中  下的选项自动排列布局：

-  **网格布局** - 用于在网格中排列表格。
-  **自动布局** - 用于排列表格以适应窗口大小。
-  **还原布局** - 用于恢复为最近保存的布局状态。

### 调整表格大小

可以使用表格右下角的红色箭头调整表格的显示大小。当保存应用程序时，不会保存显示大小。

还可以使用工具栏中的自动设置显示大小选项：


-  **全部折叠** - 用于最小化所有表格，仅显示表格名称。
-  **显示链接字段** - 用于缩减所有表格的大小，以显示表格名称以及与其他表格关联的所有字段。
-  **全部展开** - 用于最大化所有表格，以显示表格中的所有字段。

## 4.3 数据模型查看器中的表格和字段预览


在数据模型查看器中，可以在屏幕底部的面板中预览所有数据表格。在预览中，可以快速检查表格或字段的内容。如果选择字段，还可以快速添加维度和度量。

表格预览将显示前 10 个表格字段及其值，字段预览将显示个别字段及其前 10 个值。此外，预览面板中还显示元数据，例如信息密度和标签。

可以使用两种方式显示和隐藏预览面板：

- 在数据模型查看器工具栏中点击 
- 点击**预览**标题



*Direct Discovery* 数据不会在预览  中显示。

### 显示表格预览

执行以下操作：

- 在数据模型查看器中点击表格标题。

随即显示预览面板，其中包含所选表格的字段和值。

Item master		数据预览				
行	827	Item Number	Product Group	Product Line	Product Sub Group	Product Type
字段	5	10001	Beverages	Drink	Juice	Pure Juice Beverages
关键项	1	10002	Beverages	Drink	Flavored Drinks	Drinks
标签:	\$key \$numeric \$integer \$ascii \$text	10003	Beverages	Drink	Flavored Drinks	Drinks
		10004	Beverages	Drink	Soda	Carbonated Beverages
		10005	Beverages	Drink	Soda	Carbonated Beverages
		10006	Beverages	Drink	Soda	Carbonated Beverages

## 显示字段预览

执行以下操作：

- 在数据模型查看器中点击表格字段。

随即显示预览面板，其中包含所选字段及其值。

Product Group		数据预览			
密度	100%	Item Number	Product Group	Product Line	Product Sub Group
子集比率	100%	10001	Beverages	Drink	Juice
有重复项	真	10002	Beverages	Drink	Flavored Drinks
非重复值总和	15	10003	Beverages	Drink	Flavored Drinks
当前的非重复值	15	10004	Beverages	Drink	Soda
非空值	827	10005	Beverages	Drink	Soda
标签:	\$ascii \$text	10006	Beverages	Drink	Soda

## 4.4 使用数据模型查看器创建主维度

在使用未发布的应用程序时，可以创建主维度，以便可以重复使用它们。发布应用程序的用户将有权访问其库中的主维度，但不能修改它们。数据模型查看器在已发布应用程序中不可用。

执行以下操作：


- 在数据模型查看器中，选择一个字段，并打开**预览**面板。
- 单击**添加为维度**。  
随即打开**新建维度**对话框，其中包含选定字段。选定字段的名称也用作维度的默认名称。
- 根据需要更改此名称，并选择性添加说明和标记。
- 单击**添加维度**。
- 单击**完成**以关闭对话框。

此维度现已保存到资源面板的主项目标签中。



您可以快速添加多个维度作为主条目，只需在添加每个维度后单击**添加维度**。在完成操作时，单击**完成**。



*Direct Discovery* 表格通过数据模型查看器中的  表示。

### 4.5 使用数据模型查看器创建主度量

在使用未发布的应用程序时，可以创建主度量，以便可以重复使用它们。发布应用程序的用户将有权访问其库中的主度量，但不能修改它们。

执行以下操作：

1. 在数据模型查看器中，选择一个字段，并打开**预览**面板。
2. 单击**添加为度量**。  
随即打开**创建新度量项**对话框，其中包含选定字段。选定字段的名称也用作度量的默认名称。
3. 为度量输入表达式。
4. 根据需要更改此名称，并选择性添加说明和标记。
5. 单击**创建**。


此度量现已保存到资源面板的主项目标签中。


### 4.6 数据模型查看器工具栏


在数据模型查看器中，可以在屏幕顶部的工具栏中找到以下工具：




导航菜单包含以下选项：

 应用概述

 数据加载编辑器


 打开应用中心




可以通过单击 ，在新标签中打开**应用概述**和**数据加载编辑器**。



菜单包含以下选项：

 帮助









 关于



内部表视图



源表视图

-  全部折叠
-  显示链接字段
-  全部展开
-  包含以下选项的布局菜单：
  -  网格布局
  -  自动布局
  -  还原布局
-  切换预览

## 5 使用区域权限管理安全性

可以在数据加载脚本中使用区域访问来处理安全性。这样，可以使用单个文件来保存多个用户或用户组的数据。**Qlik Sense** 将为身份验证和授权以及动态减少数据使用区域权限中的信息，以使用户仅可查看自己的数据。

安全性由文件自己建立在其内部，这意味着下载的文件也在一定程度上受到保护。但是，如果安全性要求高，那么应避免文件下载和脱机使用，并且文件仅应由 **Qlik Sense** 服务器发布。由于所有数据都保存在一个文件中，因此这个文件的大小可能会非常大。

### 5.1 脚本中的区域

通过一个或数个安全表来管理访问控制，其加载方式与 **Qlik Sense** 常规加载数据相同。也可将这些表格存储在常规数据库中。在存取区域内给定了管理安全性表格的脚本语句，它在脚本内由**区域权限**发起。

如果存取区域被定义在脚本内，则加载应用程序数据的脚本部分应被放置于其他区域，这是由**区域应用程序**发起的。

示例：

```
Section Access;
LOAD * inline
[ACCESS,USERID
ADMIN, A
USER,U ];
Section Application;
LOAD... .. from... ..
```

### 区域权限系统字段

访问级别被指派给一个或数个表格中的用户，并加载在区域权限内。这些表格可以包含数个不同的用户指定系统字段，通常包括 **USERID**，该字段定义存取级别，**ACCESS**。所有区域权限系统字段将被用于深度验证或授权。以下描述的是全套区域权限系统字段：

无，全部或任何安全性字段组合可以载入区域权限。因此没有必要使用用户 ID-一个授权可以使用其他字段，例如：仅序列号。

**ACCESS** 定义相应用户拥有哪些访问权限。

**USERID** 包含与 **Qlik Sense** 用户名对应的字符串。**Qlik Sense** 将从代理获取登录信息并将之与此字段中的值相比较。

**GROUP** 包含与 **Qlik Sense** 中的组对应的字符串。**Qlik Sense** 将针对此组解析由代理提供的用户。

**OMIT** 包含将为此指定用户省略的字段名称。通配符可能会被使用，并且字段可能为空。一个简单做法就是使用子字段。



不得对关键字段应用 **OMIT**，因为这将更改底层数据结构。这可能会产生逻辑岛和计算不一致性。

Qlik Sense 将代理提供的用户与 UserID 进行比较并对照表格中的组来解析用户。如果用户属于允许访问的组或者用户匹配，则将获得对应用程序的访问权。

由于相同内部逻辑是 Qlik Sense 的特点也用于存取区域中，安全性字段可能放置于不同表格中。区域权限内 **LOAD** 或 **SELECT** 语句列出的全部字段均必须大写。使用 **Upper** 函数将数据库内任何包含小写字母的字段名全部转换为大写，如此方才会被 **LOAD** 或 **SELECT** 语句读取。

通配符 \* 被解释为字段中所有 列出的 值，即表格中的值列表。如果用于脚本存取部分加载的表格中的系统字段之一 **USERID** 或 **GROUP**，则星号将被视为此字段的所有 也包括未列出的 可能值。



当从 QVD 文件加载数据时，使用上限函数将减慢加载速度。



如果已启用了区域权限，那么无法使用此处列出的区域权限系统字段名称作为您的数据模型中的字段名称。

### 示例：

在此示例中，仅财务组中的用户可以打开文档。

ACCESS

GROUP

USER

Finance

## 5.2 动态数据缩减

应用程序中的部分数据可以对基于区域权限登录的用户隐藏，Qlik Sense 以此支持动态数据缩减功能：

- 可以通过使用系统字段 **OMIT** 来隐藏字段 列 。
- 可通过将区域权限数据与真实数据进行链接来隐藏记录 行 ；通过在区域权限和区域应用程序中选择一个或多个具有公共名称的字段来控制要显示/排除的值的选项。用户登录 Qlik Sense 后将复制区域权限中字段的选项至区域应用程序的任意字段中，并使用相同的字段名字段名必须大写 。做好选择后，Qlik Sense 将永久隐藏用户选择之外的所有数据。



用于转换上述描述和字段中所有字段值的全部字段名必须大写，因为所有字段名和字段值在区域权限内会默认转换为大写。

### 示例：

```
section access;
```

```
LOAD * inline [  
ACCESS, USERID,REDUCTION, OMIT  
USER, AD_DOMAIN\ADMIN,*,  
USER, AD_DOMAIN\A,1,  
USER, AD_DOMAIN\B, 2,NUM  
USER, AD_DOMAIN\C, 3, ALPHA  
];  
section application;  
T1:  
LOAD *,  
NUM AS REDUCTION;  
LOAD  
Chr( RecNo()+ord('A')-1) AS ALPHA,  
RecNo() AS NUM  
AUTOGENERATE 3;
```

现在，字段 **REDUCTION** 大写 同时存在于区域权限和区域应用程序中 所有字段值同样是大写 。通常这两个字段会完全不同并分隔开，但是如果使用区域权限，那么这些字段将链接并减少向用户显示的记录数量。

区域权限中的 **OMIT** 字段定义应该对用户隐藏的字段。

结果将为：

- 用户 **ADMIN** 可以查看所有字段，且只有当 **REDUCTION** 为 1、2 或 3 时其他用户才可以在此示例中查看这些记录。
- 用户 **A** 可以查看所有字段，但仅在记录连接至 **REDUCTION=1** 的条件下。
- 用户 **B** 可以查看除去 **NUM** 值外的所有字段，但仅在记录连接至 **REDUCTION=2** 的条件下。
- 用户 **C** 可以查看除去 **ALPHA** 值外的所有字段，但仅在记录连接至 **REDUCTION=3** 的条件下。

### 5.3 继承访问限制

二进制加载会导致新 **Qlik Sense** 应用程序受到继承访问限制。对此新应用程序拥有 **ADMIN** 权限的人可以通过添加新的权限区域来更改新应用程序的访问权限。拥有 **USER** 权限的人可以执行脚本并更改脚本，如此可以添加自己是数据至二进制加载文件。拥有 **USER** 权限的人无法更改访问权限。这样数据库管理员还可以控制用户访问和二进制加载 **Qlik Sense** 应用程序。

## 6 使用 Direct Discovery 访问大数据集

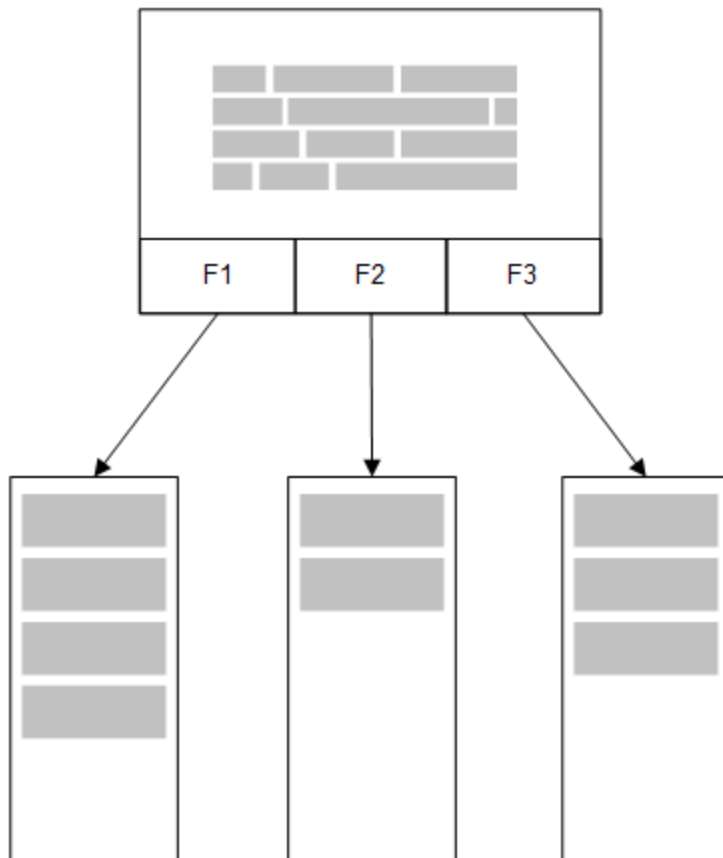
Direct Discovery 可扩展 Qlik Sense 内存中数据模型的相关功能，只需通过无缝关联大数据集和内存中数据的聚合查询访问其他源数据即可。Direct Discovery 增强了商业用户在没有限制的情况下对大数据源执行相关分析的能力。可以选择内存中数据和 Direct Discovery 数据，以使用相同的 Qlik Sense 关联颜色 绿色、白色和灰色 查看数据集之间的相关性。可视化可用于同时分析两个数据集中的数据。

可以使用特殊的脚本语法 **DIRECT QUERY** 为 Direct Discovery 选择数据。在建立 Direct Discovery 结构后，可以使用 Direct Discovery 字段以及内存中数据创建 Qlik Sense 对象。在 Qlik Sense 对象中使用 Direct Discovery 字段时，将在外部数据源中自动运行 SQL 查询。

### 6.1 Direct Discovery 数据与内存中数据之间的差别

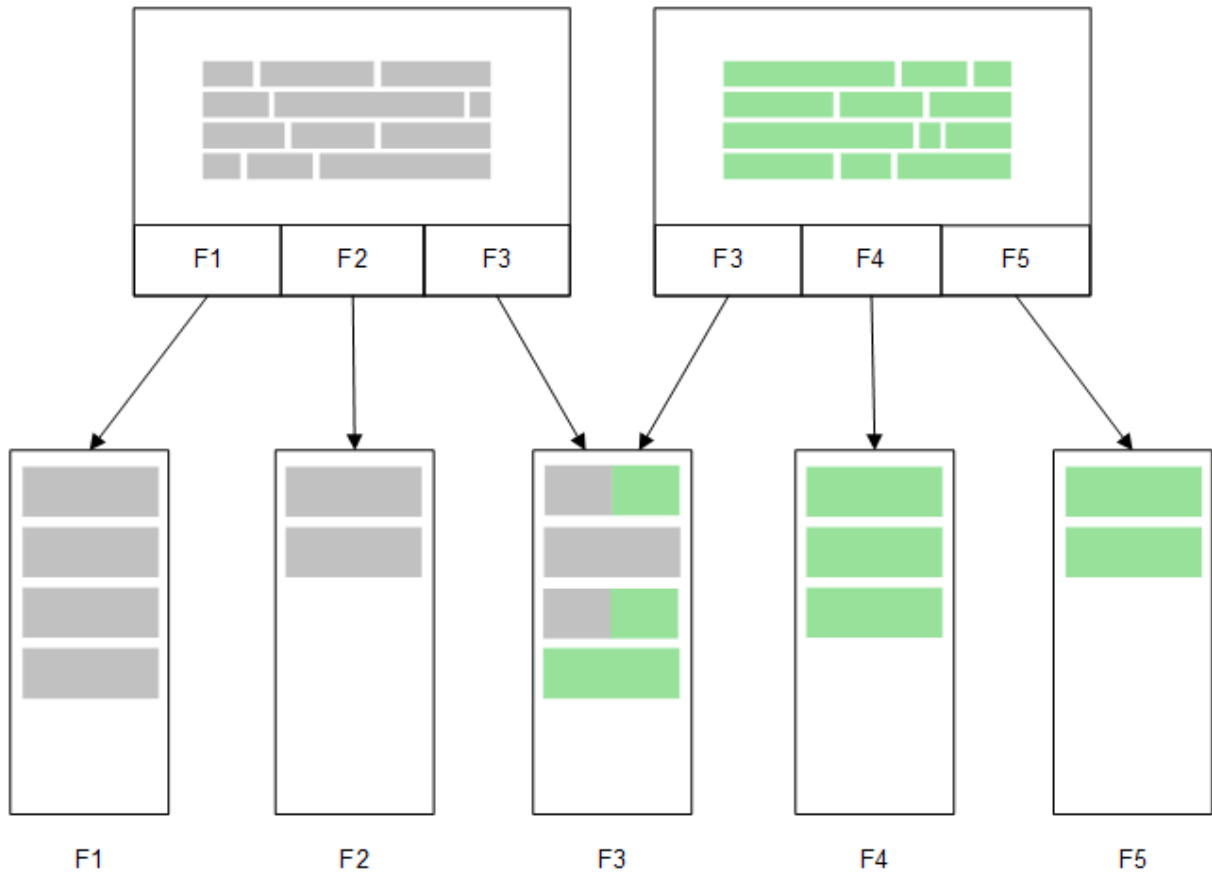
#### 内存中模型

在 Qlik Sense 内存中模型中，从加载脚本中的表格中选择的字段中的所有唯一值都已加载到字段结构，并且相关数据同时也加载到表格。字段数据和相关数据都保留在内存中。



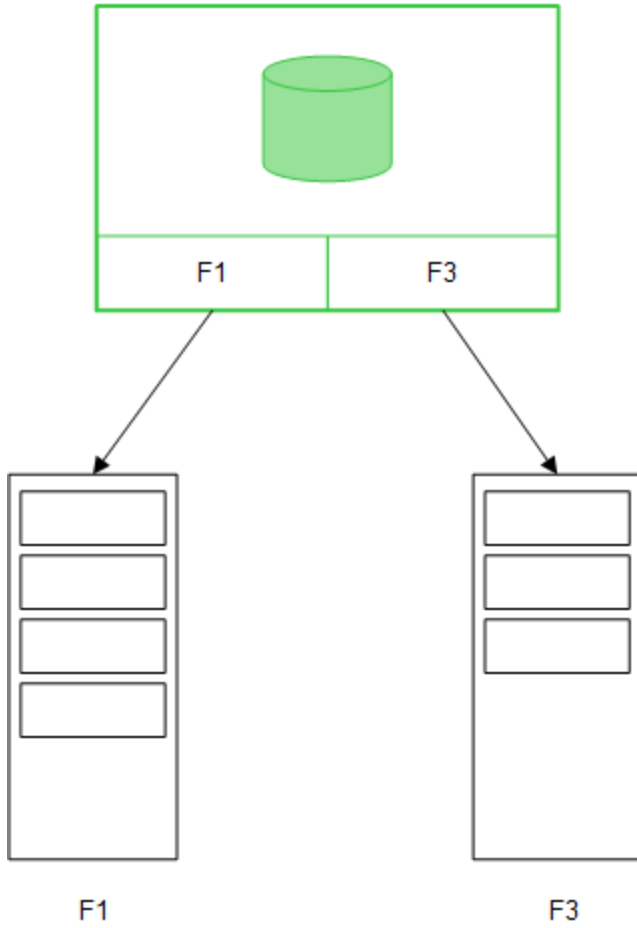
加载到内存的另一个相关表格将共享共同字段，并且该表格可能会将新的唯一值添加到共同字段，也可能会共享现有值。



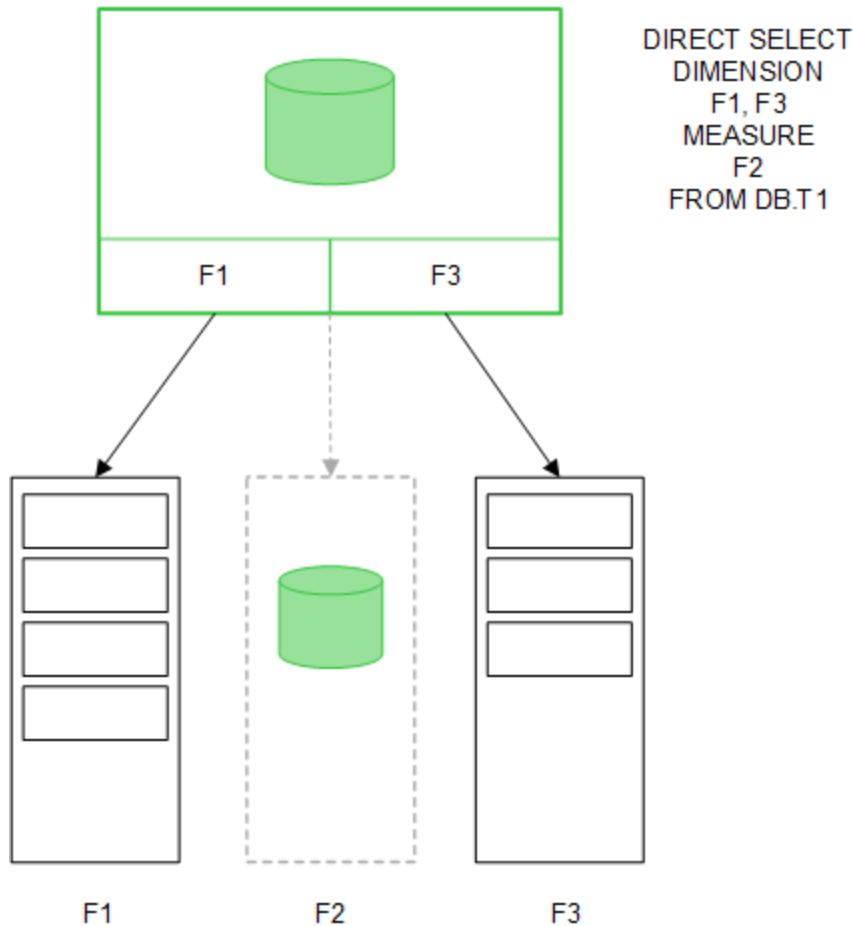


### Direct Discovery

在使用 Direct Discovery **LOAD** 语句 (**Direct Query**) 加载表格字段时, 仅使用 **DIMENSION** 字段创建相似的表格。与使用内存中字段一样, **DIMENSION** 字段的唯一值会加载到内存。但是这些字段之间的相关性会保留在数据库中。



**MEASURE** 字段值也会保留在数据库中。



在建立 Direct Discovery 结构后，Direct Discovery 字段可用于某些可视化对象，并且可用于与内存中字段关联。在使用 Direct Discovery 字段时，Qlik Sense 会自动创建相应的 SQL 查询以对外部数据运行。在选择选择项时，Direct Discovery 字段的相关数据值可用于 WHERE 条件的数据库查询。

使用每个选择项时，会重新计算包含 Direct Discovery 字段的可视化，并且会通过执行 Qlik Sense 创建的 SQL 查询在源数据库表格中进行计算。可以使用计算条件功能指定应重新计算可视化的时间。在满足条件之前，Qlik Sense 不会发送重新计算可视化的查询。

### 内存中字段与 Direct Discovery 字段之间的性能差别

与在源数据库中处理相比，内存中处理始终更快。Direct Discovery 的性能会反映处理 Direct Discovery 查询的数据库的运行系统性能。

可以为 Direct Discovery 使用标准数据库和查询调整最佳做法。所有性能调整操作都应在源数据库中完成。Direct Discovery 不能对 Qlik Sense 应用程序中的查询性能调整提供支持。但可以使用连接池功能进行数据库异步、并行调用。设置连接池功能的加载脚本语法是：

```
SET DirectConnectionMax=10;
```

Qlik Sense 缓存也可以改善整体用户体验。请参阅以下的 *缓存和 Direct Discovery* 第 53 页。

使用 **DIMENSION** 字段的 Direct Discovery 性能也可以通过从关联关系中分离某些字段来改善。为此，可在 **DIRECT QUERY** 中使用 **DETACH** 关键字。虽然不查询分离字段的关联情况，但这些字段仍然是筛选的一部分，从而缩短选择时间。

虽然 Qlik Sense 内存中字段和 Direct Discovery **DIMENSION** 字段的所有数据都保留在内存中，但它们的加载方式会影响内存中的加载速度。在有多个同值实例时，Qlik Sense 内存中字段仅保留一个字段值副本。但是，会加载所有字段数据，然后检索出重复数据。

**DIMENSION** 字段也只存储一个字段值副本，但在加载到内存前会先在数据库中检索出重复值。在处理大量数据时，就像平常一样，使用 Direct Discovery 时，加载数据的速度就像 **DIRECT QUERY** 加载一样，比通过内存中字段使用的 **SQL SELECT** 加载快得多。

### 内存中数据与数据库数据之间的差别

在与内存中数据建立关联时，**DIRECT QUERY** 区分大小写。Direct Discovery 会根据数据库的大小写敏感度从源数据库选择数据。如果数据库不区分大小写，则 Direct Discovery 查询可能会返回内存中查询没有查询的数据。例如，如果不区分大小写的数据库中存在以下数据，则查询值 "Red" 的 Direct Discovery 查询会返回所有行的内容。

ColumnA	ColumnB
red	one
Red	two
rED	three
RED	four

另一方面，选择 "Red," 的内存中选择仅返回：

Red two

Qlik Sense 可在某种程度上规范化数据，从而生成所选数据的匹配项，数据库不会匹配这些匹配项。因此，与 Direct Discovery 查询相比，内存中查询可能会生成更匹配的值。例如，在下表中，数字 "1" 的值根据其空间位置的不同有所不同：

ColumnA	ColumnB
' 1'	space_before
'1'	no_space
'1 '	space_after
'2'	two

如果在 **筛选器窗格** 中为 ColumnA 选择 "1"，其中数据在标准的 Qlik Sense 内存中，则前三行相关联：

'	space_before
1'	

'1'	no_space
'1',	space_after

如果**筛选器窗格**包含 Direct Discovery 数据，选择 "1" 可能仅关联 "no\_space"。为 Direct Discovery 数据返回的匹配项取决于数据库。某些数据仅返回 "no\_space"，某些数据 比如 SQL Server 返回 "no\_space" 和 "space\_after"。

## 缓存和 Direct Discovery

Qlik Sense 缓存可在内存中存储查询的选择状态。在选择相同类型的选择项时，Qlik Sense 会利用缓存中的查询，而不是查询源数据。在选择不同选择项时，将在数据源中执行 SQL 查询。缓存结果将在用户之间共享。

示例：

1. 用户应用最初的选择项。  
SQL 会传递到基础数据源。
2. 用户清除选择，并应用与最初的选择项相同的选择项。  
返回缓存结果，SQL 不会传递到基础数据源。
3. 用户应用不同的选择项。  
SQL 会传递到基础数据源。

可以使用 **DirectCacheSeconds** 系统变量对缓存设置时间限制。在达到时间限制后，Qlik Sense 会清除为之前的选择项生成的 Direct Discovery 查询结果的缓存。Qlik Sense 之后会在源数据中查询这些选择项，并针对指定的时间限制重新创建缓存。

Direct Discovery 查询结果的默认缓存时间是 30 分钟，除非使用 **DirectCacheSeconds** 系统变量。

## 6.2 Direct Discovery 字段类型

在 Direct Discovery 中，有三种数据字段类型：DIMENSION、MEASURE 和 DETAIL。当使用 **Direct Query** 语句在加载脚本中选择 Direct Discovery 选择项时，已在数据字段中设置类型。

所有 Direct Discovery 字段均可与内存中字段组合使用。通常，用作维度的离散值的字段应使用 DIMENSION 关键字加载，而只用于聚合的数字数据应标记为 MEASURE 字段。DIMENSION 字段不能用于对象表达式。

下表概述了 Direct Discovery 字段类型的特征和使用情况：

字段类型	内存中？	构成关联？	用于图表表达式？
DIMENSION	是	是	是

MEASURE	否	否	是
DETAIL	否	否	否

### DIMENSION 字段

DIMENSION 字段已加载到内存中，可用于创建内存中数据与 Direct Discovery 字段数据之间的关联。Direct Discovery DIMENSION 字段还用于定义图表中的维度值。

### MEASURE 字段

另一方面，MEASURE 字段需要在“元级别”识别。MEASURE 字段未加载到内存中，它们不会在数据模型查看器中出现。目的是让 MEASURE 字段中的数据在数据库中而不是在内存中聚合。但是，MEASURE 字段可用于表达式而无需更改表达式语法。因此，对最终用户而言，从数据库使用 Direct Discovery 字段是透明的。

以下聚合函数可用于 MEASURE 字段：

- **Sum**
- **Avg**
- **Count**
- **Min**
- **Max**

### DETAIL 字段

DETAIL 字段提供用户可能想要显示的信息或详细信息，但不能用于图表表达式。DETAIL 字段只能用于**计数**聚合，并且只能在**表格**中查看。指定为 DETAIL 的字段通常包含不能用任何有意义的方式聚合的数据，如注释。

所有字段均可被指定为 DETAIL 字段。

## 6.3 Direct Discovery 支持的数据源

用户可通过 32 位和 64 位连接针对以下数据源使用 Qlik Sense Direct Discovery：

- ODBC/OLEDB 数据源 - 支持所有 ODBC/OLEDB 源，包括 SQL Server、Teradata 和 Oracle。
- 支持 SQL 的自定义连接器 - 用于兼容 SQL 数据存储的 SAP SQL 连接器、自定义 QVX 连接器。

支持 32 位和 64 位连接。

### SAP

对于 SAP，Direct Discovery 只能用于 Qlik SAP SQL Connector，并且需要在 **SET** 变量中使用以下参数：

```
SET DirectFieldColumnDelimiter=' ';  
SET DirectIdentifierQuoteChar=' ';
```

SAP 使用 OpenSQL，由于是使用空格而非逗号分隔列，因此上方的 `set` 语句会导致替换，以便适应 ANSI SQL 和 OpenSQL 之间的差别。

### Google Big Query

Direct Discovery 可用于 Google Big Query，并且需要在 `set` 变量中使用以下参数：

```
SET DirectDistinctSupport=false;
SET DirectIdentifierQuoteChar='[]';
SET DirectIdentifierQuoteStyle='big query'
```

Google Big Query 不支持 **SELECT DISTINCT** 或引用的列/表格名称，并且具有使用“[]”的非 ANSI 引用配置。

### MySQL 和 Microsoft Access

Direct Discovery 可与 MySQL 和 Microsoft Access 结合使用，但鉴于这些数据源中使用的引用字符，可能需要在 `set` 变量中使用以下参数：

```
SET DirectIdentifierQuoteChar='`';
```

## 6.4 使用 Direct Discovery 时的限制

### 支持的数据类型

Direct Discovery 支持所有数据类型，不过可能存在以下情况：需要对 Qlik Sense 定义特定源数据格式。为此，可在加载脚本中使用“SET Direct...Format”语法。以下示例展示了如何定义用作 Direct Discovery 数据源的源数据库的数据格式：

示例：

```
SET DirectDateFormat='YYYY-MM-DD';
```

还有两个脚本变量可控制 Direct Discovery 如何在生成的 SQL 语句中设置货币类型值的格式：

```
SET DirectMoneyFormat (default '#.0000')
SET DirectMoneyDecimalSep (default '.')
```

这两个变量的语法与 **MoneyFormat** 和 **MoneyDecimalSep** 的语法相同，但在应用中有两个重要的不同之处：

- 由于不是显示格式，因此不应包含货币符号或千分位分隔符。
- 默认值不是按场景变化的，而是硬连接到值的。特定于场景的格式包括货币符号。

Direct Discovery 支持选择扩展 Unicode 数据，只需使用一些数据库 例如 SQL Server 所要求的扩展字符串文字 (N'<扩展字符串>') 的 SQL 标准格式。带有脚本变量 **DirectUnicodeStrings** 的 Direct Discovery 可启用此语法。将该变量设置为 "true"，可在字符串文字之前使用 "N"。

### 安全性

在使用 Direct Discovery 时，应考虑以下有关安全性的最佳做法：

- 使用带有 **Direct Discovery** 功能的相同应用程序的所有用户均使用相同的连接。不支持传递和按用户使用凭据的身份验证。
- 仅在服务器模式下支持区域权限。
- 可以使用 **NATIVE** 关键字表达式在数据库中执行自定义 **SQL** 语句，因此加载脚本中设置的数据库连接应使用以下帐户：对此数据库有只读访问权限的帐户。
- **Direct Discovery** 没有日志记录功能，但可以使用 **ODBC** 跟踪功能。
- 可以使用客户端的请求注入数据库。
- 可以从服务器日志文件中获取详细的错误消息。

### 不支持 Qlik Sense 功能

由于 **Direct Discovery** 具有交互性和 **SQL** 语法特定特性，因此不支持多种功能：

- 高级计算 集合分析、复杂表达式
- 计算维度
- 在使用 **Direct Discovery** 字段的对象上比较分析 替换状态
- 搜索工具不支持 **Direct Discovery****MEASURE** 和 **DETAIL** 字段
- 访问 **Direct Discovery** 表格的应用程序中的二进制加载
- 循环和减少
- **Direct Discovery** 表格的合成键
- 脚本中的表格命名方式不适用于 **Direct Discovery** 表格
- 在加载脚本中的 **DIRECT QUERY** 关键字后使用通配符 \* 字符 (**DIRECT QUERY \***)
- 不支持带有 **LONG** 数据类型列的 **Oracle** 数据库表格。
- 科学记数法中超出 [-9007199254740990, 9007199254740991] 范围的大整数会导致舍入错误和未定义行为。

## 6.5 Direct Discovery 中的多表格支持

您可以使用 **Direct Discovery** 通过 **ANSI SQL** 联接功能加载多个表格或视图。在单个图表中，所有度量必须派生自 **Qlik Sense** 中的同一逻辑表，但此表可以通过 **join** 语句联接的来自数据源的多个表格组合。但可以在同一图表中使用源自其他表格的维度。

例如，可以使用 **Where** 子句或 **Join** 子句联接使用 **Direct Discovery** 加载的表格。

### 使用 **Where** 子句联接 **Direct Discovery** 表格

在此脚本示例中，我们从数据库 **AW2012** 加载数据。表格 **Product** 和 **ProductSubcategory** 通过 **Where** 子句使用共同的 **ProductSubCategoryID** 字段联接。

```
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 ;
```

### 使用 Join On 子句联接 Direct Discovery 表格

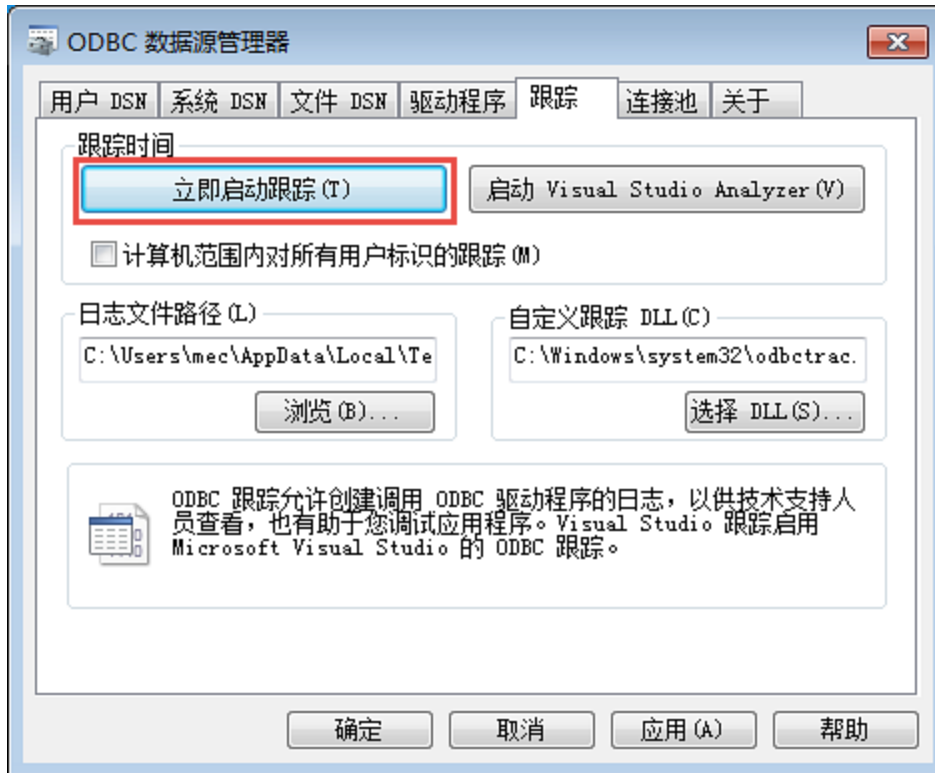
还可以使用 **Join On** 子句联接 Direct Discovery 表格。在此语句示例中，我们通过 SalesOrderID 字段将 SalesOrderHeader 表格联接到 SalesOrderDetail 表格，同时还通过 Customer ID 字段将 Customer 表格联接到 SalesOrderHeader 表格。

在此示例中，我们通过同一逻辑表创建度量，这意味着我们可以在同一图表中使用这些度量。例如，可以创建一个以 SubTotal 和 OrderQty 为度量的图表。

```
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);
```

## 6.6 记录 Direct Discovery 访问

传递到数据源的 Direct DiscoverySQL 语句可以记录到数据库连接的跟踪文件中。对于标准 ODBC 连接，从 **ODBC 数据源管理器** 开始跟踪：



生成的跟踪文件会详述通过用户选择项和交互生成的 SQL 语句。

## 7 数据建模简介

本介绍是如何将数据加载到 Qlik Sense 中，以便为本部分主题提供背景的简要说明，展示了如何实现基本的数据加载和转换。

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。在脚本中，指定要加载的字段和表格。还可以使用脚本语句和表达式操作数据结构。

数据加载期间，Qlik Sense 会确定不同表格的共同字段 关键字段 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。此应用程序是该程序功能的核心，该程序的关键特征包括无限制的数据关联方式、大量的可能维度、分析速度及其压缩大小。打开应用程序时，它存储在 RAM 内。

当应用程序未直接连接到数据源时，Qlik Sense 会一直进行分析。所以，要刷新数据，您需要重新加载脚本。

### 7.1 了解数据结构

#### 数据加载语句

数据由 **LOAD** 或 **SELECT** 语句加载。每一语句都将生成一个内部表格。表格总是可以被视为某种列表，然后每一记录 行 可视为对象类型的新实例，而每一字段 列 可视为某种特定属性或对象的属性。

#### 规则

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。
- 文件名将在后续流程中用以识别字段和进行关联。文件名需要区分大小写，这在脚本中重新命名字段时经常是必需的。

#### 执行脚本

对于典型的 **LOAD** 或 **SELECT** 语句，事件顺序大致如下所示：

1. 表达式评估
2. 用 **as** 重新命名字段
3. 用 **alias** 重新命名字段
4. 字段名限定

5. 如果字段名匹配，则映射数据
6. 将数据存储到内部表格中

### 字段

字段是 Qlik Sense 中的主要数据承载实体。字段通常包含许多值，这些值称为字段值。在数据库术语中，我们说由 Qlik Sense 处理的数据来源于数据文件。文件是由几个字段组成，每一个数据条目构成一条记录。文件、字段和记录等术语分别相当于表格、列和行。Qlik Sense AQL 逻辑仅在字段及其字段值上运作。

通过 **LOAD**、**SELECT**或**Binary** 语句，可以检索字段数据。更改字段数据唯一的方法就是重新执行脚本。真实字段值无法由用户从布局操作或自动控制。一旦读入到 Qlik Sense，只能查看和用于逻辑选择项和计算。

字段值由数字或字母 文本 数据组成。数值实际上有双重值，即数值及其当前格式化文本呈现形式。只有后者显示在工作表对象等中。

字段的内容可显示在筛选窗格中。

### 字段标签

共有三种不同类型的系统标记：脚本生成的用户无法更改的系统标记，脚本生成的可在脚本中更改的系统标记以及用户交互设置的系统标记。系统标记始终位于 **\$** 符号之后。

以下系统标记会在脚本生成结束时自动生成。用户无法更改以下标记：

- **\$system** - 表示系统字段。
- **\$key** - 表示关键字段。
- **\$keypart** - 表示字段是一个或多个合成键的组成部分。
- **\$syn** - 表示合成键。

以下标记也会在脚本生成结束时自动生成，但可以使用脚本语法更改或覆盖。

- **\$hidden** - 表示隐藏字段。
- **\$numeric** - 字段中所有 非 NULL 值均是数值。
- **\$integer** - 字段中所有 非 NULL 值均是整数。
- **\$text** - 字段中所有值都不是数值。
- **\$ascii** - 字段值仅包含 **ascii** 标准字符。
- **\$date** - 字段中的所有 非 NULL 值可被解释为日期 整数 。
- **\$timestamp** - 字段中的所有 非 NULL 值可被解释为时间戳。

用户可以启用和禁用以下标记：

- **\$dimension** - 表示建议用于图表维度和筛选器选框等的字段。
- **\$measure** - 表示建议用于表达式的字段。

用户还可以在脚本中添加自定义标记。这些自定义标记可能与任何系统标记的名称都不同。

## 系统字段

除从数据源解压缩的字段之外，系统字段还可以由 **Qlik Sense** 创建。这些字段都以“\$”开头，并且可像普通字段一样在可视化（如筛选器选框或表格）中显示。系统字段通常在执行脚本期间创建，且主要在应用程序设计中用作辅助工具。



资产面板或表达式编辑器的字段列表中不包括系统字段。如果要使用系统字段，需要通过手动输入来引用。

**示例：** `=$Field`

## 可用系统字段

以下系统字段可用：

<b>\$Table</b>	显示由脚本载入的所有内部表格。
<b>\$Field</b>	显示从表格读取的字段。
<b>\$Fields</b>	表示不同表格中的字段数。
<b>\$FieldNo</b>	显示表格中的字段位置。
<b>\$Rows</b>	表示表格中的行数。
<b>\$Info</b>	显示信息表格。如果应用程序中包括 <code>Info</code> 的名称。

## 逻辑表格

每个 **LOAD** 或 **SELECT** 语句均可生成一个表格。**Qlik Sense** 通常将生成的每个表格结果视为一个逻辑表格。但是，本规则存在两个例外：

- 如果两个或多个语句生成带有相同字段名的表格，各表格会被串联并被视为一个逻辑表格。
- 如果 **LOAD** 或 **SELECT** 语句位于以下任何限定符之后，数据会被更改或得到不同的处理：

<b>concatenate</b>	此表格会串联 添加 至另一个命名表格或最后创建的逻辑表格。
<b>crosstable</b>	此表格会从交叉表格式转换为列格式。
<b>generic</b>	此表格会被分成几个其他的逻辑表格。
<b>info</b>	此表格不会被加载为逻辑表格，但会被加载为包含文件、声音和 <b>URL</b> 等外部信息链接的信息表格。
<b>intervalmatch</b>	此表格 必须包含两列 被解释为数值间隔，该间隔与指定字段中的离散数字关联。
<b>join</b>	<b>Qlik Sense</b> 会根据共同的字段联接此表格与另一个命名表格或最后创建的逻辑表格。
<b>keep</b>	此表格会缩减到只包含与另一个命名表格或最后创建的逻辑表格共有的字段。

<b>mapping</b>	此表格 必须包含两列 会被读取为映像表,且从不与其他表格关联。
<b>semantic</b>	此表格不会加载为逻辑表格,但会作为包含不应联接的关系的语义表格而加载,例如其他相同类型对象的前任、继任及其他参考。

一旦数据完成加载,逻辑表格即会关联。

### 表格名

Qlik Sense 内部表格在其存储到 Qlik Sense 内部数据库时会被命名。例如,表格名可用于 **load...resident** 和 **peek** 语句等,并可在布局的 **\$Table** 系统字段中查看。

表格将根据以下规则进行命名:

1. 如果标签直接位于 **LOAD** 或 **SELECT** 语句之前,则将该标签用作表格名。标签后必须跟冒号。
2. 如果未指定标签,则文件名或表格名将直接跟在所使用的 **LOAD** 或 **SELECT** 语句中的关键词 **FROM** 之后。最多可使用 32 个字符。对于文件名,可忽略扩展名。
3. 加载的内联表格被命名为 **INLINExx**,其中 **xx** 为数字。第一个内联表格被命名为 **INLINE01**。
4. 自动生成的表格被命名为 **AUTOGENERATExx**,其中 **xx** 为数字。第一个自动生成的表格被命名为 **AUTOGENERATE01**。
5. 如果根据以上规则生成的表格名与先前的表格名相冲突,则名称通过 **-x** 扩展,其中 **x** 是数字。数字可以一直增加,避免冲突出现。例如,三个表格可命名为 **Budget/Budget-1** 和 **Budget-2**。

表格名有三个单独的域:**section access**、**section application**和映射表。必须分别处理在**section access**和**section application**中生成的表格名。如果引用的表格名未在该区域内找到,则 Qlik Sense 还会尝试搜索其他区域。映射表格将会单独处理,并且与其他两个表格名的域无任何连接。

### 表格标签

通过使用 **resident** 子句或包含 **peek** 函数的表达式,可以标记内部表格以供 **LOAD** 语句日后引用。此标签 可以是数字或字符的任意字符串 应在第一个 **LOAD** 语句或创建表格的 **SELECT** 语句之前。此标签后必须跟冒号“:”。

如果标签包含空格,则必须使用单引号、双引号或方括号括起来。

#### 示例 1:

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

#### 示例 2:

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

## 逻辑表格之间的关联

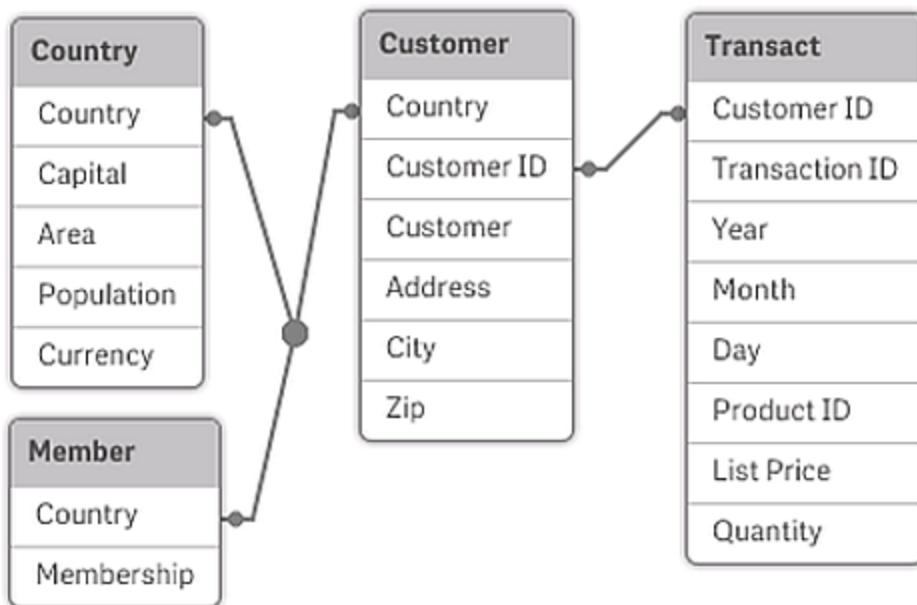
数据库可以包含许多表格。每个表格可视为某些数据的列表，列表中的每条记录均表示某类对象实例。

**示例：**

如果两个表格为不同数据的列表，例如一个是客户列表，另一个是发票列表，两个表格拥有共同的字段，如客户编号，这通常表明两个表格之间存在关系。在标准 SQL 查询工具中，两个表格几乎应始终联接。

Qlik Sense 脚本中定义的表格被称为逻辑表格。Qlik Sense 依据字段名对表格进行关联，并且在做出选择 如在筛选器选框中选择字段值 后执行联接。

这意味着关联几乎等同于联接。唯一的差异在于联接是在脚本执行时执行，逻辑表格通常是联接的结果。关联是在逻辑表格创建之后执行，关联始终在逻辑表格之间做出。



四种表格：国家/地区列表，客户列表，交易列表和会员列表，它们通过 *Country* 和 *CustomerID* 字段彼此关联。

## Qlik Sense 关联与 SQL 自然外部连接的对比

Qlik Sense 关联类似于 SQL 自然外部联接。但是，关联更普遍：SQL 中的外部联接通常是一个表格对于另一个表格的单向投影。关联总是可生成完整的双向自然外部联接。

## 关联字段中的频率信息

大多数关联字段的使用均存在一定局限性，即两个或更多表格包含共同字段。当一个字段在多个表格中出现时，则 Qlik Sense 要想知道哪些表格可用于计算数据频率会有困难。

**Qlik Sense** 可通过分析数据来查看是否有明确的方法可用来确定要计算的主表格 有时存在 ，但大部分情况下，程序只能对此进行猜测。由于错误的猜测可能是极其危险的 **Qlik Sense** 可能会出现计算错误 ，因此程序被设计为不允许在对关联字段的数据解释模糊不清时执行某些操作。

### 关联字段限制

1. 显示字段的筛选器选框中不可能显示频率信息。
2. 该字段的统计框显示大部分统计量为 **n/a**。
3. 在图表中，不可能创建包含依赖字段的频率信息 如 **Sum**、**Count** 函数和 **Average** 的函数的表达式，除非激活 **Distinct** 修饰符。每次重新加载之后，**Qlik Sense** 会扫描所有图表表达式，以查看更改数据结构后是否有任何模糊不清的结果产生。如果发现模糊不清的表达式，则警告对话框会显示，并且表达式也会被禁用。待问题得到纠正后，表达式方可启用。如果启用日志文件，则所有模糊不清的表达式会列在日志内。

### 解决方法

克服这些限制的方法非常简单。使用新名称超时加载执行频率计数的表格中的字段。然后将新字段用于带频率的筛选器选框，或者用于统计框或图表中的计算。

## 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** 通过自动创建合成键来处理此问题。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。

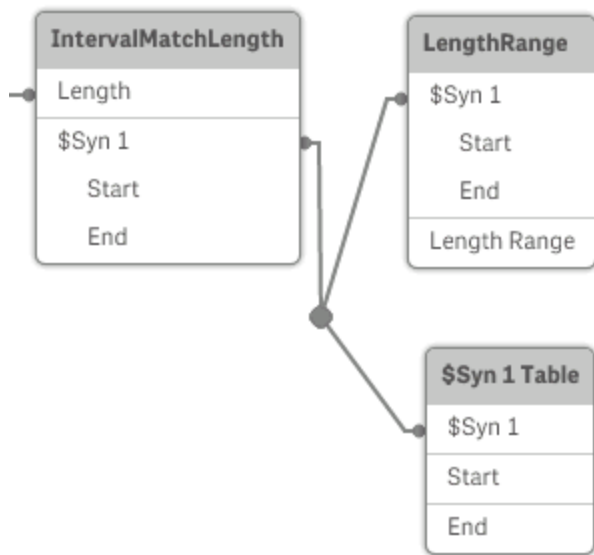
如果在加载数据时收到有关合成键的警告，建议您在数据模型查看器中查看数据结构。您应自己想想数据模型是否正确。有时 但并非总是 ，由于脚本中的错误，会出现合成键。

多个合成键通常是数据模型不正确的症状，但也不是绝对。但是，如果您具有基于其他合成键的合成键，则肯定表明数据模型不正确。



当合成键数量增加时 这取决于数据总数，表格结构和其他因素 ，**Qlik Sense** 可能会，也可能不会处理，并且最终可能会耗用过量的时间和/或内存。在此情况下，您需要通过删除所有合成键来重新编写您的脚本。





## 处理合成键

如果需要避免合成键，有多种方法可在数据加载脚本中解决此问题：

- 确认仅使用逻辑上链接两个表格的字段作为关键字。
  - 诸如“Comment”、“Remark”和“Description”等字段可能存在于多个表格，但不相关，因此不应用作关键字。
  - 诸如“Date”、“Company”和“Name”等字段可能存在于多个表格中并且具有相同值，但仍具有不同角色（订单日期/发货日期、客户公司/供应商公司）。在此情况下，不应将它们用作关键字。
- 确保未使用冗余字段，即，确保仅必要的字段进行连接。如果数据被用作密钥，确保不加载多个内部表格的相同日期的 `year`、`month` 或 `day_of_month`。
- 如有必要，通常使用 `AutoNumber` 脚本函数中的字符串 `concatenation` 自行创建非复合关键字段。

## Qlik Sense 中的数据类型

Qlik Sense 可以正确处理文本字符串，数字，日期，时间，时间戳以及货币。它们可以众多不同的格式排序和分类，并且可用于计算。这意味着，日期，时间和时间戳可以相互加减。

## Qlik Sense 中的数据呈现形式

为了理解 Qlik Sense 中的数据解释和数字格式化，有必要知道程序如何在内部存储数据。所有加载到 Qlik Sense 内的数据都有以下两种呈现形式：字符串和数字。

1. 字符串呈现形式始终可用，是指列表框和其他工作表对象中显示的内容。列表框中数据的格式（数字格式）仅会对字符串呈现形式造成影响。
2. 仅当数据可以解释为有效数字时才可使用数字呈现形式。数字呈现形式可用于所有数值计算和数字排序。

如果读取进一个字段的几个数据项目拥有相同的数字呈现形式，它们全都被视为相同数值，并且全都共享遇到的第一个字符串呈现形式。示例：按此顺序读取的数字 1.0, 1 和 1.000 全都拥有数字呈现形式 1 和初始字符串表示 1.0。

### 数字解释

在加载包含数字、货币或日期的数据时，根据是否定义了数据类型，将以不同方式来解释数据。本节描述如何在两种不同情况下解释数据。

#### 带类型信息的数据

使用 ODBC 加载的数据库中包含带定义数据类型的数字的字段将根据其各自的格式在 Qlik Sense 中进行处理。他们的字符串呈现形式将是带适当应用格式的数字。

Qlik Sense 会记住字段的原始数字格式，即使数字格式在属性面板中的**数字格式**下面针对某个度量进行了更改。

不同数据类型的默认格式为：

- 整数、浮点数：数字的默认设置
- 货币：货币的默认设置
- 时间、日期、时间戳：ISO 标准格式

数字和货币的默认设置可使用脚本数字解释变量或操作系统来设置 **控制面板** 定义。

#### 不带类型信息的数据

对于不带数据源的特定格式信息的数据 例如，文本文件的数据或带一般格式的 ODBC 数据，情况变得更加复杂。最终结果取决于至少六个不同因数：

1. 数据写入源数据库的方法
2. 数字，时间和日期等操作系统设置。 **控制面板**
3. 脚本中解释变量的可选数字的使用
4. 脚本中可选解释函数的使用
5. 脚本中可选格式函数的使用
6. 文档中的数字格式控制

Qlik Sense 尝试将输入数据解释为数字、日期、时间等等。只要系统默认设置用于数据，Qlik Sense 会自动完成解释和显示格式化，并且用户无需更改脚本或 Qlik Sense 中的任何设置。找出输入数据是否正确解释非常容易：数值在列表框中右对齐，文本字符串左对齐。

以下方案会默认使用，直至找到完全匹配项。默认格式为操作系统 即**控制面板** 指定的格式，例如小数分隔符，年，月和日等之间的顺序，或者在一些情况下通过脚本中特定数字解释变量指定的格式。

Qlik Sense 会将数据解释为：

1. 符合数字默认格式的数字。
2. 符合日期默认格式的时间戳。
3. 符合时间和日期默认格式的时间戳。

4. 符合时间默认格式的时间。
5. 符合以下格式的时日期: yyyy-MM-dd
6. 符合以下格式的时间戳: YYYY-MM-DD hh:mm[:ss[.fff]]。
7. 符合以下格式的时间: hh:mm [:ss[.fff]]。
8. 符合货币默认格式的货币。
9. 以“,”作为千分位分隔符的数字和以“.”作为小数分隔符的数字, 前提为假设小数分隔符和千分位分隔符都不设置为“,”。
10. 以“,”作为千分位分隔符的数字和以“.”作为小数分隔符的数字, 前提为假设小数分隔符和千分位分隔符都不设置为“,”。
11. 文本字符串。最后一项测试从未失败: 只要可以读取数据, 则总可以将其解释为字符串。

加载文本文件的数字时会发生一些解释问题, 例如错误的千分位分隔符或小数分隔符可能会导致 Qlik Sense 错误解释数字。首先应该检查脚本中的数字解释变量是否正确定义, 然后检查**控制面板**中的系统设置是否正确。

当 Qlik Sense 解释数据为日期或时间, 则可以在可视化的属性面板中更改日期或时间格式。

由于数据的预定义格式不存在, 因此同一字段中的不同记录当然可能会包含不同格式的数据。例如, 在一个字段内找到有效日期, 整数和文本是可能的。因此, 数据不会格式化, 但会以原始样式显示。

## 日期和时间解释

Qlik Sense 会将数据中发现的每个日期, 时间和时间戳存储为日期序列号。日期序列号用于日期, 时间和时间戳, 并且依据日期和时间实体进行算术运算。因此可增减日期和时间, 还可比较时间间隔等。

日期序列号是指 1899 年 12 月 30 号以来的 实值 天数, 即 Qlik Sense 格式与 Microsoft Excel 和其他程序使用的 1900 日期系统 1900 年 3 月 1 日至 2100 年 2 月 28 日 完全相同。例如, 33857 相当于 1992 年 9 月 10 日。一旦超出此范围, Qlik Sense 会使用相同的日期系统按公历延长。

时间序列号是介于 0 与 1 之间的数字。序列号 0.00000 相当于 00:00:00, 而 0.99999 相当于 23:59:59。混合数字表示日期和时间: 序列号 2.5 相当于 1900 年 1 月 1 日的中午 12:00。

但是日期按照字符串的格式显示。默认情况下使用**控制面板**中所做的设置。使用脚本中的数字解释变量, 或者借助格式化函数也可以设置日期格式。最后, 还可以在工作表对象的属性工作表中重新格式化数据。

### 示例 1:

1997-08-06	存储为:	35648
9:00	存储为:	0.375
1997-08-06 9:00	存储为:	35648.375

以及反过来。

35648	数字格式 'D/M/YY' 显	6/8/97
-------	-----------------	--------

示为：

0.375 数字格式 'hh.mm' 显示 09.00  
为：

Qlik Sense 会遵循一套规则，尝试解释日期，时间和其他数据类型。但是最终结果受上述众多因素的影响。

### 示例 2:

这些示例假定采用以下默认设置：

- 数字小数位分隔符：。
- 缩写日期格式：YY-MM-DD
- 时间格式：hh:mm

以下表格显示了当数据读取到脚本中无特殊解释函数时 Qlik Sense 的不同呈现形式：

源数据	Qlik Sense 默认解释	“YYYY-MM-DD”日期格式	“MM/DD/YYYY”日期格式	“hh:mm”时间格式	“# ##0.00”数字格式
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	8/6/97	0:00	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

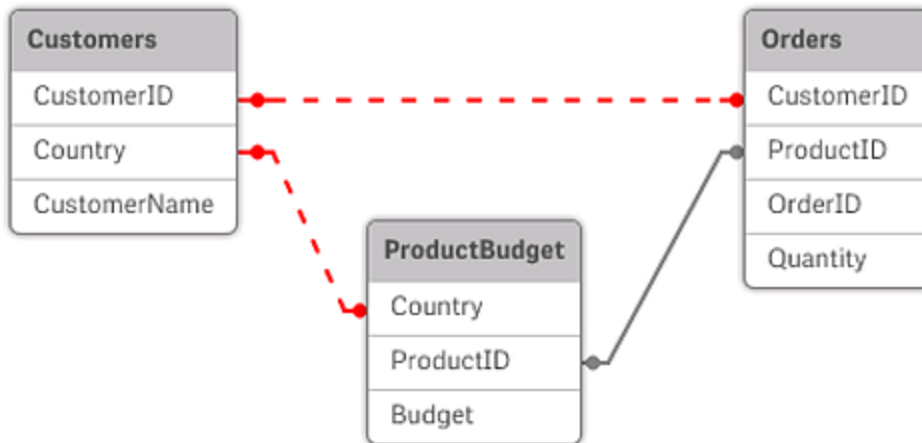
以下表格显示了当数据读取到在脚本中使用 date#(A, 'M/D/YY') 解释函数的 Qlik Sense 时的不同呈现形式：

源数据	Qlik Sense 默认解释	“YYYY-MM-DD”日期格式	“MM/DD/YYYY”日期格式	“hh:mm”时间格式	“# ##0.00”数字格式
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	8/6/97	8/6/97	0:00	35 648.00

## 7.2 了解循环引用

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。

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。



带循环引用的三个表格

Qlik Sense 通过松散耦合表来打破循环，解决循环引用的问题。如果 Qlik Sense 在执行加载脚本时发现循环数据结构，将会显示一个警告对话框，并设置一个或多个表格为松散组合。Qlik Sense 通常会尝试设置循环中的最长表格成为松散耦合表，因为它往往是交易表格，通常也是应该作松散处理的表格。在数据模型查看器中，松散耦合表通过连接到其他表格的红色虚线表示。

示例：

从三个图表中加载的数据包括以下：

- 一些国家足球队的名字
- 一些城市的足球俱乐部
- 一些欧洲国家的城市

The image shows three data tables: NationalTeams, Clubs, and Cities.

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

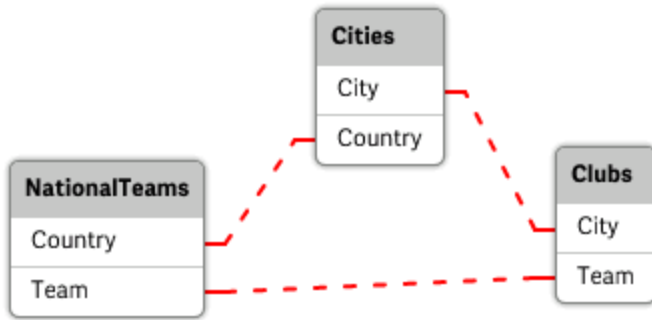
Country	City
Germany	Hamburg
Germany	Munich
Italy	Milano
Italy	Rome
Italy	Turin
Spain	Barcelona
Spain	Madrid

源数据表格的视图

此数据结构并不是很好，因为字段名 *Team* 有两个不同的使用目的：国家队名和地方俱乐部名。不可能的逻辑情况因表格中的数据而呈现于前。

当加载表格到 Qlik Sense 时，Qlik Sense 确定不重要的数据连接，然后松散处理表格。

选择**数据模型查看器**可查看 Qlik Sense 解释数据连接关系的方式：




循环引用的视图用红色虚线表示。

城市与及它们所属的国家现在处理成不同国家的国家队和不同城市的当地俱乐部的松散耦合表。

## 解决循环引用

一旦循环引用出现，您需要通过分配一个唯一的相同名称命名字段来编辑数据加载脚本。

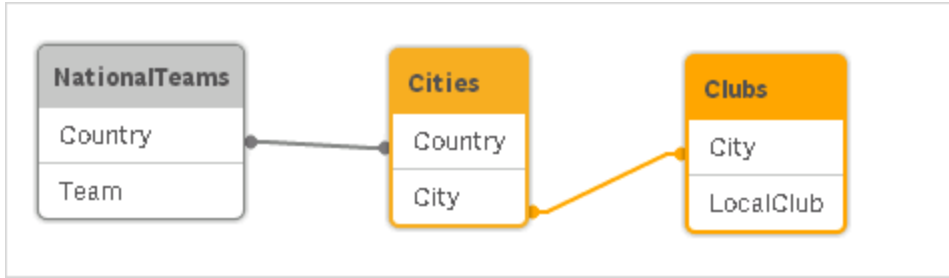
执行以下操作：

1. 打开数据加载编辑器。
2. 为其中一个重复字段名称编辑 **LOAD** 语句。  
在此例中，拥有当地球队及其城市的 **LOAD** 语句将会为 *Team* 编辑新名称，例如 *LocalClub*。更新的 **LOAD** 语句读作：  
`LOAD City, Team as LocalClub`
3. 点击工具栏中的  以在应用程序中重新加载数据。

您现在已经有处理所有表格的逻辑了。在此例中，在选择 *Germany* 后，国家队名，德国城市名和每个城市的当地俱乐部名与之相关：

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

打开**表格查看器**后，可以看到常规联系已替换松散组合联系：



## 7.3 重新命名字段

有时需要重新命名字段以便获取所需关联。重新命名字段的三个主要原因为：

1. 两个字段的命名不同，尽管表示的意思相同：

- 字段 *ID* 在 *Customers* 表格中
- 字段 *CustomerID* 在 *Orders* 表格中

两个字段都表示特定客户的识别码，因此两个字段的命名应相同，如均为 *CustomerID*。

2. 两个字段的命名相同，但实际上表示的意思不同：

- 字段 *Date* 在 *Invoices* 表格中
- 字段 *Date* 在 *Orders* 表格中

两个字段最好应分别重新命名为 *InvoiceDate* 和 *OrderDate*。

3. 在数据库中有可能出现拼写错误或有关大小写字母的不同惯例。

字段可以在脚本中重新命名，因此不需要更改原始数据。可以通过两种方法按照示例重新命名字段。

### 示例 1: 使用 **alias** 语句

**LOAD** 或 **SELECT** 语句可以放在 **alias** 语句之后。

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

### 示例 2: 使用 **as** 说明符

**LOAD** 或 **SELECT** 语句可以包含 **as** 说明符。

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

## 7.4 串联表格

### 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。

示例：

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

结果内部表格包含字段 **a**、**b** 和 **c**。记录数是表格 1 和表格 2 中的记录数的总和。



字段的数量和名称必须完全相同。两个语句的排列顺序是任意的。

### 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后创建的表格。

示例：

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

结果内部表格包含字段 **a**、**b** 和 **c**。结果表中的记录数表格 1 和表格 2 中的记录数的总和。表格 2 记录中的字段 **b** 的值为 **NULL**。



字段的数量和名称必须完全相同。除非先前加载表格的名称在 **concatenate** 语句内有具体说明，否则 **concatenate** 前缀会使用最后一个创建的表格。因而此时两个语句的排列顺序并不是任意的。

### 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。使用 **noconcatenate** 语句可以防止串联。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。

示例：

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

## 7.5 从之前加载的表格中加载数据

您可以在 **LOAD** 语句中使用 **Resident** 断言，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 ，这很有用。

示例：

在本例中，数据解释在 **Resident** 中执行，因为它在初始 **Crosstable LOAD** 中无法完成。

```
PreBudget:  
Crosstable (Month, Amount, 1)  
LOAD Account,
```



```

Jan,
Feb,
Mar,
...
From Budget;

Budget:
Noconcatenate
LOAD
  Account,
  Month(Date#(Month,'MMM')) as Month,
  Amount
Resident PreBudget;

Drop Table PreBudget;

```



一种使用 **Resident** 的常见情况是，您希望使用临时表格来计算或筛选数据。一旦您使用临时表格达到了目的，就会使用 **Drop table** 语句将其删除。

## Resident 或前面的 LOAD?

在大多数情况下，通过使用前面的 **LOAD** 语句可以获得相同的结果，也就是从以下 **LOAD** 或 **LOAD** 语句加载的 **SELECT** 语句 无需指定您通常要指定的源限定符，如 **From** 或 **Resident** 。前面的 **LOAD** 通常是更快的选项，但是在某些情况下，您需要使用 **Resident LOAD**：

- 在处理 **LOAD** 语句之前，如果您要使用 **Order\_by** 子句对记录进行排序。
- 如果您要使用以下任何前缀，在这种情况下，前面的 **LOAD** 不受支持：
  - **Crosstable**
  - **Join**
  - **Intervalmatch**

## 前置 LOAD

前置 **LOAD** 可让您在一次遍历中加载表格，但仍然需要定义多个连续的转换。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。

可以使用 **Resident** 实现相同的结果，但在大多数情况下前置 **LOAD** 运行速度更快。

前置 Load 的另一个好处是可将计算保存在一个地方，并在上面部署的 **LOAD** 语句中进行重复使用。



以下前缀不能与前置 **LOAD** 结合使用：**Join**、**Crosstable** 和 **Intervalmatch**。

**示例 1:** 可通过 **SELECT** 语句加载转换数据。

如果使用 **SELECT** 语句从数据库加载数据，则不能使用 Qlik Sense 函数解释 **SELECT** 语句中的数据。解决方法是添加 **LOAD** 语句，这样就可以针对上面的 **SELECT** 语句执行数据转换。

在本例中，我们在 **LOAD** 语句中使用 Qlik Sense 函数 **Date#** 解释存储作为字符串的日期，并使用上一个 **SELECT** 语句解释数据源。

```
LOAD Date#(OrderDate,'YYYYMMDD') as OrderDate;  
SQL SELECT OrderDate FROM ... ;
```

**示例 2: 可通过重复使用计算简化脚本。**

在本例中，我们在脚本中多次使用了计算：

```
LOAD ...,  
    Age(FromDate + IterNo() - 1, BirthDate) as Age,  
    Date(FromDate + IterNo() - 1) as ReferenceDate  
Resident Policies  
    while IterNo() <= ToDate - FromDate + 1 ;
```

通过在第一次遍历中引入计算，我们在前置 **LOAD** 的 **Age** 函数中进行了重复使用：

```
LOAD ..., ReferenceDate,  
    Age(ReferenceDate, BirthDate) as Age;  
LOAD *,  
    Date(FromDate + IterNo() - 1) as ReferenceDate  
Resident Policies  
    while IterNo() <= ToDate - FromDate + 1 ;
```

## 8 数据建模的最佳实践

本部分介绍根据数据构建方式和需要实现的数据模型将数据加载到 Qlik Sense 应用程序中的各种不同方式。

通常，将数据加载到应用程序的方式可以按提取、转换和加载过程介绍：

### 1. 提取

第一步是从数据源系统提取数据。在脚本中，使用 **SELECT** 或 **LOAD** 语句定义此步骤。这些语句之间的差别是：

- **SELECT** 用于从 ODBC 数据源或 OLE DB 提供程序选择数据。**SELECT SQL** 语句是由数据提供程序 而非 Qlik Sense 评估。
- **LOAD** 用于从文件、脚本中定义的数据、之前加载的表格、网页、后续 **SELECT** 语句的结果中加载数据，或通过自动生成数据加载数据

### 2. 转换

转换阶段涉及通过使用脚本函数和规则派生所需数据模型结构来操作数据。典型的操作包括：

- 计算新值
- 转换编码值
- 重新命名字段
- 联接表格
- 聚合值
- 透视
- 数据验证

### 3. Load

在最后一步中，可以运行脚本来加载定义到应用程序中的数据模型。

您的目标应是创建数据模型，以便在 Qlik Sense 中有效地处理数据。通常，这意味着您的目标应是不使用任何循环引用合理地规范化星型模型和雪花模型，即将每个实体保存在单独的表格中的模型。换句话说，典型的数据模型如下所示：

- 中心事实表格包含维度和数字的关键项，用于计算度量 如单位数量、销售金额和预算金额 。
- 周围表格包含维度及其所有属性 如产品、客户、类别、日历和提供商 。



在许多情况下，通过在加载脚本中创建更丰富的数据模型或在图表表达式中执行聚合可以解决一个任务，如聚合。作为一般规则，如果在加载脚本中保存数据转换，将会体验到更好的性能。



一种比较好的做法是在文件中描绘数据模型。这可以通过提供数据结构帮助您提取所需的数据和执行转换。

数据模型中的每个表格通常都对应一个 **SELECT** 或 **LOAD** 语句。这些语句之间的差别是：

- **SELECT** 用于从 ODBC 数据源或 OLE DB 提供程序选择数据。**SELECT** SQL 语句是由数据提供程序 而非 Qlik Sense 评估。
- **LOAD** 用于从文件、脚本中定义的数据、之前加载的表格、网页、后续 **SELECT** 语句的结果中加载数据，或通过自动生成数据加载数据

## 8.1 数据和字段的指导原则

以下是在使用 Qlik Sense 时需要知道的某些约定和限制。例如：数据表格和字段的上限，以及 Qlik Sense 中加载数据的最大数量。

### 加载数据数量的指导原则

可以加载到 Qlik Sense 中的数据量主要受计算机主内存量的限制。

### 数据表格和字段的上限

请注意，如果创建很大的应用程序，Qlik Sense 应用程序一个字段的非重复值不能超过 2,147,483,648 个。

可以加载的字段和数据表格数以及表格单元格和表格行数仅受 RAM 限制。

### 加载脚本段的建议限制

每个加载脚本段使用的建议最大字符数为 50,000 个。

### 数字格式和时间格式的约定

在许多解释和格式化函数中，可以通过使用格式代码设置数字和日期的格式。本主题介绍对数字、日期、时间或时间戳设置格式所使用的惯例。这些惯例同时适用于脚本和图表函数。

#### 数字格式

- 要表示一个具体数位的数字，可以使用符号 "0" 代表每一个数位。
- 要表示一个可能数位，可以使用符号 "#"。如果格式只包含小数点左侧的 #，则小于 1 的数字前面带一个小数点。
- 要标记千位分隔符的位置或小数分隔符的位置，可以使用适用的千位分隔符和小数分隔符。

格式代码用于定义分隔符的位置。无法在格式代码中设置分隔符。在脚本中，对于这种情况可以使用 **DecimalSep** 和 **ThousandSep** 变量。

可以对任何位置的数字使用千位分隔符分组数位。例如，一个格式字符串 "0000-0000-0000" 千位分隔符="-" 可以用于显示一个 12 位部件编号 "0012-4567-8912"。

示例：

# ##0	描述带千位分隔符的整数数字。
###0	描述带千位分隔符的整数数字。
0000	描述至少四位数的整数数字。如数字 123 将显示为 0123。

<b>0.000</b>	描述带三个小数位的数字。
<b>0.0##</b>	描述带至少 1 个小数位和最多三个小数位的数字。

### 特殊数字格式

Qlik Sense 可以解释及格式化任何基数在 2 和 36 之间的数字，包括二进制、八进制及十六进制。它也可处理罗马格式。

<b>二进制格式</b>	要表示二进制格式，格式代码前面应带 (bin) 或 (BIN)。
<b>八进制格式</b>	要表示八进制格式，格式代码前面应带 (oct) 或 (OCT)。
<b>十六进制格式</b>	要表示十六进制格式，格式代码前面应带 (hex) 或 (HEX)。如果使用大写版本，格式化将使用 A-F 如 14FA。非大写版本将导致格式化使用 a-f 如 14fa。解释对两种变量都可用，忽略代码的大小写。
<b>十进制格式</b>	允许使用 (dec) 或 (DEC) 表示小数格式，但是没有必要。
<b>自定义基数格式</b>	要使用介于 2 和 36 之间的任何基数表示一个格式，格式代码前面应以 (rxx) 或 (Rxx) 为开头，其中 xx 是两位数数字，表示要使用的基数。如果使用大写 R，则大于 10 的基数中的字母在 Qlik Sense 格式化时将变成大写如 14FA。非大写的 r 将导致使用非大写字母格式化如 14fa。解释对两种变量都可用，忽略代码的大小写。注意，(r02) 与 (bin) 相等，(R16) 与 (HEX) 相等，以此类推。
<b>罗马格式</b>	要表示罗马数字，格式代码前面应带 (rom) 或 (ROM)。如果使用大写版本，格式化将使用大写字母如 MMXVI。非大写版本将导致格式化使用小写字母如 mmxvi。解释对两种变量都可用，忽略代码的大小写。罗马字母负数将产生一个负号，零将产生 0。罗马格式将忽略小数位。

示例：

<b>num(199, 'bin')</b>	返回	11000111
<b>num(199, 'oct')</b>	返回	307
<b>num(199, 'hex')</b>	返回	c7
<b>num(199, '(HEX)')</b>	返回	C7
<b>num(199, '(r02)')</b>	返回	11000111
<b>num(199, '(r16)')</b>	返回	c7
<b>num(199, '(R16)')</b>	返回	C7
<b>num(199, '(R36)')</b>	返回	5J
<b>num(199, '(rom)')</b>	返回	cxcix
<b>num(199, '(ROM)')</b>	返回	CXCIX

## 日期

可以使用下列符号格式化日期。可以使用任意分隔符。

<b>D</b>	要描述日期，每一个数位使用符号 "D"。
<b>M</b>	要描述月份，使用符号 "M"。 <ul style="list-style-type: none"> <li>• 使用 "M" 或 "MM" 表示一位或两位数字。</li> <li>• "MMM" 表示以字母表示的短月份名称，由操作系统定义，或由脚本中覆盖系统变量 <b>MonthNames</b> 定义。</li> <li>• "MMMM" 表示以字母表示的长月份名称，由操作系统定义，或由脚本中覆盖系统变量 <b>LongMonthNames</b> 定义。</li> </ul>
<b>Y</b>	要描述年，每一个数位使用符号 "Y"。
<b>W</b>	要描述周几，使用符号 "W"。 <ul style="list-style-type: none"> <li>• "W" 将以一个数位方式返回天的数字 如 0 代表周一。</li> <li>• "WW" 将返回两位数的数字 如 02 表示周三。</li> <li>• "WWW" 将显示周名称的短版本 如 Mon，由操作系统定义，或由脚本中覆盖系统变量 <b>DayNames</b> 定义。</li> <li>• "WWWW" 将显示周名称的长版本 如 Monday，由操作系统定义，或由脚本中覆盖系统变量 <b>LongDayNames</b> 定义。</li> </ul>

示例：使用 31st March 2013 作为示例日期

<b>YY-MM-DD</b>	描述如 13-03-31 的日期。
<b>YYYY-MM-DD</b>	描述如 2013-03-31 的日期。
<b>YYYY-MMM-DD</b>	描述如 2013-Mar-31 的日期。
<b>DD MMMM YYYY</b>	描述如 31 March 2013 的日期。
<b>M/D/YY</b>	描述如 3/31/13 的日期。
<b>W YY-MM-DD</b>	描述如 6 13-03-31 的日期。
<b>WWW YY-MM-DD</b>	描述如 Sat 13-03-31 的日期。
<b>WWWW YY-MM-DD</b>	描述如 Saturday 13-03-31 的日期。

## 时间

可以使用下列符号格式化时间。可以使用任意分隔符。

<b>h</b>	要描述小时，每一个数位使用符号 "h"。
<b>m</b>	要描述分钟，每一个数位使用符号 "m"。
<b>s</b>	要描述秒，每一个数位使用符号 "s"。

<b>f</b>	要描述分秒，每一个数位使用符号 "f"。
<b>tt</b>	要以 AM/PM 格式描述时间，在时间后使用符号 "tt"。

示例：使用 **18.30** 作为示例时间：

<b>hh:mm</b>	描述如 18:30 的时间
<b>hh.mm.ss.ff</b>	描述如 18.30.00.00 的时间
<b>hh:mm:tt</b>	描述如 06:30:pm 的时间

## 时间戳

在时间戳中使用的符号与以上在日期和时间中使用的符号相同。

示例：使用 **31th March 2013 18.30** 作为示例时间戳

<b>YY-MM-DD hh:mm</b>	描述如 13-03-31 18:30 的时间戳
<b>M/D/Y hh.mm.ss.ffff</b>	描述如 3/31/13 18.30.00.0000 的时间戳

## 8.2 使用 QVD 文件

QVD (QlikView Data) 文件是包含从 Qlik Sense 中导出的数据表的文件。QVD 是本地 Qlik 格式只可由 Qlik Sense 或 QlikView 写入及读取。当从脚本中读取数据时该文件格式可提升速度，同时又非常紧凑。从 QVD 文件读取数据通常比从其他数据源读取快 10 到 100 倍。

QVD 文件可以用两种模式读取：标准 快速 和优化 超快。所选模式由脚本引擎自动确定。尽管字段可以重命名，但仅当全部加载字段在无任何转换的形式下读取 操作字段公式 时才可以使用优化模式。导致 Qlik Sense 解压记录的 **where** 子句也将禁用优化加载。

QVD 文件正好包含一个数据表，由三个部分组成：

1. 页眉。



如果使用 QlikView 生成 QVD 文件，则页眉为在表格中描述字段且格式完整的 XML 页眉 UTF-8 字符集、后续信息的布局以及其他元数据。

2. 字节填充格式符号表。
3. 位填充格式实际表。

## QVD 文件的目的是

QVD 文件具有许多用途。至少可轻易识别四种主要的用途。在任何给定的情况下都可以应用不只一个：

### 提高加载速度

通过缓冲 QVD 文件中输入数据的不改变或缓慢改变部分，执行大型数据集脚本可变得相当快。

### 降低数据库服务器上的加载量

从外部数据源提取的数据量也可大幅度降低。这可减少外部数据库的工作量和网络流量。而且，当几个脚本共享相同的数据时，只需要将这些数据从源数据库加载到 QVD 文件一次即可。其他应用程序可以利用此 QVD 文件中相同的数据。

### 合并多个应用程序数据

使用 **binary** 脚本语句，可能只能从单个应用程序将数据加载到另一个应用程序，但使用 QVD 文件，脚本可以合并任何数量的应用程序。例如，这样能够使应用程序合并不同业务部门的类似数据。

### 增量加载

在许多常见情况下，QVD 功能可用于简化增量加载，通过从不断扩大的数据库中加载新记录。

### 创建 QVD 文件

可通过两种方法创建 QVD 文件：

1. 使用脚本中的 **store** 命令进行显式创建并命名。在脚本中说明要将先前读取的表格或其部分导出到您选定位置上的一个明确命名文件。
2. 从脚本中自动创建并维护。通过为 **LOAD** 或 **SELECT** 语句加入 **buffer** 前缀，Qlik Sense 将自动创建 QVD 文件，但重新加载数据时需使用某些条件代替初始数据源。

在结果 QVD 文件之间没有不同，如在读取速度方面等。

### 从 QVD 文件读取数据

可以通过以下方法读入或访问 QVD 文件：

1. 加载 QVD 文件作为显式数据源。QVD 文件可由脚本中的 **LOAD** 语句引用，与任何其他类型的文本文件一样 **csv**、**fix**、**dif**、**biff** 等。

示例：

```
LOAD * from xyz.qvd (qvd);  
LOAD Name, RegNo from xyz.qvd (qvd);  
LOAD Name as a, RegNo as b from xyz.qvd (qvd);
```

2. 自动加载缓冲 QVD 文件。当在 **LOAD** 或 **SELECT** 语句上使用 **buffer** 前缀时，无需明确说明读取的语句。Qlik Sense 将自行确定使用 QVD 文件数据的程度这一点同通过原始 **LOAD** 或 **SELECT** 语句获取数据相反。
3. 通过脚本访问 QVD 文件。许多脚本函数 都以 **qvd** 开头 都可用于检索一个 XML 文件的 QVD 标题中发现的数据的不同信息。



## 8.3 使用 Join 和 Keep 合并表格

将两个表格组合成一个表格的连接操作。组合后的表格记录是由原始表格的记录所组成，通常在将两个表格的记录合并到任何一个表格中时，一个常见值可用于一个或几个常见字段，这就是所谓的自然联接。在 Qlik Sense 中，联接可在脚本中创建，以生成逻辑表格。

联接已处于脚本中的表格是完全有可能的。Qlik Sense 逻辑随后无法看到单独的表格，但能看到联接结果，即单一的内部表格。某些情况下这是必要的，但也存在一些缺点：

- 加载表格往往会变得更大，这将使 Qlik Sense 运行得更慢。
- 一些信息可能会丢失：原始表格中的频率 记录数 可能不再可用。

**Keep** 功能可以在表格存入 Qlik Sense 之前将两个表格中的一个或两个缩减为表格数据的交集，旨在减少需要使用显式联接的情况。



在本文档中，术语“联接”通常指创建内部表格前所作的联接。但是，创建内部表格后所作的关联在本质上也是联接。

### SQL SELECT 语句中的联接

借助一些 ODBC 驱动程序可以在 **SELECT** 语句中进行联接。这相当于用 **Join** 前缀进行联接。

但是，大部分 ODBC 驱动程序无法进行完整 双向 外部联接。它们只能进行左向或右向外部联接。左右 向外部联接仅包括联接密钥存在于左 右 表格的组合。完整外部联接包括所有组合。Qlik Sense 会自动进行完整外部联接。

此外，在 **SELECT** 语句中进行联接远比在 Qlik Sense 中进行联接要复杂很多。

示例：

```
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;
```

本 **SELECT** 语句联接包含虚构公司订单的表格，并带有订单详情表格。这是右外部联接，意味着 *OrderDetails* 的所有记录包括在内，带有 *OrderID* 的订单不在 *Orders* 表格中。但是，在 *Orders* 表格中但在 *OrderDetails* 中的订单不包括在内。

### Join

进行联接的最简单方法是在脚本中使用 **Join** 前缀，以联接内部表格与另一个命名表格或最后创建的表格。该联接是外部联接，创建两个表格的所有可能的数值组合。

示例：

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

结果内部表格包含字段 a、b、c 和 d。根据这两个表格的字段值不同，记录的数量也会有所不同。



联接的字段名称必须完全相同。联接的字段数可以是任意的。表格通常包含一个或几个共同字段。没有共同字段会致使表格生成笛卡儿积。所有字段均相同也是可能的，但通常没有意义。除非先前加载表格的名称在 **Join** 语句内有具体说明，否则 **Join** 前缀会使用最后一个创建的表格。因而此时两个语句的排列顺序并不是任意的。

## Keep

数据加载脚本中的显式 **Join** 前缀可完全联接这两个表格。结果会生成一个表格。在许多情况下，这种联接将产生很大的表格。Qlik Sense 的其中一个主要功能就是使表格之间形成关联，而不是联接这些表格，这种关联可以大大减少使用内存，提高处理速度并且灵活多变。保存功能旨在减少需要使用显式联接的情况。

两个 **LOAD** 或 **SELECT** 语句之间的 **Keep** 前缀会将两个表格中的一个或两个表格缩减为表格数据交集，然后才将其存储到 Qlik Sense。**Keep** 前缀必须是 **Inner**、**Left** 或 **Right** 关键字之一。选择表格记录的方法与相应联接方法相同。但是，这两个表格并未联接，而是分别命名后存储在 Qlik Sense 中。

## Inner

数据加载脚本中的 **Join** 和 **Keep** 前缀可以位于前缀 **Inner** 之后。

如果用于 **Join** 之前，说明两个表格之间的联接应为内部联接。由此生成的表格所包含的两表格之间的组合必带有两表格的完整数据集。

如果用于 **Keep** 之前，说明首先应使两个表格缩减为它们自身的共同交集，然后才可在 Qlik Sense 中存储这些表格。

示例：

在这些示例中，我们使用源表格 Table1 和 Table2：

Table1		Table2	
A	B	A	C
1	aa	1	xx
2	cc	4	yy
3	ee		

Inner 示例源表格

首先，我们使用两个表格中合并的数据在表格上执行 **Inner Join**，从而使 QVTable 仅包含一行，即两个表格中仅有的记录。

```
QVTable:
SELECT * from Table1;
inner join SELECT * from Table2;
```

## QVTable

A	B	C
1	aa	xx

*Inner Join 示例*

如果我们执行 **Inner Keep**，则会获得两个表格。这两个表格通过通过字段 **A** 关联。

```
QVTab1:
SELECT * from Table1;
QVTab2:
inner keep SELECT * from Table2;
```

QVTab1		QVTab2	
A	B	A	C
1	aa	1	xx

*Inner Keep 示例*

## Left

数据加载脚本中的 **Join** 和 **Keep** 前缀可以位于前缀 **left** 之后。

如果用于 **Join** 之前，说明两个表格之间的联接应为左联接。由此生成的表格所包含的两表格之间的组合仅带有第一个表格的完整数据集。

如果用于 **Keep** 之前，说明首先应使第二个表格缩减为其与第一个表格间的共同交集，然后才可在 Qlik Sense 中存储此表格。

示例：

在这些示例中，我们使用源表格 Table1 和 Table2:

Table1		Table2	
A	B	A	C
1	aa	1	xx
2	cc	4	yy
3	ee		

*Left 示例源表格*

首先，我们在表格上执行 **Left Join**，从而产生 QVTable，其中包含 Table1 中的所有行 与 Table2 中匹配的字段合并。

```
QVTable:
SELECT * from Table1;
```

```
left join SELECT * from Table2;
```

### QVTable

A	B	C
1	aa	xx
2	cc	—
3	ee	—

*Left Join 示例*

如果我们执行 **Left Keep**，则会获得两个表格。这两个表格通过通过字段 A 关联。

```
QVTab1:
```

```
SELECT * from Table1;
```

```
QVTab2:
```

```
left keep SELECT * from Table2;
```

### QVTab1

A	B
1	aa
2	cc
3	ee

### QVTab2

A	C
1	xx

*Left Keep 示例*

## Right

数据加载脚本中的 **Join** 和 **Keep** 前缀可以位于前缀 **right** 之后。

如果用于 **Join** 之前，说明两个表格之间的联接应为右联接。由此生成的表格所包含的两表格之间的组合仅带有第二个表格的完整数据集。

如果用于 **Keep** 之前，说明首先应使第一个表格缩减为其与第二个表格间的共同交集，然后才可在 Qlik Sense 中存储此表格。

**示例：**

在这些示例中，我们使用源表格 Table1 和 Table2:

### Table1

A	B
1	aa
2	cc
3	ee

### Table2

A	C
1	xx
4	yy

*Right 示例源表格*

首先，我们在表格上执行 **Right Join**，从而产生 **QVTable**，其中包含 **Table2** 中的所有行 与 **Table1** 中匹配航的字段合并。

```
QVTable:
SELECT * from Table1;
right join SELECT * from Table2;
```

**QVTable**

A	B	C
1	aa	xx
4	-	yy

*Right Join 示例*

如果我们执行 **Left Keep**，则会获得两个表格。这两个表格通过通过字段 **A** 关联。

```
QVTab1:
SELECT * from Table1;
QVTab2:
right keep SELECT * from Table2;
```

**QVTab1**

A	B
1	aa

**QVTab2**

A	C
1	xx
4	yy

*Right Keep 示例*

## 8.4 将映射用作联接的替代

**Join** 前缀在 **Qlik Sense** 中是一种可在数据模型中合并多个数据表格的强大方式。其中一个缺点是合并的表格可以变得很大，且可能会导致出现性能问题。在使用 **Join** 的替代的情况下，需要使用映射从另一个表格中查找单个值。这可以节省加载导致计算速度缓慢和可能会造成计算错误的 unnecessary 的数据，因为联接可以更改表格中的记录数。

映射表包含两列：比较字段 输入 和映射值字段 输出。

在本例中，我们使用订单表格 **Orders**，并且需要知道客户所在的国家/地区，此信息存储在客户表格中 **Customers**。

*Orders 数据表*

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 数据表

CustomerID	Name	Country	...
1	DataSales	Spain	...
2	BusinessCorp	Italy	...
3	TechCo	Germany	...
4	Mobecho	France	...

要查找客户所在的国家/地区 Country，我们需要映射表格，如下所示：

CustomerID	Country
1	Spain
2	Italy
3	Germany
4	France

我们称为 MapCustomerIDtoCountry 的映射表已在脚本中定义，如下所示：

```
MapCustomerIDtoCountry:
Mapping LOAD CustomerID, Country From Customers ;
```

接下来是在加载订单表格时使用 **ApplyMap** 函数应用映射：

```
Orders:
S *,
    ApplyMap('MapCustomerIDtoCountry', CustomerID, null()) as Country
From Orders ;
```

**ApplyMap** 函数的第三个参数用于定义当在映射表中找不到值时要返回的内容，在本例中为 **Null()**。

最终生成的表格如下所示：

OrderID	OrderDate	ShipperID	Freight	CustomerID	Country
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

## 8.5 使用交叉表

交叉表是常见的表格类型，特点是在两个标题数据正交列表之间显示值矩阵。看上去会像以下表格。

### 示例 1:

ex1.xlsx

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

如果此表格只是简单加载到 Qlik Sense，结果将为 *Year* 使用一个字段，每个月份各一字段。通常，这并非您希望看到的结果。您可能更希望生成三个字段：一个用于每一标题类别 *Year* 和 *Month*，一个用于矩阵内的数据值。

通过将 **crosstable** 前缀添加到 **LOAD** 或 **SELECT** 语句可实现此操作。

```
crosstable (Month, Sales) LOAD * from ex1.xlsx;
```

这可以在 Qlik Sense 中创建下列结果：

Year	Month	Sales
2008	Apr	12
2008	Feb	65
2008	Jan	45
2008	Jun	22
2008	Mar	78
2008	May	78
2009	Apr	22
2009	Feb	23
2009	Jan	11
2009	Jun	85
2009	Mar	22

交叉表通常位于许多限定列之后，这些列将被直接读取。在此例中，将会出现一个限定列，即 *Year*：

### 示例 2:

ex2.xlsx

Salesman	Year	Jan	Feb	Mar	Apr	May	Jun
----------	------	-----	-----	-----	-----	-----	-----

A	2008	45	65	78	12	78	22
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

在此例中，左边有两个限定列，紧跟矩阵列之后。限定列的数量可表述为 **crosstable** 前缀的第三个参数，如下所述：

```
crosstable (Month, Sales, 2) LOAD * from ex2.xlsx;
```

这可以在 Qlik Sense 中创建下列结果：

Salesman ▲	Year	Month	Sales
A	2008	Apr	12
A	2008	Feb	65
A	2008	Jan	45
A	2008	Jun	22
A	2008	Mar	78
A	2008	May	78
A	2009	Apr	22
A	2009	Feb	23
A	2009	Jan	11
A	2009	Jun	85
A	2009	Mar	??

## 8.6 通用数据库

通用数据库是一个表格，其中字段名作为字段值储存在一列中，而字段值储存在第二列。通用数据库通常用于不同对象的属性。

看看以下 **GenericTable** 示例。这是一个包含两个对象，一个球和一个框的通用数据库。显然，某些属性 如颜色和线宽 是两个对象的常见属性，而其他属性 如直径、高度、长度和宽度 并不常见。

*GenericTable*

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

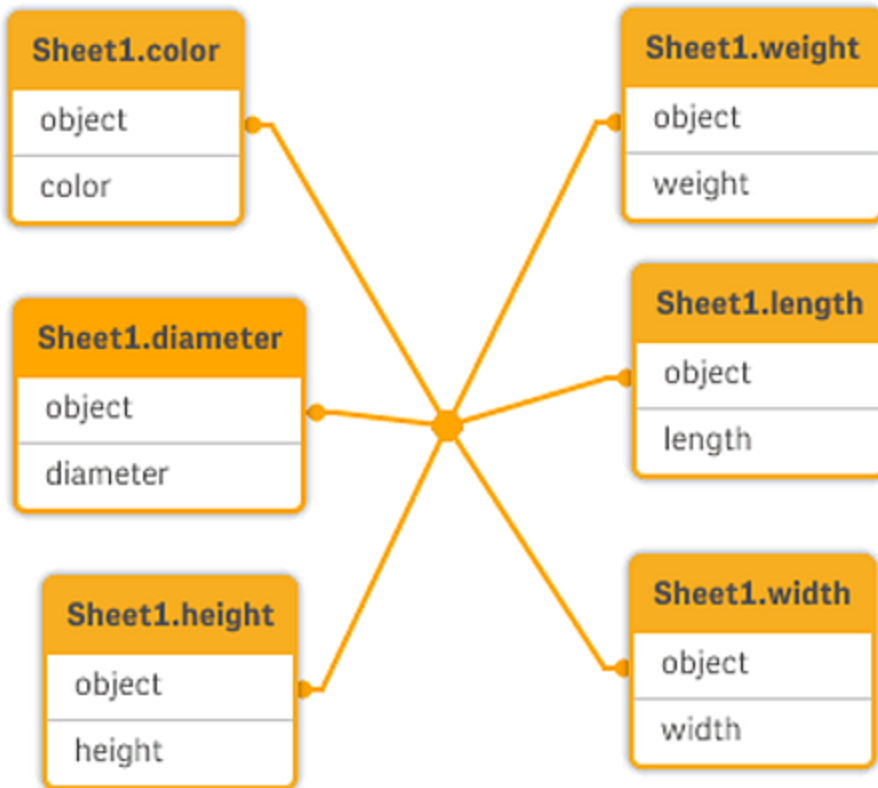
一方面，难以存储用于给出各列本身属性的数据，因为许多属性并不与特定对象相关联。

另一方面，混合长度，颜色和线宽显示，这会看上去显得杂乱。

如果使用标准方式将此数据库加载到 Qlik Sense，并在表格中显示数据，则如下所示：

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

但是，如果作为通用数据库加载此表格，会将第二列和第三列拆分到不同的表格中，然后为第二列的每一个单值生成一个字段：



此操作的语法很简单：

示例：

```
Generic SELECT* from GenericTable;
```

无论是使用 **LOAD** 还是 **SELECT** 语句加载通用数据库，这都无关紧要。

## 8.7 离散数据的匹配间隔

**LOAD** 或 **SELECT** 语句的 **intervalmatch** 前缀用于链接离散数值和一个或多个时间间隔数字。此功能十分强大，例如，其可用于生产环境，如下例所示。

示例：

仔细察看以下两个表格。第一个表格显示了不同订单生产的开始时间和结束时间。第二个表格显示了几件离散事件。我们如何才能将离散事件和这些订单相关联，以便我们了解，例如找出哪些订单受到干扰的影响，哪些订单依据哪次轮班处理？

*Table OrderLog*

Start	End	Order
01:00	03:35	A
02:30	07:58	B
03:04	10:27	C
07:23	11:43	D

Table EventLog

Time	Event	Comment
00:00	0	Start of shift 1
01:18	1	Line stop
02:23	2	Line restart 50%
04:15	3	Line speed 100%
08:00	4	Start of shift 2
11:43	5	End of production

通常首先加载两个表格，然后将字段 *Time* 链接到通过 *Start* 和 *End* 字段定义的时间间隔：

```
SELECT * from OrderLog;
SELECT * from EventLog;
Intervalmatch (Time) SELECT Start,End from OrderLog;
```

现在，可以在 Qlik Sense 中创建表格，如下所示：

Time	Event	Comment	Order	Start	End
0:00	0	Start of shift 1	-	-	-
1:18	1	Line stop	A	1:00	3:35
2:23	2	Line restart 50%	A	1:00	3:35
4:15	3	Line speed 100%	B	2:30	7:58
4:15	3	Line speed 100%	C	3:04	10:...
8:00	4	Start of shift 2	C	3:04	10:...
8:00	4	Start of shift 2	D	7:23	11:...
11:43	5	End of production	D	7:23	11:...

我们现在可以很容易看到订单 *A* 主要受生产线停止影响，但减缓生产线速度也可能会影响订单 *B* 和 *C*。*Shift 2* 只能处理订单 *C* 和 *D* 的一部分。

请在使用 **intervalmatch** 语句时务必注意以下几点：

- 在使用 **intervalmatch** 语句之前，必须已将包含离散数据点 上例中的 *Time* 的字段读取到 Qlik Sense。**intervalmatch** 语句不会从数据库表格中读取此字段！
- 读入 **intervalmatch LOAD** 或 **SELECT** 语句的表格必须始终包含两个完整字段 上例中的 *Start* 和 *End*。要建立指向其他字段的链接，您必须读取时间间隔字段以及位于单独的 **LOAD** 或 **SELECT** 语句中的字段 上例中第一个 **SELECT** 语句。

- 时间间隔始终是封闭区间，即间隔的端点包括在时间间隔之中。非数值限制会导致省略时间间隔 未定义 ，同时 NULL 限制会无限扩展时间间隔 无限 。
- 时间间隔可以重叠，离散值可以链接所有匹配的时间间隔。

## 使用扩展的 **intervalmatch** 语法解决维度变化缓慢问题

扩展的 **intervalmatch** 语法可以用于处理源数据中著名的维度变化缓慢的问题。

样本脚本：

```
SET NullInterpret='';

IntervalTable:
LOAD Key, ValidFrom, Team
FROM 'lib://dataqv/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;
nullinterpret 语句仅在从表格文件中读取数据时才需要，因为缺失值被定义为空白字符串而不是 NULL 值。
```

从 *IntervalTable* 加载数据会生成以下表格：

Key	FirstDate	Team
000110	2011-01-21	Southwest
000110		Northwest
000120		Northwest
000120	2013-03-05	Southwest
000120	2013-03-05	Northwest
000120	2013-01-06	Southwest

**nullasvalue** 语句允许将 NULL 映射到所列出的字段。

通过使用 **previous** 和 **order by** 创建 *Key*、*FirstDate* 和 *LastDate* 属性字段，然后使用此关键字表格替换已删除的 *IntervalTable*。

从 *Transact* 加载数据会生成以下表格：

Key	Name	Date	Sales
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** 语句前面的 **intervalmatch** 语句将使用链接 *Transact* 表格的关键组合字段替代上述关键字段，生成以下表格：

Key	Team	Name	FirstDate	LastDate	Date	Sales
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

## 8.8 层次结构

此外，不平衡的  $n$  级层次结构通常用于呈现数据中的地理维度或组织维度。通常，这些层次结构类型存储在相邻节点表格，即在一个表格中，其中每一条记录对应一个节点并具有包含父节点引用的字段。

NodeID	ParentNodeID	Title
1		- General manager
2	1	Region manager
3	2	Branch manager
4	3	Department manager

此类表格内的节点存储在一个记录上，但其仍拥有数量任意的子节点。表格可能包含更多描述节点属性的字段。

相邻节点表格可用于维护，但在日常工作中很难使用。相反，在查询和分析中会使用其他呈现形式。扩展节点表格是一种常见的呈现形式，其中的所有层次结构级别均存储于单独的字段内。扩展节点表格层级可轻松用于 例如 树形结构。**hierarchy** 关键字可用于数据加载脚本，以将相邻节点表格转换为扩展节点表格。

示例：

```
Hierarchy (NodeID, ParentNodeID, Title, 'Manager') LOAD
  NodeID,
  ParentNodeID,
  Title
FROM 'lib://data/hierarchy.txt' (txt, codepage is 1252, embedded labels, delimiter is ',', msq);
```

NodeID	ParentNodeID	Title	Title1	Title2	Title3	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

扩展节点表格有一个问题，就是很难将级别字段用于搜索或选择，因为需要预备哪些级别需要搜索或选择的相关知识。上级表格是解决此问题的不同呈现形式。这种呈现形式又被称之为桥接表格。

上级表格包含一条记录，其用于显示数据中的层次结构上下级关系。此表格包含子节点以及祖先节点的关键字和名称，即每条记录均描述特定节点属于哪个节点。**hierarchybelongsto** 关键字可用于数据加载脚本，以将相邻节点表格转换为上级表格。

## 8.9 货币符号扩展

货币符号扩展可定义脚本或表达式运用的文本替换。这一过程称为扩展，即使新文档其实更短。替换发生在验证脚本语句或表达式之前。严格说来，其实就是宏扩展。

宏扩展总是以“\$(开始”，以)”结束，括号里的内容定义文本替换将如何完成。不要混淆脚本宏和下面提到的作为货币符号扩展的宏扩展。

货币符号扩展可与以下任何一个对象一起使用：

- 变量
- 参数
- 表达式



货币符号扩展受其可以计算的表达式的数量限制。无法计算任何超过 1000 的表达式。

### 使用变量进行货币符号扩展

如果要在脚本或表达式中使用变量进行文本替换，需使用以下语法：

```
$ (variablename)
```

`$(variablename)` 扩展至变量中的值。如果 `variablename` 不存在，那么该扩展将为空字符串。

对于数字变量扩展，使用以下语法：

```
$ (#variablename)
```

其总是生成合法小数点来表示变量的数字值，可能带有指数符号 对于非常大/小的数值 。如果 `variablename` 不存在或不包含数字值，那宏扩展将被扩展至 `0`。

示例：

```
SET DecimalSep=',';
LET X = 7/2;
货币符号扩展 $(X) 将扩展至 3,5，而 $(#X) 将扩展至 3.5。
```

示例：

```
Set MyPath=C:\MyDocs\Files\;
...
LOAD * from $(MyPath)abc.csv;
数据将从 C:\MyDocs\Files\abc.csv 加载。
```

示例：

```
Set CurrentYear=1992;
...
SQL SELECT * FROM table1 WHERE Year=$(CurrentYear);
将选中 Year=1992 的行。
```

示例：

```
Set vConcatenate = ;
For each vFile in FileList('.\*.txt')
  Data:
  $(vConcatenate)
  LOAD * FROM [$(vFile)];
  Set vConcatenate = Concatenate ;
Next vFile
```

在本例中，目录中的所有 `.txt` 文件将使用 **Concatenate** 前缀加载。如果字段略有不同 在这种情况下自动串联不起作用，则可能必须采用这种方法。`vConcatenate` 变量初始设置为空字符串，因为 **Concatenate** 前缀在第一次加载中无法使用。如果目录包含三个名为 `file1.txt`、`file2.txt` 和 `file3.txt` 的文件，**LOAD** 语句将在三次迭代过程中扩展至：

```
LOAD * FROM[.\file1.txt];
Concatenate LOAD * FROM[.\file2.txt];
Concatenate LOAD * FROM[.\file3.txt];
```

### 使用参数进行货币符号扩展

参数可用于货币符号扩展。该变量必须包含形式参数，如 `$1`、`$2`、`$3` 等。当扩展变量时，参数应在以逗号分隔的列表中显示。

示例：

```
Set MUL='$1*$2';  
Set X=$(MUL(3,7)); // 返回“3*7” 在 X 中  
  
Let X=$(MUL(3,7)); // 返回 21 在 X 中
```

如果形式参数的数量超过实际参数数量，只有与实际参数相对应的格式参数才会被扩展。如果实际参数超过形式参数，多余的实际参数将会被忽略。

示例：

```
Set MUL='$1*$2';  
Set X=$(MUL); // 返回“$1*$2” 在 X 中  
  
Set X=$(MUL(10)); // 返回“10*$2” 在 X 中  
  
Let X=$(MUL(5,7,8)); // 返回 35 在 X 中
```

参数 \$0 返回通过实际调用的参数的数量。

示例：

```
set MUL='$1*$2 $0 par';  
set X=$(MUL(3,7)); // 返回“3*7 2 par” 在 X 中
```

### 使用表达式进行货币符号扩展

表达式可用于货币符号扩展。括号内的内容必须以等号开头：

```
$(=expression )
```

表达式将被验证，数值也将用于表达式。

示例：

```
$(=Year(Today())); // 返回包含当年年份的字符串。  
  
$(=Only(Year)-1); // 返回所选年份之前的一年。
```

包括文件

使用货币符号扩展包括文件。相应语法为：

```
$(include=filename )
```

上述文本会替换为等号之后指定的文件的内容。此功能在文本文件中存储脚本或部分脚本时极为有用。

示例：

```
$(include=C:\Documents\MyScript.qvs);
```

### 8.10 在脚本中使用引号

可以通过多种不同方法在脚本语句中使用引号。



## 在 LOAD 语句内

在 **LOAD** 语句内，以下符号应用作引号：

	说明	符号	ASCII	示例
字段名称	双引号	" "	34	"字符串"
	方括号	[ ]	91, 93	[字符串]
	重音符号	``	96	`字符串`
字符串文字	单引号	' '	39	'字符串'

## 在 SELECT 语句中

对于 **SELECT** 语句 由 ODBC 驱动程序解释，情况或许稍有不同。通常，对于字段名和表格名应使用直双引号 (Alt + 0034)，对于文字应使用直单引号 (Alt + 0039)，并避免使用重音符号。但是，一些 ODBC 驱动程序不仅接受重音符号，而且首选引号。在这种情况下，生成的 **SELECT** 语句包含重音引号。

### Microsoft Access 引号示例

Microsoft Access ODBC Driver 3.4 包括在 Microsoft Access 7.0 中 在分析 **SELECT** 语句时接受以下引号：

字段名和表格名： [ ] " " ``  
字符串文字： ' '

其他数据库可能具有不同的约定。

## 在 LOAD 语句外

在 **LOAD** 语句外，Qlik Sense 期望在表达式中使用双引号表示变量引用，而不是字段引用。如果使用双引号，则应将括起来的字符串解释为变量和将要使用的变量的值。

## 上下文外字段引用和表格引用

一些脚本函数引用已创建的字段，或者 **LOAD** 语句输出中的字段，如 **Exists()** 和 **Peek()**。这些字段引用被称为上下文外字段引用，与引用上下文中的字段的源字段引用相对，即 **LOAD** 语句输入表中的字段。

应将上下文外字段引用和表格引用看作是文字，因此需要使用单引号。

## 名称和文字之间的差异

比较以下示例，名称和文字之间的差异会变得更加清晰：

示例：

```
'Sweden' as Country
```

如果在 **LOAD** 或 **SELECT** 语句中将此表达式用作字段列表的一部分，则会将文本字符串“Sweden”作为字段值加载到 Qlik Sense 字段“Country”。

示例：

### "land" as Country

如果在 **LOAD** 或 **SELECT** 语句中将此表达式用作字段列表的一部分，则会将数据库字段的内容和名为“land”的表格列作为字段值加载到 Qlik Sense 字段“Country”。这意味着会将 *land* 视为字段引用。

## 数值和字符串文字之间的差异

比较以下示例，数值和字符串文字之间的差异会变得更加清晰：

示例：

'12/31/96'

如果将此字符串用作表达式的一部分，则会在第一步中将其解释为文本字符串“12/31/96”，这反过来也可以将其解释为日期，如果日期格式为“MM/DD/YY”。在这种情况下，会将它存储作为同时包含数字和文字表示形式的双值。

示例：

12/31/96

如果将此字符串用作表达式的一部分，则会将它解释为 12 除以 31 除以 96 的数字。

## 8.11 数据清理

当从不同表格加载数据时，要注意表示相同事物的字段值并不总是拥有一致的名称。由于缺乏一致性不仅让人烦恼，而且还会阻碍关联性，因而此问题必须得到解决。这一问题通过创建映射表比较字段值可以得到完美解决。

### 映射表

通过 **mapping load** 或 **mapping select** 加载的表格的处理方式不同于其他表格。它们存储在内存的单独区域内，并在脚本执行期间仅用作映像表。脚本执行完后，它们会自动被丢弃。

规则：

- 映像表必须拥有列，第一列包含比较值，第二列包含所需的映射值。
- 两列必须命名，且名称之间不存在关联。列名称不得与常规内部表格的字段名称相联系。

### 使用映射表

当加载几个列有国家/地区的表格时，您会发现，同一个国家/地区具有几个不同的名称。在此示例中，U.S.A. 列为 US、U.S. 和 United States。

为了避免在串联表中以三种不同的记录表示 United States，可以创建一个与所示表格类似的表格，然后将其作为映像表加载。

整个脚本外观如下：

```
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 语句加载文件 MappingTable.txt 作为包含标签 CountryMap 的映射表。
```

**map** 语句使用之前加载的映射表 *Country* 启用字段 *CountryMap* 的映射。

**LOAD** 语句加载表格 *CountryA* 和 *CountryB*。这些表格由于具有相同的字段设置将会串联起来 包括字段 *Country*，该字段的字段值用来与映射表的第一列进行比较。然后，您会发现字段值 *US*、*U.S.* 和 *United States* 将被映射表第二列的值所替换，即 *USA*。

在生成由 Qlik Sense 表格存储的字段的事件链中，自动映射是最后环节。对于典型的 **LOAD** 或 **SELECT** 语句，事件顺序大致如下所示：

1. 表达式评估
2. 用 **as** 重新命名字段
3. 用 **alias** 重新命名字段
4. 表格名限定性条件 如果适用。
5. 如果字段名匹配，则映射数据

这意味着并非每次遇到作为表达式组成部分的字段名时都会执行映射，而是在当值存储在 Qlik Sense 表格中的字段名之下时才执行映射。

要禁用映射，可以使用 **unmap** 语句。

对于表达式级别映射，可以使用 **applymap** 函数。

对于子字符串级别映射，可以使用 **mapsubstring** 函数。

### 8.12 数据中的通配符

也可能在数据中使用通配符。存在两种不同的通配符：一个是星号，其被看作表示该字段的所有值；以及可选符号，其被看作表示该字段的所有剩余值。

#### 星号

星号，被解释为字段中所有 列出的 值，即表格中的值列表。如果用于脚本存取部分加载的表格中的系统字段之一 *USERID*、*PASSWORD*、*NTNAME* 或 *SERIAL*，则星号将被视为此字段的所有 也包括未列出的 可能值。

星号不可用于信息文件。此外，星号还不能用于关键字段，即用于连接表格的字段。

除非是显式指定，否则无星号可用。

## OtherSymbol

在许多情况下，都需要一种在表格中呈现所有其他值的方法，即呈现未在加载数据中显式出现的全部值的方法。为此，可使用称为 **OtherSymbol** 的特殊变量。要将 **OtherSymbol** 定义为被视为“所有其他值”，请使用以下语法：

```
SET OTHERSYMBOL=<sym>;
```

在 **LOAD** 或 **SELECT** 语句之前。<sym> 可以是任意字符串。

内部表格中出现定义符号时，将导致 Qlik Sense 将此符号定义为之前未加载到包含它的字段中的所有值。因此，出现 **OtherSymbol** 后字段中的值将被忽略。

为了重置此功能，可以使用：

```
SET OTHERSYMBOL=;
```

示例：

*Table Customers*

CustomerID	Name
1	ABC Inc.
2	XYZ Inc.
3	ACME INC
+	Undefined

*Table Orders*

CustomerID	Name
1	1234
3	1243
5	1248
7	1299

在脚本上述第一个载入的表格之前插入以下语句：

```
SET OTHERSYMBOL=+;
```

例如，当点击 *OrderID 1299* 时，除 1、2 或 3 以外的所有 *CustomerID* 引用都将导致在 *Name* 下显示 *Undefined*。



**OtherSymbol** 不可用于创建表格之间的外部连接。

## 8.13 NULL 值处理

由于数据库查询和/或表格之间的连接，当某个字段无数据产生时，结果通常为 NULL 值。

Qlik Sense 逻辑将以下值视为真正的 NULL 值：

- 从 ODBC 连接返回的 NULL 值
- 由于在数据加载脚本中强制表格串联而创建的 NULL 值
- 由于在数据加载脚本中进行联接而创建的 NULL 值
- 由于生成在表格中显示的字段值组合而创建的 NULL 值



通常可以将这些 NULL 值用于关联和选择内容，但使用 **NullAsValue** 语句时除外。

文本文件根据定义不能包含 NULL 值。

### 从 ODBC 关联/选择空值

从 ODBC 数据源可以关联和/或选择空。为此，可用已定义的脚本。可以使用以下语法：

```
SET NULLDISPLAY=<sym>;
```

<sym> 符号将替代来自数据输入最低级别上 NULL 数据源的全部 ODBC 值。<sym> 可以是任意字符串。

要重置此功能到默认解释，请使用以下语法：

```
SET NULLDISPLAY=;
```



使用 **NULLDISPLAY** 仅会影响来自 ODBC 数据源中的数据。

如果您希望将 Qlik Sense 逻辑将 NULL 连接返回的 ODBC 值解释为一个空白字符串，请在任意 **SELECT** 语句之前添加以下内容到脚本：

```
SET NULLDISPLAY="";
```



这里的 " 实际上是两个单引号，中间没有任何内容。

### 从文本文件创建空值

可以定义一个符号，令其在文本文件或 **inline** 子句中出现时可被解释为真实空值。使用以下语句：

```
SET NULLINTERPRET=<sym>;
```

这里 <sym> 符号将被解释为 NULL 值 <sym> 可以是任意字符串。

要重置此功能到默认解释，请使用：

```
SET NULLINTERPRET=;
```



使用 **NULLINTERPRET** 仅可影响来自文本文件和内联子句的数据！

## 表达式中 NULL 值的传送

NULL 值将根据一些合乎逻辑并非常合理的规则借助表达式传送。

### 函数

一般规则是当参数处于函数所定义的范围之外，则函数返回 NULL 值。

示例：

```
asin(2)      返回  NULL
log(-5)     返回  NULL
round(A,0)  返回  NULL
```

因此看出，以上示例遵循的规则是任何一个评估所必需的参数为 NULL 值，函数通常返回 NULL 值。

示例：

```
sin(NULL)    返回  NULL
chr(NULL)    返回  NULL
if(NULL, A, B) 返回  NULL
if(True, NULL, A) 返回  NULL
if(True, A, NULL) 返回  A
```

第二条规则的例外情况是逻辑函数类型测试。

示例：

```
isnull(NULL) 返回  True (-1)
isnum(NULL)   返回  False (0)
```

### 算法和字符串运算符

如果在这些运算符的任何一边遇到 NULL 值，则返回 NULL 值。

示例：

A	+	NULL	returns	NULL
A	-	NULL	返回	NULL
A	/	NULL	返回	NULL
A	*	NULL	返回	NULL
NULL	/	A	返回	NULL
0	/	NULL	返回	NULL
0	*	NULL	返回	NULL
A	&	NULL	返回	A

### 关系运算符

如果在这些运算符的任何一边遇到 NULL 值，则系统运用特殊规则。

#### 示例：

NULL	rel.op	NULL	返回	NULL
A	<>	NULL	返回	True (-1)
A	<	NULL	返回	False (0)
A	<=	NULL	返回	False (0)
A	=	NULL	返回	False (0)
A	>=	NULL	返回	False (0)
A	>	NULL	返回	False (0)