



Verwalten von Daten

Qlik Sense®

2.2

Copyright © 1993-2016 QlikTech International AB. Alle Rechte vorbehalten.



Copyright © 1993-2016 QlikTech International AB. Alle Rechte vorbehalten.

Qlik®, QlikTech®, Qlik Sense®, QlikView®, Sense® und das Qlik-Logo sind in mehreren Ländern eingetragene Marken von QlikTech International AB oder werden in anderer Weise von QlikTech International AB als Marken verwendet. Weitere hier genannte Marken sind Marken ihrer jeweiligen Eigentümer.

1	Über dieses Dokument	11
2	Verwalten von Daten	12
3	Verwaltung von Daten in der App mit dem Datenmanager	13
3.1	Hinzufügen einer neuen Datentabelle	13
3.2	Bearbeiten einer Datentabelle	13
3.3	Löschen einer Datentabelle	14
3.4	Verwalten von Datentabellenverknüpfungen	14
3.5	Änderungen anwenden und Daten neu laden	14
3.6	Vorgänge in Datenmanager rückgängig machen und wiederherstellen	15
3.7	Zusammengefasste Tabellen	15
	Trennen einer zusammengefassten Tabelle	16
3.8	Interaktion zwischen Datenmanager und Datenladeskript	16
3.9	Hinzufügen von Daten aus Dateien und Datenbanken	17
	Hinzufügen von Daten aus einer vorhandenen Datenquelle	17
	Hinzufügen von Daten aus einer neuen Datenquelle	18
	Anhängen von Datendateien	18
	Anhängen einer neuen Datei	18
	Löschen einer angehängten Datei	19
	Aktualisieren einer angehängten Datei	19
	Synthetische Schlüssel	20
	Welche Datenquellen sind für mich verfügbar?	20
3.10	Auswählen von Datenfeldern aus Dateien und Datenbanken	21
	Auswählen von Daten aus einer Datenbank	21
	Auswählen von Daten aus einer Microsoft Excel-Tabelle	22
	Auswählen von Daten aus einer Tabelle	23
	Auswählen von Einstellungen für Dateitypen	24
	Tabellen mit Trennzeichen	24
	Dateiformateinstellungen	24
	Datendateien mit fester Datensatzlänge	25
	Positionen der Feldumbrüche festlegen	25
	Dateiformateinstellungen	25
	HTML-Dateien	25
	Dateiformateinstellungen	25
	XML-Dateien	25
	QVD-Dateien	26
	QVX-Dateien	26
	KML-Dateien	26
	Zurück zum vorherigen Schritt (Daten hinzufügen)	26
3.11	Auswählen von Daten aus Qlik DataMarket	26
	Anwenden einer Lizenz auf Qlik DataMarket	27
	Hinzufügen von Daten aus Qlik DataMarket	28
	Auswählen von Qlik DataMarket-Dimensionen, Kennzahlen und Zeiträumen	29
	Auswählen von hierarchischen Daten aus Qlik DataMarket	29
	Auswählen von übergeordneten und untergeordneten Zweigen	30

Auswählen von übergeordneten Zweigen, die keine Daten enthalten	31
Auswählen von aggregierten Daten aus Qlik DataMarket	33
3.12 Bearbeiten einer Tabelle	34
Umbenennen von Tabellen	34
Felder umbenennen	35
Feldtyp und Anzeigeformat ändern	35
Sicherstellen, dass ein Datum- oder Timestamp-Feld richtig erkannt wird	36
Das Anzeigeformat von Datum- und Timestamp-Feldern verändern	36
Tabellen aus der Datenquelle aktualisieren	36
Ein berechnetes Feld hinzufügen	37
Aktualisierung eines berechneten Feldes	37
Welche Funktionen kann ich in einer berechneten Feldformel verwenden?	38
Stringfunktionen für die Verwendung in einer berechneten Feldformel	38
Datumfunktionen für die Verwendung in einer berechneten Feldformel	39
Formatierungs- und Interpretationsfunktionen für die Verwendung in einer berechneten Feldformel	41
Rückgängigmachen und Wiederherstellen	42
3.13 Verwalten von Datentabellenverknüpfungen	43
Anzeigen von Empfehlungen	43
Zwei Datenquellen enthalten ein Feld mit verwandten Daten, aber unterschiedlichen Namen	44
Zwei Datenquellen enthalten Felder mit demselben Namen, aber nicht zusammenhängenden Daten	44
Auswählen einer empfohlenen Verknüpfung	45
Auflösen der Verknüpfung	45
Zwei Tabellen enthalten mehr als ein gemeinsames Feld	45
Rückgängigmachen und Wiederherstellen	45
Beschränkungen	46
4 Laden von Daten mit dem Datenladeskript	47
4.1 Interaktion zwischen Datenmanager und dem Datenladeskript	47
4.2 Arbeiten mit dem Dateneditor	48
Schnellstart	49
Symbolleisten	49
Hauptsymbolleiste	50
Editor-Symbolleiste	50
Verbinden mit Datenquellen	51
Datenverbindungsarten	51
Erstellen einer neuen Datenverbindung	51
Datenverbindungseinstellungen	52
Löschen einer Datenverbindung	54
Bearbeiten einer Datenverbindung	55
Einfügen eines Verbindungsstrings	55
Auswählen von Daten aus einer Datenverbindung	55
Verweisen auf eine Datenverbindung im Skript	56
Wo ist die Datenverbindung gespeichert?	56

Laden von Daten aus Dateien	57
Vorbereitung von Excel-Dateien für das Laden mit Qlik Sense	57
Daten aus Datenbanken laden	58
Datensätze aus einer ODBC-Datenbank einlesen	58
ODBC	59
Verschieben von Apps mit ODBC-Datenverbindungen	60
Sicherheitsaspekte beim Verbinden mit dateibasierten ODBC-Datenverbindungen	60
OLE DB	61
Logik in Datenbanken	61
Auswählen von Daten im Dateneditor	61
Auswählen von Daten aus einer Datenbank	61
Auswählen von Daten aus einer Microsoft Excel-Tabelle	62
Auswählen von Daten aus einer Tabelle	63
Auswählen von Einstellungen für Dateitypen	64
Anzeigen einer Skriptvorschau	67
Einbinden von LOAD-Befehlen	67
Bearbeiten des Datenladeskripts	67
Zugriff auf die Syntaxhilfe für Befehle und Funktionen	67
Zugriff auf das Hilfe-Portal	67
Verwenden der Auto-Vervollständigungsfunktion	68
Verwenden der Popup-Hilfe für Funktionen	68
Einfügen eines vorbereiteten Test-Skripts	68
Einrücken von Code	68
Text suchen und ersetzen	68
Text suchen	68
Text ersetzen	69
Kommentieren im Skript	69
Auskommentieren	70
Einkommentieren	70
Entsperren eines automatisch generierten Abschnitts	70
Gesamten Code auswählen	71
Gliedern des Skriptcodes	71
Erstellen eines neuen Skriptabschnitts	71
Löschen eines Skriptabschnitts	71
Umbenennen eines Skriptabschnitts	72
Reihenfolge der Skriptabschnitte ändern	72
Debuggen des Datenladeskripts	72
Debug-Symboleiste	72
Ausgabe	73
Variablen	73
Festlegen einer Variable als Favorit	73
Filtern von Variablen	74
Haltepunkte	74
Hinzufügen eines Haltepunkts	74
Löschen von Haltepunkten	74
Aktivieren und Deaktivieren von Haltepunkten	75

Speichern des Ladeskripts	75
Ausführen des Skripts zum Laden der Daten	75
4.3 Verständnis der Skript-Syntax und der Datenstrukturen	76
Extrahieren, Umformen und Laden	76
Befehle zum Laden von Daten	77
Regeln	77
Die Ausführung des Skripts	77
Felder	78
Abgeleitete Felder	78
Definieren der Kalenderfelddefinitionen	78
Zuordnen von Datenfeldern zum Kalender mithilfe von Derive	79
Verwenden der abgeleiteten Datumsfelder in einer Visualisierung	79
Tags für Felder	79
Systemfelder	79
Verfügbare Systemfelder	80
Umbenennen von Feldern	81
Interne Tabellen	82
Tabellennamen	82
Tabellenbezeichnungen	83
Verknüpfungen zwischen Tabellen	84
Qlik Sense-Verknüpfung im Vergleich zu SQL Natural Outer Join	84
Häufigkeiten in Schlüsselfeldern	85
Problemumgehung	85
Synthetische Schlüssel	85
Handhabung synthetischer Schlüssel	86
Zirkelbezüge und Auflösen von Zirkelbezügen	87
Auflösen von Zirkelbezügen	88
Zusammenfassung von Tabellen	89
Automatische Zusammenfassung	89
Manuelle Zusammenfassung	90
Zusammenfassung von Tabellen verhindern	90
Laden von Daten aus bereits geladenen Tabellen	90
Resident oder vorangehender LOAD-Befehl?	91
Resident LOAD	91
Vorangehender Load-Befehl	92
Datentypen in Qlik Sense	93
Speicherung in Qlik Sense	93
Interpretation von Zahlen	93
Daten mit Informationen zum Datentyp	93
Daten ohne Informationen zum Datentyp	94
Interpretation von Datum und Uhrzeit	95
Aufrufen von Variablen	97
Textvariablen und numerische Variablen	97
Aufrufen von Variablen mit Parametern	98
Aufrufen von Variablen mit Formeln	99
Verwenden von Anführungszeichen im Skript	99

In LOAD-Befehlen	99
In SELECT-Befehlen	100
Microsoft Access – Beispiel für Anführungszeichen	100
Außerhalb von LOAD-Befehlen	100
Feldreferenzen und Tabellenreferenzen ohne Kontext	100
Unterschied zwischen Namen und Literalen	100
Unterschied zwischen Zahlen und String-Literalen	101
Verwenden von einfachen Anführungszeichen innerhalb eines Strings	101
Wildcards in den Daten	101
Stern-Symbol	101
OtherSymbol	102
NULL-Werte im Programm	103
Überblick	103
NULL-Werte aus ODBC-Datenquellen	103
Erstellen von NULL-Werten aus Textdateien	103
NULL-Werte in Formeln	104
Funktionen	104
Arithmetische Operatoren und String-Operatoren	105
Relationale Operatoren	105
4.4 Richtlinien für Daten und Felder	105
Richtlinien für die Menge an geladenen Daten	105
Obergrenzen für Datentabellen und -felder	106
Empfohlene Grenzen für Abschnitte des Ladeskripts	106
Konventionen für Zahlen- und Uhrzeitformate	106
Zahlenformate	106
Spezielle Zahlenformate	107
Datumsangaben	108
Uhrzeiten	109
Zeitstempel	109
4.5 Arbeiten mit QVD-Dateien	109
Anwendung von QVD-Dateien	110
Erstellen von QVD-Dateien	110
Daten aus QVD-Dateien einlesen	111
QVD-Format	111
4.6 Sicherheitsverwaltung über Zugriffsabschnitt	111
Abschnitte im Skript	112
Sicherheitsfelder	112
Einschränkung des Datenbestands	114
Übertragung von Zugriffsbeschränkungen	115
4.7 Zugriff auf große Datensätze mit Direct Discovery	115
Unterschiede zwischen Direct Discovery und im Speicher befindlichen Daten	115
In-Memory-Modell	115
Direct Discovery	117
Leistungsunterschiede zwischen im Speicher befindlichen und Direct Discovery- Feldern	119

Unterschiede zwischen Daten im Speicher und Datenbankdaten	120
Caching und Direct Discovery	121
Feldtypen für Direct Discovery	122
DIMENSION-Felder	122
MEASURE-Felder	122
DETAIL-Felder	123
In Direct Discovery unterstützte Datenquellen	123
SAP	123
Google Big Query	123
MySQL und Microsoft Access	123
DB2, Oracle und PostgreSQL	124
Sybase und Microsoft SQL Server	124
Einschränkungen in der Verwendung von Direct Discovery	124
Unterstützte Datentypen	124
Datensicherheit	124
Qlik Sense-Funktion wird nicht unterstützt	125
Unterstützung mehrerer Tabellen in Direct Discovery	125
Verbinden von Direct Discovery-Tabellen mit einer Where-Bedingung	126
Verbinden von Direct Discovery-Tabellen mit Join On-Bedingungen	126
Verwenden von Unterabfragen in Direct Discovery	127
Szenario 1: niedrige Kardinalität	128
Szenario 2: Verwenden von Unterabfragen	129
Protokollieren von Zugriff auf Direct Discovery	130
5 Anzeigen des Datenmodells	131
5.1 Symbolleiste	131
5.2 Verschieben und Anpassen der Größe von Tabellen im Datenmodell	132
Verschieben von Tabellen	132
Anpassen der Größe von Tabellen	133
5.3 Vorschau von Tabellen und Feldern im Datenmodell	133
Vorschau einer Tabelle anzeigen	133
Vorschau eines Feldes anzeigen	134
5.4 Erstellen einer Master-Dimension aus dem Datenmodell	134
5.5 Erstellen einer Master-Kennzahl aus dem Datenmodell	135
6 Beste Praktiken für Datenmodelle	136
6.1 Umwandeln von Datenspalten in Zeilen	136
6.2 Umwandeln von Datenzeilen in Felder	136
6.3 Laden von Daten, die in hierarchischen Ebenen organisiert sind, z. B. in einem Organisationschema.	137
6.4 Laden von ausschließlich neuen oder aktualisierten Datensätzen aus einer großen Datenbank	138
6.5 Zusammenführen von Daten aus zwei Tabellen mit einem gemeinsamen Feld	138
6.6 Einordnung eines diskreten Werts in ein Intervall	138
6.7 Handhabung von inkonsistenten Feldwerten	139
6.8 Laden räumlicher Daten zur Visualisierung von Daten mit einer Karte	140

6.9 Laden von neuen und aktualisierten Datensätzen mit inkrementellem Laden	140
Nur angehängte Datensätze	141
Nur eingefügte Datensätze (keine Aktualisierung oder Löschung)	141
Neu eingefügte und geänderte Datensätze (keine Löschung)	141
Neu eingefügte, geänderte und gelöschte Datensätze	142
6.10 Kombinieren von Tabellen mit Join und Keep	143
Joins innerhalb eines SQL SELECT-Befehls	143
Join	144
Keep	144
Inner	145
Inner Join	145
Inner Keep	145
Left	146
Right	147
6.11 Mapping als Alternative zu Joining	148
6.12 Arbeiten mit Kreuztabellen	149
Pivotierung einer Kreuztabelle mit einer Spalte aufheben	149
Pivotierung einer Kreuztabelle mit zwei Spalten aufheben	150
6.13 Generische Datenbanken	151
6.14 Einordnung diskreter Werte in Intervalle	153
Intervalmatch-Beispiel	153
Behebung von Problemen mit sich langsam verändernden Dimensionen mit der erweiterten Syntax von intervalmatch	155
Beispiel:	155
6.15 Erstellen eines Datumsintervalls aus einem einzigen Datum	157
6.16 Hierarchiedaten laden	159
6.17 Laden von Kartendaten	160
Erstellen einer Karte aus Daten in einer KML-Datei	161
Erstellen einer Karte aus Punktdaten in einer Excel-Datei	162
Punktdatenformate	162
Anzahl der dargestellten Punkte	164
6.18 Datenpflege durch Mapping	164
Mapping-Tabellen	165
Es gilt:	165
Mit Mapping-Tabellen arbeiten	165
7 Fehlerbehebung – Laden von Daten	167
7.1 Datenverbindung funktioniert nicht mehr, nachdem SQL Server neu gestartet wird	167
7.2 Warnung zu synthetischen Schlüsseln beim Laden von Daten	167
7.3 Warnung zu Zirkelbezügen beim Laden von Daten	168
7.4 Datenauswahlprobleme mit einer OLE DB-Datenquelle	168
7.5 Zeichensatzprobleme mit Nicht-ANSI-codierten Datendateien	168
7.6 Benutzerdefinierter Konnektor funktioniert nicht	169
Der benutzerdefinierte Konnektor ist nicht ordnungsgemäß installiert	169
Der benutzerdefinierte Konnektor ist nicht für Qlik Sense angepasst	169

7.7 Das Datenladeskript wird ohne Fehler ausgeführt, die Daten werden aber nicht geladen	169
Ein Befehl ist nicht mit einem Semikolon abgeschlossen	170
Einfaches Anführungszeichen innerhalb eines Strings	170
7.8 Spalten werden bei Auswahl von Daten aus einer Datei mit festen Satzlängen nicht wie erwartet angeordnet	170
7.9 Fehlermeldung "Ungültiger Pfad" beim Anhängen einer Datei	171
7.10 Fehler beim Laden einer App, die aus einem QlikView-Dokument konvertiert wurde	171
Im Skript werden absolute Dateipfadverweise verwendet	171
Im Skript werden nicht unterstützte Funktionen oder Befehle verwendet	171
7.11 Probleme bei der Verbindungsherstellung mit Microsoft Excel-Dateien über ODBC oder dem Laden solcher Dateien.	172
7.12 Das Anhängen einer Datei durch Ablegen in Daten hinzufügen funktioniert nicht	172

1 Über dieses Dokument

Nachdem Sie eine Qlik Sense-App erstellt haben, besteht der erste Schritt darin, einige Daten hinzuzufügen, die Sie untersuchen und analysieren können. In diesem Dokument wird beschrieben, wie Daten hinzugefügt und verwaltet werden, wie ein Datenladeskript für erweiterte Datenmodelle erstellt wird und wie das resultierende Datenmodell in der Datenmodellansicht angezeigt wird. Außerdem werden beste Praktiken für die Erstellung von Datenmodellen in Qlik Sense erläutert.

Es gibt zwei Möglichkeiten zum Hinzufügen von Daten zur App.

- **Datenmanager**

Sie können Daten aus Ihren eigenen Datenquellen oder aus anderen Quellen wie Qlik DataMarket hinzufügen, ohne eine Skriptsprache lernen zu müssen. Datenauswahlen können bearbeitet werden und Sie können Unterstützung beim Erstellen von Datenverknüpfungen in Ihrem Datenmodell erhalten.

- **Dateneditor**

Sie können unter Verwendung der Qlik Sense -Datenladeskriptsprache ein Datenmodell mit ETL (Extract, Transform & Load)-Vorgängen erstellen. Die Skriptsprache ist leistungsfähig und ermöglicht es Ihnen, komplexe Umformungen vorzunehmen und ein skalierbares Datenmodell zu erstellen.



Detaillierte Referenzen zur Skript- und Tabellenfunktionen finden Sie in der Datei Skriptsyntax und Diagrammfunktionen.

Dieses Dokument wurde aus der Online-Hilfe für Qlik Sense erstellt. Es ermöglicht das Lesen der Hilfe im Offline-Modus und ein einfaches Ausdrucken von Seiten und enthält im Vergleich zur Online-Hilfe keine zusätzlichen Informationen.

Die Online-Hilfe, zusätzliche Guides und mehr finden Sie auf help.qlik.com/sense.

2 Verwalten von Daten

Nachdem Sie eine Qlik Sense-App erstellt haben, besteht der erste Schritt darin, einige Daten hinzuzufügen, die Sie untersuchen und analysieren können. In diesem Abschnitt wird beschrieben, wie Daten hinzugefügt und verwaltet werden, wie ein Datenladeskript für erweiterte Datenmodelle erstellt wird und wie das resultierende Datenmodell in der Datenmodellansicht angezeigt wird. Außerdem werden beste Praktiken für die Erstellung von Datenmodellen in Qlik Sense erläutert.

Es gibt zwei Möglichkeiten zum Hinzufügen von Daten zur App.

- **Datenmanager**

Sie können Daten aus Ihren eigenen Datenquellen oder aus anderen Quellen wie Qlik DataMarket hinzufügen, ohne eine Skriptsprache lernen zu müssen. Datenauswahlen können bearbeitet werden und Sie können Unterstützung beim Erstellen von Datenverknüpfungen in Ihrem Datenmodell erhalten.

- **Dateneditor**

Sie können unter Verwendung der Qlik Sense-Datenladeskriptsprache ein Datenmodell mit ETL (Extract, Transform & Load)-Vorgängen erstellen. Die Skriptsprache ist leistungsfähig und ermöglicht es Ihnen, komplexe Umformungen vorzunehmen und ein skalierbares Datenmodell zu erstellen.



*Sie können ein im **Datenmanager** erstelltes Datenmodell in ein Datenladeskript konvertieren, das im **Dateneditor** weiterentwickelt werden kann. Es ist jedoch nicht möglich, ein Datenladeskript in ein **Datenmanager**-Datenmodell zu konvertieren. Das **Datenmanager**-Datenmodell und die im Datenladeskript definierten Datentabellen können zwar nebeneinander existieren, aber die Problembehebung mit dem Datenmodell erschweren.*

3 Verwaltung von Daten in der App mit dem Datenmanager

Im **Datenmanager** können Sie Daten aus Ihren eigenen Datenquellen oder Qlik DataMarket hinzufügen und verwalten, damit Sie diese Daten in Ihrer App verwenden können. Sie erhalten eine Übersicht über alle Daten in der App, und zwar unabhängig davon, ob Sie diese mit **Daten hinzufügen** oder durch Laden mit dem Datenladeskript hinzugefügt haben. Jede Tabelle wird mit dem Tabellennamen, der Anzahl der Datenfelder und dem Namen der Datenquelle angezeigt.

3.1 Hinzufügen einer neuen Datentabelle

Durch das Anklicken von **Daten hinzufügen** im **Datenmanager** oder im ☰-Menü können Sie Ihrer App schnell eine Datentabelle hinzufügen.


Sie können von den nachfolgenden Typen von Datenquellen Daten hinzufügen:

Verbindungen	Wählen Sie aus Datenverbindungen, die von Ihnen oder einem Administrator festgelegt wurden, und aus Ordnern, aus denen Sie bereits Daten ausgewählt haben.
Meine Daten verbinden	Wählen Sie aus einer neuen Datenquelle wie ODBC oder OLE DB-Datenbanken, Datendateien, Webdateien oder benutzerdefinierten Konnektoren.
Qlik DataMarket	Wählen Sie aus normalisierten Daten aus öffentlichen und kommerziellen Datenbanken.
Datei anhängen	Laden Sie eine Datendatei von Ihrem Computer hoch und hängen Sie sie an Ihre App an.

3.2 Bearbeiten einer Datentabelle

Sie können alle Datentabellen bearbeiten, die Sie mit **Daten hinzufügen** hinzugefügt haben. Sie können die Felder in der Datentabelle umbenennen und die Felder aus der Datenquelle aktualisieren. (Zudem kann ein berechnetes Feld hinzugefügt werden und es können Felder aus der Analyseansicht entfernt werden).


Gehen Sie folgendermaßen vor:

1. Klicken Sie bei der Datentabelle, die Sie bearbeiten möchten, auf  .
Wenn der Datentabelleneditor geöffnet ist, können Sie beliebige Bearbeitungen und Transformationen vornehmen.
2. Klicken Sie auf **Schließen**, um zur Datenmanageransicht zurückzukehren.

Die Tabelle ist jetzt mit **Aktualisierung ausstehend** markiert und die Änderungen werden beim nächsten Laden der Daten auf die App-Daten angewendet.



Sie können nur Datentabellen bearbeiten, die über **Daten hinzufügen** hinzugefügt wurden.


Wenn Sie in einer Tabelle, die mithilfe des Ladeskripts geladen wurde, auf  klicken, wird der Dateneditor geöffnet.

Siehe: *Arbeiten mit dem Dateneditor (Seite 48)*

3.3 Löschen einer Datentabelle

Sie können nur Datentabellen löschen, die über **Daten hinzufügen** hinzugefügt wurden. Datentabellen, die mithilfe des Ladeskripts geladen wurden, können nur über eine Bearbeitung des Skripts im Dateneditor entfernt werden.

Gehen Sie folgendermaßen vor:

- Klicken Sie bei der Datentabelle, die Sie löschen möchten, auf .

Nun ist die Tabelle mit **Löschen ausstehend** markiert und wird beim nächsten Laden der Daten entfernt.

Sie können Löschvorgänge mit  bzw.  rückgängig machen und wiederherstellen.



Wenn Sie Felder aus der Datentabelle in einer Visualisierung verwendet haben, verursacht das Entfernen der Datentabelle einen Fehler, der in der App angezeigt wird.

3.4 Verwalten von Datentabellenverknüpfungen

Sie können Tabellenverknüpfungen anpassen und benutzerdefinierte Verknüpfungen zwischen den Tabellen definieren, indem Sie die Daten profilieren.

Gehen Sie folgendermaßen vor:

1. Klicken Sie auf **Verknüpfungen**.

Tabellen-Verknüpfungsempfehlungen werden mit allen möglichen Tabellenpaaren einschließlich der gewählten Tabellen sowie mit allen anderen Tabellenpaaren mit Warnungen angezeigt. Sie können die Verknüpfungen für Tabellenpaare ändern und benutzerdefinierte Verknüpfungen festlegen.




Siehe: *Verwalten von Datentabellenverknüpfungen (Seite 43)*

3.5 Änderungen anwenden und Daten neu laden

Die Änderungen, die Sie im **Datenmanager** vorgenommen haben, werden erst angezeigt, nachdem Sie die Daten erneut geladen haben. Wenn Sie die Daten neu laden wird die App auch alle Daten in der App aus externen Datenquellen neu laden.

3 Verwaltung von Daten in der App mit dem Datenmanager

Wenn die Daten im **Datenmanager** nicht mit den App-Daten synchron sind, ist die Schaltfläche **Daten laden** grün und alle neuen, aktualisierten oder gelöschten Tabellen werden blau markiert und ein Symbol zeigt den Status der Tabelle an:



- Tabellen, die mit **Löschen ausstehend**  markiert sind, werden gelöscht.
- Tabellen, die mit **Aktualisierung ausstehend**  markiert sind, werden mit den Feldern aktualisiert, die hinzugefügt, umbenannt oder entfernt wurden bzw. wird die Tabelle umbenannt.
- Tabellen, die mit **Hinzufügen ausstehend**  markiert sind, werden hinzugefügt.

Gehen Sie folgendermaßen vor:

- Klicken Sie auf **Daten laden**, um alle Daten in der App erneut zu laden.

Die App-Daten werden nun mit den Änderungen aktualisiert, die Sie im **Datenmanager** vorgenommen haben.

3.6 Vorgänge in **Datenmanager** rückgängig machen und wiederherstellen

Wenn Sie in im **Datenmanager** Änderungen vornehmen, können Sie Ihre Aktionen rückgängig machen oder wiederherstellen, indem Sie unten rechts im Bildschirm auf  bzw.  klicken oder die Tastaturkombinationen Ctrl + Z und Ctrl + Y verwenden.

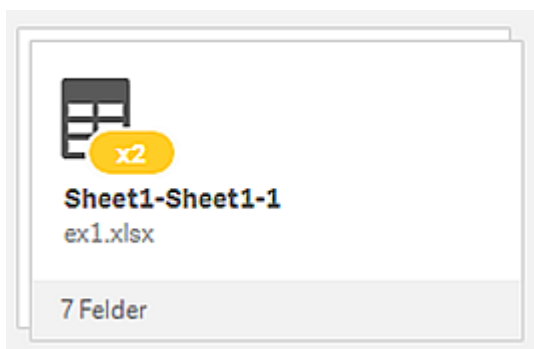
Das Protokoll der Aktionen wird gelöscht, wenn Sie:

- Die Ansicht ändern, zum Beispiel von der Tabellenübersicht zu **Verknüpfungen** wechseln.
- Daten laden.
- Den **Datenmanager** schließen.

3.7 Zusammengefasste Tabellen

Stimmen bei zwei oder mehr Tabellen die Zahl der Felder und die Feldnamen genau überein, fasst Qlik Sense diese Tabellen automatisch zu einer Tabelle zusammen.

Eine Tabelle, die das Ergebnis zusammengefasster Tabellen ist, wird mit einem Symbol angezeigt, dass darauf verweist, wie viele Tabellen zusammengefasst worden sind.




3 Verwaltung von Daten in der App mit dem Datenmanager

Siehe: *Zusammenfassung von Tabellen (Seite 89)*

Trennen einer zusammengefassten Tabelle

Falls Sie unabsichtlich eine einzelne zusammengefasste Tabelle erzeugt haben, können Sie die zusammengefasste Tabelle in ihre Komponententabellen trennen.

Gehen Sie folgendermaßen vor:

1. Markieren Sie die zusammengefasste Tabelle.
2. Klicken Sie auf .
Sie sehen eine Vorschau der Tabellen, die infolge der Trennung erzeugt werden.
3. Klicken Sie auf **Trennen**.

Nun wird die Tabelle in ihre Komponententabellen getrennt. Alle Felder in den Komponententabellen werden qualifiziert, das heißt, sie werden umbenannt und enthalten den Tabellennamen sowie ein Punktzeichen als Präfix.

Beispiel:

Table1 und Table2 enthalten beide die Felder Field1 und Field2. Wenn Sie sie im **Datenmanager** hinzufügen, werden sie zu der Tabelle Table1-Table2 mit denselben Feldern zusammengefasst: Field1 und Field2.

Wenn Sie Table1-Table2 trennen, erhalten Sie zwei Tabellen:

- Table1 mit den Feldern Table1.Field1 und Table1.Field2
- Table2 mit den Feldern Table2.Field1 und Table2.Field2

3.8 Interaktion zwischen **Datenmanager** und Datenladeskript

Wenn Sie Datentabellen im **Datenmanager** hinzufügen, wird ein Datenladeskriptcode generiert. Sie können den Skriptcode im **Automatisch generierten Abschnitt** des Dateneditors sehen. Sie können den generierten Skriptcode auch entsperren und bearbeiten. Dann werden die Datentabellen jedoch nicht mehr im **Datenmanager** verwaltet. Im Ladeskript definierte Datentabellen werden nicht im **Datenmanager** verwaltet. Die Tabellen werden zwar in der Datenübersicht angezeigt, können aber nicht im **Datenmanager** gelöscht oder bearbeitet werden und für mit dem Skript geladene Tabellen werden keine Verknüpfungsempfehlungen angezeigt.

Sie können Skriptbereiche hinzufügen und Code entwickeln, die/der das Datenmodell verbessert, das in **Datenmanager** erstellt wurde und mit diesem interagiert, allerdings müssen Sie in einigen Bereichen Vorsicht walten lassen. Der Skriptcode, den Sie schreiben, kann mit dem **Datenmanager**-Datenmodell kollidieren und in den folgenden Fällen zu Problemen führen:

3 Verwaltung von Daten in der App mit dem Datenmanager

- Umbenennen oder Ablegen von Tabellen, die im Skript mit **Datenmanager** hinzugefügt wurden.
- Ablegen von Feldern aus Tabellen, die mit **Datenmanager** hinzugefügt wurden.
- Zusammenführung von Tabellen, die mit **Datenmanager** hinzugefügt wurden und im Skript geladenen Tabellen.
- Verwendung der Qualifizieren-Funktion mit Feldern in Tabellen, die **Datenmanager** hinzugefügt wurden.
- Laden von Tabellen, die mit **Datenmanager** hinzugefügt wurden, mit **Resident** im Skript.

3.9 Hinzufügen von Daten aus Dateien und Datenbanken

Durch Klicken auf **Daten hinzufügen** im **Datenmanager** oder im Menü ☰ können Sie Ihrer App schnell Daten hinzufügen.

Hinzufügen von Daten aus einer vorhandenen Datenquelle

Sie können Daten aus Verbindungen auswählen, die von Ihnen oder einem Administrator definiert wurden. Dabei kann es sich um eine Datenbank, einen Ordner mit Datendateien oder einen benutzerdefinierten Konnektor zu einer externen Datenquelle wie Salesforce handeln. Wenn Sie die Daten mit **Meine Daten verbinden** aus einer neuen Datenquelle hinzufügen, wird eine Verbindung mit der Datenquelle unter **Verbindungen** erstellt, was das Hinzufügen von mehr Daten aus derselben Datenquelle vereinfacht.

Gehen Sie folgendermaßen vor:

1. Klicken Sie auf **Daten hinzufügen**.
2. Klicken Sie auf **Verbindungen**.
3. Wählen Sie die Verbindung aus.
4. Wählen Sie, aus welcher Datenquelle Daten hinzugefügt werden sollen. Dies variiert je nach Typ der Datenquelle.
 - Dateibasierte Datenquellen: Wählen Sie eine Datei aus.
 - Datenbanken: Legen Sie fest, welche Datenbank verwendet werden soll.
 - Webdateien: Geben Sie die URL der Webdatei ein.
 - Andere Datenquellen: Vom Konnektor festgelegt.
5. Wählen Sie die zu ladenden Tabellen und Felder aus.
6. Wählen Sie, ob Sie weiter mit den Daten arbeiten oder Daten in die App laden möchten:
 - Klicken Sie auf **Daten vorbereiten**, um weiter Datenquellen hinzuzufügen, Daten zu transformieren oder mögliche Probleme im **Datenmanager** zu lösen.
 - Klicken Sie auf **Daten laden und abschließen**, um die hinzugefügten Daten in die App zu laden und zum Arbeitsblatt zu gehen. Sie können auch alle vorhandenen Daten aus Datenquellen laden.

Wenn ein schwerwiegendes Problem erkannt wird, müssen Sie die Probleme im **Datenmanager** lösen, bevor Sie weiter Daten in die App laden können.

Hinzufügen von Daten aus einer neuen Datenquelle

Sie können Daten aus einer Datenquelle auswählen, die Sie zuvor noch nicht verwendet haben. Es stehen eine Reihe von Datenquellentypen zur Verfügung.

Gehen Sie folgendermaßen vor:

1. Klicken Sie auf **Daten hinzufügen**.
2. Klicken Sie auf **Meine Daten verbinden**.
3. Wählen Sie, welche Art von Datenquelle verwendet werden soll.
4. Wählen Sie, aus welcher Datenquelle Daten hinzugefügt werden sollen.
 - Dateibasierte Datenquellen: Wählen Sie eine Datei aus.
 - Datenbanken: Legen Sie fest, welche Datenbank verwendet werden soll.
 - Webdateien: Geben Sie die URL der Webdatei ein.
 - Andere Datenquellen: Vom Konnektor zur Datenbank festgelegt.
5. Wählen Sie die zu ladenden Tabellen und Felder aus.
6. Wählen Sie, ob Sie weiter mit den Daten arbeiten oder Daten in die App laden möchten:
 - Klicken Sie auf **Daten vorbereiten**, um weiter Datenquellen hinzuzufügen, Daten zu transformieren oder mögliche Probleme im **Datenmanager** zu lösen.
 - Klicken Sie auf **Daten laden und abschließen**, um die hinzugefügten Daten in die App zu laden und zum Arbeitsblatt zu gehen. Dann werden auch alle vorhandenen Daten aus Datenquellen erneut geladen.

Wenn ein schwerwiegendes Problem erkannt wird, müssen Sie die Probleme im **Datenmanager** lösen, bevor Sie weiter Daten in die App laden können.

Anhängen von Datendateien

Wenn Sie Qlik Sense auf einem Server verwenden und Datendateien von Ihrem Computer nutzen möchten, können Sie diese hochladen und einer App anhängen. Eine angehängte Datei ist nur in der jeweiligen App verfügbar, der sie angehängt wurde. Es besteht keine Verbindung zu Ihrer Originaldatei und das bedeutet, dass Sie die angehängte Datei aktualisieren müssen, falls Sie die Originaldatei aktualisiert hatten.



Datei anhängen ist in Qlik Sense Desktop nicht verfügbar.

Anhängen einer neuen Datei

Gehen Sie folgendermaßen vor:

1. Klicken Sie auf **Daten hinzufügen**.
2. Klicken Sie auf **Datei anhängen**.
3. Legen Sie eine Datendatei ab oder wählen Sie eine zu ladende Datei von Ihrem Computer per Mausklick aus.

3 Verwaltung von Daten in der App mit dem Datenmanager

Wenn Sie versuchen eine Datei mit einem Namen anzuhängen, deren Name mit einer bereits angehängten Datei identisch ist, können Sie die angehängte Datei gegen die neue Datei austauschen.



Jede angehängte Datei muss einen einzigartigen Dateinamen haben.

4. Wählen Sie die zu ladenden Tabellen und Felder aus.
Siehe: *Auswählen von Datenfeldern aus Dateien und Datenbanken (Seite 21)*
5. Wählen Sie, ob Sie weiter mit den Daten arbeiten oder Daten in die App laden möchten:
 - Klicken Sie auf **Daten vorbereiten**, um weiter Datenquellen hinzuzufügen, Daten zu transformieren oder mögliche Probleme im **Datenmanager** zu lösen.
 - Klicken Sie auf **Daten laden und abschließen**, um die hinzugefügten Daten in die App zu laden und zum Arbeitsblatt zu gehen. Sie können auch alle vorhandenen Daten aus Datenquellen laden.

Wenn ein schwerwiegendes Problem erkannt wird, müssen Sie die Probleme im **Datenmanager** lösen, bevor Sie weiter Daten in die App laden können.

Löschen einer angehängten Datei



*Wenn Sie eine angehängte Datei löschen, die in der App verwendet wird, können Sie die App erst wieder öffnen, nachdem Sie die entsprechenden Dateibezüge im **Datenmanager** oder im **Ladeskript** im **Dateneditor** entfernt haben. Ladeskripts werden im **Dateneditor** bearbeitet.*

Gehen Sie folgendermaßen vor:

1. Klicken Sie auf **Daten hinzufügen**.
2. Klicken Sie auf **Datei anhängen**.
3. Klicken Sie neben der zu löschenden Datei auf **✕**.
4. Klicken Sie auf **✕**, um **Daten hinzufügen** zu schließen.

Aktualisieren einer angehängten Datei

Eine angehängte Datei ist nur in der jeweiligen App verfügbar, der sie angehängt wurde. Es besteht keine Verbindung zu Ihrer Originaldatei und das bedeutet, dass Sie die angehängte Datei aktualisieren müssen, falls Sie die Originaldatei aktualisiert hatten und die neuen Daten in die App übernommen werden sollen.

Gehen Sie folgendermaßen vor:

- Hängen Sie die Datei erneut an und bestätigen Sie, dass Sie die angehängte Datei durch die aktualisierte Datei ersetzen möchten. Die aktualisierte Datei muss denselben Namen haben wie die angehängte Datei.

Synthetische Schlüssel

Wenn zwei oder mehr Datentabellen zwei oder mehr gemeinsame Felder haben, ist von einer zusammengesetzten Schlüsselbeziehung auszugehen. Um dies zu vermeiden, erstellt Qlik Sense automatisch synthetische Schlüssel. Synthetische Schlüssel enthalten alle möglichen Kombinationen der Werte zweier oder mehrerer Schlüsselfelder.

Siehe: *Synthetische Schlüssel (Seite 85)*

Wenn das Hinzufügen einer Tabelle zu einem der folgenden Fälle führt, ist **Daten vorbereiten** die einzige verfügbare Option, wenn Sie mit dem Hinzufügen von Daten fertig sind.

- Ein synthetischer Schlüssel, der mehr als fünf Felder enthält, wird erstellt.
- Mehr als zehn synthetische Schlüssel werden erstellt.
- Verschachtelte synthetische Schlüssel werden erstellt, das heißt synthetische Schlüssel, die andere synthetische Schlüssel enthalten.

Diese Fälle weisen darauf hin, dass Sie die Datentabellen anpassen müssen, um die Probleme zu beheben.

Welche Datenquellen sind für mich verfügbar?

Sie können Daten aus den nachfolgenden Typen von Datenquellen hinzufügen:

- **Verbindungen**
Fügen Sie Daten aus Datenverbindungen, die von Ihnen oder einem Administrator festgelegt wurden, und aus Ordnern, aus denen Sie bereits Daten ausgewählt haben, hinzu.
- **Meine Daten verbinden**
Fügen Sie Daten aus einer neuen Datenquelle wie ODBC oder OLE DB-Datenbanken, Datendateien, Webdateien oder benutzerdefinierten Konnektoren hinzu.



Sie können Datendateien auch hinzufügen, indem Sie sie im Qlik Sense Desktop-Fenster ablegen.

- **Qlik DataMarket**
Wählen Sie aus normalisierten Daten aus öffentlichen und kommerziellen Datenbanken.
- **Datei anhängen**
Laden Sie eine Datendatei von Ihrem Computer hoch und hängen Sie sie an Ihre App an.

Welche Datenquellentypen für Sie verfügbar sind, hängt von verschiedenen Faktoren ab:

- **Zugriffseinstellungen**
Die Administratoreinstellungen bestimmen, mit welchen Datenquellentypen Sie sich verbinden können.
- **Installierte benutzerdefinierte Konnektoren**

3 Verwaltung von Daten in der App mit dem Datenmanager

Qlik Sense bietet eine integrierte Unterstützung für viele Datenquellen. Zum Verbinden mit zusätzlichen Datenquellen benötigen Sie möglicherweise einen benutzerdefinierten Konnektor, der von Qlik oder einem Drittanbieter bereitgestellt wird. Benutzerdefinierte Konnektoren müssen installiert werden, bevor Sie sie verwenden können.

- Verfügbarkeit lokaler Dateien

Lokale Dateien auf dem Desktopcomputer sind nur in Qlik Sense Desktop verfügbar. Bei Serverinstallationen von Qlik Sense sind sie nicht verfügbar.



Wenn Sie lokale Dateien in eine Serverinstallation von Qlik Sense laden möchten, müssen Sie die Dateien an die App anhängen oder die Dateien in einen Ordner übertragen, der für den Qlik Sense-Server verfügbar ist – vorzugsweise in einen Ordner, der bereits als Ordnerdatenverbindung definiert wurde.

3.10 Auswählen von Datenfeldern aus Dateien und Datenbanken

Sie können auswählen, welche Tabellen und Felder beim Hinzufügen von Daten oder beim Bearbeiten einer Tabelle verwendet werden. Einige Datenquellen wie CSV-Dateien enthalten nur eine einzige Tabelle, wohingegen andere Datenquellen wie Microsoft Excel-Tabellen oder -Datenbanken mehrere Tabellen enthalten können.

Wenn eine Tabelle eine Kopfzeile enthält, werden Feldnamen normalerweise automatisch erkannt. In manchen Fällen müssen Sie jedoch die Einstellung für **Feldnamen** ändern. Möglicherweise müssen Sie auch noch andere Tabellenoptionen ändern, z. B. die **Größe des Headers** oder den **Zeichensatz**, damit die Daten richtig interpretiert werden. Die Tabelleneinstellungen unterscheiden sich je nach Datenquelle.

Auswählen von Daten aus einer Datenbank

Wenn Sie Daten aus einer Datenbank hinzufügen, kann die Datenquelle mehrere Tabellen enthalten.

Gehen Sie folgendermaßen vor:

1. Wählen Sie in der Dropdown-Liste eine **Datenbank** aus.
2. Wählen Sie den **Inhaber** der -Datenbank.
3. Wählen Sie die erste Tabelle aus, aus der Daten geladen werden sollen. Sie können alle Felder der Tabelle auswählen, indem Sie das Kästchen neben dem Tabellennamen aktivieren.
4. Wählen Sie durch Aktivieren der Kästchen neben den einzelnen Feldern die Felder aus, die Sie laden möchten.

3 Verwaltung von Daten in der App mit dem Datenmanager



Sie können den Feldnamen bearbeiten, indem Sie auf einen vorhandenen Feldnamen klicken und einen neuen Namen eingeben. Dies kann sich darauf auswirken, wie die Tabelle mit anderen Tabellen verlinkt ist, da sie standardmäßig über gemeinsame Felder verbunden werden.

5. Wenn Sie mit der Datenauswahl fertig sind, können Sie auf eine von zwei verschiedenen Arten fortfahren:
 - Klicken Sie auf **Datenprofil erstellen**, um mit der Datenprofilerstellung fortzufahren und Empfehlungen für Tabellenbeziehungen zu sehen.
 - Klicken Sie auf **Laden und Abschließen**, um die ausgewählten Daten ohne Durchführung einer Datenprofilerstellung zu laden und die Erstellung der Visualisierungen zu starten. Die Tabellen werden mithilfe natürlicher Verknüpfungen verlinkt, d. h. über gemeinsam benannte Felder.

Auswählen von Daten aus einer Microsoft Excel-Tabelle

Wenn Sie Daten aus einer Microsoft Excel-Tabelle hinzufügen, kann die Datei mehrere Tabellen enthalten. Jedes Arbeitsblatt wird als separate Tabelle geladen. Eine Ausnahme liegt vor, wenn das Arbeitsblatt dieselbe Feld-/Spaltenstruktur wie ein anderes Arbeitsblatt oder eine geladene Tabelle aufweisen. In diesem Fall werden die Tabellen zusammengefasst.

Gehen Sie folgendermaßen vor:

1. Achten Sie darauf, dass im Arbeitsblatt die richtigen Einstellungen verwendet werden:
 - Feldnamen** Wählen Sie, ob die Tabelle **Eingebettete Feldnamen** oder **Keine Feldnamen** enthält. In der ersten Zeile eines Excel-Arbeitsblatts stehen normalerweise die eingebetteten Feldnamen. Wird **Keine Feldnamen** ausgewählt, werden die Felder A, B, C usw. benannt.
 - Größe des Headers** Wenn Sie die Zeilenzahl einstellen, die für den Tabellen-Header ausgelassen werden soll, werden Zeilen, die keine allgemeinen Informationen enthalten, nicht in einem Spaltenformat angezeigt.

Beispiel

Mein Arbeitsblatt sieht folgendermaßen aus:

Machine:	AEJ12B		
Date:	2015-10-05 09		
Timestamp	Order	Operator	Yield
2015-10-05 09:22	00122344	A	52
2015-10-05 10:31	00153534	A	67
2015-10-05 13:46	00747899	B	86

3 Verwaltung von Daten in der App mit dem Datenmanager

In diesem Fall müssen Sie diese Einstellungen verwenden:

Feldnamen **Eingebettete Feldnamen.**

Größe des Headers 2

2. Wählen Sie das erste Arbeitsblatt, aus dem Daten geladen werden sollen. Sie können alle Felder des Arbeitsblatts auswählen, indem Sie das Kästchen neben dem Arbeitsblattnamen aktivieren.
3. Wählen Sie durch Aktivieren der Kästchen neben den einzelnen Feldern die Felder aus, die Sie laden möchten.



Sie können den Feldnamen bearbeiten, indem Sie auf einen vorhandenen Feldnamen klicken und einen neuen Namen eingeben. Dies kann sich darauf auswirken, wie die Tabelle mit anderen Tabellen verlinkt ist, da sie standardmäßig über gemeinsame Felder verbunden werden.

4. Wenn Sie mit der Datenauswahl fertig sind, können Sie auf eine von zwei verschiedenen Arten fortfahren:
 - Klicken Sie auf **Datenprofil erstellen**, um mit der Datenprofilerstellung fortzufahren und Empfehlungen für Tabellenbeziehungen zu sehen.
 - Klicken Sie auf **Laden und Abschließen**, um die ausgewählten Daten ohne Durchführung einer Datenprofilerstellung zu laden und die Erstellung der Visualisierungen zu starten. Die Tabellen werden mithilfe natürlicher Verknüpfungen verlinkt, d. h. über gemeinsam benannte Felder.

Auswählen von Daten aus einer Tabelle

Sie können Daten aus einer größeren Anzahl von Datendateien auswählen.

Gehen Sie folgendermaßen vor:

1. Achten Sie darauf, dass das entsprechende Dateiformat unter **Dateiformat** ausgewählt wird.
2. Achten Sie darauf, dass in der Datei die richtigen Einstellungen verwendet werden. Die Dateieinstellungen unterscheiden sich je nach Dateityp.
3. Wählen Sie durch Aktivieren der Kästchen neben den einzelnen Feldern die Felder aus, die Sie laden möchten. Sie können auch alle Felder einer Datei auswählen, indem Sie das Kästchen neben dem Arbeitsblattnamen aktivieren.



Sie können den Feldnamen bearbeiten, indem Sie auf einen vorhandenen Feldnamen klicken und einen neuen Namen eingeben. Dies kann sich darauf auswirken, wie die Tabelle mit anderen Tabellen verlinkt ist, da sie standardmäßig über gemeinsame Felder verbunden werden.

4. Wenn Sie mit der Datenauswahl fertig sind, können Sie auf zwei verschiedene Arten fortfahren:

3 Verwaltung von Daten in der App mit dem Datenmanager

- Klicken Sie auf **Datenprofil erstellen**, um mit der Datenprofilerstellung fortzufahren und Empfehlungen für Tabellenbeziehungen zu sehen.
- Klicken Sie auf **Laden und Abschließen**, um die ausgewählten Daten ohne Durchführung einer Datenprofilerstellung zu laden und die Erstellung der Visualisierungen zu starten. Die Tabellen werden mithilfe natürlicher Verknüpfungen verlinkt, d. h. über gemeinsam benannte Felder.

Auswählen von Einstellungen für Dateitypen

Tabellen mit Trennzeichen

Diese Einstellungen werden für Tabellen mit Trennzeichen validiert, die eine einzelne Tabelle enthalten, in der jeder Datensatz durch eine Zeilenschaltung getrennt ist und jedes Feld durch ein Trennzeichen getrennt wird, wie zum Beispiel in einer CSV-Datei.

Dateiformateinstellungen

Dateiformat	<p>Wählen Sie Mit Trennzeichen oder Feste Satzlängen.</p> <p>Wenn Sie eine Auswahl treffen, spiegelt sich dies im Dialog "Daten auswählen" wider.</p>
Feldnamen	<p>Wählen Sie, ob die Tabelle Eingebettete Feldnamen oder Keine Feldnamen enthält.</p>
Delimiter	<p>Legen Sie das Trennzeichen für die Tabelle fest.</p>
Anführungszeichen	<p>Legen Sie fest, wie Anführungszeichen verarbeitet werden sollen:</p> <p>Keine = Anführungszeichen sind nicht zulässig</p> <p>Standard = standardmäßige Verwendung von Anführungszeichen (die Anführungszeichen können als erste und letzte Zeichen eines Feldwerts verwendet werden)</p> <p>MSQ = Modern Style Quoting (Anführungszeichen in modernem Stil, die mehrzeiligen Inhalt in Feldern erlauben)</p>
Größe des Headers	<p>Stellen Sie die Zeilenanzahl ein, die für den Tabellen-Header ausgelassen werden soll.</p>
Zeichensatz	<p>Wählen Sie den in den Tabellen verwendeten Zeichensatz aus.</p>
Kommentar	<p>Datendateien können Kommentare zwischen Datensätzen enthalten. Sie sind am Zeilenanfang durch ein oder zwei Sonderzeichen wie z. B. // gekennzeichnet.</p> <p>Geben Sie ein oder mehrere Zeichen zur Kennzeichnung von Kommentarzeilen ein. Zeilen, die mit dem/den hier angegebenen Zeichen beginnen, werden von Qlik Sense nicht geladen.</p>
EOF ignorieren	<p>Wählen Sie EOF ignorieren, wenn bei Ihren Daten Dateieinde-Zeichen als Bestandteile des Feldwerts enthalten sind.</p>

3 Verwaltung von Daten in der App mit dem Datenmanager

Datendateien mit fester Datensatzlänge

Datendateien mit fester Datensatzlänge enthalten eine einzige Tabelle, in der jeder Datensatz (jede Datenzeile) Spalten mit fester Feldgröße – üblicherweise mit Leer- oder Tabulatorzeichen – enthält.

Positionen der Feldumbrüche festlegen

Sie können die Positionen der Feldumbrüche auf zwei Weisen festlegen:

- Geben Sie die Feldumbruchpositionen durch Kommas voneinander getrennt manuell in **Position der Feldumbrüche** ein. Jede Position markiert den Anfang eines Felds.

Beispiel: 1,12,24

- Aktivieren Sie **Feldumbrüche**, um die Positionen der Feldumbrüche interaktiv in der Felddatenvorschau zu bearbeiten. **Positionen der Feldumbrüche** wird mit den ausgewählten Positionen aktualisiert. Sie haben folgende Möglichkeiten:
 - Klicken Sie in die Felddatenvorschau, um einen Feldumbruch einzufügen.
 - Klicken Sie auf einen Feldumbruch, um ihn zu löschen.
 - Ziehen Sie einen Feldumbruch, um ihn zu verschieben.

Dateiformateinstellungen

Feldnamen	Wählen Sie, ob die Tabelle Eingebettete Feldnamen oder Keine Feldnamen enthält.
Größe des Headers	Stellen Sie die Größe des Headers auf die Zeilenzahl ein, die für den Tabellen-Header ausgelassen werden soll.
Zeichensatz	Wählen Sie den in der Tabellendatei verwendeten Zeichensatz aus.
Tab-Länge	Wählen Sie die Anzahl der Leerzeichen aus, für die ein Tabulatorzeichen in den Tabellen stehen soll.
Satzlänge	Wählen Sie die Anzahl der Zeilen aus, über die ein Datensatz in der Tabellendatei angezeigt werden soll. Standard ist 1.

HTML-Dateien

HTML-Dateien können mehrere Dateien enthalten. Qlik Sense interpretiert alle Elemente mit einem <TABLE>-Tag als Tabelle.

Dateiformateinstellungen

Feldnamen	Wählen Sie, ob die Tabelle Eingebettete Feldnamen oder Keine Feldnamen enthält.
Zeichensatz	Wählen Sie den in den Tabellen verwendeten Zeichensatz aus.

XML-Dateien

Sie können Daten laden, die im XML-Format gespeichert sind.

3 Verwaltung von Daten in der App mit dem Datenmanager

Es gibt keine speziellen Dateiformateinstellungen für XML-Dateien.

QVD-Dateien

Sie können Daten laden, die im QVD-Format gespeichert sind. QVD ist ein natives Qlik Format, das nur von Qlik Sense oder QlikView gelesen oder geschrieben werden kann. Das Dateiformat ist für besonders schnelles Laden aus einem Qlik Sense-Skript optimiert, aber gleichzeitig sehr kompakt.

Es gibt keine speziellen Dateiformateinstellungen für QVD-Dateien.

QVX-Dateien

Sie können Daten laden, die im Qlik data eXchange (QVX)-Format gespeichert sind. QVX-Dateien werden mit benutzerdefinierten Konnektoren erzeugt, die mit dem Qlik QVX SDK entwickelt wurden.

Es gibt keine speziellen Dateiformateinstellungen für QVX-Dateien.

KML-Dateien

Sie können Kartendateien laden, die im KML-Format gespeichert sind, und diese in Kartenvisualisierungen verwenden.

Es gibt keine speziellen Dateiformateinstellungen für KML-Dateien.

Zurück zum vorherigen Schritt (**Daten hinzufügen**)

Sie können beim Hinzufügen von Daten zum vorherigen Schritt zurückkehren.

Gehen Sie folgendermaßen vor:

- Klicken Sie auf , um zum vorherigen Schritt des Vorgangs **Daten hinzufügen** zurückzukehren.

3.11 Auswählen von Daten aus Qlik DataMarket

Mit Qlik DataMarket können Sie Daten aus externen Quellen hinzufügen. Qlik DataMarket stellt eine umfangreiche Sammlung an aktuellen und einsatzbereiten Daten aus externen Quellen bereit, die direkt in Qlik Sense geöffnet werden können. Qlik DataMarket stellt aktuelle und historische Wetter- und Demografiedaten, Wechselkurse verschiedener Währungen sowie wirtschaftliche und gesellschaftliche Daten bereit.

Manche Qlik DataMarket-Daten sind kostenlos verfügbar. Datenpakete der Kategorie **Premium** sind gegen einen Abonnementpreis verfügbar. Nach Eingabe der Abonnementzugangsdaten werden die Premium-Datenpakete als **Lizenziert** gekennzeichnet.



Premium-Datensätze sind in Qlik Sense Desktop nicht verfügbar.

Qlik DataMarket-Daten können separat untersucht oder mit Ihren eigenen Daten integriert werden. Das Anreichern interner Daten mit Qlik DataMarket kann oft zu umfassenderen Erkenntnissen führen.

3 Verwaltung von Daten in der App mit dem Datenmanager

Qlik DataMarket-Daten sind auf dem gleichen Stand wie die Quelle, aus der sie stammen. Die Häufigkeit, mit der Quelldaten aktualisiert werden, variiert. Wetter- und Marktdaten werden üblicherweise mindestens einmal täglich aktualisiert, während Bevölkerungsstatistiken meist jährlich aktualisiert werden. Die meisten makroökonomischen Indikatoren wie Arbeitslosigkeit, Preisindizes und Handel werden monatlich veröffentlicht. Alle Aktualisierungen werden normalerweise am selben Tag in Qlik DataMarket verfügbar gemacht.

Datenauswahlen in Qlik Sense sind persistent, sodass die neuesten verfügbaren Daten aus Qlik DataMarket geladen werden, wann immer das Skript ausgeführt wird.

Die meisten Qlik DataMarket-Daten sind sowohl global als auch landesspezifisch. Daten zur Bevölkerung sind zum Beispiel für mehr als 200 Länder und Gebiete verfügbar. Zudem bietet Qlik DataMarket verschiedene Daten für Staaten/Länder und Regionen in den USA und Europa.

Anwenden einer Lizenz auf Qlik DataMarket

Bevor Sie Qlik DataMarket-Daten verwenden können, müssen Sie die allgemeinen Geschäftsbedingungen akzeptieren. Wenn Sie Premium-Datenpakete bezahlt haben, müssen Sie außerdem Zugangsdaten eingeben, um Daten in diesen Paket verwenden zu können. Auf einige DataMarket-Daten können Sie ohne eine Lizenz zugreifen. Einige Daten sind eingeschränkt und es wird eine Lizenz benötigt. Die Daten, für welche eine Lizenz benötigt wird, sind in den Auswahldialogen mit **Premium** gekennzeichnet. Nach Eingabe der Zugangsdaten werden die Premium-Daten als **Lizenziert** gekennzeichnet.



Die allgemeinen Geschäftsbedingungen für Qlik DataMarket müssen bei der Verwendung von Qlik Sense Desktop nicht akzeptiert werden. Zugangsdaten sind ebenfalls nicht erforderlich, da Premium-Datensätze bei Qlik Sense Desktop nicht verfügbar sind.

Gehen Sie folgendermaßen vor:

1. Öffnen Sie Qlik Management Console.
2. Wählen Sie Qlik DataMarket auf der Registerkarte **Lizenzen und Token** aus.
3. Wählen Sie **Ich akzeptiere die allgemeinen Geschäftsbedingungen** aus.
4. Wählen Sie ein **Abonnement**, entweder **Kostenloses Paket** oder **Lizenziertes Paket**.
5. Wenn Sie das **lizenzierte Abonnement** wählen, geben Sie bitte Ihre Zugangsdaten ein:
Benutzername
Benutzerorganisation
Seriennummer
Kontrollnummer
6. Nachdem Sie Ihre Zugangsdaten eingegeben haben, erweitern Sie **LEF-Zugriff** und klicken Sie **LEF erhalten und Vorschau der Lizenz anzeigen**, um eine LEF-Datei vom Qlik SenseLEF-Server herunterzuladen.



Fehler beim Abrufen von LEF von Server wird angezeigt, wenn die Seriennummer oder Kontrollnummer falsch ist.

Wenn der Zugriff auf Ihre Installationsseite eingeschränkt ist, können Sie den Qlik-Lizenzserver nicht erreichen. Alternativ können Sie die LEF-Daten einer LEF-Datei kopieren und in das Textfeld einfügen. Ihr Ansprechpartner bei Qlik kann ebenfalls eine Lizenzdatei bereitstellen.



Wenn der Internetzugriff eingeschränkt ist, können Sie den Qlik DataMarket-Server für die Datenauswahl nicht erreichen. Damit Sie Zugriff erlangen, müssen Sie die Qlik DataMarket-Domäne der "Whitelist" der zugelassenen Domänen hinzufügen. Die DataMarket-Domäne ist datamarket-api.qlik.com.

7. Klicken Sie auf **Übernehmen**.

Hinzufügen von Daten aus Qlik DataMarket

Wenn Sie Daten aus Qlik DataMarket hinzufügen, wählen Sie Kategorien aus und filtern Sie dann die Felder der verfügbaren Daten in diesen Kategorien. Die DataMarket-Kategorien enthalten große Datenmengen. Das Filtern ermöglicht es Ihnen, Untergruppen der Daten zu generieren und die Menge geladener Daten zu reduzieren.

Gehen Sie folgendermaßen vor:

1. Klicken Sie auf **Daten hinzufügen**.
2. Klicken **Qlik DataMarket** im Schritt **Datenquelle auswählen**, damit die Qlik DataMarket-Pakete angezeigt werden.
3. Wählen Sie ein Datenpaket aus.



Um die mit *Premium* markierten Pakete zu verwenden, müssen Sie Ihre Zugangsdaten eingeben. Nachdem Sie die Zugangsdaten eingegeben haben, werden die Premium-Pakete als *Lizenziert* markiert und stehen zur Datenauswahl zur Verfügung.

4. Wählen Sie im Schritt **Datensatz auswählen** einen Datensatz aus.
Abhängig vom DataMarket-Paket, das Sie ausgewählt haben, müssen Sie möglicherweise verschiedene Unterkategorien im Paket auswählen, um zu einzelnen Datensätzen zu gelangen, die Daten enthalten. Datensätze sind am Symbol **i** neben der Datensatzbeschreibung erkennbar. Wenn Sie auf das Symbol **i** klicken, werden die Metadaten für den Datensatz angezeigt. Durch die Auswahl des Datensatzes wird eine Verbindung zu dieser spezifischen Datensammlung hergestellt.
5. Wählen Sie im Schritt **Auswahl der zu ladenden Daten** mindestens einen Filter für jede Dimension, Kennzahl und jeden Zeitraum aus.

3 Verwaltung von Daten in der App mit dem Datenmanager

Im Bereich auf der linken Seite werden die Dimensionen, Kennzahlen und Zeiträume angezeigt. Wenn Sie im linken Bereich eine Dimension, Kennzahl oder einen Zeitraum anklicken, werden die zugehörigen Werte im rechten Bereich angezeigt.

6. Wählen Sie, ob Sie weiter mit den Daten arbeiten oder Daten in die App laden möchten:
 - Klicken Sie auf **Daten vorbereiten**, um weiter Datenquellen hinzuzufügen, Daten zu transformieren oder mögliche Probleme im **Datenmanager** zu lösen.
 - Klicken Sie auf **Daten laden und abschließen**, um die hinzugefügten Daten in die App zu laden und zum Arbeitsblatt zu gehen. Dann werden auch alle vorhandenen Daten aus Datenquellen erneut geladen.

Wenn ein schwerwiegendes Problem erkannt wird, müssen Sie die Probleme im **Datenmanager** lösen, bevor Sie weiter Daten in die App laden können.

Auswählen von Qlik DataMarket-Dimensionen, Kennzahlen und Zeiträumen

Qlik DataMarket-Datensätze enthalten Dimensionen, Kennzahlen und Zeiträume. Dimensionen definieren den Datenkontext. Beispiele sind Land, Altersgruppe oder Geschlecht. Kennzahlen enthalten numerische Daten für den in Dimensionen festgelegten Kontext. Zeiträume definieren Jahre oder andere Zeitspannen, in denen die Kennzahlen betrachtet werden.

Datensätze enthalten mindestens eine Dimension und eine Kennzahl und alle haben Zeitdimensionen. Bevor Sie Daten zu einer App hinzufügen können, müssen Sie mindestens eine Dimension und eine Kennzahl auswählen und den Zeitraum festlegen. Wenn Sie Dimensionen auswählen, müssen Sie Dimensionen einschließen, die Daten enthalten. Wenn Daten hierarchisch strukturiert sind, besteht die Möglichkeit, dass ein übergeordneter Zweig keine Daten enthält.

In einigen Datensätzen ist es nicht notwendig, eine Kennzahl zu erstellen, weil der Datensatz nur eine Kennzahl enthält. Kennzahlauswahlen werden nur angezeigt, wenn mehr als eine Kennzahl zur Auswahl steht. Der Datensatz **US per capita personal income by state** zeigt zum Beispiel nur die geografische Dimension und den Zeitraum an, weil es nur eine Kennzahl im Datensatz gibt – das private Pro-Kopf-Einkommen.

Es gibt auch Datensätze, für die keine Dimension ausgewählt werden muss. Die Datensätze **US federal interest rate** und **US consumer price index for urban consumers** erfordern nur die Auswahl eines Zeitraums, weil es in diesen Datensätzen nur eine Dimension und eine Kennzahl gibt. Im ersten Fall ist die Kennzahl die Bundeszinsrate und die Dimension sind die USA. Im zweiten Fall ist die Kennzahl der Verbraucherpreisindex und die Dimension sind die städtischen Verbraucher in den USA.



*Datensätze können Daten aus Quellen enthalten, die nicht genannt werden dürfen. In solchen Fällen wird ein Dimensionsfeld **Zensiert belassen** benannt. So können Sie Kennzahlendaten aus einer Quelle verwenden, ohne ihre Herkunft preiszugeben. Wenn Sie zum Beispiel möchten, dass die Summe einer Kennzahl alle verfügbaren Daten enthält, können Sie das Feld **Zensiert belassen** in der Dimension auswählen.*

Auswählen von hierarchischen Daten aus Qlik DataMarket

3 Verwaltung von Daten in der App mit dem Datenmanager

Viele Qlik DataMarket Datensätze enthalten hierarchisch strukturierte Dimensionen und Kennzahlen. Hierarchisch strukturierte DataMarket-Datensätze enthalten Hierarchien mit zwei und drei Ebenen. Wie Auswahlen in diesen Hierarchien gemacht werden, hängt von den Daten auf den einzelnen Ebenen ab.

Auswählen von übergeordneten und untergeordneten Zweigen

In manchen Hierarchien können Auswahlen auf jeder und allen Ebenen erfolgen. Ein Beispiel: Der Datensatz **Selected development indicators** enthält die Dimension **Geographical area** mit drei Ebenen.

The screenshot shows the 'Select data to load' interface. On the left, there are filters for 'Dimensions' (Geographical area: 0/222), 'Measures' (Indicator: 0/20), and 'Time' (Year: 0/2). On the right, the 'Geographical area' dimension is expanded, showing a tree view with three levels: 'World', 'Region', and 'Country'. The 'World' level is selected, and the 'North America' region is also selected. The 'Country' level is not selected.

Eine Auswahl aus **World**, **Region** oder **Country** sind jede für sich gültig. Jede Auswahl, in welcher die höchste Ebene enthalten ist, lädt alle Daten der Regionen und Länder, und zwar auch dann, wenn zusätzlich einzelne Regionen und Länder ausgewählt wurden. Wird aber nur eine Region ausgewählt, wird nur diese Region der Welt geladen. Der folgende Bildschirm zeigt das Ergebnis, wenn nur die Region **North America** ohne **World** oder ein bestimmtes Land ausgewählt wird. Als Kennzahl wurde **CO2 emissions (metric tons per capita)** ausgewählt.

World	Region	Country	Year	CO2 emissions (met ...
World	North America		2011	16.7319858017

Wenn Sie sowohl **World** als auch **North America** auswählen, werden die Daten für World getrennt von jenen für North America angezeigt.

World	Region	Country	Year	CO2 emissions (met ...
World			2011	4.94464893026
World	North America		2011	16.7319858017

3 Verwaltung von Daten in der App mit dem Datenmanager

Wenn Sie **Canada** aus **Country** auswählen, werden getrennte Daten für World, die Region North America und Canada angezeigt.

Selected development indicators				
Add field ▼				
Select data from source				
Done ✓				
World	Region	Country	Year	CO2 emissions (metr...
World			2011	4.94464893026
World	North America		2011	16.7319858017
World	North America	Canada	2011	14.1358133791

Wenn Sie **Canada** aus **Country** auswählen, jedoch nicht **North America**, werden keine aggregierten Daten für North America geladen. Nur die Daten für Canada werden für die Region North America geladen.

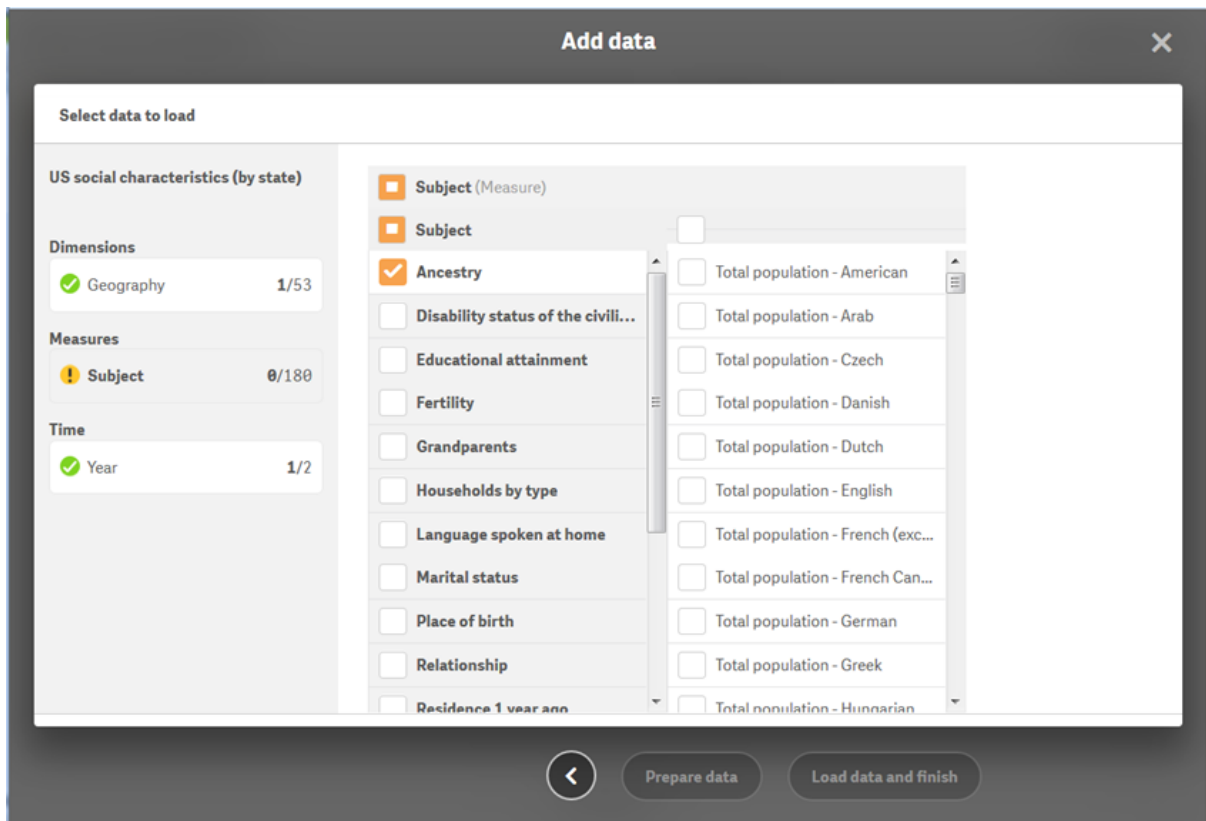
Selected development indicators				
Add field ▼				
Select data from source				
Done ✓				
World	Region	Country	Year	CO2 emissions (metr...
World			2011	4.94464893026
World	North America	Canada	2011	14.1358133791

Auswählen von übergeordneten Zweigen, die keine Daten enthalten

In einigen Datensätzen enthält der übergeordnete Zweig einer Hierarchie selbst keine Daten und kann daher nicht ausgewählt werden. (Felder in Zweigen ohne Daten werden fett dargestellt.) Beispiel: Im Datensatz **US social characteristics (by state)** stellt eine Auswahl, die nur ein übergeordnetes Objekt in der Kennzahl **Subject** enthält, keine vollständige Auswahl dar. Im folgenden Bildschirm ist **Subject** in der linken Spalte gelb markiert, obwohl eine Auswahl (**Ancestry**) getroffen wurde. Eine Auswahl muss in der untergeordneten Spalte erfolgen.

Solange die Auswahl nicht vollständig ist, sind die Schaltflächen für den nächsten Schritt deaktiviert. Sie werden erst aktiviert, wenn Sie ein Feld in der rechten Spalte der Kennzahl **Subject** auswählen. Das sind die Felder, die Daten enthalten, und die Auswahl von einem oder mehreren dieser Felder schließt die Auswahl für die Kennzahl **Subject** ab.

3 Verwaltung von Daten in der App mit dem Datenmanager



Wenn übergeordnete Zweige keine Daten enthalten, müssen Sie dort keine Auswahl vornehmen. Im Datensatz, der im Bildschirm oben gezeigt ist, könnten Sie die Auswahl von **Ancestry** aufheben und einfach nur eines der Felder von **Total population** im untergeordneten Zweig auswählen. Die Auswahl eines Feldes im übergeordneten Zweig hat den Vorteil, dass dann im untergeordneten Zweig die für die im übergeordneten Zweig getroffene Auswahl relevanten Daten hervorgehoben werden.

Wenn Sie ein Feld aus einem übergeordneten Zweig auswählen, das keine Daten enthält und dann ein nicht zugehöriges Feld aus dem untergeordneten Zweig auswählen, wird das Feld des untergeordneten Zweigs für die Datenauswahl herangezogen. Wenn im **Datenmanager** die Tabelle geladen wird, werden keine Daten des nicht zugehörigen übergeordneten Zweigs angezeigt. Wenn zum Beispiel aus dem Datensatz **US social characteristics (by state)** das Feld **Total population** und im übergeordneten Zweig ein nicht zugehöriges Feld wie **Marital status** ausgewählt werden, sieht die Tabelle im **Datenmanager** wie folgt aus:

US social characteristics (by state)			
Country	State	Year	Total population - Danish
United States		2013	1360858.0
United States	California	2013	182221.0

Auswählen von aggregierten Daten aus Qlik DataMarket

Einige Qlik DataMarket-Datensätze enthalten Dimensionen mit einem aggregierten Feld. Der folgende Datensatz **Weltbevölkerung nach Land** enthält zum Beispiel die Dimension **Geschlecht** mit einem aggregierten Gesamtwert sowie getrennten Gesamtwerten für Frauen und Männer. Wenn Sie die gesamte Dimension auswählen – das Kontrollkästchen oben – werden alle Felder in der Dimension ausgewählt. Sie können aber auch einzelne Felder auf den unteren Ebenen auswählen.

Select data to load

World population by country

Dimensions

- Country 2/228
- Sex 0/3
- Age group 21/22

Time

- Year 1/2

- Sex
- Sex ag... Sex
- Total Female
- Male

Wenn nur **Geschlecht aggregieren** ausgewählt wird, enthalten die Daten einen Gesamtwert für alle Männer und Frauen, nicht jedoch die einzelnen Gesamtwerte von Männern und Frauen. Sie könnten also keine Visualisierungen mit den jeweiligen Daten für Männer und Frauen erstellen.

Um die Daten nach Männern und Frauen aufzuschlüsseln, müssten Sie **Geschlecht** auswählen. Dann könnten Sie Visualisierungen erstellen, die getrennte Gesamtwerte für die männliche und weibliche Population anzeigen. Und wenn Sie nur am Gesamtwert eines Geschlechts interessiert sind, könnten Sie **Männlich** oder **Weiblich** auswählen.

Das aggregierte Feld ist enthalten, auch wenn Sie es nicht explizit auswählen. Der Wert im aggregierten Feld entspricht dem Wert der einzelnen ausgewählten Felder. Bei den **Geschlecht**-Daten entspricht **Geschlecht aggregieren** der Gesamtanzahl Frauen, wenn das Feld **Weiblich** das einzige ausgewählte **Geschlecht**-Feld ist.

Wenn Sie eine Visualisierung mit der Dimension **Geschlecht** erstellen, zeigen die Daten die Summen für Männer und Frauen getrennt an. Zum Beispiel wird die gesamte männliche und weibliche Bevölkerung von Argentinien im Balkendiagramm als zwei Balken dargestellt.

3 Verwaltung von Daten in der App mit dem Datenmanager




Würden Sie im Balkendiagramm die Dimension **Geschlecht aggregieren** anstelle von **Geschlecht** auswählen, würde ein Balken angezeigt, der die Summe aller Männer und Frauen repräsentiert.

Wenn Sie nur **Geschlecht aggregieren** ohne **Geschlecht** oder eines der Geschlechter auswählen, können Sie die Daten in der Visualisierung nicht nach Geschlecht aufgeschlüsselt darstellen. Das oben gezeigte Balkendiagramm würde einen einzigen Balken für die Summe aller Männer und Frauen anstelle zweier Balken für **Weiblich** und **Männlich** enthalten, wie im Bildschirm oben dargestellt.

3.12 Bearbeiten einer Tabelle

Sie können eine Tabelle, die der App in der **Datenmanager**-Übersicht hinzugefügt worden war, bearbeiten, um die Tabelle umbenennen oder um Felder zu transformieren.

- Klicken Sie bei der Datentabelle, die Sie bearbeiten möchten, auf .

Umbenennen von Tabellen

Wenn Sie eine Tabelle im **Datenmanager** hinzufügen, wird der Tabelle der standardmäßige Name basierend auf dem Namen der Datenbanktabelle, der Datendatei oder beispielsweise dem Excel-Arbeitsblatt zugewiesen. Wenn der Name nicht beschreibend oder ungeeignet ist, können Sie die Tabelle umbenennen.

Gehen Sie folgendermaßen vor:

3 Verwaltung von Daten in der App mit dem Datenmanager

1. Klicken Sie auf den Tabellennamen.
2. Bearbeiten Sie den Tabellennamen.
3. Drücken Sie die Eingabetaste oder klicken Sie außerhalb des Tabellennamens.



Eine zusammengeführte Tabelle kann nicht umbenannt werden.

Felder umbenennen

Sie können Felder in einer Tabelle umbenennen, weil Sie beispielsweise einen verständlicheren Namen wünschen oder das Feld mit einem Feld in einer anderen Tabelle verknüpfen möchten. Felder mit dem gleichen Namen werden automatisch verknüpft.

Gehen Sie folgendermaßen vor:

1. Klicken Sie auf das Feld, das Sie umbenennen möchten oder wählen Sie aus der Dropdown-Liste in der Zeile unter dem Feldnamen **Umbenennen** aus.
2. Geben Sie den neuen Namen ein.






Die Feldnamen müssen in jeder Tabelle einzigartig sein. Wenn Sie den Feldnamen ändern und er mit dem Feldnamen in einer anderen Tabelle identisch ist, werden die Tabellen automatisch mittels dieses Feldes verknüpft.

3. Drücken Sie die Eingabetaste oder klicken Sie außerhalb des Felds.

Das Feld wird umbenannt.

Feldtyp und Anzeigeformat ändern

Wenn Daten hinzugefügt werden, interpretiert Qlik Sense den Feldtyp jedes Feldes automatisch. Aktuell werden die folgenden Feldtypen unterstützt:

-  **Allgemein**
-  **Datum**
-  **Timestamp**

Wenn die Daten nicht richtig als Datum oder Timestamp interpretiert wurden, können Sie das Eingabeformat anpassen. Sie können auch das Anzeigeformat von Datum- und Zeitstempel-Feldern verändern.


Das Anzeigeformat von Datum- und Zeitstempel-Feldern kann jedoch nicht in allen Fällen verändert werden.

- Die Tabelle ist das Ergebnis der Zusammenfassung von zwei oder mehr Tabellen.
- Das Feld wird bereits als Datum oder Timestamp erkannt.


3 Verwaltung von Daten in der App mit dem Datenmanager

Sicherstellen, dass ein Datum- oder Timestamp-Feld richtig erkannt wird

Wenn ein Datum- oder Timestamp-Feld nicht als Datum oder Timestamp erkannt wird, wird es mit

 **Allgemein** gekennzeichnet und Sie können sicherstellen, dass es richtig interpretiert wird.

Gehen Sie folgendermaßen vor:

1. Klicken Sie auf  unter der Feldüberschrift. Der Datenformat-Dialog wird geöffnet.
2. Ändern Sie den **Feldtyp** zu **Datum** oder **Timestamp**.
3. Ändern Sie den Formatstring in **Eingabeformat**, um die Daten richtig zu interpretieren. Sie können ein vorbereitetes Format aus der Dropdown-Liste verwenden oder Ihren eigenen Formatstring schreiben.
Siehe: *Konventionen für Zahlen- und Uhrzeitformate (Seite 106)*





Es ist hier nicht möglich, in einem Formatstring ein einfaches Anführungszeichen zu verwenden.

4. Wenn Sie ein anderes Anzeigeformat als das standardmäßige Format in Ihrer App verwenden möchten, schreiben oder wählen Sie einen Formatstring in **Anzeigeformat**.
Wenn Sie das Feld leer lassen, wird das standardmäßige Anzeigeformat verwendet.
5. Klicken Sie außerhalb des Datenformat-Dialogs, um den Dialog zu schließen und das Feld zu aktualisieren.

Das Anzeigeformat von Datum- und Timestamp-Feldern verändern

Jede App verfügt über standardmäßige Anzeigeformate für die Felder Datum und Timestamp. Sie können das Anzeigeformat von individuellen Datum- und Timestamp-Feldern verändern.

Gehen Sie folgendermaßen vor:

1. Klicken Sie unter der Feldüberschrift auf  (Datum) oder  (Timestamp). Der Datenformat-Dialog wird geöffnet.
2. Ändern Sie den Formatstring in **Anzeigeformat**. Verwenden Sie entweder ein vorbereitetes Format aus der Dropdown-Liste oder schreiben Sie Ihren eigenen Formatstring.
3. Klicken Sie außerhalb des Datenformat-Dialogs, um den Dialog zu schließen und das Feld zu aktualisieren.

Tabellen aus der Datenquelle aktualisieren

Möglicherweise möchten Sie die Auswahl der Felder aus der Datenquelle verändern. Beispielsweise müssen Sie ein fehlendes Feld hinzufügen oder die Datenquelle wurde mit zusätzlichen Feldern aktualisiert. Dann können Sie die Tabelle aus der Datenquelle aktualisieren.

Gehen Sie folgendermaßen vor:

3 Verwaltung von Daten in der App mit dem Datenmanager

1. Klicken Sie auf **Datenquelle auswählen**.
Der Datenauswahlassistant öffnet mit Ihrer aktuellen Auswahl.
2. Nehmen Sie die notwendigen Änderungen an der Auswahl vor.
3. Klicken Sie auf **Laden und Abschließen**.

Nun wird die Tabelle entsprechend Ihrer Auswahl mit den Feldern aktualisiert.

Ein berechnetes Feld hinzufügen

In vielen Fällen müssen Sie die geladenen Felddaten gegebenenfalls anpassen oder transformieren. Beispielsweise müssen Sie einen Vor- und Nachnamen zu einem vollständigen Namen zusammenfassen, ein Teil aus einer Produktnummer extrahieren, das Datenformat konvertieren oder zwei Zahlen multiplizieren.

In vielen vergleichbaren Fällen können Sie berechnete Felder hinzufügen. Ein berechnetes Feld ist eine Formel, mit welcher das Ergebnis des Felds definiert wird. Sie können Funktionen, Felder und Operatoren in der Formel verwenden. Sie können lediglich auf die Felder in der Tabelle verweisen, die Sie bearbeiten.

Gehen Sie folgendermaßen vor:

1. Klicken Sie auf **Feld hinzufügen**.
Der Editor für das berechnete Feld wird geöffnet.
2. Geben Sie den Namen des berechneten Feldes unter **Name** ein.
3. Definieren Sie die Formel des berechneten Feldes. Dazu gibt es zwei verschiedene Möglichkeiten.
 - Verwenden Sie die Felder **fx (Funktionen)**, **≡ (Felder)** und **⊖ (Operatoren)**-Listen, um die Formel auszuwählen und einzufügen.
Das von Ihnen gewählte Element wird an der Cursorposition in **Formel** angezeigt.
 - Geben Sie die Formel für das berechnete Feld in **Formel** ein.
Während der Eingabe erhalten Sie Unterstützung hinsichtlich Funktionen und Feldern.
Die unter **Formel** angezeigten Nachrichten helfen Ihnen bei der Erstellung der richtigen Formel.
4. Klicken Sie auf **Erstellen**, um das berechnete Feld zu erstellen und den Editor für das berechnete Feld zu schließen.
Die Schaltfläche **Erstellen** wird erst aktiviert, nachdem Sie einen Namen angegeben haben und die Formel hinsichtlich der richtigen Syntax überprüft wurde.



Einer Tabelle, die das Ergebnis von zwei oder mehr zusammengeführten Tabellen ist, können keine berechneten Felder hinzugefügt werden.

Aktualisierung eines berechneten Feldes

Sie können den Namen eines berechneten Feldes oder dessen Formel ändern.

Gehen Sie folgendermaßen vor:

1. Wählen Sie im Dropdown-Menü neben dem Feldnamen **Aktualisieren**.
Der Editor für berechnete Felder wird geöffnet.

3 Verwaltung von Daten in der App mit dem Datenmanager

2. Bearbeiten Sie den Namen des berechneten Feldes unter **Name**, falls Sie den Namen ändern möchten.
3. Bearbeiten Sie die Formel des berechneten Feldes.
4. Klicken Sie auf **Aktualisieren**, um das berechnete Feld zu aktualisieren und den Editor für das berechnete Feld zu schließen.
Die Schaltfläche **Aktualisieren** wird erst aktiviert, nachdem Sie einen Namen angegeben haben und die Formel hinsichtlich der richtigen Syntax überprüft wurde.

Welche Funktionen kann ich in einer berechneten Feldformel verwenden?

Wenn Sie eine berechnete Feldformel verwenden, können Sie die hier aufgeführten Funktionen verwenden. Es handelt sich um einen Subsatz der Formeln, die im Datenladeskript vorhanden sind.


Die Formel kann Daten aus mehreren Datensätzen aggregieren oder Inter-Record-Funktionen verwenden, um auf Daten in anderen Datensätzen zu verweisen.

Stringfunktionen für die Verwendung in einer berechneten Feldformel

Diese Funktionen können für die Bearbeitung oder Extrahierung von Daten im Textstringformat verwendet werden.

Funktion	Beschreibung
Großschreibung	Capitalize() liefert den String mit allen Wörtern mit großen Anfangsbuchstaben.
FindOneOf	FindOneOf() durchsucht einen String, um die Position des Vorkommens eines beliebigen Zeichens aus einem Satz von bereitgestellten Zeichen zu finden. Die Position des ersten Vorkommens eines beliebigen Zeichens aus dem Suchsatz wird ausgegeben, sofern kein drittes Argument (mit einem Wert über 1) angegeben wird. Wird keine Übereinstimmung gefunden, ist das Ergebnis 0 .
Links	Left() liefert einen String bestehend aus den ersten n Zeichen des Eingabestrings (links außen), bei dem die Zahl der Zeichen durch das zweite Argument bestimmt wird.
Len	Len() liefert die Länge des Eingabestrings.
Lower	Lower() konvertiert alle Zeichen im Eingabestring zu Kleinbuchstaben.
LTrim	LTrim() liefert den Eingabestring ohne führende Leerzeichen.
Mid	Mid() liefert den Teil des Eingabestrings, der bei der Position des durch das zweite Argument, 'start', definierten Zeichens beginnt, wobei die Stringlänge durch das dritte Argument 'count' definiert wird. Wird 'count' weggelassen, wird die Position des ersten Vorkommens ausgegeben. Die Nummerierung des ersten Zeichens im Eingabestring beginnt bei 1.
Ersetzen	Replace() liefert den String s, wobei ein bestimmter Eingabestring bei jedem Vorkommen durch einen anderen ersetzt wird. Es erfolgt nur ein Durchlauf von links nach rechts.

3 Verwaltung von Daten in der App mit dem Datenmanager

Funktion	Beschreibung
Rechts	Right() liefert einen String bestehend aus den letzten (rechts außen stehenden) Zeichen des Eingabestrings, bei dem die Zahl der Zeichen durch das zweite Argument bestimmt wird.
RTrim	RTrim() liefert den Eingabestring ohne abschließende Leerzeichen.
SubField	Subfield() wird zur Extrahierung von Teilstring-Komponenten aus einem übergeordneten Stringfeld verwendet, bei dem die Ursprungsdatensatzfelder aus zwei oder mehr Teilen bestehen, die durch ein Trennzeichen getrennt sind.
<div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <i>Wenn Sie die Subfield-Funktion mit zwei Parametern verwenden, können weitere Datensätze erzeugt werden, die ggf. nicht im Tabelleneditor angezeigt werden.</i></div>	
TextBetween	TextBetween() liefert den Text im Eingabestring, der zwischen den Zeichen erscheint, die als Trennzeichen festgelegt wurden.
Trim	Trim() liefert den Eingabestring ohne führende und abschließende Leerzeichen.
Upper	Upper() wandelt alle Zeichen im Eingabestring für alle Textzeichen in der Formel in Großbuchstaben um. Zahlen und Symbole werden ignoriert.

Datumfunktionen für die Verwendung in einer berechneten Feldformel

Qlik Sense Funktionen für Datum und Zeit werden für die Umwandlung und Konvertierung von Daten- und Zeitwerten verwendet.

Die Funktionen basieren auf einer laufenden Nummer von Datum und Zeit, die der Zahl an Tagen seit dem 30. Dezember 1899 entspricht. Die ganze Zahl steht für den Tag und der Bruchteil für die Uhrzeit des Tages.

Qlik Sense verwendet den numerischen Wert des Arguments und somit gilt eine Zahl auch als Argument, wenn sie nicht als Datum oder Uhrzeit formatiert ist. Entspricht der Parameter nicht dem numerischen Wert, zum Beispiel, weil es sich um einen String handelt, versucht Qlik Sense, den String gemäß der Umgebungsvariablen von Datum und Uhrzeit zu interpretieren.

Wenn das verwendete Datumsformat im Argument nicht mit jenem übereinstimmt, das in der Systemvariablen **DateFormat** konfiguriert worden ist, kann Qlik Sense das Datum nicht korrekt interpretieren. In diesem Fall können Sie entweder die Systemeinstellungen ändern oder Sie verwenden die Interpretationsfunktion.

Funktion	Beschreibung
addmonths	Diese Funktion liefert das Datum, das n Monate nach startdate liegt bzw. n Monate vor n , wenn startdate negativ ist.
Tag	Diese Funktion liefert den Tag als ganze Zahl, wenn expression entsprechend dem Standardformat als Datum interpretiert wird.

3 Verwaltung von Daten in der App mit dem Datenmanager

Funktion	Beschreibung
Monat	Diese Funktion gibt einen dualen Wert zurück: ein Monatsname gemäß Definition in der Umgebungsvariable MonthNames sowie eine Ganzzahl zwischen 1-12. Der Monat berechnet sich durch die Datumsinterpretation der Formel entsprechend dem Standardformat.
Monatsende	Diese Funktion liefert den Zeitstempel der letzten Millisekunde des letzten Tags des Monats, in dem date liegt. Das Ergebnis wird entsprechend dem im Skript definierten DateFormat formatiert.
monthname	Diese Funktion liefert einen Wert mit dem Monat (entsprechend der Skriptvariable MonthNames formatiert) und dem Jahr mit einem numerischen Wert, der dem Zeitstempel der ersten Millisekunde des ersten Tags des Monats entspricht.
monthstart	Diese Funktion liefert den Zeitstempel der ersten Millisekunde des ersten Tages des Monats, in dem date liegt. Das Ergebnis wird entsprechend dem im Skript definierten DateFormat formatiert.
quarterend	Diese Funktion liefert den Zeitstempel der letzten Millisekunde des Quartals, in dem date liegt. Das Ergebnis wird entsprechend dem im Skript definierten DateFormat formatiert.
quartername	Diese Funktion liefert den Zeitstempel der ersten Millisekunde des ersten Tags des Quartals. Das Ergebnis wird als Kombination von Monaten (entsprechend der Skriptvariablen MonthNames) und Jahr formatiert.
quarterstart	Diese Funktion liefert den Zeitstempel der ersten Millisekunde des Quartals, in dem date liegt. Das Ergebnis wird entsprechend dem im Skript definierten DateFormat formatiert.
Woche	Diese Funktion liefert die Kalenderwoche als Ganzzahl gemäß ISO 8601. Die Kalenderwoche berechnet sich durch die Datumsinterpretation der Formel entsprechend dem Standardformat.
Wochenende	Diese Funktion liefert den Zeitstempel der ersten Millisekunde des letzten Tags der Kalenderwoche (beginnend mit Sonntag), in der date liegt. Das Ergebnis wird entsprechend dem im Skript definierten DateFormat formatiert.
Wochenname	Diese Funktion liefert den Zeitstempel der ersten Millisekunde der Kalenderwoche, in der date liegt. Das Ergebnis wird als Kombination von Jahr und Wochennummer formatiert.
Wochenstart	Diese Funktion liefert den Zeitstempel der ersten Millisekunde des ersten Tags der Kalenderwoche (beginnend mit Montag), in der date liegt. Das Ergebnis wird entsprechend dem im Skript definierten DateFormat formatiert.
Jahr	Diese Funktion liefert das Jahr als ganze Zahl, wenn expression entsprechend der Standardinterpretation als Datum interpretiert wird.
Jahresende	Diese Funktion liefert den Zeitstempel der letzten Millisekunde des letzten Tages des Jahres, in dem date liegt. Das Ergebnis wird entsprechend dem im Skript definierten DateFormat formatiert.

3 Verwaltung von Daten in der App mit dem Datenmanager

Funktion	Beschreibung
Jahresanfang	Diese Funktion liefert einen Zeitstempel, der dem Beginn des ersten Jahres mit date entspricht. Das Ergebnis wird entsprechend dem im Skript definierten DateFormat formatiert.

Formatierungs- und Interpretationsfunktionen für die Verwendung in einer berechneten Feldformel

Die Formatierungsfunktionen nutzen den numerischen Wert der Eingabeformel und wandeln diesen in einen Textwert um. Die Interpretationsfunktionen gehen genau andersherum vor: sie werten Stringformeln als Zahlen aus und geben das Format der resultierenden Zahl an. Der Ausgabewert ist in beiden Fällen ein dualer Wert, mit einem Text- und einem numerischen Wert.

Beachten Sie beispielsweise bitte die Unterschiede in der Ausgabe zwischen den Funktionen **Date** und **Date#**.

Funktion	Ausgabe (Text)	Ausgabe (numerisch)
Date#('20140831', 'YYYYMMDD')	20140831	41882
Date(41882, 'YYYY.MM.DD')	2014.08.31	41882

Diese Funktionen sind nützlich, wenn Ihre Daten Datenfelder enthalten, die nicht als Datum interpretiert werden, weil das Format von der Datumformateinstellung in Qlik Sense abweicht. In diesem Fall kann die Verschachtelung von Funktionen sinnvoll sein:

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

Dann wird das DateInput-Feld entsprechend dem Eingabeformat YYYYMMDD interpretiert und gibt das von Ihnen gewünschte Format aus: YYYY.MM.DD.

Funktion	Beschreibung
Datum	Date() formatiert eine Formel als Datum und verwendet dafür das vorgegebene Format aus den Systemvariablen des Datenladeskripts, des Betriebssystems oder eines Formatcodes, wenn dieser zur Verfügung steht.
Date#	Date# evaluiert eine Formel als Datum und verwendet dafür das Format, das im zweiten Argument angegeben ist, wenn dies zur Verfügung steht.
Dual	Dual() kombiniert eine Zahl und einen String in einem einzelnen Datensatz, sodass die Zahlendarstellung des Datensatzes zur Sortierung und Berechnung verwendet und der Stringwert zu Darstellungszwecken genutzt werden kann.
Intervall	Interval() formatiert eine Zahl als Zeitintervall und verwendet dafür das vorgegebene Format aus den Systemvariablen des Datenladeskripts, des Betriebssystems oder eines Formatcodes, wenn dieser zur Verfügung steht.

3 Verwaltung von Daten in der App mit dem Datenmanager

Funktion	Beschreibung
Intervall#	Intervall#() evaluiert eine Textformel als Zeitintervall und verwendet dafür das vorgegebene Format des Betriebssystems, das standardmäßig vorgegebene Format oder das Format, das im zweiten Argument angegeben ist, wenn dies zur Verfügung steht.
Geld	Money() formatiert eine Formel numerisch als Geldwert und verwendet dafür, wenn kein Formatcode zur Verfügung steht, das vorgegebene Format aus den Systemvariablen des Datenladeskripts oder des Betriebssystems sowie optionale Dezimal- und Tausendertrennzeichen.
Geld#	Money#() wandelt einen Textstring in einen Geldwert um und verwendet dafür, wenn kein Formatcode zur Verfügung steht, das vorgegebene Format des Ladeskripts oder des Betriebssystems. Angepasste Dezimal- und Tausendertrennzeichen sind optionale Parameter.
Num	Num() formatiert eine Formel, wenn kein Formatcode zur Verfügung steht, numerisch im vorgegebenen Zahlenformat der Systemvariablen des Datenladeskripts oder des Betriebssystems sowie mit optionalen Dezimal- und Tausendertrennzeichen.
Num#	Num#() wandelt einen Textstring in einen numerischen Wert um, und zwar in das Zahlenformat, das im Datenladeskript oder im Betriebssystem festgelegt wurde. Angepasste Dezimal- und Tausendertrennzeichen sind optionale Parameter.
Text	Text() interpretiert die Formel als Text, auch wenn eine numerische Interpretation möglich ist.
Time	Time() formatiert eine Formel als Zeitwert und verwendet dafür, wenn kein Formatcode zur Verfügung steht, das vorgegebene Format aus den Systemvariablen des Datenladeskripts oder des Betriebssystems.
Time#	Time#() evaluiert eine Formel als Zeitwert und verwendet dafür, wenn kein Formatcode zur Verfügung steht, das vorgegebene Zeitformat des Datenladeskripts oder des Betriebssystems.
Timestamp	TimeStamp() formatiert eine Formel als Daten- und Zeitwert und verwendet dafür, wenn kein Formatcode zur Verfügung steht, das vorgegebene Zeitstempelformat aus den Systemvariablen des Datenladeskripts oder des Betriebssystems.
Timestamp#	TimeStamp#() evaluiert eine Formel als Daten- und Zeitwert und verwendet dafür, wenn kein Formatcode zur Verfügung steht, das vorgegebene Zeitstempelformat des Datenladeskripts oder des Betriebssystems.

Rückgängigmachen und Wiederherstellen

Sie können Tabellenbearbeitungsvorgänge mit  bzw.  rückgängig machen und wiederherstellen.

Der Aktionsverlauf wird gelöscht, wenn Sie den Tabelleneditor schließen.

3.13 Verwalten von Datentabellenverknüpfungen

Sie haben die Möglichkeit, bestehende Verknüpfungsprobleme mit den im Datenmanager geladenen Datendateien zu lösen, wenn Sie Daten hinzufügen. Qlik Sense verknüpft Tabellen automatisch basierend auf gemeinsamen Feldnamen, aber in manchen Fällen müssen Sie die Verknüpfung anpassen. Hier ein Beispiel:

- Wenn Sie zwei Felder mit demselben Dateninhalt aber unterschiedlichen Feldnamen aus zwei verschiedenen Tabellen geladen haben, ist es wahrscheinlich eine gute Idee, den Feldern identische Namen zu geben, um die Tabellen miteinander zu verknüpfen.
- Wenn Sie zwei Felder mit unterschiedlichem Dateninhalt aber identischen Feldnamen aus zwei verschiedenen Tabellen geladen haben, müssen Sie mindestens eines der Felder umbenennen, um sie als separate Felder zu laden.

Gehen Sie folgendermaßen vor:

- Klicken Sie auf **Verknüpfungen**, um zur Datenmanageransicht zurückzukehren.

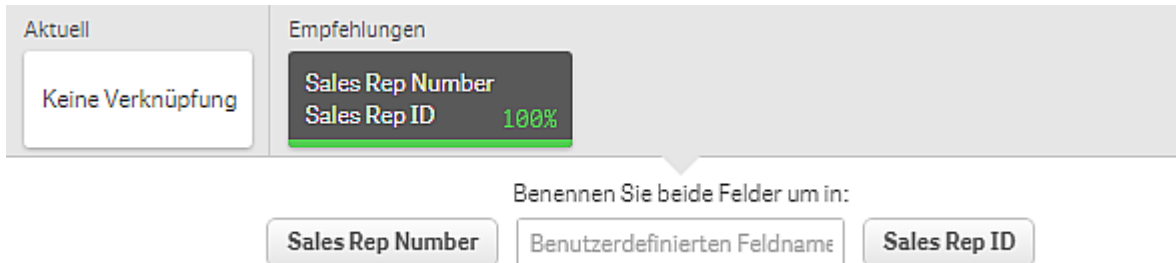
Qlik Sense führt eine Profilerstellung der Daten durch, die Sie laden möchten, um Ihnen beim Lösen der Tabellenverknüpfungsprobleme zu helfen. Vorhandene schlechte Verknüpfungen und potenziell gute Verknüpfungen werden hervorgehoben und Sie erhalten Hilfe bei der Auswahl von zu verknüpfenden Feldern, basierend auf der Analyse der Daten.

Anzeigen von Empfehlungen

Verknüpfungsempfehlungen werden in einer Liste angezeigt und Sie können mit den Schaltflächen **^** und **∨** zwischen den Empfehlungen und Warnungen navigieren. Wenn die Liste Warnungen enthält, bestehen Verknüpfungsprobleme, die behoben werden müssen.



Zwei Datenquellen enthalten ein Feld mit verwandten Daten, aber unterschiedlichen Namen



In diesem Fall enthalten die beiden Tabellen ein gemeinsames Feld, das in den beiden Tabellen unterschiedlich benannt ist. **Aktuell** zeigt, dass die Tabellen nicht verknüpft sind. **Vorschläge** zeigt jedoch, dass es zwei Felder mit ähnlichen Daten und hoher Übereinstimmung gibt.

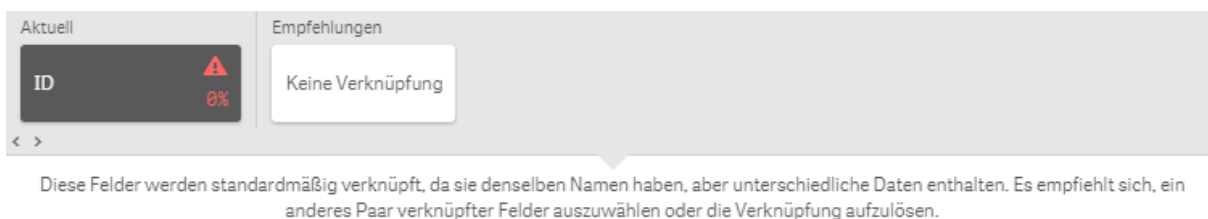
Um eine Verknüpfung zwischen den Tabellen zu erstellen, müssen Sie eines der oder beide Felder umbenennen, sodass sie denselben Namen haben.

Gehen Sie folgendermaßen vor:

1. Wählen Sie das Feldpaar aus, das Ihnen am geeignetsten erscheint. Üblicherweise ist dies das Paar mit der höchsten Übereinstimmung.
2. Wählen Sie den Feldnamen aus, den Sie verwenden möchten, oder geben Sie einen neuen benutzerdefinierten Feldnamen ein.

Die Felder werden nun umbenannt, damit sie denselben Namen haben, und die Tabellen werden verknüpft, wenn Sie die Daten laden.

Zwei Datenquellen enthalten Felder mit demselben Namen, aber nicht zusammenhängenden Daten



Sales					Stores		
ID	Amount	Item	Customer	Date	ID	Location	City
0034	12	023423	AF Fruit	2015-03-12	AFG112	West Mall	Newville
0035	5	023476	Johnson General	2015-02-12	AFG122	Central Square	Newville
0036	4	025643	NBGH	2015-02-14	KFG231	High Street	Springfield

3 Verwaltung von Daten in der App mit dem Datenmanager

In diesem Fall hat die Datenprofilerstellung gezeigt, dass die beiden Tabellen Felder mit nicht zusammenhängenden Daten, aber demselben Namen enthalten, was durch eine niedrige Übereinstimmung angezeigt wird. Das ID-Feld in der Tabelle Sales könnte für jede Reihenfolge ein eindeutiger Identifikator sein, während das ID-Feld in der Tabelle SalesOrderHeader der Identifikator einer Auftragsüberschrift ist. Wenn Sie die Tabellen laden, ohne das Problem zu beheben, werden sie verknüpft. Dies kann jedoch zu einem problematischen Datenmodell führen.

Um sicherzustellen, dass das Datenmodell korrekt ist, stehen Ihnen Möglichkeiten zur Behebung des Problems zur Verfügung. Sie sollten die Option auswählen, die die genaueste Verknüpfung darstellt. Dabei handelt es sich nicht zwingend um die Empfehlung mit der höchsten Übereinstimmung in %.

- Wählen Sie eine empfohlene Verknüpfung aus.
- Lösen Sie die Verknüpfung auf.

Auswählen einer empfohlenen Verknüpfung

In diesem Beispiel liegen gemäß der Datenprofilerstellung zwei empfohlene Verknüpfungen vor: ID-RevisionNumber und ID-TerritoryID. Wenn eine davon die richtige Verknüpfung ist, müssen Sie die Namen der verknüpften Felder angleichen.

Gehen Sie folgendermaßen vor:

1. Wählen Sie das empfohlene Feldpaar aus.
2. Wählen Sie den Feldnamen aus, den Sie verwenden möchten, oder geben Sie einen neuen benutzerdefinierten Feldnamen ein.

Die Felder werden nun umbenannt und die Tabellen werden verknüpft, wenn Sie Daten geladen haben.

Auflösen der Verknüpfung

Sind die zwei Tabellen nicht miteinander verbunden, aber beide enthalten ein Feld mit demselben Namen, dann müssen Sie die Verknüpfung auflösen.

Gehen Sie folgendermaßen vor:

1. Wählen Sie **Keine Verknüpfung** aus.
2. Klicken Sie auf **Aktuelle Verknüpfung auflösen**.

Die Felder werden nun umbenannt, indem sie mit dem Tabellennamen genauer bestimmt werden, in diesem Fall Sales.ID und SalesOrderHeader. Die Tabellen werden nicht verknüpft, wenn Sie Daten geladen haben.

Zwei Tabellen enthalten mehr als ein gemeinsames Feld

Wenn zwei Tabellen mehr als ein gemeinsames Feld enthalten, das eine Verknüpfung erstellen würde, erstellt Qlik Sense einen synthetischen Schlüssel. Wenn dies während der Datenprofilerstellung auftritt, erhalten Sie Empfehlungen, entweder eines der Felder als Schlüssel beizubehalten, indem die anderen gemeinsamen Felder umbenannt werden, oder die Tabellenverknüpfung aufzulösen.

Rückgängigmachen und Wiederherstellen

Sie können Verknüpfungsänderungen mit  bzw.  rückgängig machen und wiederherstellen.

3 Verwaltung von Daten in der App mit dem Datenmanager

Der Aktionsverlauf wird gelöscht, wenn Sie **Verknüpfungen** schließen.

Beschränkungen

In manchen Fällen werden keine Verknüpfungsempfehlungen angezeigt. Dies hängt von der Struktur der geladenen Tabellen und den Daten in diesen Tabellen ab:

- N:N-Beziehungen sollten nicht als Verknüpfungen verwendet werden.
- Wenn die Daten nicht in beide Richtungen gut übereinstimmen, wird keine Verknüpfung empfohlen. Ein Beispiel dafür wäre eine kleine Tabelle mit einigen wenigen Feldwerten, die zu 100 % mit einem Feld in einer großen Tabelle übereinstimmen, während die Übereinstimmung in die andere Richtung wesentlich kleiner ausfällt.
- Verbundschlüssel-Verknüpfungen werden nicht empfohlen.
- Der Datenmanager analysiert nur jene Tabellen, die mit **Daten hinzufügen** hinzugefügt wurden. Tabellen, die mit dem Datenladeskript hinzugefügt wurden, werden nicht in die Verknüpfungsempfehlungen übernommen.

4 Laden von Daten mit dem Datenladeskript

Diese Einleitung gibt einen kurzen Überblick darüber, wie Sie Daten mithilfe von Datenladeskripten in Qlik Sense laden können.

Qlik Sense benutzt ein Datenladeskript, das im Dateneditor verwaltet wird, mit dem Sie eine Verbindung zur Datenbank herstellen können, um verschiedene Datenquellen abzurufen. Im Skript werden die zu ladenden Felder und Tabellen festgelegt. Es ist möglich, die Datenstruktur zu ändern. Dies erfolgt über Skriptbefehle und Formeln.

Während der Ausführung des Skripts legt Qlik Sense allgemeine Felder aus verschiedenen Tabellen (Schlüselfelder) fest, um die Daten zu verknüpfen. Die Datenstruktur der Dateien in der App können Sie im Datenmodell überprüfen. Änderungen an der Datenstruktur können durch Umbenennen der Felder vorgenommen werden, um unterschiedliche Verknüpfungen zwischen den Tabellen zu erhalten.

Nachdem die Daten in Qlik Sense geladen sind, werden sie in der App gespeichert. Die App ist das Kernstück des Programms, auf dem die gesamte Funktionalität aufbaut: die Verknüpfung von Tabellen und Datensätzen, eine unbegrenzte Zahl von Dimensionen bei kompakter Dateigröße und die Abfrage per Mausklick in Sekundenbruchteilen. Wenn die App geöffnet ist, werden die darin enthaltenen Daten im RAM gehalten.

Die Analyse wird in Qlik Sense immer dann durchgeführt, wenn die App nicht direkt mit den Datenquellen verbunden ist. Für eine Aktualisierung der Daten müssen Sie das Skript daher erneut ausführen.

4.1 Interaktion zwischen **Datenmanager** und dem Datenladeskript

Wenn Sie Datentabellen im **Datenmanager** hinzufügen, wird ein Datenladeskriptcode generiert. Sie können den Skriptcode im **Automatisch generierten Abschnitt** des Dateneditors sehen. Sie können den generierten Skriptcode auch entsperren und bearbeiten. Dann werden die Datentabellen jedoch nicht mehr im **Datenmanager** verwaltet. Im Ladeskript definierte Datentabellen werden nicht im **Datenmanager** verwaltet. Die Tabellen werden zwar in der Datenübersicht angezeigt, können aber nicht im **Datenmanager** gelöscht oder bearbeitet werden und für mit dem Skript geladene Tabellen werden keine Verknüpfungsempfehlungen angezeigt.

Sie können Skriptbereiche hinzufügen und Code entwickeln, die/der das Datenmodell verbessert, das in **Datenmanager** erstellt wurde und mit diesem interagiert, allerdings müssen Sie in einigen Bereichen Vorsicht walten lassen. Der Skriptcode, den Sie schreiben, kann mit dem **Datenmanager**-Datenmodell kollidieren und in den folgenden Fällen zu Problemen führen:

- Umbenennen oder Ablegen von Tabellen, die im Skript mit **Datenmanager** hinzugefügt wurden.
- Ablegen von Feldern aus Tabellen, die mit **Datenmanager** hinzugefügt wurden.
- Zusammenführung von Tabellen, die mit **Datenmanager** hinzugefügt wurden und im Skript geladenen Tabellen.

4 Laden von Daten mit dem Datenladeskript

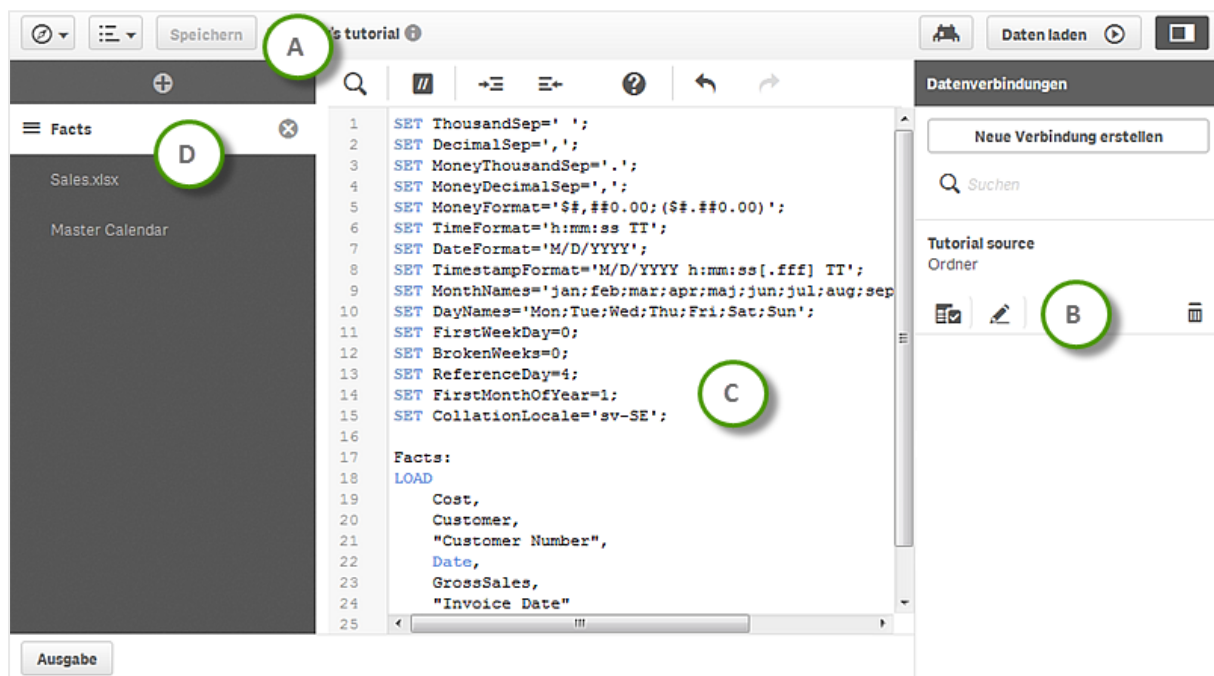
- Verwendung der Qualifizieren-Funktion mit Feldern in Tabellen, die **Datenmanager** hinzugefügt wurden.
- Laden von Tabellen, die mit **Datenmanager** hinzugefügt wurden, mit **Resident** im Skript.


4.2 Arbeiten mit dem Dateneditor

In diesem Abschnitt wird beschrieben, wie Sie den Dateneditor zum Erstellen oder Bearbeiten eines Datenladeskripts verwenden können, mit dem Sie Ihr Datenmodell in die App laden können.

Das Datenladeskript verknüpft eine App mit einer Datenquelle und lädt Daten von der Datenquelle in die App. Nachdem Sie die Daten geladen haben, können sie in der App analysiert werden. Sie können Datenladeskripte im Dateneditor anlegen, bearbeiten und ausführen.

Ein Skript kann manuell eingegeben oder automatisch generiert werden. Komplexe Skriptbefehle müssen allerdings – zumindest teilweise – von Hand eingegeben werden.




A	Die Symbolleiste mit den am häufigsten verwendeten Befehlen für den Dateneditor: Navigationsmenü, das globale Menü, Speichern ,  (Debuggen) und Daten laden . Die Symbolleiste zeigt außerdem den Speicher- und Datenladestatus der App an.	
B	Unter Datenverbindungen können Sie Shortcuts zu den Datenquellen (Datenbanken oder Dateien auf Servern) speichern, die Sie häufig verwenden. Von hier aus können Sie auch die Auswahl starten, welche Daten geladen werden.	

C	Im Texteditor können Sie Skript-Code schreiben und bearbeiten. Jede Zeile im Skript wird nummeriert. Außerdem ist das Skript nach Syntaxkomponenten farbcodiert. Die Symbolleiste des Texteditors enthält Befehle für Suchen und Ersetzen , Hilfe-Modus , Rückgängig und Wiederherstellen . Das ursprüngliche Skript enthält bereits einige regional vordefinierte Variableneinstellungen, zum Beispiel <code>SET thousandSep=</code> , die Sie im Allgemeinen nicht bearbeiten müssen.	
D	Teilen Sie Ihr Skript in Abschnitte auf, damit es leichter zu lesen und zu verwalten ist. Die Abschnitte werden von oben nach unten ausgeführt.	
E	In der Ausgabe werden alle Meldungen angezeigt, die während der Skriptausführung generiert wurden.	

Schnellstart

Wenn Sie eine Datei oder Tabellen aus einer Datenbank laden möchten, müssen Sie in den **Datenverbindungen** die folgenden Schritte ausführen:


1. **Neue Verbindung erstellen**, die auf die Datenquelle verweist (falls die Datenverbindung noch nicht vorhanden ist).
2.  Daten aus der Verbindung auswählen.

Nach Abschluss des Auswahldialogs durch **Skript einfügen** können Sie **Daten laden** auswählen, um das Datenmodell in Ihre App zu laden.















*Detaillierte Referenzen zur Skript- und Tabellenfunktionen finden Sie in der Datei **Skriptsyntax und Diagrammfunktionen**.*









Symbolleisten

Mithilfe der Symbolleisten können Sie globale Aktionen wie Rückgängig/Wiederholen, Debuggen und Suchen/Ersetzen für Ihr Datenladeskript durchführen. Sie können auch auf **Daten laden**  klicken, um die Daten in Ihrer App erneut zu laden.

Hauptsymboleiste

	<p>Navigationsmenü mit folgenden Optionen:</p> <ul style="list-style-type: none">  App-Übersicht  Datenmodell  Hub öffnen
	<p>Menü mit folgenden Optionen:</p> <ul style="list-style-type: none">  Daten hinzufügen  Hilfe  Info
Speichern	Speichern Sie die Änderungen.
	Klicken Sie zum Ein- oder Ausblenden der App-Details auf das Infosymbol.
	Skript debuggen.
Daten laden 	Führen Sie das Skript aus und laden Sie die Daten neu. Die App wird vor dem Neuladen automatisch gespeichert.
	Ansicht Datenverbindungen ein-/ausblenden.

Editor-Symboleiste

	Text im Skript suchen und ersetzen.
	Auskommentieren/einkommentieren
	Einrücken
	Ausrücken
	<p>Syntaxhilfe-Modus aktivieren. Im Hilfe-Modus können Sie auf ein Syntax-Schlüsselwort (blau markiert) im Editor klicken, um detaillierte Syntaxhilfe anzuzeigen.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  <i>Das Skript kann im Hilfe-Modus nicht bearbeitet werden.</i> </div>
	Die letzte Änderung im aktuellen Abschnitt rückgängig machen (kann mehrmals ausgeführt werden). Dasselbe erreichen Sie mit der Tastenkombination Strg+Z.
	Den Zustand vor dem letzten Rückgängig im aktuellen Abschnitt wiederherstellen. Dasselbe erreichen Sie mit der Tastenkombination Strg+Y.

Verbinden mit Datenquellen

Datenverbindungen bieten im Dateneditor eine Möglichkeit, die Verknüpfungen zu den Datenquellen zu speichern, die Sie normalerweise verwenden: Datenbanken, lokale Dateien oder Dateien auf Servern. Unter **Datenverbindungen** werden die Verbindungen, die Sie gespeichert haben, in alphabetischer Reihenfolge aufgelistet. Sie können das Such-/Filterfeld durch Eingabe eines bestimmten Namens oder Typs dazu verwenden, die Liste der Verbindungen einzugrenzen.

Datenverbindungsarten

Es gibt die folgenden Arten von Verbindungen:

- Standard-Konnektoren:
 - **ODBC**-Datenbankverbindungen.
 - **OLE DB**-Datenbankverbindungen.
 - Ordner-Verbindungen, die einen Pfad für lokale oder Netzwerkdateien festlegen. Der Ordner **Angehängte Dateien**, den Sie weder löschen noch bearbeiten können, enthält Dateien, die hochgeladen und der App angehängt werden.
 - Web-Datei-Verbindungen, die Daten aus Dateien auswählen, die sich unter einer Web-URL befinden.
- Benutzerdefinierte Konnektoren:

Mit benutzerdefinierten Konnektoren können Sie eine Verbindung zu Datenquellen herstellen, die von Qlik Sense nicht direkt unterstützt werden. Benutzerdefinierte Konnektoren werden mithilfe des QVX SDK entwickelt oder von Qlik oder den Entwicklern eines Drittanbieters bereitgestellt. Bei einer Standardinstallation von Qlik Sense stehen keine benutzerdefinierten Konnektoren zur Verfügung.



Ihnen werden nur Datenverbindungen angezeigt, die Ihnen gehören oder für die Sie Lese- oder Schreibzugriff haben. Wenden Sie sich bitte, falls erforderlich, an Ihren Qlik Sense-Systemadministrator, um Zugriff zu erhalten.

Erstellen einer neuen Datenverbindung

Gehen Sie folgendermaßen vor:

1. Klicken Sie auf **Neue Verbindung erstellen**.
2. Wählen Sie den Typ der Datenquelle aus der Dropdown-Liste aus, den Sie erstellen möchten. Das Dialogfeld mit den Einstellungen für die spezifische ausgewählte Datenquelle wird geöffnet.
3. Geben Sie die Einstellungen für die Datenquelle ein und klicken Sie auf **Erstellen**, um die Datenverbindung zu erstellen.

An den Verbindungsnamen werden Ihr Benutzername und Ihre Domäne angehängt, um sicherzustellen, dass er eindeutig ist.

4 Laden von Daten mit dem Datenladeskript

Die Datenverbindung wird nun mit Ihnen als Standardbenutzer erstellt. Wenn Sie möchten, dass andere Benutzer die Verbindung in einer Serverinstallation verwenden können, müssen Sie die Zugriffsrechte der Verbindung in Qlik Management Console bearbeiten.



Die Einstellungen der von Ihnen erstellten Verbindung werden nicht automatisch aktualisiert, wenn die Einstellungen der Datenquelle verändert werden. Sie müssen also beim Speichern von Benutzernamen und Passwörtern vorsichtig sein – besonders dann, wenn Sie Einstellungen der integrierten Windows-Sicherheit und des DSN-Datenbank-Logins ändern.



*Wird **Neue Verbindung erstellen** nicht angezeigt, verfügen Sie nicht über die Berechtigungen zum Hinzufügen von Datenverbindungen. Wenden Sie sich bitte, falls erforderlich, an Ihren Qlik Sense-Systemadministrator, um Zugriff zu erhalten.*

Datenverbindungseinstellungen

Jeder Datenverbindungstyp verfügt über spezifische Einstellungen, die Sie konfigurieren müssen. In diesem Abschnitt werden die Einstellungen für die Datenverbindungstypen beschrieben. Die benutzerdefinierten Konnektoreinstellungen werden in der Dokumentation des benutzerdefinierten Konnektors beschrieben.



ODBC-Datenverbindungseinstellungen

Einstellung	Beschreibung
Benutzer-DSN System-DSN	Wählen Sie, mit welcher Art von DSN eine Verbindung hergestellt werden soll. Bei Benutzer-DSN -Quellen müssen Sie festlegen, ob ein 32-Bit-Treiber in Verbindung mit 32-Bit-Verbindung verwenden eingesetzt wird. System-DSN -Verbindungen können nach 32 Bit oder 64 Bit gefiltert werden.
Benutzername	Benutzername für die Verbindung, sofern von der Datenquelle angefordert.
Passwort	Das Passwort für die Verbindung, wenn die Datenquelle dies erfordert.
Name	Name der Datenverbindung.


OLE DB-Datenverbindungseinstellungen

Einstellung	Beschreibung
Provider	Wählen Sie in der Liste der verfügbaren Provider Provider aus. Nur verfügbar, wenn Sie eine neue Verbindung erstellen.

4 Laden von Daten mit dem Datenladeskript

Einstellung	Beschreibung
Datenquelle	<p>Geben Sie den Namen der Datenquelle an, zu der die Verbindung aufgebaut werden soll. Hierbei kann es sich um einen Servernamen oder in manchen Fällen den Pfad zur Datenbankdatei handeln. Das hängt vom OLE DB-Provider ab, den Sie verwenden. Nur verfügbar, wenn Sie eine neue Verbindung erstellen.</p> <p>Beispiel:</p> <p>Falls Sie Microsoft Office 12.0 Access Database Engine OLE DB Provider ausgewählt haben, geben Sie den Dateinamen der Access-Datenbankdatei an, einschließlich des vollständigen Dateipfads:</p> <p><i>C:\Users\{user}\Documents\Qlik\Sense\Apps\Tutorial source files\Sales.accdb</i></p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;">  <i>Falls die Verbindung mit der Datenquelle fehlschlägt, wird eine Warnmeldung angezeigt.</i> </div>
Verbindungsstring	<p>Der Verbindungsstring für die Verbindung mit der Datenquelle. Dieser Verbindungsstring enthält Referenzen zum Provider und der Datenquelle. Nur verfügbar, wenn Sie eine Verbindung bearbeiten.</p>
In Windows integrierte Sicherheit	<p>Bei dieser Option verwenden Sie bestehende Windows-Login-Daten des Benutzers, der den Qlik Sense-Service ausführt.</p>
Bestimmter Benutzername und Passwort	<p>Bei dieser Option müssen Sie Benutzername und Passwort für die Datenquelle eingeben.</p>
Benutzername	<p>Benutzername für die Verbindung, sofern von der Datenquelle angefordert.</p> <p>Lassen Sie das Feld leer, wenn Sie die in Windows integrierte Sicherheit verwenden oder die Datenquelle keine Anmeldedaten erfordert.</p>
Passwort	<p>Das Passwort für die Verbindung, wenn die Datenquelle dies erfordert.</p> <p>Lassen Sie das Feld leer, wenn Sie die in Windows integrierte Sicherheit verwenden oder die Datenquelle keine Anmeldedaten erfordert.</p>
Load Datenbank auswählen...	<p>Wenn Sie die Verbindung testen möchten, klicken Sie auf Laden und dann auf Datenbank auswählen..., um die Datenbank zur Festlegung der Datenverbindung auszuwählen.</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;">  <i>Sie können weiterhin alle verfügbaren Datenbanken der Datenquelle verwenden, wenn Sie Daten aus der Datenverbindung auswählen.</i> </div>
Name	<p>Name der Datenverbindung.</p>

Ordner-Datenverbindungseinstellungen


Einstellung	Beschreibung
Pfad	<p>Geben Sie den Pfad zum Ordner mit den Datendateien ein. Sie haben zwei Möglichkeiten:</p> <ul style="list-style-type: none">• Den Ordner auswählen• Geben Sie einen gültigen lokalen Pfad ein <p>Beispiel: C:\data\MyData\</p> <ul style="list-style-type: none">• Geben Sie einen UNC-Pfad ein <p>Beispiel: \\myserver\filedir\</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <i>Ein zugeordnetes Netzwerklaufwerk kann in dem Pfad nicht verwendet werden.</i></div>
Name	Name der Datenverbindung.

Web-Datei-Datenverbindungseinstellungen

Einstellung	Beschreibung
URL	<p>Vollständige URL zu der Web-Datei, mit der Sie die Verbindung herstellen möchten.</p> <p>Wenn Sie sich mit einer FTP-Datei verbinden müssen, benötigen Sie möglicherweise Sonderzeichen, beispielsweise: oder @ im Benutzernamen- und Passwort-Teil der URL. Dann müssen Sie die Sonderzeichen durch ein Prozentzeichen und den hexadezimalen ASCII-Code des entsprechenden Zeichens ersetzen. Beispielsweise müssen Sie: gegen '%3a' und '@' gegen '%40' austauschen.</p>
Name	Name der Datenverbindung.


Löschen einer Datenverbindung

Gehen Sie folgendermaßen vor:

1. Klicken Sie auf der Datenverbindung, die Sie löschen möchten, auf .
2. Bestätigen Sie das Löschen der Verbindung.


Die Datenverbindung wurde gelöscht.



Wird  nicht angezeigt, verfügen Sie nicht über die Berechtigungen zum Löschen der Datenverbindung. Wenden Sie sich bitte, falls erforderlich, an Ihren Qlik Sense-Systemadministrator, um Zugriff zu erhalten.

Bearbeiten einer Datenverbindung

Gehen Sie folgendermaßen vor:


1. Klicken Sie bei der Datenverbindung, die Sie bearbeiten möchten, auf .
2. Bearbeiten Sie die Datenverbindung. Die Datenverbindungsdetails richten sich nach der Art der Verbindung.
3. Klicken Sie auf **Speichern**.

Die Datenverbindung wurde aktualisiert.



Nach dem Bearbeiten des Namens einer Datenverbindung müssen Sie auch alle vorhandenen Referenzen (lib://) zu dieser Verbindung im Skript bearbeiten, wenn die Referenz zur Verbindung weiterhin bestehen soll.



Wird  nicht angezeigt, verfügen Sie nicht über die Berechtigungen zum Aktualisieren der Datenverbindung. Wenden Sie sich bitte an Ihren Qlik Sense-Systemadministrator, falls erforderlich.

Einfügen eines Verbindungsstrings

Verbindungsstrings sind für **ODBC**-, **OLE DB**- und benutzerdefinierte Verbindungen erforderlich.

Gehen Sie folgendermaßen vor:

- Klicken Sie auf der Verbindung, in die ein Verbindungsstring eingefügt werden soll, auf .


Der Verbindungsstring für die ausgewählte Datenverbindung wird an der aktuellen Position im Dateneditor eingefügt.



Sie können einen Verbindungsstring auch einfügen, indem Sie eine Datenverbindung ziehen und an der Position im Skript ablegen, an der Sie diese einfügen möchten.

Auswählen von Daten aus einer Datenverbindung

Wenn Sie Daten aus einer Datenverbindung zum Einlesen in die App auswählen möchten, gehen Sie folgendermaßen vor:

1. **Neue Verbindung erstellen**, die auf die Datenquelle verweist (falls die Datenverbindung noch nicht vorhanden ist).
2.  Daten aus der Verbindung auswählen.

Verweisen auf eine Datenverbindung im Skript

Sie können die Datenverbindung verwenden, um auf Datenquellen in Befehlen und Funktionen im Skript zu verweisen, typischerweise dort, wo Sie sich auf einen Dateinamen mit einem Pfad beziehen möchten.

Die Syntax für den Bezug auf eine Datei ist `'lib://(Verbindungsname)/(Dateiname_einschließlich_Pfad)'`

Beispiel 1: Laden einer Datei aus einer Ordner-Datenverbindung

In diesem Beispiel wird die Datei `orders.csv` aus dem angegebenen Speicherort in die MyData-Datenverbindung geladen.

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

Beispiel 2: Laden einer Datei aus einem Unterordner

In diesem Beispiel wird die Datei `Customers/cust.txt` aus dem Datenverbindungsordner DataSource geladen. Customers ist ein Unterordner in dem Speicherort, der für die MyData-Datenverbindung angegeben wurde.

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

Beispiel 3: Laden aus einer Web-Datei

In diesem Beispiel wird eine Tabelle aus der Web-Datei-Datenverbindung PublicData geladen, die den Link zur tatsächlichen URL enthält.

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

Beispiel 4: Laden aus einer Datenbank

In diesem Beispiel wird die Tabelle `Sales_data` aus der Datenbankverbindung MyDataSource geladen.

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

Wo ist die Datenverbindung gespeichert?

Verbindungen werden mit Qlik Sense Repository Service gespeichert. Sie können Datenverbindungen mit Qlik Management Console in einer Qlik Sense-Serverbereitstellung verwalten. Mit Qlik Management Console können Sie Datenverbindungen löschen, Zugriffsrechte festlegen und andere Systemverwaltungsaufgaben durchführen.



In Qlik Sense Desktop werden alle Verbindungen ohne Verschlüsselung in der App gespeichert. Dies umfasst mögliche Details zu Benutzernamen, Passwort und dem Dateipfad, den Sie beim Erstellen der Verbindung angegeben haben. Alle diese Informationen sind also möglicherweise in Nur-Text verfügbar, wenn Sie die App mit einem anderen Nutzer teilen. Das müssen Sie berücksichtigen, wenn Sie eine App zur Freigabe einrichten.

Laden von Daten aus Dateien

Qlik Sense kann Daten aus Dateien in verschiedenen Formaten lesen:

- Textdateien, in denen die Daten in den Feldern durch Trennzeichen wie Kommas, Tabulator oder Strichpunkte getrennt sind (Comma-Separated-Variable-Dateien (CSV)).
- HTML-Tabellen
- Excel-Dateien (außer passwortgeschützte Excel-Dateien)
- XML-Dateien
- Native Dateiformate von Qlik QVD und QVX
- Dateien mit fester Datensatzlänge
- DIF-Dateien (Data Interchange Format) DIF-Dateien könne nur mit dem Dateneditor geladen werden)

Die Feldnamen werden hierbei meistens in der ersten Zeile angegeben

Es gibt verschiedene Möglichkeiten zum Laden von Daten aus Dateien:

- Hinzufügen von Daten
- Auswahl von Daten aus einer Datenverbindung im Dateneditor.
Statt einer manuellen Eingabe im Dateneditor können die zu ladenden Daten automatisch über den Dialog **Daten auswählen** ausgewählt werden.
- Laden von Dateien aus einer Datei durch das Schreiben eines Skriptcodes.
Zum Herunterladen der Dateien wird im Skript der **LOAD**-Befehl verwendet. **LOAD**-Befehle können alle Formeln im Skript beinhalten.
Daten aus anderen Qlik Sense-Apps können mithilfe von Binärbefehlen eingelesen werden, siehe **Binary**-Befehl.

Vorbereitung von Excel-Dateien für das Laden mit Qlik Sense

Wenn Sie Microsoft Excel-Dateien in Qlik Sense laden möchten, gibt es viele Funktionen, die Sie verwenden können, um Ihre Daten in das Datenladeskript umzuwandeln und zu bereinigen. Es ist jedoch möglicherweise praktischer, die Quelldaten direkt in der Tabellendatei in Microsoft Excel vorzubereiten. In diesem Abschnitt finden Sie einige Tipps, die Ihnen bei der Vorbereitung der Tabellenkalkulation auf das Laden in Qlik Sense mit minimalem Aufwand für die Skripterstellung helfen sollen.

Verwenden von Spaltenüberschriften

In Excel verwendete Spaltenüberschriften werden automatisch als Feldnamen verwendet, wenn Sie bei der Auswahl der Daten in Qlik Sense die Option **Eingebettete Feldnamen** aktivieren. Zeilenumbrüche sollten in den Bezeichnungen vermieden werden und die Kopfzeile sollte die erste Zeile im Arbeitsblatt sein.

Formatieren Ihrer Daten

Das Laden einer Excel-Datei in Qlik Sense gestaltet sich einfacher, wenn der Inhalt ohne Formatierung in der Tabelle vorliegt. Folgendes sollte vermieden werden:

- Aggregationen wie Summen oder Anzahlen. Aggregationen können in Qlik Sense festgelegt und berechnet werden.

4 Laden von Daten mit dem Datenladeskript

- Duplizierte Kopfzeilen.
- Zusätzliche Informationen wie Kommentare, die nicht zu den Daten gehören. Am besten ist es, eine gesonderte Spalte nur für Kommentare anzulegen, die beim Laden der Datei in Qlik Sense übersprungen werden kann.
- Kreuztabellen-Datenlayout. Wenn Sie beispielsweise eine Spalte pro Monat haben, sollten Sie stattdessen eine Spalte mit der Bezeichnung "Monat" erstellen und die Daten in 12 Zeilen eingeben, jeweils eine Zeile pro Monat. Sie können die Tabelle immer noch im Kreuztabellenformat in Qlik Sense aufrufen.
- Bei untergeordneten Kopfzeilen wie beispielsweise "Abteilung A", auf welche die Zeilen für Abteilung A folgen, sollte stattdessen eine Spalte namens "Abteilung" erstellt werden und diese mit den Namen der einzelnen Abteilungen ausgefüllt werden.
- Verbundene Zellen. Listen Sie einzelne Werte stattdessen in einzelnen Zellen auf.
- Leere Zellen, bei denen der Wert durch den vorherigen Wert impliziert wird. Bei wiederholten Werten müssen die leeren Zellen aufgefüllt werden, damit jede Zelle einen Datenwert enthält.

Verwenden benannter Bereiche

Wenn Sie nur einen Teil des Arbeitsblatts einlesen möchten, können Sie einen Spalten- und Zeilenbereich auswählen und diesen in Excel als benannten Bereich definieren. Qlik Sense kann Daten aus benannten Bereichen sowie aus Tabellen laden.

Typischerweise können Sie Rohdaten als einen benannten Bereich definieren und alle zusätzlichen Kommentare und Legenden außerhalb des benannten Bereichs behalten. Dadurch können die Daten einfacher in Qlik Sense geladen werden.

Passwortschutz entfernen

Passwortgeschützte Dateien werden von Qlik Sense nicht unterstützt.

Daten aus Datenbanken laden

Sie können mithilfe der folgenden Konnektoren Daten aus kommerziellen Datenbanken in Qlik Sense einlesen:


- **Standardkonnektoren** über die Schnittstelle Microsoft ODBC oder OLE DB. Zur Verwendung von ODBC muss ein entsprechender Treiber installiert werden, der Ihr DBMS unterstützt, und Ihre Datenbank muss als ODBC-Datenquelle konfiguriert werden. Dies kann in der Windows-**Systemsteuerung** über den Dialog **ODBC-Datenquellen-Administrator** erfolgen.
- **Benutzerdefinierte Konnektoren**, speziell zum Laden von Daten aus einem DBMS in Qlik Sense konzipiert.

Datensätze aus einer ODBC-Datenbank einlesen

Am einfachsten lassen sich Daten aus einer Datenbank wie Microsoft Access oder einer beliebigen anderen Datenbank, auf die über eine ODBC-Datenquelle zugegriffen werden kann, mithilfe des Dialogs für die Datenauswahl im Dateneditor laden.

Gehen Sie dazu wie folgt vor:

4 Laden von Daten mit dem Datenladeskript

1. Sie benötigen die ODBC-Datenquelle für die gewünschte Datenbank. Dies wird über den **ODBC-Datenquellen-Administrator** in der Windows-**Systemsteuerung** konfiguriert. Wenn Sie noch nicht über eine Quelle verfügen, müssen Sie sie hinzufügen und konfigurieren, beispielsweise durch den Verweis auf eine Microsoft Access-Datenbank.
2. Öffnen Sie den Dateneditor.
3. Erstellen Sie eine **ODBC**-Datenverbindung, die auf die in Schritt 1 erwähnte ODBC-Verbindung verweist.
4. Klicken Sie an der Datenverbindung auf , um den Dialog für die Datenauswahl zu öffnen.

Jetzt können Sie Daten aus der Datenbank auswählen und den zum Laden der Daten erforderlichen Skript-Code einfügen.

ODBC

Für den Zugriff auf DBMS (Database Management System) über ODBC mit Qlik Sense bieten sich Ihnen zwei Optionen.

- Installation eines ODBC-Treibers für das entsprechende DBMS und Erstellung einer DSN-Datenquelle. Dies wird in diesem Abschnitt beschrieben.
- Installation des Qlik ODBC Connector-Pakets, mit dem Sie die Datenquelle in Qlik Sense definieren können, ohne dass Sie den Microsoft Windows **ODBC-Datenquellen-Administrator** verwenden müssen.

Alternativ können Sie die Daten aus der Datenbank exportieren und in einem Dateiformat speichern, das Qlik Sense direkt einlesen kann.

Normalerweise stellt das Betriebssystem eine Reihe von ODBC-Treibern mit Microsoft Windows bereit. Zusätzliche Treiber können Sie im Softwarehandel kaufen, aus dem Internet herunterladen oder vom Hersteller des Datenbanksystems (DBMS) beziehen. Zum Teil sind diese Treiber kostenlos verfügbar.

Die hier beschriebene ODBC-Schnittstelle bezieht sich auf einen Client-Computer. Soll ODBC für den Zugriff auf eine Multi-User-Datenbank auf einem Netzwerkserverserver verwendet werden, brauchen Sie unter Umständen zusätzliche DBMS-Software, die den Clients Zugriff auf die Datenbank auf dem Server verschafft. Wenden Sie sich bitte auch hier an den DBMS-Hersteller.

Installieren von ODBC-Treibern

Damit Qlik Sense auf Ihre Datenbank zugreifen kann, muss ein passender ODBC-Treiber für Ihr DBMS (DataBase Management System) installiert sein. Weitere Informationen finden Sie in der Dokumentation des DBMS, das Sie verwenden.

64-Bit- und 32-Bit-Versionen der ODBC-Konfiguration

Eine 64-Bit-Version des Microsoft Windows-Betriebssystems enthält folgende Versionen des Microsoft Open DataBase Connectivity (ODBC) Data Source Administrator Tools (*Odbcad32.exe*):

- Die 32-Bit-Version der *Odbcad32.exe*-Datei finden Sie im Ordner `%systemdrive%\Windows\SysWOW64`.
- Die 64-Bit-Version der Datei *Odbcad32.exe* finden Sie im Ordner `%systemdrive%\Windows\System32`.

Erstellen von ODBC-Datenquellen

Um auf die Daten in Ihrer Datenbank zugreifen zu können, müssen Sie diese als ODBC-Datenquelle konfigurieren. Dies können Sie gleich bei der ODBC-Installation oder zu einem späteren Zeitpunkt vornehmen.



*Zunächst muss die Entscheidung erfolgen, ob für die Datenquellen die Option **Benutzer-DSN** oder **System-DSN** (empfohlen) gewählt werden soll. Sie können Benutzer-Datenquellen nur mit den richtigen Login-Daten aufrufen. Bei einer Serverinstallation müssen Sie normalerweise Systemdatenquellen erstellen, damit die Datenquellen für andere Benutzer freigegeben werden können.*

Gehen Sie folgendermaßen vor:

1. Öffnen Sie *Odbcad32.exe*.
2. Rufen Sie zum Erstellen einer Systemdatenquelle die Dialogseite **System-DSN** auf.
3. Klicken Sie auf **Hinzufügen**.

Sie gelangen in den Dialog **Neue Datenquelle erstellen**. Dort finden Sie eine Liste aller installierten ODBC-Treiber.

4. Ist der passende ODBC-Treiber aufgeführt, markieren Sie ihn und klicken Sie auf **Fertig**.
Es öffnet sich ein Dialog, der sich speziell auf den ausgewählten Treiber bezieht.
5. Benennen Sie dort die Datenquelle und legen Sie die gewünschten Parameter fest.
6. Klicken Sie auf **OK**.

Beste Praktiken für ODBC-Datenverbindungen

Verschieben von Apps mit ODBC-Datenverbindungen

Wenn Sie eine App zwischen Qlik Sense-Sites/Qlik Sense Desktop-Installationen verschieben, sind die Datenverbindungen enthalten. Wenn die App ODBC -Datenverbindungen enthält, müssen Sie sich vergewissern, dass die zugehörigen ODBC-Datenquellen auch bei der neuen Bereitstellung vorhanden sind. Die ODBC-Datenquellen müssen identisch benannt und konfiguriert sein und auf dieselben Datenbanken oder Dateien verweisen.

Sicherheitsaspekte beim Verbinden mit dateibasierten ODBC-Datenverbindungen

ODBC-Datenverbindungen mit dateibasierten Treibern legen den Pfad zur verbundenen Datendatei im Verbindungsstring offen. Der Pfad kann offen gelegt werden, wenn die Verbindung im Datenauswahldialog oder in bestimmten SQL-Abfragen bearbeitet wird.

Wenn dies nicht gewünscht wird, empfiehlt es sich, die Datendatei mit einer Ordner-Datenverbindung zu verbinden, falls möglich.

OLE DB

Qlik Sense unterstützt die OLE DB(Object Linking and Embedding, Database)-Schnittstelle, um Verbindungen zu externen Datenquellen herzustellen. OLE DB ermöglicht den Zugriff auf eine große Anzahl von externen Datenbanken.

Sicherheitsaspekte beim Verbinden mit dateibasierten OLE DB-Datenverbindungen

OLE DB-Datenverbindungen mit dateibasierten Treibern legen den Pfad zur verbundenen Datendatei im Verbindungsstring offen. Der Pfad kann offen gelegt werden, wenn die Verbindung im Datenauswahldialog oder in bestimmten SQL-Abfragen bearbeitet wird.

Wenn dies nicht gewünscht wird, empfiehlt es sich, die Datendatei mit einer Ordnerdatenverbindung zu verbinden, falls möglich.

Logik in Datenbanken

In ein Qlik Sense-Dokument können mehrere Tabellen aus Datenbanken gleichzeitig eingelesen werden. Wenn ein Feld in mehreren dieser Tabellen vorkommt, werden die Tabellen über dieses Schlüsselfeld logisch miteinander verknüpft.

Wenn ein Wert ausgewählt wird, werden alle mit der/den Auswahl(en) kompatiblen Werte als optional angezeigt. Alle anderen Werte sind ausgeschlossen.

Die Auswahl von Werten in mehreren Feldern wird im Sinne einer logischen UND-Verknüpfung interpretiert.

Die Auswahl mehrerer Werte innerhalb eines Felds wird standardmäßig als logisches ODER interpretiert.

Unter gewissen Umständen lässt sich dies ändern und sie wird als logisches UND gewertet.

Auswählen von Daten im Dateneditor

Sie können im Dateneditor über **Daten auswählen** wählen, welche Felder aus Dateien oder Datenbanktabellen geladen werden sollen und welche Ansichten der Datenquelle Sie verwenden möchten. Sie können die Felder im Dialog nicht nur auswählen, sondern auch umbenennen. Nach der Auswahl der Felder können Sie den generierten Skriptcode in Ihr Skript einfügen.


Einige Datenquellen wie CSV-Dateien enthalten nur eine einzige Tabelle, wohingegen andere Datenquellen wie Microsoft Excel-Tabellen oder -Datenbanken mehrere Tabellen enthalten können.

Öffnen Sie **Daten auswählen**, indem Sie in einer Datenverbindung im Dateneditor auf  klicken.

Auswählen von Daten aus einer Datenbank

Beim Hinzufügen von Daten aus einer Datenbank kann die Datenquelle mehrere Tabellen enthalten.

Gehen Sie folgendermaßen vor:

1. Klicken Sie in einer Ordner-Datenverbindung im Dateneditor auf . Der Dateiauswahldialog wird angezeigt.
2. Wählen Sie in der Dropdown-Liste eine **Datenbank** aus.

4 Laden von Daten mit dem Datenladeskript

3. Wählen Sie den **Autor** der Datenbank aus.
Die Liste der **Tabellen** wird mit Ansichten und Tabellen aufgefüllt, die in der ausgewählten Datenbank verfügbar sind.
4. Wählen Sie die erste Tabelle oder Ansicht aus, aus der Daten geladen werden sollen. Sie können alle Felder der Tabelle auswählen, indem Sie das Kästchen neben dem Tabellennamen aktivieren. **Felder** wird mit dem verfügbaren Tabelleninhalt aktualisiert und Sie können weitere Felder auswählen.
5. Wählen Sie durch Aktivieren der Kästchen neben den einzelnen Feldern die Felder aus, die Sie laden möchten.



Sie können den Feldnamen bearbeiten, indem Sie auf einen vorhandenen Feldnamen klicken und einen neuen Namen eingeben. Dies kann sich darauf auswirken, wie die Tabelle mit anderen Tabellen verlinkt ist, da sie standardmäßig über gemeinsame Felder verbunden werden.



Es ist nicht möglich, Felder im Assistenten für die Datenauswahl umzubenennen und dabei die Felder zu filtern. Sie müssen zunächst das Suchwort im Textfeld löschen.




Es ist nicht möglich, zwei Felder in derselben Tabelle gleich zu benennen.

6. Wenn Sie die Daten ausgewählt haben, gehen Sie folgendermaßen vor:
 - Klicken Sie auf **Skript einfügen**.
Das Datenauswahlfenster wird geschlossen und die LOAD /SELECT-Befehle werden im Skript gemäß Ihren Auswahlen eingefügt.

Auswählen von Daten aus einer Microsoft Excel-Tabelle

Wenn Sie Daten aus einer Microsoft Excel-Tabelle hinzufügen, kann die Datenquelle mehrere Tabellen enthalten. Jedes Arbeitsblatt wird als separate Tabelle geladen. Eine Ausnahme liegt vor, wenn das Arbeitsblatt dieselbe Feld-/Spaltenstruktur wie ein anderes Arbeitsblatt oder eine geladene Tabelle aufweisen. In diesem Fall werden die Tabellen zusammengefasst.

Gehen Sie folgendermaßen vor:

1. Klicken Sie in einer Ordner-Datenverbindung im Dateneditor auf .
Der Dateiauswahldialog wird angezeigt.
2. Wählen Sie das erste Arbeitsblatt, aus dem Daten geladen werden sollen. Sie können alle Felder des Arbeitsblatts auswählen, indem Sie das Kästchen neben dem Arbeitsblattnamen aktivieren.
3. Achten Sie darauf, dass im Arbeitsblatt die richtigen Einstellungen verwendet werden:

4 Laden von Daten mit dem Datenladeskript

- Feldnamen** Wählen Sie, ob die Tabelle **Eingebettete Feldnamen** oder **Keine Feldnamen** enthält. In der ersten Zeile eines Excel-Arbeitsblatts stehen normalerweise die eingebetteten Feldnamen. Wird **Keine Feldnamen** ausgewählt, werden die Felder A, B, C usw. benannt.
- Größe des Headers** Wenn Sie die Zeilenzahl einstellen, die für den Tabellen-Header ausgelassen werden soll, werden Zeilen, die keine allgemeinen Informationen enthalten, nicht in einem Spaltenformat angezeigt.

Beispiel

Mein Arbeitsblatt sieht folgendermaßen aus:

Machine:	AEJ12B		
Date:	2015-10-05 09		
Timestamp	Order	Operator	Yield
2015-10-05 09:22	00122344	A	52
2015-10-05 10:31	00153534	A	67
2015-10-05 13:46	00747899	B	86

In diesem Fall müssen Sie diese Einstellungen verwenden:

- Feldnamen** **Eingebettete Feldnamen.**
- Größe des Headers** 2

4. Wählen Sie durch Aktivieren der Kästchen neben den einzelnen Feldern die Felder aus, die Sie laden möchten.



Sie können den Feldnamen bearbeiten, indem Sie auf einen vorhandenen Feldnamen klicken und einen neuen Namen eingeben. Dies kann sich darauf auswirken, wie die Tabelle mit anderen Tabellen verlinkt ist, da sie standardmäßig über gemeinsame Felder verbunden werden.

5. Wenn Sie die Daten ausgewählt haben, gehen Sie folgendermaßen vor:

- Klicken Sie auf **Skript einfügen**.

Das Datenauswahlfenster wird geschlossen und die LOAD /SELECT-Befehle werden im Skript gemäß Ihren Auswahlen eingefügt.



*Sie können über die Microsoft Excel-Schnittstelle auch eine ODBC-Datei als Datenquelle verwenden. In diesem Fall müssen Sie eine **ODBC-Datenverbindung** anstelle einer **Ordner-Datenverbindung** verwenden.*


Auswählen von Daten aus einer Tabelle

Sie können Daten aus einer größeren Anzahl von Datendateien auswählen:

4 Laden von Daten mit dem Datenladeskript

- Textdateien, in denen die Daten in den Feldern durch Trennzeichen wie Kommas, Tabulator oder Strichpunkte getrennt sind (Comma-Separated-Variable-Dateien (CSV)).
- HTML-Tabellen
- XML-Dateien
- KML-Dateien
- Native Dateiformate von Qlik QVD und QVX
- Dateien mit fester Datensatzlänge
- DIF-Dateien (Data Interchange Format)

Gehen Sie folgendermaßen vor:

1. Klicken Sie in einer Ordner-Datenverbindung im Dateneditor auf .
Der Dateiauswahldialog wird angezeigt.
2. Achten Sie darauf, dass das entsprechende Dateiformat unter **Dateiformat** ausgewählt wird.
3. Achten Sie darauf, dass in der Datei die richtigen Einstellungen verwendet werden. Die Dateieinstellungen unterscheiden sich je nach Dateityp.
4. Wählen Sie durch Aktivieren der Kästchen neben den einzelnen Feldern die Felder aus, die Sie laden möchten. Sie können auch alle Felder einer Datei auswählen, indem Sie das Kästchen neben dem Arbeitsblattnamen aktivieren.



Sie können den Feldnamen bearbeiten, indem Sie auf einen vorhandenen Feldnamen klicken und einen neuen Namen eingeben. Dies kann sich darauf auswirken, wie die Tabelle mit anderen Tabellen verlinkt ist, da sie standardmäßig über gemeinsame Felder verbunden werden.

5. Wenn Sie die Daten ausgewählt haben, gehen Sie folgendermaßen vor:
 - Klicken Sie auf **Skript einfügen**.
Das Datenauswahlfenster wird geschlossen und die LOAD /SELECT-Befehle werden im Skript gemäß Ihren Auswahlen eingefügt.

Auswählen von Einstellungen für Dateitypen

Tabellen mit Trennzeichen

Diese Einstellungen werden für Tabellen mit Trennzeichen validiert, die eine einzelne Tabelle enthalten, in der jeder Datensatz durch eine Zeilenschaltung getrennt ist und jedes Feld durch ein Trennzeichen getrennt wird, wie zum Beispiel in einer CSV-Datei.

Dateiformat

Wählen Sie **Mit Trennzeichen** oder **Feste Satzlängen**.

Wenn Sie eine Auswahl treffen, spiegelt sich dies im Dialog "Daten auswählen" wider.

4 Laden von Daten mit dem Datenladeskript

Feldnamen	Wählen Sie, ob die Tabelle Eingebettete Feldnamen oder Keine Feldnamen enthält.
Delimiter	Legen Sie das Trennzeichen für die Tabelle fest.
Anführungszeichen	Legen Sie fest, wie Anführungszeichen verarbeitet werden sollen: Keine = Anführungszeichen sind nicht zulässig Standard = standardmäßige Verwendung von Anführungszeichen (die Anführungszeichen können als erste und letzte Zeichen eines Feldwerts verwendet werden) MSQ = Modern Style Quoting (Anführungszeichen in modernem Stil, die mehrzeiligen Inhalt in Feldern erlauben)
Größe des Headers	Stellen Sie die Zeilenzahl ein, die für den Tabellen-Header ausgelassen werden soll.
Zeichensatz	Wählen Sie den in den Tabellen verwendeten Zeichensatz aus.
Kommentar	Datendateien können Kommentare zwischen Datensätzen enthalten. Sie sind am Zeilenanfang durch ein oder zwei Sonderzeichen wie z. B. // gekennzeichnet. Geben Sie ein oder mehrere Zeichen zur Kennzeichnung von Kommentarzeilen ein. Zeilen, die mit dem/den hier angegebenen Zeichen beginnen, werden von Qlik Sense nicht geladen.
EOF ignorieren	Wählen Sie EOF ignorieren , wenn bei Ihren Daten Dateiende-Zeichen als Bestandteile des Feldwerts enthalten sind.

Datendateien mit fester Datensatzlänge

Datendateien mit fester Datensatzlänge enthalten eine einzige Tabelle, in der jeder Datensatz (jede Datenzeile) Spalten mit fester Feldgröße – üblicherweise mit Leer- oder Tabulatorzeichen – enthält.

Sie können die Positionen der Feldumbrüche auf zwei Weisen festlegen:

- Geben Sie die Feldumbruchpositionen durch Kommas voneinander getrennt manuell in **Position der Feldumbrüche** ein. Jede Position markiert den Anfang eines Felds.

Beispiel: 1,12,24

- Aktivieren Sie **Feldumbrüche**, um die Positionen der Feldumbrüche interaktiv in der Felddatenvorschau zu bearbeiten. **Positionen der Feldumbrüche** wird mit den ausgewählten Positionen aktualisiert. Sie haben folgende Möglichkeiten:
 - Klicken Sie in die Felddatenvorschau, um einen Feldumbruch einzufügen.
 - Klicken Sie auf einen Feldumbruch, um ihn zu löschen.
 - Ziehen Sie einen Feldumbruch, um ihn zu verschieben.

4 Laden von Daten mit dem Datenladeskript

Feldnamen	Wählen Sie, ob die Tabelle Eingebettete Feldnamen oder Keine Feldnamen enthält.
Größe des Headers	Stellen Sie die Größe des Headers auf die Zeilenzahl ein, die für den Tabellen-Header ausgelassen werden soll.
Zeichensatz	Wählen Sie den in der Tabellendatei verwendeten Zeichensatz aus.
Tab-Länge	Wählen Sie die Anzahl der Leerzeichen aus, für die ein Tabulatorzeichen in den Tabellen stehen soll.
Satzlänge	Wählen Sie die Anzahl der Zeilen aus, über die ein Datensatz in der Tabellendatei angezeigt werden soll. Standard ist 1.

HTML-Dateien

HTML-Dateien können mehrere Dateien enthalten. Qlik Sense interpretiert alle Elemente mit einem <TABLE>-Tag als Tabelle.

Feldnamen	Wählen Sie, ob die Tabelle Eingebettete Feldnamen oder Keine Feldnamen enthält.
Zeichensatz	Wählen Sie den in den Tabellen verwendeten Zeichensatz aus.

XML-Dateien

Sie können Daten laden, die im XML-Format gespeichert sind.

Es gibt keine speziellen Dateiformateinstellungen für XML-Dateien.

QVD-Dateien

Sie können Daten laden, die im QVD-Format gespeichert sind. QVD ist ein Qlik-spezifisches Dateiformat und es kann nur aus Qlik Sense oder QlikView heraus generiert und nur von diesen eingelesen werden. Das Dateiformat ist für besonders schnelles Laden aus einem Qlik Sense-Skript optimiert, aber gleichzeitig sehr kompakt.

Es gibt keine speziellen Dateiformateinstellungen für QVD-Dateien.

QVX-Dateien

Sie können Daten laden, die im Qlik data eXchange (QVX)-Format gespeichert sind. QVX-Dateien werden mit benutzerdefinierten Konnektoren erzeugt, die mit dem Qlik QVX SDK entwickelt wurden.

Es gibt keine speziellen Dateiformateinstellungen für QVX-Dateien.

KML-Dateien

Sie können Kartendateien laden, die im KML-Format gespeichert sind, und diese in Kartenvisualisierungen verwenden.

Es gibt keine speziellen Dateiformateinstellungen für KML-Dateien.

Anzeigen einer Skriptvorschau

Die einzufügenden Befehle werden in der Skriptvorschau angezeigt. Sie können diese ausblenden, indem Sie auf **Skriptvorschau anzeigen** klicken.

Einbinden von LOAD-Befehlen

Wenn **Load-Befehl einbinden** ausgewählt ist, werden SELECT-Befehle generiert, denen LOAD-Befehle vorangestellt sind und bei denen die SELECT-Befehle als Input verwendet werden.



Wenn Sie Felder in einer Tabelle umbenennen, wird unabhängig von dieser Einstellung automatisch ein LOAD-Befehl eingefügt.

Bearbeiten des Datenladeskripts

Sie schreiben das Skript im Texteditor des Dateneditors. Hier können Sie manuelle Änderungen an den **LOAD/SELECT**-Befehlen vornehmen, die Sie mit den Datenauswahl-Popups generiert haben, und ein neues Skript eingeben.

Die einzelnen Elemente des Skripts, das mithilfe der Qlik Sense-Skriptsyntax geschrieben werden muss, lassen sich durch die Darstellung in verschiedenen Farben leicht unterscheiden. Kommentare sind grün hervorgehoben, während Qlik Sense-Syntax-Schlüsselwörter blau hervorgehoben sind. Die Zeilen sind durchnummeriert.

Es gibt im Editor eine Reihe von Funktionen, die Sie beim Entwickeln des Ladeskripts unterstützen. Sie werden in diesem Abschnitt beschrieben.

Zugriff auf die Syntaxhilfe für Befehle und Funktionen

Die detaillierte Syntaxhilfe für ein Qlik Sense-Syntax-Schlüsselwort kann auf mehrere Arten aufgerufen werden:

Zugriff auf das Hilfe-Portal

Sie haben zwei Möglichkeiten, um die detaillierte Hilfe im Qlik Sense-Hilfe-Portal aufzurufen.

- Klicken Sie in der Symbolleiste auf , um den Syntaxhilfe-Modus aufzurufen. Im Syntaxhilfe-Modus können Sie auf ein Syntax-Schlüsselwort (blau markiert und unterstrichen) klicken, um die Syntaxhilfe aufzurufen.



Das Skript kann im Syntaxhilfe-Modus nicht bearbeitet werden.

- Platzieren Sie den Mauszeiger auf oder am Ende des Schlüsselworts und drücken Sie Strg+H.

Verwenden der Auto-Vervollständigungsfunktion

Wenn Sie anfangen, ein Qlik Sense-Skript-Schlüsselwort einzugeben, wird eine Auto-Vervollständigungsliste passender Schlüsselwörter angezeigt, aus denen Sie wählen können. Die Auswahl wird immer weiter eingegrenzt, während Sie weiter tippen.

Gehen Sie folgendermaßen vor:

1. Geben Sie die ersten Buchstaben des Funktionsnamens ein.
2. Wählen Sie ein Schlüsselwort aus der Liste aus, indem Sie darauf klicken oder die Eingabetaste drücken.

Verwenden der Popup-Hilfe für Funktionen

Wenn Sie nach einer Qlik Sense-Skriptfunktion eine öffnende Klammer eingeben, wird eine Popup-Hilfe mit der Syntax der Funktion angezeigt, einschließlich Parametern, Rückgabewerttypen und zusätzlichen Befehlen.

Einfügen eines vorbereiteten Test-Skripts

Sie können ein vorbereitetes Testskript einfügen, das einen Satz von Inline-Datenfeldern lädt. Auf diese Weise können Sie schnell einen Datensatz zu Testzwecken erstellen.

Gehen Sie folgendermaßen vor:





- Drücken Sie *Ctrl+00*

Der Test-Skriptcode wird ins Skript eingefügt.

Einrücken von Code

Sie können den Code zur Verbesserung der Lesbarkeit einrücken.

Gehen Sie folgendermaßen vor:

1. Wählen Sie einen oder mehrere Zeilen aus, deren Einrückung geändert werden soll.
2. Klicken Sie auf  , um den Text nach rechts zu rücken (Einzug vergrößern), oder klicken Sie auf  , um den Text nach links zu rücken (Einzug verkleinern).



Sie können auch Tastenkombinationen verwenden:

Tabulator (Einzug vergrößern)

Umschalt+Tabulator (Einzug verkleinern)




Text suchen und ersetzen


Sie können in allen Skriptabschnitten Text suchen und ersetzen.

Text suchen

Daten im Dateneditor öffnen Gehen Sie folgendermaßen vor:

4 Laden von Daten mit dem Datenladeskript

1. Klicken Sie in der Symbolleiste auf .
Der Dropdown-Dialog "Suchen" wird angezeigt.
2. Geben Sie in das Suchfeld den gewünschten Suchtext ein.
Die Suchergebnisse werden im aktuellen Abschnitt des Skriptcodes angezeigt. Neben der Abschnittsbezeichnung wird außerdem die Anzahl der im Text gefundenen Instanzen angegeben.
3. Navigieren Sie durch die Ergebnisse, indem Sie auf  und  klicken.




Klicken Sie in der Symbolleiste auf , um den Dialog "Suchen" zu schließen.




*Sie können außerdem **In allen Abschnitten suchen** auswählen, damit alle Skriptabschnitte durchsucht werden. Neben jeder Abschnittsbezeichnung wird die Anzahl der im Text gefundenen Instanzen angegeben. Groß- und Kleinschreibung wird bei der Suche nicht berücksichtigt.*

Text ersetzen

Gehen Sie folgendermaßen vor:

1. Klicken Sie in der Symbolleiste auf .
Der Dropdown-Dialog "Suchen" wird angezeigt.
2. Geben Sie den gesuchten Text im Feld "Suchen" ein.
3. Geben Sie den Ersetzungstext im Feld "Ersetzen" ein und klicken Sie auf **Ersetzen**.
4. Klicken Sie auf , um die nächste Instanz des gesuchten Texts aufzurufen und führen Sie eine der folgenden Aktionen aus:
 - Klicken Sie auf **Ersetzen**, um Text zu ersetzen.
 - Klicken Sie auf , um die nächste Instanz zu finden.

Klicken Sie in der Symbolleiste auf , um den Dialog "Suchen" zu schließen.



*Sie können auch auf **Alle in Abschnitt ersetzen** klicken, um alle Instanzen des Suchtexts im aktuellen Skriptabschnitt zu ersetzen. Bei der Ersetzen-Funktion wird die Groß- und Kleinschreibung beachtet und der ersetzte Text wird so geschrieben, wie er im Ersetzen-Feld angegeben wurde. Eine Meldung mit Informationen dazu wird angezeigt, wie viele Instanzen ersetzt wurden.*


Kommentieren im Skript

Sie können Kommentare in den Skriptcode einfügen oder Teile des Skriptcodes durch Auskommentieren deaktivieren. Sämtlicher Text, der in einer Zeile rechts von // (zwei Schrägstrichen) steht, wird als Kommentar betrachtet und beim Ausführen des Skripts nicht berücksichtigt.

In der Symbolleiste des Dateneditors befindet sich eine Verknüpfung für das Ein- und Auskommentieren von Code. Diese Funktion fungiert als Schalter. Das bedeutet, wenn der markierte Code auskommentiert ist, wird er kommentiert und umgekehrt.

Auskommentieren


Gehen Sie folgendermaßen vor:

1. Wählen Sie eine oder mehrere Zeilen des Codes aus, die nicht auskommentiert sind, oder platzieren Sie den Cursor an den Anfang einer Zeile.
2. Klicken Sie auf  oder drücken Sie Strg+K.

Der ausgewählte Code ist jetzt auskommentiert und deaktiviert.

Einkommentieren

Gehen Sie folgendermaßen vor:

1. Wählen Sie eine oder mehrere Zeilen des Codes aus, die auskommentiert sind oder platzieren Sie den Cursor an den Anfang einer auskommentierten Zeile.
2. Klicken Sie auf  oder drücken Sie Strg+K.

Der markierte Code wird nun ausgeführt.



Es gibt weitere Möglichkeiten, um Kommentare in den Skript-Code einzufügen:

- Verwenden des **Rem**-Befehls.
- Das Umschließen eines Code-Abschnitts durch `/*` und `*/`.

Beispiel:

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

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

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

Entsperren eines automatisch generierten Abschnitts

Wenn Sie mittels **Daten hinzufügen** Daten eingegeben haben, wird im Datenladeskript ein Abschnitt mit dem Namen **Automatisch generierter Abschnitt** angezeigt, der den zum Laden der Daten erforderlichen Skriptcode enthält. Das automatisch generierte Skript ist schreibgeschützt, Sie können den Schutz aber aufheben. Das bedeutet, dass Sie zum Definieren dieses Teils Ihres Datenmodells vom **Datenmanager** zum Datenladeskript wechseln. Dies kann nützlich sein, wenn Sie komplexere Datenoperationen ausführen müssen, um das für Ihre App benötigte Datenmodell zu erhalten.



*Wenn Sie den Schutz des automatisch generierten Skripts aufgehoben haben, können Sie die Auswahl dieser Daten nicht mehr länger mithilfe des **Datenmanagers** bearbeiten. Falls Sie mithilfe des **Datenmanagers** neue Daten hinzufügen, wird ein neuer, automatisch generierter Skriptabschnitt erstellt.*

Gehen Sie folgendermaßen vor:

4 Laden von Daten mit dem Datenladeskript

1. Wählen Sie den Abschnitt mit dem Namen **Automatisch generierter Abschnitt** aus.
2. Klicken Sie auf **Entsperren**.

Sie können jetzt das gesamte Datenladeskript bearbeiten, um Ihr Datenmodell anzupassen.



Es wird empfohlen, den Abschnitt umzubenennen, nachdem Sie ihn entsperrt haben.

Gesamten Code auswählen

Sie können den gesamten Code im aktuellen Skriptabschnitt auswählen.

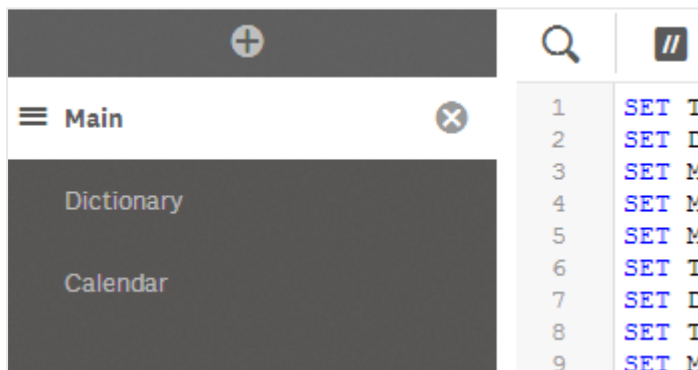
Gehen Sie folgendermaßen vor:

- Drücken Sie **Strg+A**

Der gesamte Skriptcode im aktuellen Abschnitt wird ausgewählt.

Gliedern des Skriptcodes

Sie können das Skript in Abschnitte aufteilen, um die Struktur zu gliedern. Das Skript wird in der Reihenfolge der Abschnitte von oben nach unten ausgeführt. Ein Skript muss mindestens einen Abschnitt enthalten.



Erstellen eines neuen Skriptabschnitts

Sie können neue Skriptabschnitte einfügen, um den Code zu gliedern.

Gehen Sie folgendermaßen vor:

- Klicken Sie auf **+**.

Der neue Abschnitt wird nach dem derzeit ausgewählten Abschnitt eingefügt.

Löschen eines Skriptabschnitts


Sie können einen Skriptabschnitt, einschließlich des gesamten Codes im Abschnitt, löschen.



Das Löschen eines Skriptabschnitts kann nicht rückgängig gemacht werden.

4 Laden von Daten mit dem Datenladeskript

Gehen Sie folgendermaßen vor:

- Klicken Sie auf  neben der Abschnittsdialogseite, um den Abschnitt zu löschen. Sie müssen den Löschvorgang bestätigen.
Der Abschnitt wird gelöscht.

Umbenennen eines Skriptabschnitts

Sie können einen Skriptabschnitt umbenennen.

Gehen Sie folgendermaßen vor:


1. Klicken Sie auf den Namen des Abschnitts und geben Sie einen anderen Namen ein.
2. Drücken Sie die Eingabetaste oder klicken Sie außerhalb des Abschnitts, wenn Sie fertig sind.

Der Abschnitt wird umbenannt.

Reihenfolge der Skriptabschnitte ändern

Sie können die Reihenfolge von Abschnitten ändern, um die Reihenfolge der Skriptausführung anzupassen.

Gehen Sie folgendermaßen vor:


1. Wählen Sie den Abschnitt aus, den Sie verschieben möchten.
2. Platzieren Sie den Cursor auf  und ziehen Sie den Abschnitt, um die Reihenfolge zu ändern.

Die Reihenfolge der Abschnitte wird geändert.

Debuggen des Datenladeskripts

Sie können die Debugging-Hilfsprogramme im Dateneditor verwenden, um Ihr Skript mithilfe von Haltepunkten Schritt für Schritt ausführen zu lassen. Auf diese Weise können Sie die Werte von Variablen und die Ausgabe der Skriptausführung überprüfen. Sie haben die Auswahlmöglichkeit, sich bestimmte oder alle **Ausgaben**, **Variablen** und **Haltepunkte** anzusehen.

So rufen Sie das Debugger-Fenster auf:

- Klicken Sie in der Symbolleiste des Dateneditors auf .
Das Debugger-Fenster wird unten im Dateneditor geöffnet.




Während der Debug-Modus ausgeführt wird, können Sie keine Verbindungen erstellen oder bearbeiten, keine Daten auswählen und das Skript nicht speichern oder ausführen. Der Debug-Modus beginnt mit dem Start der Debug-Ausführung, bis das Skript ausgeführt oder die Ausführung beendet wurde.


Debug-Symbolleiste


Das Debug-Fenster des Dateneditors enthält eine Symbolleiste mit den folgenden Optionen zur Steuerung der Debug-Ausführung:

4 Laden von Daten mit dem Datenladeskript

Datensätze max.	<p>Durch die Aktivierung dieser Beschränkung wird festgelegt, wie viele Zeilen mit Daten aus jeder Datenquelle geladen werden. Diese Option kann die Ausführungszeit verkürzen, wenn Sie mit großen Datenquellen arbeiten.</p> <p>Geben Sie die Anzahl der zu ladenden Zeilen ein.</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <i>Dies gilt nur für physische Datenquellen. Automatisch generierte und Inline-Loads werden beispielsweise nicht beschränkt.</i></div>
▶	Starten Sie die Ausführung im Debug-Modus bzw. fahren Sie mit ihr fort, bis der nächste Haltepunkt erreicht wird.
▶▶	Fahren Sie mit der nächsten Codezeile fort.
■	Beenden Sie die Ausführung hier. Wenn Sie beenden, bevor der gesamte Code ausgeführt wurde, enthält das resultierende Datenmodell nur Daten bis zu der Codezeile, bei der die Ausführung beendet wurde.

Ausgabe

In der **Ausgabe** werden alle Meldungen angezeigt, die während der Debug-Ausführung generiert wurden. Sie können wahlweise festlegen, dass das Scrollen der Ausgabe beim Anzeigen neuer Meldungen verhindert wird. Klicken Sie dazu auf .

Das Ausgabemenü () umfasst außerdem die folgenden Optionen:



Löschen	Klicken Sie auf diese Option, um alle Ausgabemeldungen zu löschen.
Gesamten Text auswählen	Klicken Sie auf diese Option, um alle Ausgabemeldungen auszuwählen.
Nach ganz unten scrollen	Klicken Sie auf diese Option, um zur letzten Ausgabemeldung zu scrollen.

Variablen

Über **Variablen** werden alle reservierten Variablen, Systemvariablen und im Skript definierten Variablen aufgelistet und die bei der Skriptausführung aktuellen Werte werden angezeigt.

Festlegen einer Variable als Favorit

Wenn Sie während der Ausführung bestimmte Variablen überprüfen möchten, können Sie diese als Favoriten festlegen. Ihre Favoriten der Variablen werden ganz oben in der Variablenliste angezeigt und durch einen gelben Stern markiert. Sie können Variablen folgendermaßen als Favorit festlegen:

- Klicken Sie neben der Variable auf .
- Das Symbol  erscheint jetzt in Gelb und die Variable oben in der Variablenliste.

Filtern von Variablen

Sie können einen Filter anwenden, damit nur der ausgewählte Variablentyp angezeigt wird. Verwenden Sie dazu die folgenden Optionen im Variablenmenü (☰):


Alle Variablen anzeigen	Klicken Sie auf diese Option, um alle Variablen anzuzeigen.
Systemvariablen anzeigen	Klicken Sie auf diese Option, um Systemvariablen anzuzeigen. Systemvariablen werden von Qlik Sense definiert, Sie können den Wert der Variablen jedoch im Skript ändern.
Reservierte Variablen anzeigen	Klicken Sie auf diese Option, um reservierte Variablen anzuzeigen. Reservierte Variablen werden von Qlik Sense definiert und der Wert kann nicht geändert werden.
Benutzerdefinierte Variablen anzeigen	Klicken Sie auf diese Option, um benutzerdefinierte Variablen anzuzeigen. Benutzerdefinierte Variablen wurden von Ihnen im Skript definiert.

Haltepunkte

Sie können Ihrem Skript Haltepunkte hinzufügen, um die Debug-Ausführung in bestimmten Codezeilen anzuhalten und die Werte der Variablen und die Ausgabemeldungen an diesem Punkt zu überprüfen. Beim Erreichen eines Haltepunkts können Sie entweder die Ausführung anhalten, bis zum nächsten Haltepunkt fortfahren oder zur nächsten Codezeile springen. Alle Haltepunkte des Skripts werden aufgelistet und als Referenz sind Abschnitt und Zeilennummer angegeben.

Hinzufügen eines Haltepunkts

Sie können einen Haltepunkt folgendermaßen zu einer Codezeile hinzufügen:



- Klicken Sie im Skript auf den Bereich direkt rechts von der Zeilennummer, für die Sie einen Haltepunkt hinzufügen möchten.
Ein  neben der Zeilennummer gibt an, dass sich in dieser Zeile ein Haltepunkt befindet.




Sie können Haltepunkte auch bei geschlossenem Debug-Fenster hinzufügen.

Löschen von Haltepunkten

Sie können einen Haltepunkt mithilfe einer der folgenden Optionen löschen:

- Klicken Sie im Skript neben der gewünschten Zeilennummer auf .
- Klicken Sie in der Liste der Haltepunkte neben einem Haltepunkt auf .

Sie können auch auf  klicken und **Alle löschen** auswählen, um alle Haltepunkte aus dem Skript zu löschen.

Aktivieren und Deaktivieren von Haltepunkten

Wenn Sie einen Haltepunkt erstellen, wird dieser standardmäßig aktiviert. In der Liste der Haltepunkte wird dies neben dem Haltepunkt durch ✓ gekennzeichnet. Sie können einzelne Haltepunkte in der Liste der Haltepunkte aktivieren oder deaktivieren.

Die folgenden Optionen stehen Ihnen im Haltepunktemenü (☰) zur Verfügung:

- **Alle aktivieren**
- **Alle deaktivieren**

Speichern des Ladeskripts

Beim Speichern eines Skripts wird die gesamte App gespeichert, aber die Daten werden nicht automatisch erneut geladen.

Gehen Sie folgendermaßen vor:

- Klicken Sie in der Symbolleiste des Dateneditors auf **Speichern**, um das Skript zu speichern.

Das Skript wurde zwar gespeichert, allerdings enthält die App weiterhin alte Daten aus dem vorherigen Öffnen, was in der Symbolleiste angezeigt wird. Um die App mit neuen Daten zu aktualisieren, klicken Sie in der Symbolleiste des Dateneditors auf **Daten laden** (↻).

Bei der Speicherung eines Skripts wird dieses automatisch auf Syntaxfehler überprüft. Syntaxfehler werden im Code markiert und bei allen Skriptabschnitten mit Syntaxfehlern wird neben der Abschnittsbezeichnung ⓘ angezeigt.



Das Skript wird beim Laden von Daten automatisch in der App gespeichert.

Ausführen des Skripts zum Laden der Daten

Klicken Sie in der Symbolleiste auf **Daten laden** (↻), um das Skript auszuführen und die Daten erneut in die App zu laden. Die App wird automatisch vor dem Laden der Daten gespeichert.

Der Dialog **Datenladefortschritt** wird angezeigt und Sie können den Ladevorgang **abbrechen**. Wenn der Datenladevorgang abgeschlossen ist, wird der Dialog mit dem Status (**Erfolgreich abgeschlossen** oder **Laden der Daten fehlgeschlagen**) und einer Zusammenfassung möglicher Fehler und Warnungen, wie zum Beispiel für synthetische Schlüssel, aktualisiert. Die Zusammenfassung wird auch in der **Ausgabe** angezeigt, wenn Sie sie nach dem Schließen des Dialogs aufrufen möchten.



*Wenn Sie möchten, dass der Dialog **Datenladefortschritt** generell nach einer erfolgreichen Ausführung automatisch geschlossen wird, wählen Sie **Nach erfolgreicher Ausführung schließen aus**.*

4.3 Verständnis der Skript-Syntax und der Datenstrukturen

Extrahieren, Umformen und Laden

Allgemein kann die Art und Weise, wie Daten in die App geladen werden, über den Prozess für das Extrahieren, Umformen und Laden erklärt werden:

1. Extrahieren

Der erste Schritt ist die Extrahierung von Daten aus dem Datenquellensystem. Im Skript definieren Sie dies anhand der Befehle **SELECT** oder **LOAD**. Im Folgenden die Unterschiede zwischen diesen Befehlen:

- **SELECT** wählt Daten aus einer ODBC-Datenquelle oder einem OLE DB-Provider aus. Der Befehl **SELECT SQL** wird vom Datenprovider evaluiert und nicht von Qlik Sense.
- **LOAD** lädt Daten aus einer Datei, Daten, die im Skript definiert sind, aus einer zuvor geladenen Tabelle, von einer Webseite, aus dem Ergebnis eines nachfolgenden **SELECT**-Befehls oder durch automatisches Generieren der Daten.

2. Umformen

Beim Umformen werden die Daten mithilfe der Skriptfunktionen und Regeln zur Ableitung der gewünschten Datenmodellstruktur manipuliert. Zu den typischen Operationen zählen:

- Berechnung neuer Werte
- Konvertieren codierter Werte
- Felder umbenennen
- Zusammenfügen von Tabellen
- Aggregierungswerte
- Schwenken
- Datenvalidierung

3. Laden

Im letzten Schritt wird das Skript ausgeführt, um das definierte Datenmodell in die App zu laden.

Sie sollten ein Datenmodell erstellen, mit dem Sie die Daten in Qlik Sense effizient nutzen können. Im Regelfall sollten Sie sich für ein relativ normales Schema in Form von Stern oder Schneeflocke ohne Zirkelbezüge entscheiden. Hierbei handelt es sich um ein Modell, in dem jedes Element in einer getrennten Tabelle gespeichert wird. Ein häufig verwendetes Datenmodell sieht wie folgt aus:

- eine zentrale Tabelle mit Fakten, die Schlüssel zu den Dimensionen und Zahlen zur Berechnung der Kennzahlen enthält (z. B. Anzahl an Abschnitten, Umsatzzahlen und Budgetzahlen).
- umliegende Tabellen mit den Dimensionen mit allen Attributen (wie Produkte, Kunden, Kategorien, Kalender und Lieferanten).



In vielen Fällen ist es möglich, eine Aufgabe wie eine Aggregation entweder durch das Erstellen eines umfangreicheren Datenmodells im Ladeskript oder durch das Ausführen von Aggregationen in der Diagrammformel zu lösen. Als Faustregel ist die Performance besser, wenn Sie die Datentransformationen auf das Ladeskript beschränken.



Eine gute Methode ist es, sich das Datenmodell vorher auf Papier zu skizzieren. Sie können dadurch die Struktur besser definieren, welche Daten extrahiert und welche Transformationen durchgeführt werden sollen.

Befehle zum Laden von Daten

Daten werden durch **LOAD**- oder **SELECT**-Befehle eingelesen. Jeder dieser Befehle erstellt eine interne Tabelle. Eine Tabelle ist immer eine Liste von Objekten. Jeder Datensatz (Zeile) ist ein neues Exemplar des Objekttyps und jedes Feld (Spalte) ist ein spezifisches Attribut oder eine Eigenschaft des Objekts.

Im Folgenden die Unterschiede zwischen diesen Befehlen:

- **SELECT** wählt Daten aus einer ODBC-Datenquelle oder einem OLE DB-Provider aus. Der Befehl **SELECT SQL** wird vom Datenprovider evaluiert und nicht von Qlik Sense.
- **LOAD** lädt Daten aus einer Datei, Daten, die im Skript definiert sind, aus einer zuvor geladenen Tabelle, von einer Webseite, aus dem Ergebnis eines nachfolgenden **SELECT**-Befehls oder durch automatisches Generieren der Daten.

Regeln

Beim Laden von Daten in Qlik Sense gelten die folgenden Regeln:

- Qlik Sense unterscheidet nicht zwischen Tabellen, die durch **LOAD**- oder **SELECT**-Befehle entstehen. Wenn mehrere Tabellen geladen werden, spielt es daher keine Rolle, ob die Daten durch **LOAD**, **SELECT** oder eine Kombination beider Befehle geladen wurden.
- Die Reihenfolge der Felder in der zugrunde liegenden Datenbank ist für Qlik Sense nicht relevant.
- Feldnamen dienen dazu, Felder zu identifizieren und Verknüpfungen herzustellen. Groß-/Kleinschreibung muss beachtet werden. Daher ist es häufig erforderlich, Felder im Skript umzubenennen.

Die Ausführung des Skripts

Bei einem typischen **LOAD**- oder **SELECT**-Befehl werden die Arbeitsschritte in folgender Reihenfolge ausgeführt:

1. Auswertung von Formeln
2. Umbenennen von Feldern mit **as**
3. Umbenennen von Feldern mit **alias**

4. Qualifizierung von Feldern
5. Daten-Mapping bei Übereinstimmung des Feldnamens
6. Speichern der internen Tabelle

Felder

Felder sind die primären datentragenden Elemente in Qlik Sense. Felder enthalten normalerweise eine Reihe von Werten, sogenannte Feldwerte. Wir sprechen davon, dass die von Qlik Sense verarbeiteten Daten aus Datendateien stammen. Eine Datei besteht aus mehreren Feldern, in denen jeder Eintrag ein Datensatz ist. Die Begriffe Datei, Feld und Datensatz sind gleichbedeutend mit Tabelle, Spalte bzw. Zeile. Die Qlik Sense-AQL-Logik arbeitet anhand der Felder und Werte.

Felddaten werden über die Skript-Befehle **LOAD**, **SELECT** oder **Binary** abgerufen. Die einzige Möglichkeit, neue Felder anzulegen oder die Werte eines Felds zu verändern, ist die erneute Ausführung des Skripts. Weder über die Automation-Schnittstelle noch über die Eigenschaftsdialoge des Dokuments oder der Objekte können Werte von Feldern geändert werden. Nach dem Einlesen in Qlik Sense dienen Werte zur Anzeige und Recherche von Informationen und zur Berechnung statistischer Kennzahlen aller Art.

Felder enthalten numerische oder alphanumerische Daten (Text). Numerische Werte werden in zwei verschiedenen Formen gespeichert: als Zahl und als String. Die Zahl wird für Berechnungen benötigt, der String repräsentiert das Zahlenformat, das Sie in Listboxen und anderen Objekten angezeigt sehen.

Der Inhalt eines Felds kann in einem Filterfenster dargestellt werden.

Abgeleitete Felder

Wenn Feldgruppen miteinander in Bezug stehen oder Felder Informationen enthalten, die in kleinere, bei der Erstellung von Dimensionen und Kennzahlen relevante Teile aufgeschlüsselt werden können, lassen sich Felddefinitionen zum Generieren der abgeleiteten Felder erstellen. Ein Beispiel wäre ein Datumsfeld, aus dem sich mehrere Attribute, wie Jahr, Monat, Kalenderwoche oder Wochentag ableiten lassen. Alle diese Attribute können in einer Dimensionsformel mithilfe der Datumsfunktionen von Qlik Sense berechnet werden. Alternativ lässt sich jedoch eine gemeinsame Kalenderdefinition für alle Felder des Datumstyps erstellen. Felddefinitionen werden im Datenladeskript gespeichert.

Definieren der Kalenderfelddefinitionen

Sie verwenden den **Declare**-Befehl, um eine Definition für die abgeleiteten Felder zu erstellen. Hier legen Sie die verschiedenen Attribute des Felds fest – in diesem Fall Attribute mit Datumsbezug. Jedes Feld wird folgendermaßen beschrieben: `<expression> As field_name tagged tag`. Das Festlegen von einem oder mehreren Tags ist optional, kann sich jedoch auf die Sortierreihenfolge des abgeleiteten Felds auswirken. Verwenden Sie \$1 zur Bezugnahme auf das Datenfeld, aus dem die abgeleiteten Felder generiert werden sollen.

Calendar:

```
DECLARE FIELD DEFINITION TAGGED '$date'  
  Parameters  
    first_month_of_year = 1  
  Fields  
    Year($1) As Year Tagged ('$numeric'),  
    Month($1) as Month Tagged ('$numeric'),  
    Date($1) as Date Tagged ('$date'),
```

4 Laden von Daten mit dem Datenladeskript

```
week($1) as week Tagged ('$numeric'),
weekday($1) as weekday Tagged ('$numeric'),
DayNumberOfYear($1, first_month_of_year) as DayNumberOfYear Tagged ('$numeric')
;
```

Zuordnen von Datenfeldern zum Kalender mithilfe von **Derive**

Der nächste Schritt ist die Verwendung des **Derive**-Befehls für die Zuordnung der vorhandenen Datenfelder zum Kalender. Dadurch werden die abgeleiteten Felder erstellt. Sie können dies auf drei verschiedene Möglichkeiten im Datenladeskript erreichen.

- Zuordnung der spezifischen Felder nach Feldname.
`DERIVE FIELDS FROM FIELDS OrderDate,ShippingDate USING Calendar;`
- Zuordnung aller Felder mit einem oder mehreren spezifischen Tags für Felder.
`DERIVE FIELDS FROM EXPLICIT TAGS '$date' USING Calendar;`
- Zuordnung aller Felder, die mit einem der Tags der Felddefinition versehen sind (\$date in unserem Beispiel).
`DERIVE FIELDS FROM IMPLICIT TAG USING Calendar;`

In diesem Fall können Sie eines von den dreien der hier aufgeführten Beispiele verwenden.

Verwenden der abgeleiteten Datumsfelder in einer Visualisierung

Qlik Sense erkennt abgeleitete Datumsfelder, wenn Sie eine Kalenderdefinition erstellt haben und die Felder wie in diesem Beispiel zugeordnet haben. Sie sind im Abschnitt **Datums- und Uhrzeitfelder** des Extras-Fensters **Felder** verfügbar. Alle abgeleiteten Felder sind auch im Formel-Editor und bei der Erstellung und Bearbeitung von Dimensionen zu finden.

Tags für Felder

Mithilfe von Feldtags können den Feldern in Ihrem Datenmodell Metadaten hinzugefügt werden. Es gibt zwei verschiedene Arten von Feldtags:

- Systemfeld-Tags
Systemfeld-Tags werden automatisch generiert, wenn das Skript ausgeführt und die Daten geladen werden. Manche der Tags können im Skript geändert werden. System-Tags geht stets ein \$-Zeichen voran.
- Benutzerdefinierte Feldtags
Sie können mithilfe des **Tag**-Befehls den Feldern im Datenladeskript benutzerdefinierte Feldtags hinzufügen. Benutzerdefinierte Tags können niemals denselben Namen wie System-Tags haben.

Systemfelder

Zusätzlich zu den aus der Datenquelle extrahierten Feldern werden von Qlik Sense automatisch Systemfelder angelegt. Diese Felder beginnen alle mit "\$" und können wie gewöhnliche Felder in Visualisierungen wie dem Filterfenster oder einer Tabelle angezeigt werden. Die Systemfelder werden in der Regel während der Skriptaufführung erzeugt und dienen zur Unterstützung bei der App-Gestaltung.

4 Laden von Daten mit dem Datenladeskript



Systemfelder sind in den Feldlisten des Extras-Fensters oder im Formel-Editor nicht enthalten. Wenn Sie ein Systemfeld verwenden möchten, müssen Sie es durch manuelle Eingabe referenzieren.

Beispiel: =\$Field

Verfügbare Systemfelder

Die folgenden Systemfelder sind verfügbar:

Tag	Beschreibung	Kann im Skript geändert werden.
\$system	Systemfeld, das von Qlik Sense während der Skriptauführung generiert wird.	Nein
\$key	Schlüsselfeld, das eine Verbindung zwischen zwei oder mehreren Tabellen darstellt.	Nein
\$keypart	Das Feld ist Teil eines synthetischen Schlüssels.	Nein
\$syn	Synthetischer Schlüssel	Nein
\$hidden	Verborgenes Feld, d. h., es wird in einer Feldauswahl bei der Erstellung von Visualisierungen, Dimensionen und Kennzahlen nicht angezeigt. Sie können verborgene Felder auch in Formeln verwenden, müssen aber den Feldnamen eintippen. Sie können mithilfe der Systemvariablen HidePrefix und HideSuffix festlegen, welche Felder ausgeblendet werden sollen.	Ja
\$numeric	Alle Werte des Felds, die nicht NULL sind, sind numerisch.	Ja
\$integer	Alle Werte des Felds, die nicht NULL sind, sind ganze Zahlen.	Ja
\$text	Das Feld enthält keine numerischen Werte.	Ja
\$ascii	Die Werte des Feldes enthalten nur ASCII-Zeichen.	Ja
\$date	Alle Werte des Felds, die nicht NULL sind, können als Datum (ganze Zahl) interpretiert werden.	Ja
\$timestamp	Alle Werte des Felds, die nicht NULL sind, können als Zeitstempel interpretiert werden.	Ja
\$Table	Zeigt alle internen Tabellen an, die vom Skript geladen wurden.	Nein

4 Laden von Daten mit dem Datenladeskript

Tag	Beschreibung	Kann im Skript geändert werden.
\$Field	Die Namen der eingelesenen Felder.	Nein
\$Fields	Die Anzahl der Felder in verschiedenen Tabellen.	Nein
\$Rows	Die Spaltennummern der Felder.	Nein
\$Info	Die Anzahl der Zeilen in den Tabellen.	Nein

Umbenennen von Feldern

Gelegentlich ist es sinnvoll oder auch notwendig, Felder umzubenennen, damit die gewünschten Verknüpfungen zustande kommen. Die drei Hauptgründe für das Umbenennen von Feldern sind:

1. Zwei Felder haben unterschiedliche Namen, obwohl sie das gleiche Objekt bezeichnen:

- Das Feld *ID* in der Tabelle *Customers*
- Das Feld *CustomerID* in der Tabelle *Orders*

Beide Felder bezeichnen einen spezifischen Kundenkennzeichnungs-Code und müssen den gleichen Namen tragen, z. B. *CustomerID*.

2. Zwei Felder tragen den gleichen Namen, bezeichnen jedoch unterschiedliche Objekte:

- Das Feld *Date* in der Tabelle *Invoices*
- Das Feld *Date* in der Tabelle *Orders*

Die beiden Felder sollten umbenannt werden, beispielsweise in *InvoiceDate* und *OrderDate*.

3. Es können Fehler in der Schreibweise in der Datenbank oder unterschiedliche Konventionen für Groß- und Kleinschreibung vorliegen.

Da Felder im Skript umbenannt werden, müssen die ursprünglichen Daten nicht geändert werden. Wie in den Beispielen gesehen, gibt es zwei unterschiedliche Arten zum Umbenennen von Feldern.

Beispiel 1: Verwenden des alias-Befehls

Dem **LOAD**- oder **SELECT**-Befehl kann ein **alias**-Befehl vorangestellt werden.

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

Beispiel 2: Verwenden des as-Spezifizierers

Der Befehl **LOAD** oder **SELECT** kann den **as**-Spezifizierer enthalten.

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

Interne Tabellen

Durch jeden **LOAD**- oder **SELECT**-Befehl wird eine Tabelle generiert. Normalerweise betrachtet Qlik Sense jede dieser Tabellen als eine einzelne interne Tabelle. Zu dieser Regel existieren allerdings Ausnahmen:

- Sollten durch zwei oder mehrere Befehle Tabellen entstehen, die in allen Feldnamen übereinstimmen, werden die Tabellen automatisch zu einer internen Tabelle zusammengefasst.
- Wenn **LOAD**- oder **SELECT**-Befehlen einer der nachstehenden Zusätze vorangeht, verarbeitet das Programm die Daten in besonderer Weise:

concatenate	Diese Tabelle wird mit einer anderen benannten Tabelle oder mit der internen Tabelle zusammengefügt (hinzugefügt), die zuvor als letzte Tabelle erstellt wurde.
crosstable	Diese Tabelle ist nicht pivotiert. Sie wurde also von einem Kreuztabellenformat in ein Spaltenformat konvertiert.
generic	Die Tabelle wird in mehrere interne Tabellen zerlegt.
intervalmatch	Die Tabelle (die genau zwei Spalten enthalten muss) wird zur Bildung von numerischen Intervallen verwendet, die der Gruppierung von diskreten Werten in einem spezifischen Feld dienen.
join	Diese Tabelle wird von Qlik Sense mit einer anderen benannten Tabelle oder mit der zuletzt erstellten internen Tabelle über die gemeinsamen Felder zusammengeschlossen.
keep	Diese Tabelle wird auf die Felder reduziert, über die sie gemeinsam mit einer anderen benannten Tabelle oder mit der zuletzt erstellten internen Tabelle verfügt.
mapping	Diese Tabelle (die genau zwei Spalten enthalten muss) wird als Mapping-Tabelle gelesen, die niemals mit anderen Tabellen verbunden wird.
semantic	Diese Tabelle wird nicht als interne Tabelle geladen, sondern als semantische Tabelle, die Beziehungen enthält, die nicht verbunden werden dürfen, z. B. Vorgänger, Nachfolger und andere Verweise auf Objekte des gleichen Typs.

Nach dem Einlesen der Daten erstellt das Programm automatisch logische Verknüpfungen zwischen den internen Tabellen.

Tabellennamen

Qlik Sense-Tabellen wird beim Speichern in der Qlik Sense-Datenbank ein Name zugewiesen. Die Tabellennamen können z. B. für **LOAD**-Befehle mit einer **resident**-Bedingung oder mit Formeln, welche die **peek**-Funktion enthalten, verwendet werden und werden im Layout im Systemfeld *\$Table* angezeigt.

Die Benennung von Tabellen gemäß folgenden Regeln:

4 Laden von Daten mit dem Datenladeskript

1. Geht den Befehlen **LOAD** oder **SELECT** ein Tabellename unmittelbar voran, so wird dieser Name verwendet. Dem Tabellennamen muss ein Doppelpunkt folgen.

Beispiel:

```
Table1:  
LOAD a,b from c.csv;
```

2. Ist kein Tabellename angegeben, wird der Datei- oder Tabellename benutzt, der in den Befehlen **FROM** oder **LOAD** unmittelbar auf das Schlüsselwort **SELECT** folgt. Dabei werden maximal 32 Zeichen berücksichtigt. Die Erweiterung wird übersprungen, wenn der Dateiname verwendet wird.
3. Tabellen, die durch Inline-Befehl geladen werden, erhalten den Namen **INLINExx**, wobei xx für eine Nummer steht. Die erste Inline-Tabelle erhält den Namen *INLINE01*.
4. Tabellen, die automatisch generiert werden, erhalten den Namen **AUTOGENERATExx**, wobei xx für eine Nummer steht. Die erste automatisch generierte Tabelle erhält den Namen *AUTOGENERATE01*.
5. Sollte ein aufgrund der genannten Regeln generierter Tabellename bereits existieren, wird der Name der zweiten Tabelle mit -x ergänzt, wobei x für eine laufende Nummer steht. Diese wird erhöht, bis Eindeutigkeit gewährleistet ist. Beispiel: *Budget*, *Budget-1* und *Budget-2*.

Das Programm speichert Tabellennamen in drei verschiedenen Bereichen ab: **section access**, **section application** und Mapping-Tabellen. In **section access** und **section application** generierte Tabellennamen werden separat behandelt. Wenn ein Tabellename innerhalb des Abschnitts nicht gefunden wird, sucht Qlik Sense auch im anderen Abschnitt. Mapping-Tabellen werden separat behandelt und stehen in keinem Zusammenhang zu Tabellennamen aus den beiden anderen Bereichen.

Tabellenbezeichnungen

Eine Tabelle kann zu späteren Referenzzwecken benannt werden, zum Beispiel durch einen **LOAD**-Befehl mit einer **resident**-Bedingung oder Formeln mit **peek**-Funktion. Der Name kann ein beliebiger String aus Buchstaben oder Ziffern sein und muss dem ersten **LOAD**-Befehl oder **SELECT**-Befehl vorangehen, der die Tabelle erstellt. Dem Tabellennamen muss ein Doppelpunkt ":" folgen.

Namen, die Leerzeichen enthalten, müssen in einfache oder doppelte Anführungszeichen oder eckige Klammern gesetzt werden.

Beispiel 1:

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

Beispiel 2: Die Bezeichnung enthält ein Leerzeichen

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

Verknüpfungen zwischen Tabellen

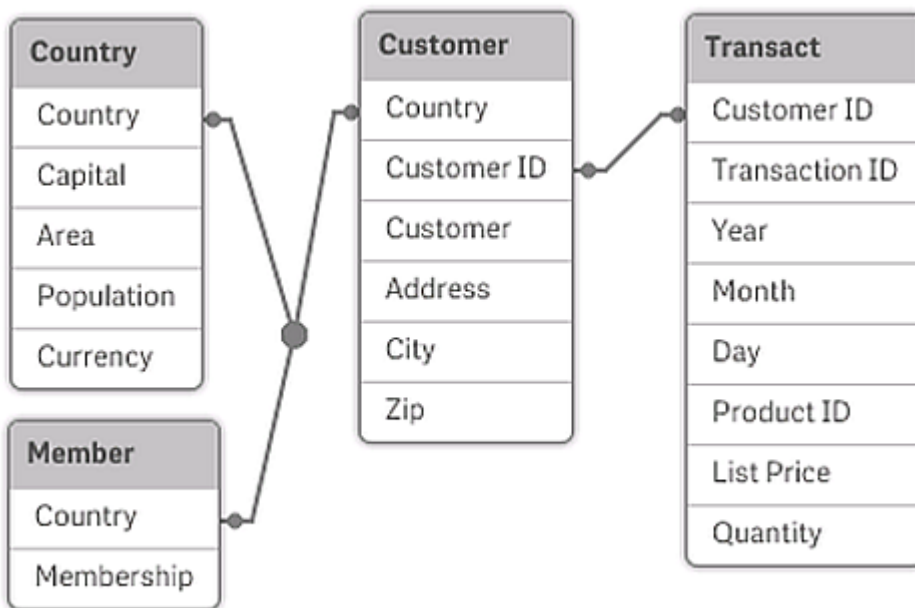
Datenbanken bestehen mitunter aus vielen Tabellen. Jede Tabelle kann als eine Liste aufgefasst werden, in der jede Zeile ein Objekt eines bestimmten Typs repräsentiert.

Beispiel:

Wenn in zwei Tabellen Objekte verschiedenen Typs aufgelistet werden, etwa eine Liste mit Kunden und eine Liste mit Rechnungen, und die beiden Listen haben ein Feld gemeinsam, z. B. Kundennummer, so besteht zumeist eine Verbindung zwischen den beiden Tabellen. SQL-Tools erkennen dies und schließen solche Tabellen durch einen Join zusammen.

Die im Qlik Sense-Skript definierten Tabellen werden interne Tabellen genannt. Qlik Sense erstellt automatisch eine Verknüpfung zwischen zwei Tabellen auf Basis der Feldnamen. Diese Verknüpfung wird aktiviert, indem die betreffenden Werte innerhalb des Dokuments ausgewählt werden, z. B. Auswahl in einem Filterfenster.

Somit ist eine Verknüpfung fast dasselbe wie ein Join. Der einzige Unterschied besteht darin, dass der Join bereits während der Ausführung des Skripts gemacht wird und als Ergebnis eine einzige interne Tabelle liefert, deren Daten aus verschiedenen Tabellen der Originaldatenquelle stammen. Verknüpfungen dagegen entstehen erst, nachdem die programminternen Tabellen angelegt sind.



Vier Tabellen: eine Länderliste, eine Kundenliste, eine Liste aller Transaktionen und eine der Mitgliedschaften, die einander über die Felder Country und CustomerID zugeordnet sind.

Qlik Sense-Verknüpfung im Vergleich zu SQL Natural Outer Join

Eine Qlik Sense-Verknüpfung hat Ähnlichkeit mit einem SQL Natural Outer Join. Die Verknüpfung ist jedoch noch umfassender: Ein Join in SQL ist normalerweise eine Projektion einer Tabelle auf eine andere und weist damit nur in eine Richtung. Die Verknüpfung dagegen entspricht einem vollständigen (bidirektionalen)

Join und weist in beide Richtungen.

Häufigkeiten in Schlüsselfeldern

Es gibt ein paar Beschränkungen bei der Verwendung der meisten assoziierten Felder, das heißt Felder, die bei zwei oder mehr Tabellen gleich sind. Wenn ein Feld in mehr als einer Tabelle vorkommt, weiß Qlik Sense nicht, welche Tabelle es für die Berechnung der Datenhäufigkeit verwenden soll.

Qlik Sense prüft zunächst, ob eine eindeutige Interpretation möglich ist. Das ist aber eher selten der Fall. Meistens kann das Programm nicht feststellen, in welcher Tabelle die Werte zur Ermittlung der Häufigkeiten gezählt werden müssen. Da ein Irrtum an dieser Stelle fatale Auswirkungen haben könnte, führt Qlik Sense Operationen, die mehrdeutige Ergebnissen bringen könnten, nicht aus.

Beschränkungen bei der Verknüpfung von Feldern

1. In dem zu dem Feld gehörigen Filterfenster können keine Häufigkeiten angezeigt werden.
2. Statistikboxen für diese Felder zeigen für die meisten statistische Größen "nicht verfügbar" an.
3. In Diagrammen können keine Formeln definiert werden, in denen Häufigkeiten von Schlüsselfeldern in die Berechnung eingehen (zum Beispiel die Funktionen Sum, Count und Average), es sei denn, die Option **Distinct** ist aktiviert. Nach jedem Ladevorgang prüft Qlik Sense, ob die Formeln in den Diagrammen noch eindeutige Ergebnisse liefern. Sollte das nicht der Fall sein, gibt das Programm eine Fehlermeldung aus, und die betreffende Formel wird deaktiviert. Sie kann erst wieder aktiviert werden, wenn Eindeutigkeit gegeben ist. Wird beim Ladevorgang eine log-Datei erstellt, sind dort die deaktivierten Formeln aufgelistet.

Problemumgehung

Diese Einschränkungen lassen sich leicht umgehen. Laden Sie das betreffende Feld noch einmal unter einem anderen Namen aus der Tabelle, aus der die Häufigkeiten ermittelt werden sollen. Dieses neue Feld können Sie für jegliche Operationen benutzen: Anzeige von Häufigkeiten, Statistiken, Diagramme usw.

Synthetische Schlüssel

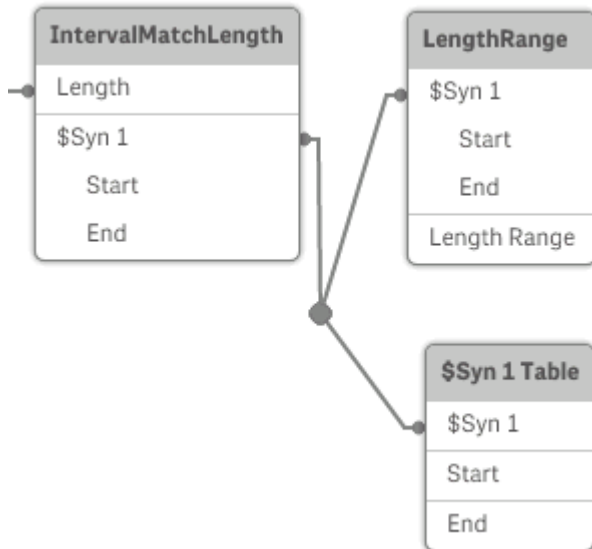
Wenn zwei oder mehr Datentabellen zwei oder mehr gemeinsame Felder haben, ist von einer zusammengesetzten Schlüsselbeziehung auszugehen. Um dies zu vermeiden, erstellt Qlik Sense automatisch synthetische Schlüssel. Synthetische Schlüssel enthalten alle möglichen Kombinationen der Werte zweier oder mehrerer Schlüsselfelder.

Wenn beim Laden von Daten ein Warnhinweis zu synthetischen Schlüsseln angezeigt wird, sollten Sie die Datenstruktur im Datenmodell anzeigen. Überprüfen Sie, ob das Datenmodell korrekt ist oder nicht. Manchmal ist es das, häufig besteht der synthetische Schlüssel jedoch wegen eines Fehlers im Skript.

Mehrere synthetische Schlüssel sind oft ein Anzeichen für ein fehlerhaftes Datenmodell, müssen dies jedoch nicht sein. Ein sicheres Zeichen für ein fehlerhaftes Datenmodell ist jedoch, wenn synthetische Schlüssel vorhanden sind, die auf anderen synthetischen Schlüsseln basieren.



Bei großen Datenmengen und komplizierten Tabellenstrukturen kann die Zahl solcher Kombinationen natürlich sehr groß werden, und es kann vorkommen, dass Qlik Sense sie nicht ohne Weiteres verarbeiten kann. In einem solchen Fall müssen Sie Ihr Skript überarbeiten und alle synthetischen Schlüssel entfernen.



Handhabung synthetischer Schlüssel

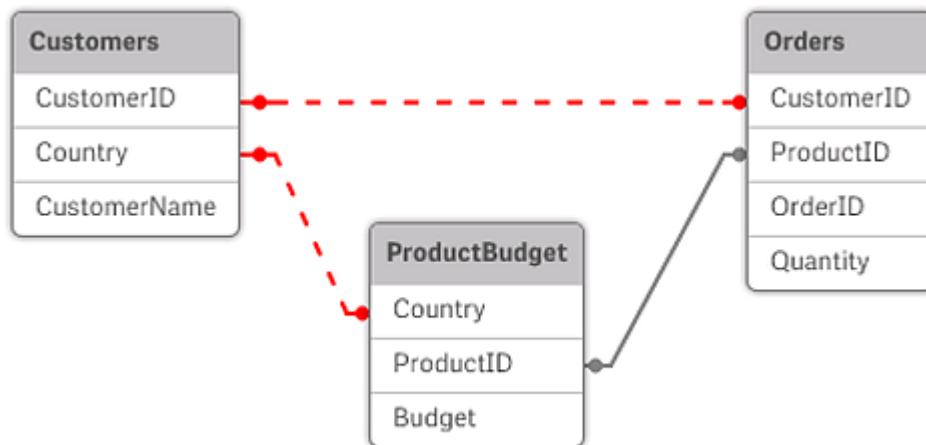
Wenn Sie synthetische Schlüssel vermeiden müssen, können Sie dies im Datenladeskript auf verschiedene Weise erreichen:

- Vergewissern Sie sich, dass nur Felder, die zwei Tabellen logisch verknüpfen, als Schlüssel verwendet werden.
 - Felder wie "Kommentar", "Anmerkung" oder "Beschreibung" können in mehreren Tabellen vorhanden sein, ohne zueinander in Bezug zu stehen. Sie sollten deshalb nicht als Schlüssel verwendet werden.
 - Felder wie "Datum", "Unternehmen" und "Name" können in mehreren Tabellen vorhanden sein und identische Werte aufweisen, aber dennoch unterschiedliche Funktionen erfüllen (Bestell-/Versanddatum, Unternehmen des Kunden/Unternehmen des Lieferanten). In solchen Fällen sollten sie nicht als Schlüssel verwendet werden.
- Vergewissern Sie sich, dass keine redundanten Felder verwendet werden – dass tatsächlich nur über die inhaltlich richtigen Felder Verknüpfungen entstehen. Wenn Sie beispielsweise ein Datum als Schlüssel verwenden, stellen Sie sicher, dass in den Tabellen nicht auch Jahres-, Monats- oder Tagesangaben desselben Datums vorkommen.
- Erstellen Sie, falls erforderlich, eigene synthetische Schlüssel durch Stringverknüpfung der Schlüsselfelder und mit Hilfe der Skriptfunktion "AutoNumber".

Zirkelbezüge und Auflösen von Zirkelbezügen

Sind Tabellen so miteinander verknüpft, dass Werte auf mehr als einem Weg miteinander verknüpft sind, spricht man von Zirkelbezügen oder auch zirkulären Verknüpfungen ("Loops") in der Datenstruktur.

Solche Strukturen sollten unter allen Umständen vermieden werden, da eventuell keine eindeutigen Zuordnungen zwischen den Werten mehr möglich sind.



Tabellen mit Zirkelbezug

In Qlik Sense wird das Problem von Zirkelbezügen dadurch gelöst, dass die Endlosschleife durch freie Tabellen unterbrochen wird. Wenn Qlik Sense beim Ausführen eines Ladeskripts kreisförmige Datenstrukturen erkennt, wird eine Warnung angezeigt und eine oder mehrere Tabellen werden als freie Tabellen geführt. Normalerweise versucht Qlik Sense zunächst, die Tabelle mit den meisten Datensätzen zu einer freien Tabelle aufzulösen. Im Datenmodell werden freie Tabellen durch rot gepunktete Verknüpfungen zu anderen Tabellen dargestellt.

Beispiel:

Die Daten werden aus drei Tabellen geladen. Die Daten umfassen Folgendes:

- Namen einiger Fußball-Nationalmannschaften
- Fußballvereine in einigen Städten
- Städte in einigen europäischen Ländern

4 Laden von Daten mit dem Datenladeskript

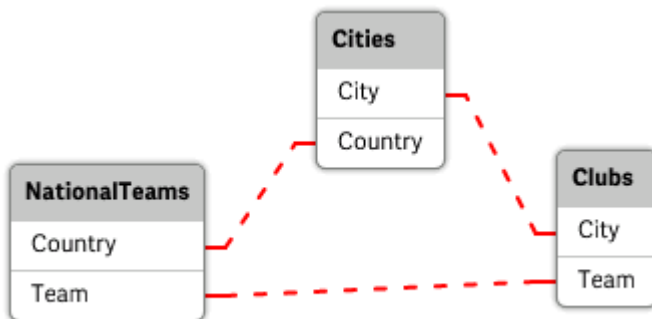
The image shows three data tables side-by-side. The first table, 'NationalTeams', has columns 'Country' and 'Team' with rows for Germany (Die Mannschaft), Italy (Azzurri), and Spain (La Roja). The second table, 'Clubs', has columns 'City' and 'Team' with rows for Barcelona (Barcelona), Hamburg (Altona), Madrid (Real Madrid), Milano (Milan), Munich (Bayern München), Rome (Lazio), and Turin (Juventus). The third table, 'Cities', has columns 'Country' and 'City' with rows for Germany (Hamburg, Munich), Italy (Milano, Rome, Turin), and Spain (Barcelona, Madrid).

Anzeigen der Quelldatentabellen

Die Datenstruktur ist nicht hochwertig, da der Feldname *Team* zwei Bedeutungen hat: Nationalmannschaften und örtliche Vereine. Die Daten in den Tabellen schaffen eine paradoxe logische Situation.

Beim Laden der Tabellen in Qlik Sense ermittelt Qlik Sense, welche Datenverbindungen die geringste Bedeutung haben und gibt die Tabelle frei.

Öffnen Sie das **Datenmodell**, um anzuzeigen, wie Qlik Sense die Relevanz der Datenverbindungen interpretiert:



Anzeigen der Zirkelbezüge mithilfe von rot gepunkteten Linien


Die Tabelle mit Städten und den Ländern, in denen sie liegen, ist jetzt eine freie Tabelle bezüglich der Tabelle mit Fußballnationalmannschaften verschiedener Länder und bezüglich der Tabelle mit örtlichen Vereinen in unterschiedlichen Städten.

Auflösen von Zirkelbezügen

Wenn Zirkelbezüge auftreten, müssen Sie das Datenladeskript bearbeiten, indem Sie einem der Felder mit identischen Namen einen eindeutigen Namen zuweisen.

4 Laden von Daten mit dem Datenladeskript

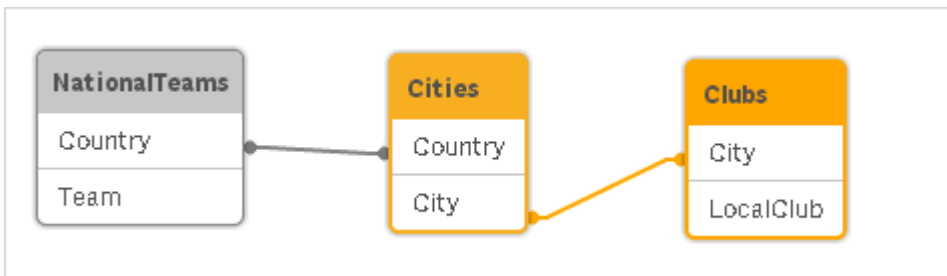
Gehen Sie folgendermaßen vor:

1. Daten im Dateneditor öffnen
2. Bearbeiten Sie den Befehl **LOAD** für einen der zweifach vorliegenden Feldnamen.
In diesem Beispiel würde der Befehl **LOAD** der Tabelle, welche die örtliche Mannschaft und die entsprechenden Städte enthält, einen neuen Namen für *Team* enthalten, z. B. *LocalClub*. Der aktualisierte Befehl **LOAD**:
`LOAD City, Team as LocalClub`
3. Klicken Sie in der Symbolleiste auf , um die Daten erneut in die App zu laden.

Sie verfügen jetzt über eine Logik, die alle Tabellen durchläuft. Wenn in diesem Beispiel *Germany* ausgewählt wird, werden die Nationalmannschaft, die deutschen Städte und die örtlichen Clubs aller Städte verknüpft:

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

Wenn Sie das **Datenmodell** öffnen, wird deutlich, dass die freien Verbindungen durch reguläre Verbindungen ersetzt wurden:



Zusammenfassung von Tabellen

Bei der Zusammenfassung werden zwei Tabellen zu einer kombiniert.

Die beiden Tabellen werden einander lediglich hinzugefügt. Die Daten werden nicht geändert und die entstehende Tabelle enthält genauso viele Datensätze wie die beiden ursprünglichen Tabellen zusammen. Es können auch mehr als zwei Tabellen zusammengefasst werden, indem der Zusammenfassungsvorgang mehrmals nacheinander ausgeführt wird.

Automatische Zusammenfassung

Stimmen bei zwei oder mehr Tabellen die Zahl der Felder und die Feldnamen genau überein, fasst Qlik Sense diese Tabellen automatisch zu einer Tabelle zusammen.

Beispiel:

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

Die entstehende Tabelle enthält die Felder a, b und c. Die Anzahl der Datensätze ist die Summe der Anzahl von Datensätzen in Tabelle 1 und Tabelle 2.



Die Anzahl der Felder und die Feldnamen müssen exakt übereinstimmen. Die Reihenfolge der Befehle ist beliebig.

Manuelle Zusammenfassung

Falls zwei oder mehr Tabellen nicht die gleichen Felder besitzen, kann eine Zusammenfassung der beiden Tabellen in Qlik Sense erzwungen werden. Dieser Vorgang erfolgt mithilfe des Zusatzes **concatenate** im Skript. Er fügt eine Tabelle mit einer anderen benannten Tabelle oder der zuletzt erstellten Tabelle zusammen.

Beispiel:

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

Die entstehende Tabelle enthält die Felder a, b und c. Die Anzahl der Datensätze der entstehenden Tabelle ist die Summe der Anzahl von Datensätzen in Tabelle 1 und Tabelle 2. Die aus Tabelle 2 stammenden Datensätze enthalten im Feld b NULL-Werte.



*Sofern im **concatenate**-Befehl mit **concatenate**-Zusatz kein anderer Tabellenname einer zuvor geladenen Tabelle angegeben ist, bezieht er sich jeweils auf die direkt zuvor angelegte Tabelle. Die Reihenfolge der Befehle ist daher nicht beliebig.*

Zusammenfassung von Tabellen verhindern

Stimmen bei zwei oder mehr Tabellen die Zahl der Felder und die Feldnamen genau überein, fasst Qlik Sense diese Tabellen automatisch zu einer Tabelle zusammen. Dies lässt sich mit dem Befehl **noconcatenate** verhindern. Die mit dem zugehörigen **LOAD**- oder **SELECT**-Befehl geladene Tabelle wird dadurch mit keiner bereits bestehenden Tabelle zusammengefasst.

Beispiel:

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

Laden von Daten aus bereits geladenen Tabellen

Es gibt zwei Möglichkeiten zum Laden und Umwandeln von Daten aus einer Tabelle, die bereits geladen wurde.

4 Laden von Daten mit dem Datenladeskript

- **Resident LOAD** – hierbei nutzen Sie das Prädikat **Resident** in einem nachfolgenden **LOAD**-Befehl, um eine neue Tabelle zu laden.
- Vorangehender Load-Befehl – hierbei laden Sie die Daten aus dem vorangehenden **LOAD**- oder **SELECT**-Befehl, ohne eine Quelle anzugeben.

Resident oder vorangehender LOAD-Befehl?

Gewöhnlich kann mit beiden Methoden dasselbe Ergebnis erzielt werden. Ein vorangehender **LOAD**-Befehl ist in der Regel die schnellere Option, aber in manchen Fällen müssen Sie einen **Resident LOAD** verwenden:

- Wenn Sie die Bedingung **Order_by** zum Sortieren der Datensätze vor der Verarbeitung des **LOAD**-Befehls verwenden.
- Wenn Sie eines der folgenden Präfixe verwenden möchten, für die ein vorangehender **LOAD**-Befehl nicht unterstützt wird:
 - **Crosstable**
 - **Join**
 - **Intervalmatch**

Resident LOAD

Sie können das Prädikat **Resident** in einem **LOAD**-Befehl verwenden, um Daten aus einer bereits geladenen Tabelle zu laden. Das ist nützlich, wenn Sie Berechnungen für Daten durchführen möchten, die mit einem **SELECT**-Befehl geladen wurden, und Sie die Qlik Sense-Funktionen wie die Verarbeitung von Datum und numerischen Werten nicht nutzen können.

Beispiel:

In diesem Beispiel erfolgt die Datumsinterpretation im **Resident Load**, da sie im ursprünglichen **Crosstable LOAD** nicht möglich ist.

```
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;
```



Resident wird auch häufig dann eingesetzt, wenn eine temporäre Tabelle für Berechnungen oder Filter verwendet werden soll. Nachdem die temporäre Tabelle ihren Zweck erfüllt hat, sollte sie mithilfe des Befehls **Drop table** gelöscht werden.

Vorangehender Load-Befehl

Mit dem vorangehenden Load-Befehl können Sie eine Tabelle in einem Arbeitsschritt laden, jedoch trotzdem aufeinanderfolgende Umformungen definieren. Im Grunde ist es ein **LOAD**-Befehl, der über den Befehl **LOAD** oder **SELECT** unten lädt, ohne einen Zusatz für die Quelle wie **From** oder **Resident** anzugeben, was normalerweise der Fall ist. Sie können auf diese Art und Weise beliebig viele **LOAD**-Befehle stapeln. Der Befehl unten wird zuerst ausgewertet, dann der Befehl darüber usw., bis der oberste Befehl ausgewertet wurde.

Sie können auch den Befehl **Resident** verwenden, in den meisten Fällen ist ein vorangehender **LOAD**-Befehl jedoch schneller.

Ein weiterer Vorteil eines vorangehenden Load-Befehls ist es, dass die Berechnung an einem Ort stattfindet und in den darüberliegenden **LOAD**-Befehlen wiederverwendet wird.



Die folgenden Präfixe können nicht mit einem vorangehenden **LOAD**-Befehl verwendet werden: **Join**, **Crosstable** und **Intervalmatch**.

Beispiel 1: Umformen von Daten, die von einem SELECT-Befehl geladen wurden

Wenn Sie Daten aus einer Datenbank mit einem **SELECT**-Befehl laden, können Sie keine Qlik Sense-Funktionen nutzen, um die Daten im **SELECT**-Befehl zu interpretieren. Sie müssen einen **LOAD**-Befehl dort, wo Sie die Datenumformung durchführen, über dem **SELECT**-Befehl hinzufügen.

In diesem Beispiel interpretieren wir ein Datum, das als String mit der **Date#**-Funktion von Qlik Sense in einem **LOAD**-Befehl mit dem vorherigen **SELECT**-Befehl als Quelle gespeichert wurde.

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

Beispiel 2: Vereinfachen Ihres Skripts durch Wiederverwenden von Berechnungen

In diesem Beispiel benutzen wir eine Berechnung mehrmals im Skript:

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

Durch Einführung der Berechnung in einem ersten Schritt können wir sie in der Age-Funktion in einem vorangehenden **LOAD**-Befehl erneut verwenden:

```
LOAD ..., ReferenceDate,  
    Age(ReferenceDate, BirthDate) as Age;
```

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

Datentypen in Qlik Sense

Qlik Sense kann Strings, Zahlen, Datumsangaben, Uhrzeiten, Zeitstempel und Geldbeträge korrekt verarbeiten. Diese können sortiert, in einer Vielzahl verschiedener Formate angezeigt und in Formeln verwendet werden. Dies bedeutet beispielsweise, dass Daten, Uhrzeiten und Zeitstempel addiert oder voneinander subtrahiert werden können.

Speicherung in Qlik Sense

Um besser zu verstehen, wie Qlik Sense numerische Daten verarbeitet, muss man wissen, wie diese intern gespeichert werden. Numerische Werte werden in Qlik Sense in zwei Formen gespeichert, als Zahl und als String.

1. Der String ist das, was Sie in Listboxen und anderen Objekten auf den Arbeitsblättern angezeigt sehen. Wenn Sie also das Zahlenformat für die Werte einer Listbox ändern, betrifft das lediglich die Stringdarstellung.
2. Eine Zahl kann das Programm nur dann speichern, wenn der Wert beim Einlesen als numerischer Wert erkannt wird. Sie wird beispielsweise für Berechnungen oder Sortierfolgen gebraucht.

Wenn mehrere Werte eines Felds durch dieselbe Zahl repräsentiert werden, interpretiert das Programm sie als ein und denselben Wert und weist ihnen die Stringdarstellung des ersten Werts zu. Beispiel: Die Zahlen 1,0 und 1 und 1,000 (in dieser Reihenfolge eingelesen) werden alle als Zahl 1 gespeichert und in der Stringdarstellung 1,0 angezeigt.

Interpretation von Zahlen

Wenn Sie Daten laden, die Zahlen, Geldbeträge oder Datumsangaben enthalten, werden diese abhängig davon, ob der Datentyp definiert ist oder nicht, unterschiedlich interpretiert. In diesem Abschnitt wird erläutert, wie die Daten in diesen beiden unterschiedlichen Fällen interpretiert werden.

Daten mit Informationen zum Datentyp

Felder, die Zahlen eines definierten Datentyps enthalten und über ODBC aus einer Datenbank eingelesen werden, behalten in Qlik Sense ihr jeweiliges Format bei und werden in der gegebenen Form angezeigt. Sie werden als String mit der Zahl in einem entsprechenden Format dargestellt.

Qlik Sense merkt sich das ursprüngliche Zahlenformat des Felds, auch wenn im Eigenschaftsfenster unter **Zahlenformat** das Zahlenformat für eine Kennzahl geändert wird.

Die Standardformate für die verschiedenen Datentypen sind:

- ganze Zahlen und Gleitkommazahlen: Standardformat für Zahlen
- Geldbeträge: Standardformat für Währungen
- Datum, Uhrzeit, Zeitstempel: ISO-Standardformat

4 Laden von Daten mit dem Datenladeskript

Die Standardformate für Zahlen und Währungen werden automatisch durch die Betriebssystem-Einstellungen (**Systemsteuerung**) oder durch Variablen zur Zahleninterpretation im Skript festgelegt.

Daten ohne Informationen zum Datentyp

Bei Daten, die ohne spezifische Formatinformation aus der Datenquelle geladen werden (z. B. aus Textdateien oder ODBC im allgemeinen Format), wird die Interpretation etwas komplizierter. Das Format der Daten im Programm hängt von mindestens sechs Faktoren ab:

1. Zahlenformate in der Datenquelle
2. Vom Betriebssystem vorgegebene Formate für Zahlen, Uhrzeit, Datum usw. (**Systemsteuerung**)
3. Skriptvariablen zur Interpretation von Zahlen
4. Interpretationsfunktionen im Skript
5. Formatfunktionen im Skript
6. Einstellungen zum Zahlenformat innerhalb des Dokuments

Qlik Sense versucht zunächst, die eingegebenen Daten als Zahl, Datum, Uhrzeit usw. zu interpretieren. Stimmen die Formate der Daten mit den vom Betriebssystem vorgesehenen überein, kann Qlik Sense den Datentyp richtig erkennen und zeigt die Werte korrekt an. In diesem Fall sind keine weiteren Einstellungen in Qlik Sense oder Änderungen des Skripts notwendig.

Standardmäßig wird das folgende Schema verwendet, bis ein vollständiges Match gefunden ist. (Standardformat bezeichnet hier die vom Betriebssystem vorgegebenen Formate, die in der **Systemsteuerung** eingestellt werden, z. B. das Dezimaltrennzeichen, Reihenfolge von Jahr, Monat und Tag usw., bzw. in Sonderfällen die von den Skript-Variablen vorgegebene Zahleninterpretation.)

Qlik Sense prüft nacheinander folgende Formate:

1. Zahl entsprechend dem Standardformat für Zahlen.
2. Datum entsprechend dem Standardformat für Datumsangaben.
3. Zeitstempel entsprechend dem Standardformat für Datum und Uhrzeit.
4. Uhrzeit entsprechend dem Standardformat für Uhrzeiten.
5. Datum entsprechend dem Format yyyy-MM-dd.
6. Zeitstempel entsprechend dem Format YYYY-MM-DD hh:mm[:ss[:fff]].
7. Uhrzeit entsprechend dem Format hh:mm[:ss[:fff]].
8. Geldbetrag entsprechend dem Standardformat für Währungen.
9. Zahl mit '.' als Dezimaltrennzeichen und ',' als Tausendertrennzeichen, sofern weder für Dezimal- noch für Tausendertrennzeichen ',' eingestellt ist.
10. Zahl mit ',' als Dezimaltrennzeichen und '.' als Tausendertrennzeichen, sofern weder für Dezimal- noch für Tausendertrennzeichen '.' eingestellt ist.
11. String. Die Anzeige als String ist immer möglich.

4 Laden von Daten mit dem Datenladeskript

Beim Einlesen von Zahlen aus Textdateien können gelegentlich Probleme auftauchen. Beispielsweise kann ein anderes Dezimal- oder Tausendertrennzeichen zu falschen Interpretationen in Qlik Sense führen. In diesem Fall sollten Sie zunächst prüfen, ob die Skriptvariablen zur Interpretation von Zahlen korrekt definiert sind und ob die Einstellungen in der **Systemsteuerung** korrekt sind.

Hat Qlik Sense die Daten als Datum oder Uhrzeit erkannt, können Sie im Eigenschaftsfenster der Visualisierung das Datums- oder Uhrzeitformat nach Ihren Wünschen ändern.

Da es kein vordefiniertes Format für die Daten gibt, können natürlich verschiedene Datensätze verschieden formatierte Daten im selben Feld enthalten. Es können sich z. B. gültige Datumsangaben, ganze Zahlen und Text in einem einzelnen Feld befinden. Deshalb zeigt das Programm die Daten zunächst in ihrem ursprünglichen Format an.

Interpretation von Datum und Uhrzeit

Qlik Sense speichert jeweils Datum, Uhrzeit und Zeitstempel in den Daten als Datenseriennummer. Die Datenseriennummer wird in arithmetischen Berechnungen auf Basis von Datum- und Zeiteinheiten für Daten, Uhrzeiten und Zeitstempel verwendet. Anhand der numerischen Werte können Datumsangaben und Uhrzeiten addiert, subtrahiert oder verglichen werden.

Die Datenseriennummer ist die (echte) Anzahl von Tagen, die seit dem 30. Dezember 1899 vergangen ist, d. h. das Qlik Sense-Format ist identisch mit dem 1900er-Datensystem, das Microsoft Excel und andere Programme verwenden, und zwar im Zeitraum zwischen 1. März 1900 und 28. Februar 2100. Der 10. September 1992 entspricht beispielsweise dem Wert 33857. Außerhalb dieses Zeitraums nutzt Qlik Sense dasselbe angepasste Datumssystem des Gregorianischen Kalenders.

Die numerische Darstellung einer Uhrzeit ist ein Dezimalbruch zwischen 0 und 1. Die Zahl 0,00000 entspricht 00:00:00, und 0,99999 entspricht 23:59:59. Dezimalzahlen größer als 1 repräsentieren Datum und Uhrzeit. Beispiel: 2,5 steht für den 1. Januar 1900, 12:00 Uhr mittags.

Die Daten werden jedoch gemäß dem Stringformat angezeigt. Standardmäßig werden die Einstellungen der **Systemsteuerung** verwendet. Es ist auch möglich, das Datenformat mit Hilfe der Variablen zur Interpretation von Zahlen im Skript oder mit Hilfe einer Formatierungsfunktion einzustellen. Schließlich ist es möglich, die Daten im Eigenschaften-Arbeitsblatt des Objekts neu zu formatieren.

Beispiel 1:

1997-08-06	wird gespeichert als:	35648
09:00	wird gespeichert als:	0.375
1997-08-06 09:00	wird gespeichert als:	35648.375

und umgekehrt.

35648	mit dem Zahlenformat 'D/M/YY' wird angezeigt als:	6/8/97
0.375	mit dem Zahlenformat 'hh.mm' wird angezeigt als:	09.00

4 Laden von Daten mit dem Datenladeskript

Qlik Sense kann nach bestimmten Vorgaben Datum, Uhrzeit und andere Datentypen korrekt interpretieren. Das Format der Anzeige hängt jedoch von verschiedenen Faktoren ab.

Beispiel 2:

In den nachfolgenden Beispielen liegen diese Standardeinstellungen zugrunde:

- Dezimaltrennzeichen: .
- Kurzes Datumsformat: YY-MM-DD
- Uhrzeit: hh:mm

Die folgende Tabelle zeigt die unterschiedlichen Darstellungen, wenn Daten ohne die spezielle Interpretationsfunktion im Skript in Qlik Sense eingelesen werden:

Quelldaten	Qlik Sense-Standardinterpretation	Datumsformat 'YYYY-MM-DD'	Datumsformat 'MM/DD/YYYY'	Uhrzeit 'hh:mm'	Zahlenformat '# ##0.00'
0.375	0.375	1899-12-30	12/30/1899	09:00	0.38
33857	33857	1992-09-10	09/10/1992	00:00	33 857.00
97-08-06	97-08-06	1997-08-06	08/06/1997	00:00	35 648.00
970806	970806	4557-12-21	12/21/4557	00:00	970 806.00
8/6/97	8/6/97	8/6/97	8/6/97	8/6/97	8/6/97

Die folgende Tabelle zeigt die unterschiedlichen Darstellungen, wenn Daten die date#(A, 'M/D/YY')-Interpretationsfunktion im Skript in Qlik Sense eingelesen werden:

Quelldaten	Qlik Sense-Standardinterpretation	Datumsformat 'YYYY-MM-DD'	Datumsformat 'MM/DD/YYYY'	Uhrzeit 'hh:mm'	Zahlenformat '# ##0.00'
0.375	0.375	0.375	0.375	0.375	0.375
33857	33857	33857	33857	33857	33857
97-08-06	97-08-06	97-08-06	97-08-06	97-08-06	97-08-06
970806	970806	970806	970806	970806	970806
8/6/97	8/6/97	1997-08-06	08/06/1997	00:00	35 648.00

Aufrufen von Variablen

Variablen sind Platzhalter für Text, die in Skripten oder Formeln verwendet werden. Dieser Vorgang wird als Aufrufen bezeichnet. Dies geschieht, bevor der Skript-Befehl ausgeführt bzw. die Formel berechnet wird. Technisch gesehen ist es ein Makroaufruf.

Die Aufrufe beginnen immer mit '\$(' und enden mit ')', wobei der Text in Klammern definiert, in welcher Form der Text ersetzt wird. Um Verwechslungen mit Skript-Makros zu vermeiden, werden Makroaufrufe im Folgenden als Variablen bezeichnet.

Aufrufe sind mit folgenden Bestandteilen möglich:

- Variablen
- Parametern
- Formeln



Beim Aufrufen von Variablen ist die Zahl der Berechnungsschritte begrenzt. Es werden nicht mehr als 1000 Berechnungsschritte vorgenommen.

Textvariablen und numerische Variablen

Wenn Sie zum Ersetzen von Text im Skript oder in einer Formel eine Variable verwenden, lautet die Syntax:

```
$(variablename)
```

`$(variablename)` nimmt den Wert der Variablen an. Existiert `variablename` nicht, ergibt sich ein leerer String.

Für numerische Variablen benutzen Sie folgende Syntax:

```
$(#variablename)
```

Es liefert immer eine gültige Repräsentation des numerischen Werts der Variable, u. U. in Exponentialschreibweise (bei sehr großen oder kleinen Zahlen). Sollte `variablename` nicht existieren oder keinen numerischen Wert enthalten, ergibt sich `0`.

Beispiel:

```
SET DecimalSep=',';  
LET X = 7/2;
```

Die Variable `$(X)` nimmt den Wert 3,5 an, während `$(#X)` den Wert 3.5 annimmt.

Beispiel:

```
set mypath=C:\MyDocs\Files\  
...  
LOAD * from $(MyPath)abc.csv;
```

Die Daten werden von `C:\MyDocs\Files\abc.csv` geladen.

Beispiel:

4 Laden von Daten mit dem Datenladeskript

```
Set CurrentYear=1992;
...
SQL SELECT * FROM table1 WHERE Year=$(CurrentYear);
Zeilen mit Year=1992 werden ausgewählt.
```

Beispiel:

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

In diesem Beispiel werden alle .txt-Dateien des Verzeichnisses mithilfe des Zusatzes **Concatenate** geladen. Dies kann erforderlich sein, wenn sich die Felder leicht unterscheiden, wodurch die automatische Zusammenfassung nicht funktioniert. Die Variable `vConcatenate` ist ursprünglich auf einen leeren String festgelegt, da der Zusatz **Concatenate** beim erstmaligen Laden nicht verwendet werden kann. Enthält das Verzeichnis drei Dateien mit den Namen `file1.txt`, `file2.txt` und `file3.txt`, wird der **LOAD**-Befehl während den drei Iterationen folgendermaßen erweitert:

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

Aufrufen von Variablen mit Parametern

Beim Aufruf einer Variablen kann mit Parametern gearbeitet werden. Die Variable erhält formale Parameter wie \$1, \$2, \$3 usw. Beim Aufruf der Variablen werden die Parameter in einer kommagetrennten Liste angegeben.

Beispiel:

```
Set MUL='$1*$2';
Set X=$(MUL(3,7)); // liefert '3*7' in X

Let X=$(MUL(3,7)); // liefert 21 in X
```

Sollten mehr formale Parameter definiert sein als tatsächliche Parameter angegeben sind, werden nur die formalen Parameter berücksichtigt, für die tatsächliche Parameter vorhanden sind. Sind mehr tatsächliche Parameter angegeben als formale Parameter definiert sind, werden die überzähligen tatsächlichen Parameter ignoriert.

Beispiel:

```
Set MUL='$1*$2';
Set X=$(MUL); // liefert '$1*$2' in X

Set X=$(MUL(10)); // liefert '10*$2' in X

Let X=$(MUL(5,7,8)); // liefert 35 in X
```

Der Parameter \$0 liefert die Zahl der übergebenen Parameter beim Aufruf der Variablen.

Beispiel:

```
set MUL='$1*$2 $0 par';  
set X=$(MUL(3,7)); // liefert '3*7 2 par' in X
```

Aufrufen von Variablen mit Formeln

Man kann auf das Definieren und Initialisieren einer Variable verzichten, wenn man eine Formel aufrufen will. Dazu muss der Inhalt der Klammern beim Aufrufen mit einem Gleichheitszeichen beginnen.

```
$(=expression )
```

Die Formel wird berechnet und das Ergebnis wie ein Wert einer Variablen behandelt.

Beispiel:

```
$(=Year(Today())); // liefert einen String mit dem aktuellen Jahr.
```

```
$(=only(Year)-1); // liefert das Jahr vor dem ausgewählten Jahr,
```

Skript-Dateien

Skript-Dateien lassen sich als Variablen einbinden. Die Syntax lautet:

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

Diese Syntax wird bei der Ausführung des Skripts durch den Text nach dem Gleichheitszeichen ersetzt. Dies ist für in Textdateien ausgelagerte Skriptteile sinnvoll.

Beispiel:

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

Verwenden von Anführungszeichen im Skript

Sie können Anführungszeichen in Skriptbefehlen auf verschiedene Weisen verwenden.

In LOAD-Befehlen

In einem **LOAD**-Befehl sollten die folgenden Symbole als Anführungszeichen verwendet werden:

	Beschreibung	Symbol	Codepoint	Beispiel
Feldnamen	Doppelte Anführungszeichen	" "	34	"String"
	Eckige Klammern	[]	91, 93	[String]
	Graviszeichen	` `	96	`String`
String-Literale	Einfache Anführungszeichen	' '	'String'	

In SELECT-Befehlen

Bei **SELECT**-Befehlen, die vom ODBC-Treiber interpretiert werden, kann die Verwendung möglicherweise etwas anders sein. Normalerweise sollten Sie für Feld- und Tabellennamen gerade, doppelte Anführungszeichen (Alt + 0034) verwenden und für Literale gerade, einfache Anführungszeichen (Alt + 0039), aber niemals die Graviszeichen. Einige ODBC-Treiber akzeptieren jedoch Graviszeichen nicht nur, sondern bevorzugen diese sogar. In solchen Fällen enthält der generierte **SELECT**-Befehl Graviszeichen.

Microsoft Access – Beispiel für Anführungszeichen

Microsoft Access ODBC Driver 3.4 (in Microsoft Access 7.0 enthalten) akzeptiert folgende Anführungszeichen beim Analysieren des **SELECT**-Befehls:

Feldnamen und Tabellennamen: [" `] " `

String-Literale: ""

Andere Datenbanken haben möglicherweise andere Konventionen.

Außerhalb von LOAD-Befehlen

Außerhalb von **LOAD**-Befehlen – an Stellen, an denen Qlik Sense eine Formel erwartet – kennzeichnen doppelte Anführungszeichen eine Variablenreferenz, keine Feldreferenz. Wenn Sie doppelte Anführungszeichen verwenden, wird der eingeschlossene String als Variable interpretiert und der Wert der Variablen wird verwendet.

Feldreferenzen und Tabellenreferenzen ohne Kontext

Einige Skriptfunktionen beziehen sich auf Felder, die bereits erstellt wurden, oder in der Ausgabe eines **LOAD**-Befehls enthalten sind, z. B. **Exists()** und **Peek()**. Diese Feldreferenzen werden als Feldreferenzen ohne Kontext bezeichnet, im Gegensatz zu Quellenfeldreferenzen, die sich auf Felder im Kontext, d. h. in der Eingabetabelle des **LOAD**-Befehls, beziehen.

Feld- und Tabellenreferenzen ohne Kontext sollten als Literale angesehen werden und erfordern deshalb einfache Anführungszeichen.

Unterschied zwischen Namen und Literalen

Die folgenden Beispiele sollen den Unterschied zwischen Namen und Literalen verdeutlichen:

Beispiel:

'Sweden' as Country

Wenn diese Formel als Teil der Feldliste in einem **LOAD**- oder **SELECT**-Befehl verwendet wird, wird der Textstring "Sweden" als Feldwert in das Qlik Sense-Feld "Country" geladen.

Beispiel:

"land" as country

4 Laden von Daten mit dem Datenladeskript

Wird diese Formel als Teil der Feldliste in einem **LOAD**- oder **SELECT**-Befehl verwendet, wird der Inhalt des Datenbankfelds oder der Tabellenspalte namens "*land*" als Feldwerte in das Qlik Sense-Feld "*Country*" geladen. Das heißt, dass *land* als Feldreferenz angesehen wird.

Unterschied zwischen Zahlen und String-Literalen

Die folgenden Beispiele sollen den Unterschied zwischen Zahlen und String-Literalen verdeutlichen.

Beispiel:

'12/31/96'

Wenn dieser String als Teil einer Formel verwendet wird, wird er in einem ersten Schritt als Textstring "12/31/96" interpretiert, was wiederum als Datum interpretiert werden kann, wenn das Datumsformat 'MM/DD/YY' lautet. In diesem Fall wird er als dualer Wert sowohl mit numerischer als auch mit Textdarstellung gespeichert.

Beispiel:

12/31/96

Wenn dieser String als Teil einer Formel verwendet wird, wird er numerisch interpretiert als 12 geteilt durch 31 geteilt durch 96.

Verwenden von einfachen Anführungszeichen innerhalb eines Strings

Wenn Sie in einem String einfache Anführungszeichen verwenden möchten, können Sie ein zusätzliches einfaches Anführungszeichen als Escape-Zeichen nutzen.

Beispiel:

Der String '10 o'clock News' wird als "10 O'clock News" interpretiert.

Wildcards in den Daten

In den Daten können Wildcards verwendet werden. Das Stern-Symbol steht für alle Werte des Feldes, und das Rest-Symbol für alle sonstigen Werte des Feldes.

Stern-Symbol

Das Stern-Symbol steht für alle existierenden (aufgeführten) Werte des Feldes, d. h. für einen beliebigen Wert in der Tabelle. Werden Wildcards in einem der sicherheitsrelevanten Felder (*USERID*, *PASSWORD*, *NTNAME* oder *SERIAL*) verwendet, interpretiert das Programm, dass sie für alle (auch für nicht explizit aufgeführte) Werte des Feldes stehen.

In Informationsdateien ist das Stern-Symbol nicht zulässig. Es darf außerdem nicht in Schlüsselfeldern verwendet werden, d. h. in Feldern, mit denen Tabellen verbunden werden.

Zunächst ist kein bestimmtes Zeichen als Stern-Symbol vorbesetzt, im Skript kann jedoch durch den star-Befehl ein solches definiert werden.

OtherSymbol

Häufig wird auch ein Symbol benötigt, das für alle verbleibenden Werte der Tabelle steht, d. h. für alle Werte, die nicht explizit in den geladenen Daten gefunden werden. Dies geschieht durch die Skriptvariable **OtherSymbol**. Damit **OtherSymbol** als "alle anderen Werte" behandelt wird, verwenden Sie folgende Syntax:

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

vor dem Befehl **LOAD** oder **SELECT**. <sym> kann ein beliebiger String sein.

Taucht das definierte Symbol in einer internen Tabelle auf, setzt Qlik Sense es für alle Werte des Felds fest, die noch nicht geladen sind. Werte, die nach dem **OtherSymbol** in der Tabelle auftreten, werden daher nicht berücksichtigt.

Um diesen Befehl rückgängig zu machen, geben Sie ein:

```
SET OTHERSYMBOL=;
```

Beispiel:

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

Ergänzen Sie das Skript vor den Ladebefehlen, welche die oben stehenden Tabellen laden, mit folgendem Befehl:

```
SET OTHERSYMBOL=+;
```

Alle Bezüge auf die *CustomerID*, die nicht 1, 2 oder 3 sind, z. B. beim Klicken auf *OrderID 1299*, ergibt *Undefined* unter *Name*.



Über das **OtherSymbol** können keine Joins zwischen Tabellen hergestellt werden.

NULL-Werte im Programm

Werden beim Einlesen einer Datenbank oder bei einem Join für ein Feld keine Werte gefunden, führt dies üblicherweise zu NULL-Werten.

Überblick

Qlik Sense liefert in folgenden Fällen NULL-Werte:

- NULL-Werte, die aus ODBC-Verbindungen stammen
- NULL-Werte, die beim manuellen Zusammenschluss von Tabellen im load-Skript entstehen
- NULL-Werte, die durch einen join-Befehl im Datenladeskript entstehen
- NULL-Werte, die bei der Kombination von Feldwerten zur Anzeige in Tabellen entstehen



*Normalerweise ist es nicht möglich, NULL-Werte auszuwählen oder für Verknüpfungen zu nutzen, es sei denn, dies ist durch den Befehl **NullAsValue** ausdrücklich zugelassen.*

Textdateien können per Definition keine NULL-Werte enthalten.

NULL-Werte aus ODBC-Datenquellen

Es ist möglich, NULL-Werte aus einer ODBC-Datenquelle auszuwählen bzw. für Verknüpfungen zu benutzen. Für diesen Zweck ist eine Skriptvariable vorgesehen. Folgende Syntax kann verwendet werden:

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

Das angegebene Symbol <sym> ersetzt alle NULL-Werte aus der ODBC-Datenquelle auf dem niedrigsten Datenlevel. <sym> kann ein beliebiger String sein.

Um diesen Befehl rückgängig zu machen, geben Sie folgende Syntax ein:

```
SET NULLDISPLAY=;
```



*Die Variable **NULLDISPLAY** ist nur für Daten aus ODBC-Datenquellen relevant.*

Wenn Qlik Sense NULL-Werte aus ODBC-Verbindungen auf diese Weise behandeln soll, definieren Sie vor dem **SELECT**-Befehl für die Variable nulldisplay den Leer-String:

```
SET NULLDISPLAY=" ;
```



" steht hier für zwei senkrechte, einfache Anführungszeichen ohne Zwischenraum.

Erstellen von NULL-Werten aus Textdateien

Es ist möglich, ein Symbol zu definieren, das als **inline**-Wert interpretiert wird, wann immer es in Textdateien oder NULL-Befehlen auftritt. Benutzen Sie dazu folgenden Befehl:

4 Laden von Daten mit dem Datenladeskript

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

Dabei ist <sym> ein beliebiger String, der als NULL-Wert interpretiert werden soll. <sym> kann ein beliebiger String sein.

Um diesen Befehl rückgängig zu machen, geben Sie Folgendes ein:

```
SET NULLINTERPRET=;
```



*Die Variable **NULLINTERPRET** ist nur für Daten aus Textdateien oder inline-Befehlen relevant.*

NULL-Werte in Formeln

NULL-Werte entstehen mitunter auch als Ergebnis von Formeln. Dies richtet sich nach einigen einfachen und logischen Gesetzen.

Funktionen

Eine Funktion ergibt grundsätzlich den NULL-Wert, wenn die Parameter oder Argumente außerhalb des Definitionsbereichs der Funktion liegen.

Beispiel:

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

Logischerweise liefert eine Funktion auch dann NULL, wenn einer der zwingend benötigten Parameter NULL ist.

Beispiel:

sin(NULL)	liefert	NULL
chr(NULL)	liefert	NULL
if(NULL, A, B)	liefert	NULL
if(True, NULL, A)	liefert	NULL
if(True, A, NULL)	liefert	A

Eine Ausnahme bilden Funktionen, die einen logischen Test durchführen.

Beispiel:

isnull(NULL)	liefert	True (-1)
isnum(NULL)	liefert	False (0)

Arithmetische Operatoren und String-Operatoren

Steht auf dieser Seite dieser Operatoren ein NULL-Wert, ist das Ergebnis NULL.

Beispiel:

A	+	NULL	liefert	NULL
A	-	NULL	liefert	NULL
A	/	NULL	liefert	NULL
A	*	NULL	liefert	NULL
NULL	/	A	liefert	NULL
0	/	NULL	liefert	NULL
0	*	NULL	liefert	NULL
A	&	NULL	liefert	A

Relationale Operatoren

Besondere Regeln gelten für den Fall, dass auf einer Seite des relationalen Operators ein NULL-Wert steht.

Beispiel:

NULL	(beliebiger relationaler Operator)	NULL	liefert	NULL
A	<>	NULL	liefert	True (-1)
A	<	NULL	liefert	False (0)
A	<=	NULL	liefert	False (0)
A	=	NULL	liefert	False (0)
A	>=	NULL	liefert	False (0)
A	>	NULL	liefert	False (0)

4.4 Richtlinien für Daten und Felder

Beim Arbeiten mit Qlik Sense müssen Sie bestimmte Konventionen und Beschränkungen berücksichtigen. Beispiel: die Obergrenze für Datentabellen und -felder sowie die maximale Menge von in Qlik Sense geladenen Daten.

Richtlinien für die Menge an geladenen Daten

Die Menge an Daten, die in Qlik Sense geladen werden kann, wird in erster Linie durch die Größe des primären Speichers des Computers begrenzt.

Obergrenzen für Datentabellen und -felder

Beachten Sie beim Erstellen sehr großer Apps, dass in Qlik Sense für eine App pro Feld maximal 2.147.483.648 distinkte Werte zulässig sind.

Die Anzahl der Felder und Datentabellen sowie die Anzahl der Tabellenzellen und -zeilen, die geladen werden können, wird nur durch den Arbeitsspeicher begrenzt.

Empfohlene Grenzen für Abschnitte des Ladeskripts

Es wird empfohlen, pro Abschnitt eines Ladeskripts maximal 50.000 Zeichen zu verwenden.

Konventionen für Zahlen- und Uhrzeitformate

In vielen Interpretations- und Formatfunktionen können Zahlen und Daten durch einen Formatcode beschrieben werden. Dieses Thema spezifiziert die Konventionen für das Zahlen-, Datums-, Zeit- oder Zeitstempelformat. Diese Konventionen gelten gleichermaßen für Skript- und Diagrammfunktionen.

Zahlenformate

- Die Zahl der angezeigten Dezimalstellen wird durch entsprechend viele Nullen "0" repräsentiert.
- Optionale Dezimalstellen werden durch das Symbol "#" dargestellt. Enthält der Formatcode vor dem Dezimaltrennzeichen nur #-Symbole, so beginnen Zahlen, die kleiner als 1 sind, mit dem Dezimaltrennzeichen.
- Um die Position des zutreffenden Dezimal- und Tausendertrennzeichens festzulegen, fügen Sie diese einfach an der gewünschten Stelle des Formatcodes ein.

Im Formatcode werden lediglich die Positionen von Tausender- und Dezimaltrennzeichen festgelegt. Es ist hier nicht möglich, Tausender- oder Dezimaltrennzeichen zu definieren. Verwenden Sie dafür die Variablen **DecimalSep** und **ThousandSep** im Skript.

Es ist möglich, beliebig viele Stellen durch das Tausendertrennzeichen abzugrenzen. Beispielsweise würde der Formatcode "0000-0000-0000" mit "-" als Tausendertrennzeichen eine zehnstellige Artikelnummer in der Form "0012-4567-8912" liefern.

Beispiele:

# ##0	liefert eine ganze Zahl mit einem Tausendertrennzeichen.
###0	liefert eine ganze Zahl ohne Tausendertrennzeichen.
0000	liefert eine ganze Zahl mit mindestens vier Stellen. Zum Beispiel wird die Zahl 123 in der Form 0123 angezeigt.
0.000	liefert eine Zahl mit drei Dezimalstellen.
0.0##	liefert eine Zahl mit mindestens einer und höchstens drei Dezimalstellen.

Spezielle Zahlenformate

Qlik Sense kann numerische Werte beliebiger Stellenwertsysteme mit einer Basis zwischen 2 und 36 interpretieren und formatieren, darunter binäre, oktale und hexadezimale Darstellung. Auch römische Zahlen werden korrekt verarbeitet.

Binäre Zahl	Um eine binäre Zahl zu signalisieren, muss der Formatcode mit (bin) oder (BIN) beginnen.
Oktale Zahl	Um eine oktale Zahl zu signalisieren, muss der Formatcode mit (oct) oder (OCT) beginnen.
Hexadezimale Zahl	Um eine hexadezimale Zahl zu signalisieren, muss der Formatcode mit (hex) oder (HEX) beginnen. Bei Verwendung der groß geschriebenen Variante wird bei der Formatierung A-F verwendet (zum Beispiel 14FA). Bei Verwendung der klein geschriebenen Variante wird bei der Formatierung a-f verwendet (zum Beispiel 14fa). Bei der Interpretation besteht kein Unterschied zwischen Groß- und Kleinbuchstaben.
Dezimalzahl	Um eine dezimale Zahl zu signalisieren, kann der Formatcode mit (dec) oder (DEC) beginnen. Dieser Zusatz ist jedoch nicht zwingend notwendig.
Format mit benutzerdefinierter Basis	Um ein Stellenwertsystem mit Basis zwischen 2 und 36 zu signalisieren, muss der Formatcode mit (rxx) oder (Rxx) beginnen, wobei xx eine zweistellige Zahl ist, die den Wert der Basis angibt. Wenn R – groß geschrieben – verwendet wird, werden Buchstaben mit einer Basis über 10 im Qlik Sense-Format groß geschrieben (zum Beispiel 14FA). Bei Verwendung der klein geschriebenen Variante (r) wird bei der Formatierung Kleinschreibung verwendet (zum Beispiel 14fa). Bei der Interpretation besteht kein Unterschied zwischen Groß- und Kleinbuchstaben. Somit ist (r02) äquivalent zu (bin), (R16) ist äquivalent zu (HEX) usw.
Römische Zahlen	Um eine römische Zahl zu signalisieren, muss der Formatcode mit (rom) oder (ROM) beginnen. Bei Verwendung der groß geschriebenen Variante wird bei der Formatierung Großschreibung verwendet (zum Beispiel MMXVI). Bei Verwendung der klein geschriebenen Variante werden bei der Formatierung Kleinbuchstaben verwendet (mmxvi). Bei der Interpretation besteht kein Unterschied zwischen Groß- und Kleinbuchstaben. Römische Zahlen werden verallgemeinert mit einem Minuszeichen für negative Zahlen und 0 für Null. Dezimalstellen entfallen bei römischen Zahlen.

Beispiele:

<code>num(199, '(bin)')</code>	liefert	11000111
<code>num(199, '(oct)')</code>	liefert	307
<code>num(199, '(hex)')</code>	liefert	c7

4 Laden von Daten mit dem Datenladeskript

<code>num(199, '(HEX'))</code>	liefert	C7
<code>num(199, '(r02'))</code>	liefert	11000111
<code>num(199, '(r16'))</code>	liefert	c7
<code>num(199, '(R16'))</code>	liefert	C7
<code>num(199, '(R36'))</code>	liefert	5J
<code>num(199, '(rom'))</code>	liefert	cxcix
<code>num(199, '(ROM'))</code>	liefert	CXCIX

Datumsangaben

Sie können die folgenden Symbole für die Formatierung eines Datums verwenden. Es können beliebige Trennzeichen verwendet werden.

D	Die numerische Angabe des Tags wird durch ein "D" für jede gewünschte Stelle repräsentiert.
M	Verwenden Sie das Symbol "M", um den Monat anzugeben. <ul style="list-style-type: none"> • Verwenden Sie "M" oder "MM" für eine bzw. zwei Stellen. • "MMM" steht für den kurzen Monatsnamen, wie er vom Betriebssystem vorgegeben oder durch die Skriptvariable MonthNames definiert ist. • "MMMM" steht für den langen Monatsnamen, wie er vom Betriebssystem vorgegeben oder durch die Skriptvariable LongMonthNames definiert ist.
Y	Die numerische Angabe des Jahrs wird durch ein "Y" für jede gewünschte Stelle repräsentiert.
W	Verwenden Sie das Symbol "W", um den Wochentag anzugeben. <ul style="list-style-type: none"> • Die dem numerischen Wochentag entsprechende Ziffer wird durch ein "W" dargestellt (Beispiel: 0 für Montag). • "WW" liefert den zweistelligen numerischen Wochentag (Beispiel: 02 für Mittwoch). • "WWW" liefert den kurzen Wochentag (zum Beispiel Mo), wie vom Betriebssystem vorgegeben oder durch die Skriptvariable DayNames definiert. • "WWWW" liefert den langen Wochentag (zum Beispiel Montag), wie vom Betriebssystem vorgegeben oder durch die Skriptvariable LongDayNames definiert.

Beispiele: (mit dem 31. März 2013 als Beispieldatum)

YY-MM-DD	liefert ein Datum in der Form 13-03-31.
YYYY-MM-DD	liefert ein Datum in der Form 3/31/13.
YYYY-MMM-DD	liefert ein Datum in der Form 2013-Mar-31.
DD MMMM YYYY	liefert ein Datum in der Form 31. März 2013.

4 Laden von Daten mit dem Datenladeskript

M/D/YY	liefert ein Datum in der Form 3/31/13.
W YY-MM-DD	liefert ein Datum in der Form 6 13-03-31.
WWW YY-MM-DD	liefert ein Datum in der Form Sa 13-03-31.
WWWY YY-MM-DD	liefert ein Datum in der Form Samstag 13-03-31.

Uhrzeiten

Sie können die folgenden Symbole für die Formatierung einer Uhrzeit verwenden. Es können beliebige Trennzeichen verwendet werden.

h	Die numerische Angabe der Stunden wird durch ein "h" für jede gewünschte Stelle repräsentiert.
m	Die numerische Angabe der Minuten wird durch ein "m" für jede gewünschte Stelle repräsentiert.
s	Die numerische Angabe der Sekunden wird durch ein "s" für jede gewünschte Stelle repräsentiert.
f	Die Sekundenbruchteile (Zehntel, Hundertstel usw.) werden durch ein "f" für jede Dezimalstelle repräsentiert.
tt	Die Uhrzeit im AM/PM-Format wird durch "tt" hinter der Uhrzeit repräsentiert.

Beispiele: (18.30 dient als Beispielzeit):

hh:mm	liefert eine Uhrzeit in der Form 18:30
hh.mm.ss.ff	liefert eine Uhrzeit in der Form 18.30.00.00
hh:mm:tt	liefert eine Uhrzeit in der Form 06:30:pm

Zeitstempel

Für Zeitstempel gilt dieselbe Notation wie bei Datum und Uhrzeit.

Beispiele: (31. März 2013 18.30 dient als Beispielzeitstempel):

YY-MM-DD hh:mm	liefert einen Zeitstempel in der Form 13-03-31 18:30
M/D/Y hh.mm.ss.ffff	liefert einen Zeitstempel in der Form 3/31/13 18.30.00.0000

4.5 Arbeiten mit QVD-Dateien

QVD (QlikView Data)-Dateien enthalten Datentabellen, die aus Qlik Sense exportiert werden. QVD ist ein natives Qlik Format, das nur von Qlik Sense oder QlikView gelesen oder geschrieben werden kann. Das

4 Laden von Daten mit dem Datenladeskript

Dateiformat ist für besonders schnelles Laden aus einem Skript optimiert, aber gleichzeitig sehr kompakt. Das Einlesen von Daten aus QVD-Dateien ist etwa 10-100 Mal schneller als das Einlesen aus anderen Datenquellen.

QVD-Dateien können in zwei Modi gelesen werden: Standard (schnell) und optimiert (schneller). Der ausgewählte Modus ergibt sich automatisch aus dem Skriptmodul. Der optimierte Modus kann nur verwendet werden, wenn alle geladenen Felder oder Teile davon ohne Umformung (Formeln, die an den Feldern aktiv werden) eingelesen werden. Felder dürfen jedoch umbenannt werden. Eine **where**-Bedingung, bei der Qlik Sense die Datensätze auspackt, deaktiviert auch die optimierte Ladung.

Anwendung von QVD-Dateien

QVD-Dateien sind in verschiedenen Situationen hilfreich. Im Wesentlichen lassen sich vier Anwendungsbereiche nennen. Oft sprechen auch gleich mehrere Gründe für den Einsatz von QVD-Dateien.

Verkürzte Ladezeiten	Daten, die sich gar nicht oder nur in großen Zeitabständen verändern, lassen sich in QVD-Dateien puffern. Dadurch wird für die Ausführung des Skripts erheblich weniger Zeit benötigt. Dieser Effekt macht sich insbesondere bei großen Datenmengen bemerkbar.
Entlastung für Datenbankserver	Die aus externen Quellen geladene Datenmenge verringert sich erheblich. Datenbankserver und Netzwerke werden entlastet, weil weniger Daten bewegt werden. Wenn mehrere Skripte dieselben Daten verwenden, müssen sie nur einmal aus der Quelldatenbank in eine QVD-Datei geladen werden. Andere Anwendungen können über diese QVD-Datei die gleichen Daten verwenden.
Integration von Daten aus mehreren Anwendungen	Mithilfe des Skript-Befehls binary lassen sich nur Daten aus einer Anwendung in eine andere laden. Sind die Daten aber in QVD-Dateien abgelegt, kann das -Skript Daten aus beliebig vielen Anwendungen kombinieren. Ein möglicher Anwendungsbereich ist z. B. der Vergleich von Daten aus verschiedenen Abteilungen eines Betriebs.
Inkrementelles Laden	In vielen Fällen kann man die QVD-Funktion für inkrementelle Ladevorgänge benutzen, bei denen nur neue Datensätze aus einer umfangreichen Datenbank gelesen werden.

Erstellen von QVD-Dateien

Eine QVD-Datei kann auf zwei verschiedene Arten erstellt werden:

1. Explizite Erstellung und Benennung mithilfe des Befehls **store** im Skript. Dazu definieren Sie im Skript, dass eine bereits eingelesene Tabelle oder ein Teil davon in eine Datei exportiert werden soll, deren Namen und Speicherort Sie bestimmen.
2. Automatische Generierung durch das Skript. Durch den Zusatz von **LOAD** vor **SELECT**- oder **buffer**-Befehlen legt Qlik Sense automatisch eine QVD-Datei an, deren Inhalt bei späteren Ladevorgängen anstelle der Originaldatenquelle eingelesen werden kann.

Die Art und Weise des Anlegens hat keinerlei Einfluss auf die Eigenschaften der QVD-Datei, d. h. die Zugriffsgeschwindigkeit ist gleich.

Daten aus QVD-Dateien einlesen

Es gibt verschiedene Möglichkeiten, QVD-Dateien einzulesen oder darauf zuzugreifen:

1. Laden einer QVD-Datei als Datenquelle. QVD-Dateien können genau wie andere Dateitypen (**LOAD**, csv, fix, dif usw.) mit Hilfe des biff-Befehls im Skript referenziert werden.

Beispiel:

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

2. Automatische Nutzung von QVD-Buffern. Wenn Sie den **buffer**-Zusatz vor einem **LOAD**- oder **SELECT**-Befehl verwenden, benötigen Sie keinen separaten Befehl zum Einlesen. Qlik Sense prüft automatisch, welche Daten aus der QVD-Datei und welche aus dem zugrunde liegenden originalen **LOAD** oder **SELECT** -Befehl gelesen werden können.
3. Zugriff auf QVD-Dateien über das Skript. Eine Reihe neuer Skriptfunktionen (ihre Namen beginnen alle mit **qvd**) helfen Ihnen, die Informationen im XML-Header der QVD-Datei auszulesen.

QVD-Format

Eine QVD-Datei enthält immer genau eine Tabelle und besteht aus drei Teilen:

1. Header.



Wenn die QVD-Datei mit QlikView erstellt wurde, handelt es sich bei dem Header um einen wohlgeformten XML-Header (im UTF-8-Zeichensatz), der die Felder in der Tabelle beschreibt, das Layout der nachfolgenden Informationen und sonstige Metadaten.

2. Symboltabellen, unter Anwendung von Bytestuffing.
3. Die eigentlichen Tabellendaten, unter Anwendung von Bitstuffing.

4.6 Sicherheitsverwaltung über Zugriffsabschnitt

Mithilfe des Zugriffsabschnitts im Datenladeskript können Sie die Sicherheit verwalten. Auf diese Weise kann eine einzelne Datei verwendet werden, die die Daten für eine Reihe von Benutzern oder Benutzergruppen enthält. Qlik Sense nutzt die Informationen in den Zugriffsdaten für die Authentifizierung und Autorisierung und schränkt diese Daten so ein, dass die Benutzer nur die für sie vorgesehenen Daten einsehen können.

Da die Zugriffskontrolle in das Dokument integriert ist, sind die Dateien auch dann geschützt, wenn sie heruntergeladen und versehentlich in falsche Hände gelangen sollten. Wenn die Sicherheitsanforderungen jedoch hoch sind, sollten das Herunterladen von Dateien und die Offline-Nutzung unterbleiben und Dateien sollten nur vom Qlik Sense-Server veröffentlicht werden. Da sämtliche Daten in einer Datei gespeichert sind, kann die Datei sehr groß werden.

Abschnitte im Skript

Qlik Sense verarbeitet die Zugriffskontrolle wie andere Daten auch, d. h. sie werden in Form einer oder mehrerer Tabellen durch das Skript eingelesen. Diese Tabellen können dadurch in einer normalen Datenbank gespeichert werden. Um eine Zugriffskontrolle einzurichten, wird das Skript in zwei Abschnitte unterteilt. Im Zugriffsabschnitt, eingeleitet durch den Befehl **Section Access**, werden die Daten zur Regelung der Zugriffskontrolle geladen.

Im Datenabschnitt, eingeleitet durch den Befehl **Section Application** werden die "normalen" Daten in das Dokument eingelesen.

Beispiel:

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

Sicherheitsfelder

Zugriffsrechte werden in einer Zugriffstabelle definiert, die im Zugriffsabschnitt des Skripts eingelesen wird. Diese Tabellen können verschiedene benutzerspezifische Systemfelder enthalten, in der Regel USERID, und das Feld, das die Berechtigungsstufe regelt, ACCESS. Alle Zugriffsabschnitt-Systemfelder werden für die Authentifizierung und die Autorisierung herangezogen. Nachfolgend finden Sie eine komplette Liste der Zugriffsabschnitt-Systemfelder.

Keines, alle oder ein Teil der Sicherheitsfelder können verwendet werden. Das Feld USERID daher ist nicht zwingend notwendig, die Zugriffskontrolle kann beispielsweise auch nur über die Lizenznummer erfolgen.

ACCESS Berechtigungsstufe des Anwenders.

Der Zugriff auf Qlik Sense-Apps kann bestimmten Anwendern oder Anwendergruppen gewährt werden. Dazu können Anwender in der Zugriffstabelle den Berechtigungsstufen ADMIN oder USER zugewiesen werden. Ist einem Anwender keine gültige Berechtigungsstufe zugeordnet, kann er die App nicht öffnen.

Personen mit ADMIN-Rechten haben Zugriff auf alle Daten in der App. Personen mit USER-Rechten können nur auf Daten zugreifen, wie sie in der Zugriffstabelle definiert sind.

USERID Enthält einen String, der einem Qlik Sense-Benutzernamen entspricht. Qlik Sense prüft die Login-Daten des Proxys und vergleicht sie mit den Werten dieses Felds.

4 Laden von Daten mit dem Datenladeskript

- GROUP** Enthält einen String, der einer Gruppe in Qlik Sense entspricht. Qlik Sense ordnet den vom Proxy bereitgestellten Benutzer gegenüber dieser Gruppe zu.
- OMIT** Enthält den Namen des Felds, das für diesen speziellen Benutzer ausgelassen werden soll. Darf der Anwender alle Felder sehen, bleibt dieses Feld leer. Es können die üblichen Wildcards verwendet werden. Es kann mit der subfield-Funktion gearbeitet werden.



Sie sollten OMIT nicht auf Schlüsselfelder anwenden, da dadurch die Datenstruktur geändert wird. Dadurch können logisch isolierte Bereiche und Berechnungsinkonsistenzen entstehen.

Qlik Sense vergleicht den vom Proxy bereitgestellten Benutzer mit UserID und ordnet den Benutzer gegenüber den Gruppen in der Tabelle zu. Wenn der Benutzer zu einer Gruppe gehört, der Zugriff eingeräumt ist, oder der Benutzer übereinstimmt, erhält er Zugriff auf die App.



Wenn Sie sich durch Festlegen einer Zugriffskontrolle selbst aus der App ausgesperrt haben, können Sie die App ohne Daten öffnen, und die Zugriffskontrolle im Datenladeskript bearbeiten. Dazu müssen Sie Zugriff auf das Datenladeskript haben, um es bearbeiten und neu laden zu können.

Da die Qlik Sense-Logik auch im Zugriffsabschnitt angewandt wird, können die Sicherheitsfelder in mehreren Tabellen gespeichert werden. Alle Felder im Zugriffsabschnitt unter **LOAD** oder **SELECT** müssen in Großbuchstaben geschrieben werden. Konvertieren Sie Feldnamen aus externen Datenquellen bei der Ausführung des Skripts mit Hilfe des Befehls **Upper** in Großbuchstaben, bevor sie von **LOAD**- oder **SELECT**-Befehlen eingelesen werden.

Wildcards (*) stehen für alle existierenden Werte des Felds, d. h. für einen beliebigen Wert in der Tabelle. Werden Wildcards in einem der sicherheitsrelevanten Felder (USERID, GROUP) verwendet, interpretiert das Programm, dass sie für alle (auch für nicht explizit aufgeführte) Werte des Felds stehen.



Wird beim Einlesen von Daten aus einer QVD-Datei der Befehl upper verwendet, beeinträchtigt dies die Ladezeit.



Wenn Sie den Zugriffsabschnitt aktiviert haben, können Sie die hier aufgeführten Zugriffsabschnitts-Systemfeldnamen nicht als Feldnamen in Ihrem Datenmodell verwenden.

Beispiel:

In diesem Beispiel können nur Benutzer in der Finanzgruppe das Dokument öffnen.

ACCESS
USER

GROUP
Finance

Einschränkung des Datenbestands

Qlik Sense unterstützt die dynamische Einschränkung des Datenbestands, durch die einige der Daten einer App für Anwender, basierend auf dem Zugriffsabschnitt-Login, ausgeblendet werden können.

- Felder (Spalten) können über das Systemfeld OMIT ausgeblendet werden.
- Durch eine erweiterte Syntax im Zugriffsabschnitt können einzelne Datensätze und Werte innerhalb der Felder vor den Anwendern verborgen werden. Diese Einschränkung erfolgt dadurch, dass ein oder mehrere Felder in beiden Abschnitten des Skripts, also im Zugriffs- und Datenabschnitt, vorkommen (beide Feldnamen müssen in Großbuchstaben geschrieben werden). Bei jedem Login versucht Qlik Sense den Abgleich der im Zugriffsabschnitt aufgeführten Werte des Felds mit dem gleichnamigen Feld im Datenabschnitt mit genau denselben Feldnamen (Feldnamen müssen GROSSGESCHRIEBEN werden). Nach abgeschlossener Auswahl werden alle übrigen Werte des Felds dauerhaft in Qlik Sense vor dem Anwender verborgen.



Alle Feldnamen, die für die Einschränkung des Datenbestands herangezogen werden sollen, müssen in Großbuchstaben geschrieben sein, da alle Daten im Zugriffsabschnitt automatisch in Großbuchstaben konvertiert werden.

Beispiel:

```
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  
USER, INTERNAL\SA_SCHEDULER,*,  
];  
section application;  
T1:  
LOAD *,  
NUM AS REDUCTION;  
LOAD  
Chr( RecNo()+ord('A')-1) AS ALPHA,  
RecNo() AS NUM  
AUTOGENERATE 3;
```

Das Feld REDUCTION existiert nun in exakt gleicher Schreibweise im Zugriffs- und im Datenabschnitt (alle Feldwerte sind in Großbuchstaben geschrieben). Normalerweise werden diese beiden Felder völlig separat verarbeitet. Durch Verwendung des Zugriffsabschnitts werden diese Felder jedoch verknüpft und die Anzahl der Datensätze, die dem Benutzer angezeigt werden, verringert.

Das Feld OMIT im Zugriffsabschnitt definiert, welche Felder ausgeblendet werden sollen.

Dadurch ergibt sich Folgendes:

- Der ADMIN-Anwender kann alle Felder, aber nur die Datensätze sehen, die andere Anwender in diesem Beispiel sehen können, wenn REDUCTION 1, 2 oder 3 ist.

4 Laden von Daten mit dem Datenladeskript

- Anwender A kann alle Felder sehen, aber nur die Datensätze, die in logischem Zusammenhang mit REDUCTION=1 stehen.
- Anwender B kann alle Felder bis auf NUM sehen, aber nur die Datensätze, die in logischem Zusammenhang mit REDUCTION=2 stehen.
- Anwender C kann alle Felder bis auf ALPHA sehen, aber nur die Datensätze, die in logischem Zusammenhang mit REDUCTION=3 stehen.



Der Kontenbenutzer von INTERNAL\ISA_SCHEDULER muss das Skript erneut ausführen, damit das Skript in einer Qlik Management Console-Aufgabe erneut geladen werden kann.

Übertragung von Zugriffsbeschränkungen

Durch Laden von Dateien mit Hilfe des binary-Befehls werden Zugriffsbeschränkungen auf die neue Qlik Sense-App übertragen.

4.7 Zugriff auf große Datensätze mit Direct Discovery

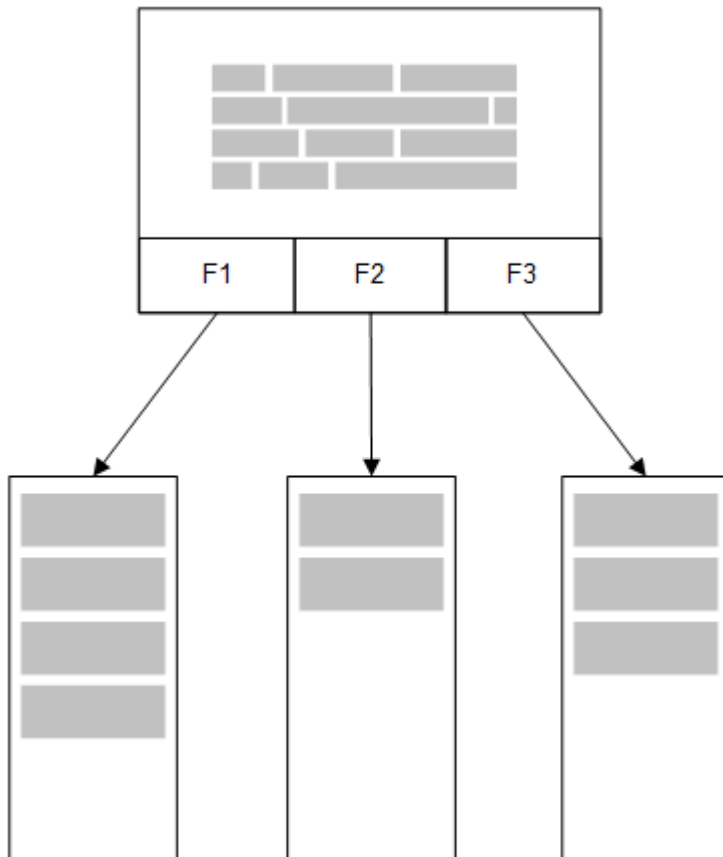
Direct Discovery erweitert die Verknüpfungsfunktionen des Qlik Sense-In-Memory-Datenmodells, indem über eine aggregierte Abfrage Zugang zu zusätzlichen Quelldaten gewährt wird. Dabei werden größere Datensätze nahtlos mit den im Speicher befindlichen Daten verknüpft. Dank Direct Discovery können Geschäftsanwender für Big-Data-Quellen ohne Beschränkungen besser assoziative Analysen durchführen. Auswahlen können sowohl für im Speicher befindliche als auch Direct Discovery-Daten vorgenommen werden, damit übergreifend über die Datensätze Verknüpfungen mit denselben Qlik Sense-Verknüpfungsfarben – Grün, Grau und Weiß – angezeigt werden können. In Visualisierungen können Daten aus beiden Datensätzen zusammen analysiert werden.

Daten werden für Direct Discovery mithilfe einer speziellen Skriptsyntax ausgewählt: **DIRECT QUERY**. Nachdem die Struktur von Direct Discovery festgelegt wurde, können Direct Discovery-Felder zusammen mit im Speicher befindlichen Daten zum Erstellen von Qlik Sense-Objekten verwendet werden. Wird ein Direct Discovery-Feld in einem Qlik Sense-Objekt verwendet, wird automatisch eine SQL-Abfrage für die externe Datenquelle ausgeführt.

Unterschiede zwischen Direct Discovery und im Speicher befindlichen Daten

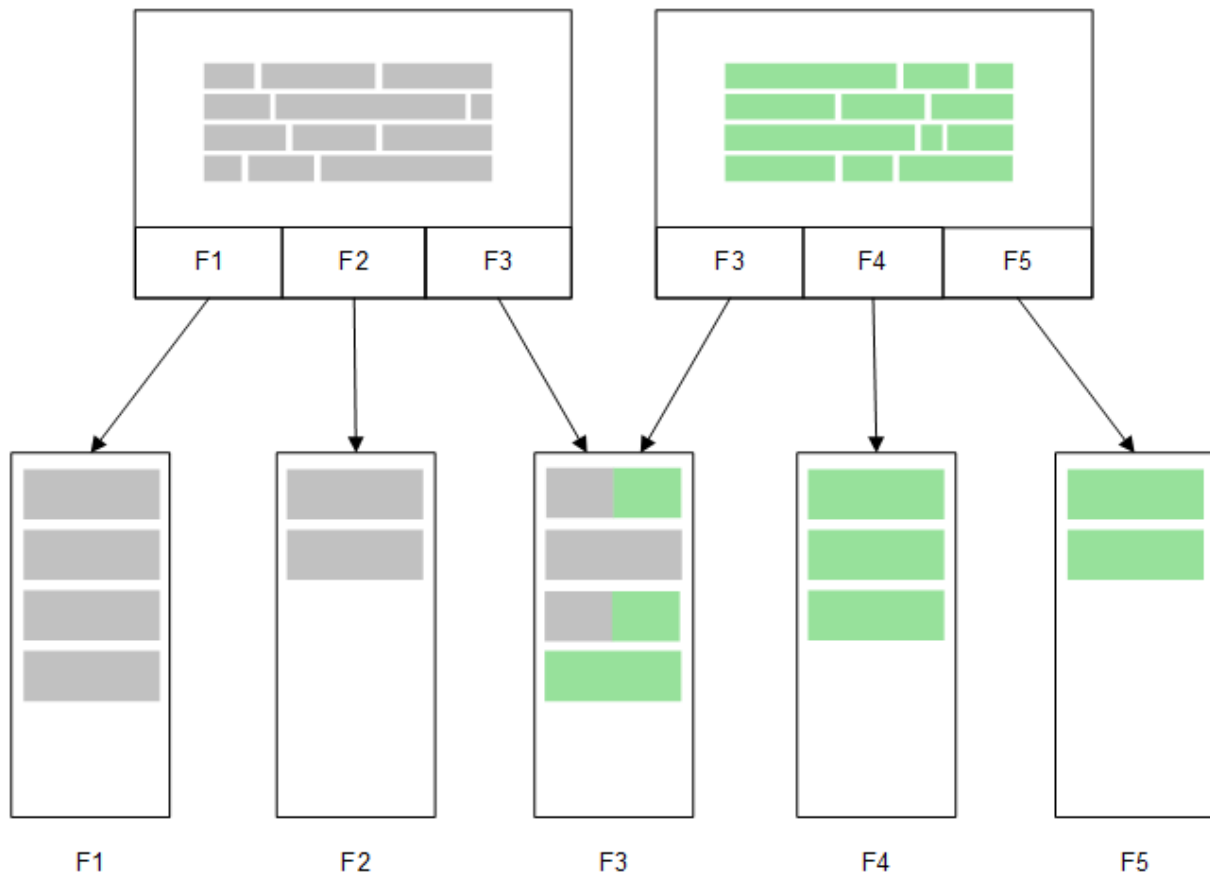
In-Memory-Modell

Im Qlik Sense In-Memory-Modell werden alle eindeutigen Werte in den aus einer Tabelle im Ladeskript ausgewählten Feldern in Feldstrukturen und die zugeordneten Daten gleichzeitig in die Tabelle geladen. Die Felddaten und die zugeordneten Daten bleiben alle im Speicher.



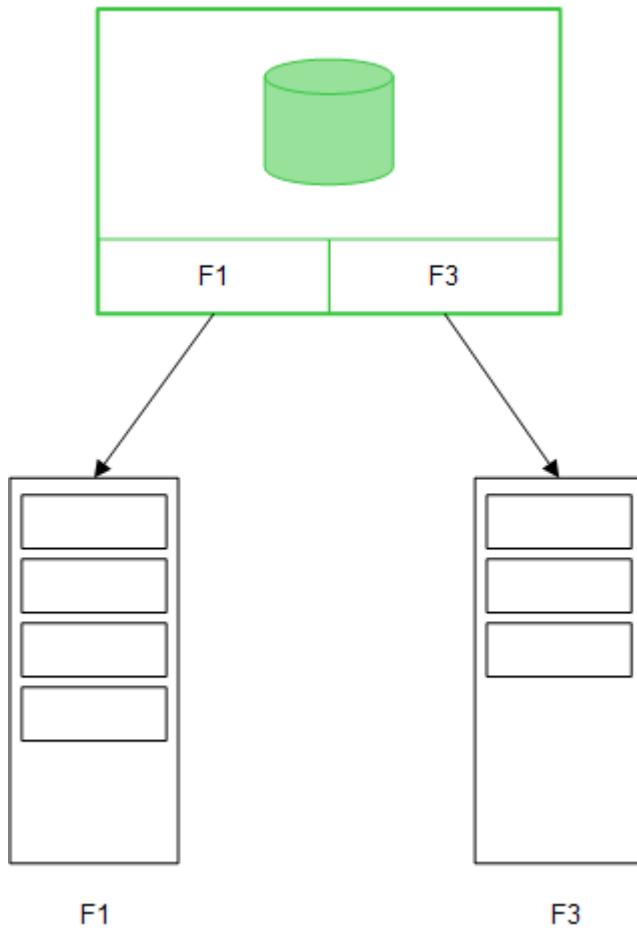
Eine zweite, betreffende Tabelle, die in den Speicher geladen wurde, würde ein gemeinsames Feld teilen und diese Tabelle könnte zum gemeinsamen Feld neue eindeutige Werte hinzufügen oder bestehende Werte teilen.

4 Laden von Daten mit dem Datenladeskript



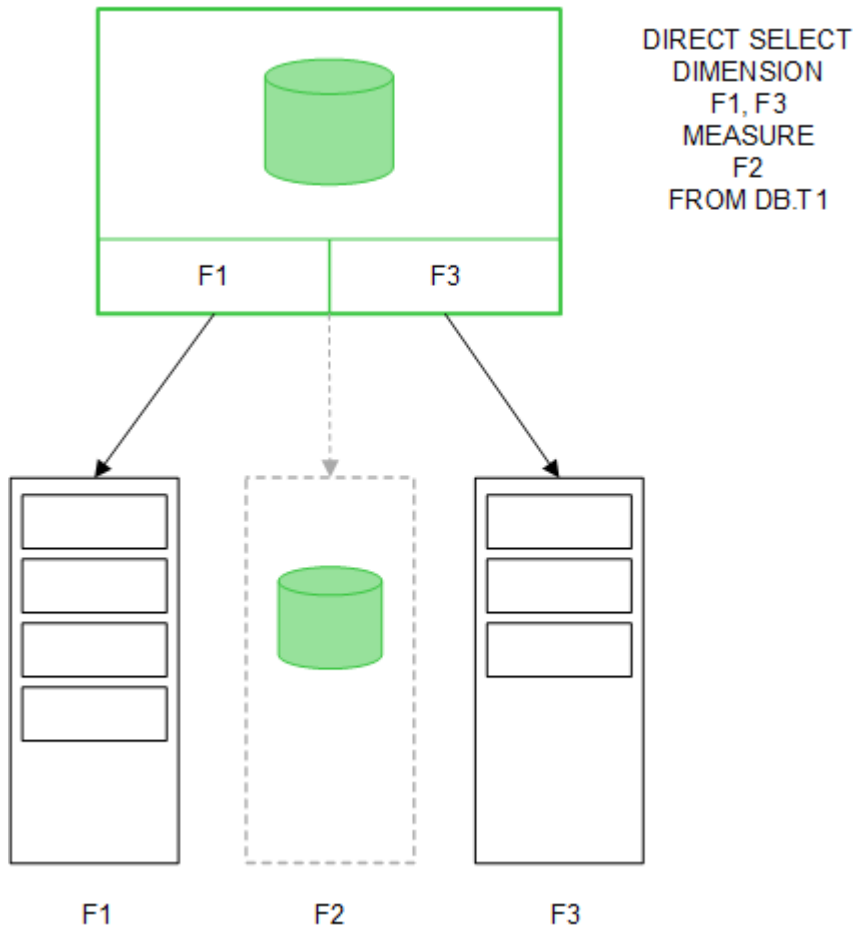
Direct Discovery

Wenn Tabellenfelder durch einen Direct Discovery **LOAD**-Befehl (**Direct Query**) geladen werden, wird eine ähnliche Tabelle nur mit den **DIMENSION**-Feldern erstellt. Wie bei den im Speicher befindlichen Feldern werden die eindeutigen Werte für die **DIMENSION**-Felder in den Speicher geladen. Aber die Verknüpfungen zwischen den Feldern werden in der Datenbank belassen.



MEASURE-Feldwerte werden auch in der Datenbank belassen.

4 Laden von Daten mit dem Datenladeskript



Nachdem die Struktur von Direct Discovery festgelegt wurde, können Direct Discovery-Felder mit bestimmten Visualisierungsobjekten verwendet werden, auch für Verknüpfungen mit im Speicher befindlichen Feldern. Wenn ein Direct Discovery-Feld verwendet wird, erstellt Qlik Sense automatisch die geeignete SQL-Abfrage für die externe Datenquelle. Bei Auswahlen werden die verknüpften Datenwerte der Direct Discovery-Felder in den WHERE-Bedingungen der Datenbankabfragen verwendet.

Für jede Auswahl werden die Visualisierungen mit Direct Discovery-Feldern neu berechnet, wobei die Berechnungen in der Tabelle der Quelldatenbank durch Ausführen der von Qlik Sense erstellten SQL-Abfrage ablaufen. Die Funktion der Berechnungsbedingung kann zur Angabe, wann Visualisierungen neu berechnet werden sollen, verwendet werden. Solange die Bedingung nicht erfüllt wird, sendet Qlik Sense keine Abfragen zur Neuberechnung der Visualisierungen.

Leistungsunterschiede zwischen im Speicher befindlichen und Direct Discovery-Feldern

Die Verarbeitung im Speicher ist stets schneller als in Quelldatenbanken. Die Leistung von Direct Discovery spiegelt die Leistung des Datenbanksystems wider, das die Direct Discovery-Abfragen verarbeitet.

Es ist möglich, eine Standarddatenbank und beste Praktiken zur Abfragenanpassung für Direct Discovery zu verwenden. Alle Leistungsanpassungen sollten in der Quelldatenbank erfolgen. Direct Discovery bietet aus

4 Laden von Daten mit dem Datenladeskript

der Qlik Sense-App für die Leistungsanpassung bei Abfragen keine Unterstützung. Mithilfe der Verbindungspooling-Funktion ist es jedoch möglich, asynchrone, parallele Abrufe aus der Datenbank durchzuführen. Die Syntax des Ladeskripts zur Einrichtung der Pooling-Funktion ist:

```
SET DirectConnectionMax=10;
```

Qlik Sense-Caching verbessert auch das gesamte Nutzererlebnis. Siehe Abbildung unten (*Caching und Direct Discovery (Seite 121)*)

Die Leistung von Direct Discovery mit **DIMENSION**-Feldern kann ebenso durch Trennen der Verknüpfung einiger Felder verbessert werden. Dies wird mit dem Schlüsselwort **DETACH** in **DIRECT QUERY** vorgenommen. Während getrennte Felder nicht nach Verknüpfungen abgefragt werden, sind sie immer noch Bestandteil der Filter, wodurch sich Auswahlzeiten verkürzen.

Während im Speicher befindliche Qlik Sense-Felder und Direct Discovery **DIMENSION**-Felder beide alle ihre Daten im Speicher halten, wirkt sich die Weise, wie sie geladen werden, auf die Ladegeschwindigkeit in den Speicher aus. Im Speicher befindliche Qlik Sense-Felder speichern nur eine Kopie eines Feldwerts, wenn es mehrere Instanzen des gleichen Werts gibt. Jedoch werden alle Felddaten geladen und die duplizierten Daten dann aussortiert.

DIMENSIONFelder speichern auch nur eine Kopie eines Feldwerts, aber die duplizierten Werte werden in der Datenbank aussortiert, bevor sie in den Speicher geladen werden. Wenn Sie mit großen Datenmengen zu tun haben, wie gewöhnlich bei der Verwendung von Direct Discovery, werden die Daten durch eine **DIRECT QUERY** viel schneller geladen als durch das für im Speicher befindliche Felder **SQL SELECT**.

Unterschiede zwischen Daten im Speicher und Datenbankdaten

DIRECT QUERY berücksichtigt Groß- und Kleinschreibung, wenn Verknüpfungen mit im Speicher befindlichen Daten vorgenommen werden. Direct Discovery wählt Daten aus Quelldatenbanken in Übereinstimmung mit der Case Sensitivity der Datenbank aus. Wenn eine Datenbank nicht Case-sensitiv ist, ergibt eine Direct Discovery-Abfrage möglicherweise Daten, die eine In-Memory-Abfrage nicht ergäbe. Zum Beispiel, wenn sich die folgenden Daten in einer Datenbank befinden, die nicht Case-sensitiv ist, ergäbe eine Direct Discovery-Abfrage des Werts "Red" alle vier Reihen.

ColumnA	ColumnB
red	one
Red	two
rED	three
RED	four

Eine Auswahl von "Red," im Speicher ergäbe hingegen nur:

```
Red two
```

Qlik Sense normalisiert Daten in einem Ausmaß, das Treffer zu ausgewählten Daten erzeugt, die Datenbanken nicht zuordnen würden. Als Folge kann eine Abfrage im Speicher mehr passende Werte ergeben als eine Direct Discovery-Abfrage. Zum Beispiel unterscheiden sich in der folgenden Tabelle die Werte für die Zahl "1" nach der Position der Leerzeichen um sie:

4 Laden von Daten mit dem Datenladeskript

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

Wenn Sie "1" in einem **Filterfenster** für ColumnA auswählen, wo sich die Daten im standardmäßigen Qlik Sense-Speicher befinden, werden die ersten drei Reihen zugeordnet:

'1'	space_before
'1'	no_space
'1 '	space_after

Wenn das **Filterfenster** Direct Discovery-Daten enthält, verknüpft die Auswahl von "1" möglicherweise nur "no_space". Die sich für Direct Discovery-Daten ergebende Treffer hängen von der Datenbank ab. Einige ergeben nur "no_space" und andere, wie SQL Server, ergeben "no_space" und "space_after".

Caching und Direct Discovery

Qlik Sense-Caching speichert Auswahlstatus von Abfragen im Speicher. Bei gleichen Arten von Auswahlen, verwendet Qlik Sense die Abfrage aus dem Cache, anstatt die Quelldaten abzufragen. Wenn eine andere Auswahl getroffen wird, erfolgt eine SQL-Abfrage in der Datenquelle. Die gecachten Ergebnisse werden unter Benutzern geteilt.

Beispiel:

1. Der Benutzer wendet die ursprüngliche Auswahl an.
SQL wird an die zugrunde liegende Datenquelle weitergegeben.
2. Der Benutzer löscht die Auswahl und wendet die gleiche Auswahl wie die ursprüngliche an.
Das Cache-Ergebnis wird ausgegeben, SQL wird nicht an die zugrunde liegende Datenquelle weitergegeben.
3. Der Benutzer wendet eine andere Auswahl an.
SQL wird an die zugrunde liegende Datenquelle weitergegeben.

Beim Caching mit der **DirectCacheSeconds**-Systemvariable kann ein Zeitlimit festgesetzt werden. Sobald das Zeitlimit erreicht wird, löscht Qlik Sense die Direct Discovery-Abfrageergebnisse aus dem Cache, die für frühere Auswahlen erzeugt wurden. Qlik Sense fragt dann die Quelldaten nach den Auswahlen ab und baut den Cache für das festgelegte Zeitlimit neu auf.

Die Standard-Cachezeit für Direct Discovery-Abfrageergebnisse ist 30 Minuten, sofern die **DirectCacheSeconds**-Systemvariable verwendet wird.

Feldtypen für Direct Discovery

Innerhalb von Direct Discovery gibt es drei Typen von Datenfeldern: DIMENSION, MEASURE und DETAIL. Die Typen werden für Datenfelder festgelegt, wenn die Direct Discovery-Auswahl mithilfe des Befehls **Direct Query** im Ladeskript getroffen wird.

Alle Direct Discovery-Felder lassen sich in Kombination mit im Speicher befindlichen Feldern nutzen. Üblicherweise sollten Felder mit diskreten Werten, die als Dimensionen verwendet werden, mit dem Schlüsselwort DIMENSION geladen werden, wohingegen numerische Werte, die nur in Aggregierungsfunktionen verwendet werden, als MEASURE-Felder markiert werden sollten. DIMENSION-Felder können nicht als Objektformeln verwendet werden.

In der folgenden Tabelle werden Eigenschaften und Verwendung der Direct Discovery-Feldtypen zusammengefasst:

Feldtyp	Im Speicher?	Bildet Verknüpfung?	In Diagrammformeln verwendet?
DIMENSION	Ja	Ja	Ja
MEASURE	Nein	Nein	Ja
DETAIL	Nein	Nein	Nein

DIMENSION-Felder

DIMENSION-Felder werden in den Speicher geladen und können verwendet werden, um Verknüpfungen zwischen im Speicher befindlichen Daten und Daten in Direct Discovery-Feldern zu erstellen. Direct Discovery-DIMENSION-Felder werden auch verwendet, um Dimensionswerte in Tabellen zu definieren.

MEASURE-Felder

MEASURE-Felder werden wiederum auf "Metaebene" erkannt. MEASURE-Felder werden nicht in den Speicher geladen (sie werden nicht im Datenmodell angezeigt). Zweck davon ist es, die Aggregation der Daten in MEASURE-Feldern in der Datenbank anstatt im Speicher zu ermöglichen. Trotzdem können MEASURE-Felder in Formeln eingesetzt werden, ohne dass die Formelsyntax verändert wird. Deshalb ist die Nutzung von Direct Discovery-Feldern aus der Datenbank für den Endbenutzer transparent.

Die folgenden Aggregierungsfunktionen können mit MEASURE-Feldern verwendet werden:

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

DETAIL-Felder

DETAIL-Felder beinhalten Informationen oder Details, die Ihnen die Anzeige, aber nicht die Verwendung in Diagrammformeln ermöglichen. DETAIL-Felder können nur in **Count**-Aggregationen verwendet und in einer **Tabelle** angezeigt werden. Als DETAIL festgelegte Felder enthalten normalerweise Daten, die wie Kommentare nicht aussagekräftig aggregiert werden können.

Beliebige Felder können als DETAIL-Feld festgelegt werden.

In Direct Discovery unterstützte Datenquellen

Qlik SenseDirect Discovery kann auf die folgenden Datenquellen angewendet werden, sowohl bei 32-Bit- wie auch bei 64-Bit-Verbindungen:

- ODBC/OLEDB-Datenquellen – Alle ODBC/OLEDB-Quellen werden unterstützt, einschließlich SQL Server, Teradata und Oracle.
- Benutzerdefinierte Konnektoren, die SQL – SAP SQL-Konnektoren unterstützen, benutzerdefinierte QVX-Konnektoren für SQL-kompatible Datenspeicher.

Sowohl 32-Bit- wie auch 64-Bit-Verbindungen werden unterstützt.

SAP

Für SAP kann Direct Discovery nur mit dem Qlik SAP SQL Connector verwendet werden. Er erfordert die folgenden Parameter in den **SET**-Variablen:

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

SAP verwendet OpenSQL, der Spalten mit einem Leerzeichen anstelle eines Kommas begrenzt. Deshalb bewirken set-Befehle einen Ersatz, um den Unterschied zwischen ANSI SQL und OpenSQL auszugleichen.

Google Big Query

Direct Discovery kann mit Google Big Query verwendet werden und erfordert die folgenden Parameter in den set-Variablen:

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

Google Big Query unterstützt **SELECT DISTINCT** und Spalten-/Tabellennamen in Anführungszeichen nicht. Es verfügt über eine nicht den ANSI-Standards entsprechende Konfiguration für Anführungszeichen, mit "[]".

MySQL und Microsoft Access

Direct Discovery kann zusammen mit MySQL und Microsoft Access verwendet werden, aber erfordert aufgrund der in diesen Quellen verwendeten Anführungszeichen möglicherweise die folgenden Parameter in den set-Variablen:

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

DB2, Oracle und PostgreSQL

Direct Discovery kann gemeinsam mit DB2, Oracle und PostgreSQL verwendet werden, aber erfordert aufgrund der in diesen Quellen verwendeten Anführungszeichen möglicherweise den folgenden Parameter in den festgelegten Variablen:

```
SET DirectIdentifierQuoteChar='\"';
```

Sybase und Microsoft SQL Server

Direct Discovery kann gemeinsam mit Sybase, Sybase und Microsoft SQL Server verwendet werden, aber erfordert aufgrund der in diesen Quellen verwendeten Anführungszeichen möglicherweise den folgenden Parameter in den festgelegten Variablen:

```
SET DirectIdentifierQuoteChar='[]';
```

Einschränkungen in der Verwendung von Direct Discovery

Unterstützte Datentypen

Alle Datentypen werden in Direct Discovery unterstützt, allerdings kann es Fälle geben, in denen bestimmte Quelldatenformate für Qlik Sense definiert werden müssen. Für die Definition von Datenformaten laden Sie das Ladeskript unter Verwendung der Syntax „SET Direct...Format“. Das folgende Beispiel zeigt, wie das Datenformat der Quelldatenbank, die als Quelle für Direct Discovery verwendet wird, zu definieren ist:

Beispiel:

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

Es gibt auch zwei Skriptvariablen zum Kontrollieren, wie Direct Discovery Geldwerte in den generierten SQL-Befehlen formatiert:

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

Die Syntax für diese beiden Variablen ist die gleiche wie für **MoneyFormat** und **MoneyDecimalSep**, aber es bestehen zwei wichtige Unterschiede in der Verwendung:

- Dies ist kein Anzeigeformat, daher sollte es keine Währungssymbole oder Tausendertrennzeichen enthalten.
- Die Standardwerte werden nicht vom Gebietsschema bestimmt, sondern sind an die Werte gebunden. (Für das Gebietsschema spezifische Formate beinhalten das Währungssymbol).

Direct Discovery kann die Auswahl erweiterter Unicode-Daten unterstützen, und zwar durch Verwendung des SQL-Standardformats für erweiterte Zeichenketten-Literale (N'<extended string>'), so wie dies für manche Datenbanken erforderlich ist, wie SQL Server. Diese Syntax kann für Direct Discovery mit der Skriptvariable **DirectUnicodeStrings** aktiviert werden. Wenn für diese Variable "true" eingestellt wird, dann ist die Verwendung von "N" vor den String-Literalen aktiviert.

Datensicherheit

Die folgenden Verhalten, die Auswirkungen auf die besten Praktiken für Datensicherheit haben können, sollten bei der Verwendung von Direct Discovery berücksichtigt werden:

4 Laden von Daten mit dem Datenladeskript

- Alle Benutzer, die die gleiche App mit der Direct Discovery-Funktion einsetzen, verwenden die gleiche Verbindung. Authentifizierungs-Durchgangsschleuse und individuelle Login-Daten pro Benutzer werden nicht unterstützt.
- Der Zugriffsabschnitt wird nur von der Server-Lösung unterstützt.
- Es ist möglich, benutzerdefinierte SQL-Befehle in der Datenbank mit einer NATIVE-Schlüsselwortformel auszuführen, sodass die im Ladeskript festgelegte Datenbankverbindung einen Account verwendet, der nur Lesezugriff auf die Datenbank hat.
- Direct Discovery hat keine Möglichkeit zur Protokollierung, aber es ist möglich, die ODBC-Ablaufverfolgung zu verwenden.
- Es ist möglich, die Datenbank mit Anfragen vom Client zu überschwemmen.
- Es ist möglich, detaillierte Fehlermeldungen aus den Serverprotokolldateien zu erhalten.

Qlik Sense-Funktion wird nicht unterstützt

Aufgrund der interaktiven und SQL-syntaxspezifischen Struktur von Direct Discovery werden mehrere Funktionen nicht unterstützt:

- Erweiterte Berechnungen (Satz-Analyse, komplexe Formeln)
- Dynamische Dimensionen
- Vergleichende Analysen (Alternative Status) für Objekte, die Direct Discovery-Felder verwenden
- Direct Discovery **MEASURE** und **DETAIL**-Felder werden bei der intelligenten Suche nicht unterstützt
- Binäres Laden aus einer Anwendung, die auf eine Direct Discovery-Tabelle zugreift
- Synthetische Schlüssel in der Direct Discovery-Tabelle
- Tabellenbenennung im Skript trifft nicht für die Direct Discovery-Tabelle zu
- Das Wildcard-Zeichen* nach einem **DIRECT QUERY**-Schlüsselwort im Ladeskript.

Beispiel: (DIRECT QUERY *)

- Oracle-Datenbanktabellen mitLONG-Datentypspalten.
- Hohe ganze Zahlen in wissenschaftlicher Schreibweise außerhalb des Bereichs [-9007199254740990, 9007199254740991]. Diese können Rundungsfehler und undefinierte Eigenschaften verursachen.

Unterstützung mehrerer Tabellen in Direct Discovery

Sie können Direct Discovery verwenden, um mehr als eine Tabelle zu laden oder mit der Verknüpfungsfunktion von ANSI SQL anzuzeigen. In einer einzelnen Tabelle müssen alle Kennzahlen aus der gleichen internen Tabelle in Qlik Sense abgeleitet sein, aber diese kann eine Kombination mehrerer, über Verknüpfungsbefehle verbundener Tabellen aus einer Quelle sein. Sie können jedoch aus anderen Tabellen bezogene Dimensionen im gleichen Diagramm verwenden.

Zum Beispiel können Sie die mit Direct Discovery durch eine **Where**- oder **Join**-Bedingung geladenen Tabellen verbinden.

- Direct Discovery kann in einem einzelnen Einzelwert/Mehrfachdimension-In-Memory-Szenario mit großen Datensätzen implementiert werden.
- Direct Discovery kann mit mehr als einer Tabelle verwendet werden, wenn eines der folgenden Kriterien erfüllt ist:
 - Die Kardinalität des Schlüsselfelds im Join ist niedrig.
 - Die Kardinalität des Schlüsselfelds im Join ist hoch, **DirectEnableSubquery** ist auf "true" gesetzt und alle Tabellen wurden mit Direct Discovery zusammengeführt.
Siehe: *Verwenden von Unterabfragen in Direct Discovery (Seite 127)*
- Direct Discovery ist nicht geeignet für die Implementierung in einem Dritte-Normalform-Szenario, wenn alle Tabellen in der Direct Discovery-Form vorliegen.

Verbinden von Direct Discovery-Tabellen mit einer **Where**-Bedingung

In diesem Beispielskript laden wir Daten aus der Datenbank AW2012. Die Tabellen Product und ProductSubcategory werden mit einer **Where**-Bedingung mithilfe des gemeinsamen Felds ProductSubCategoryID verbunden.

```
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 ;
```

Verbinden von Direct Discovery-Tabellen mit **Join On**-Bedingungen

Sie können auch **Join On**-Bedingungen zum Verbinden von Direct Discovery-Tabellen verwenden. In diesem Beispielbefehl verknüpfen wir die Tabelle SalesOrderHeader mit der Tabelle SalesOrderDetail über das Feld SalesOrderID. Ebenso verknüpfen wir die Tabelle Customer mit der Tabelle SalesOrderHeader über das Feld Customer ID.

In diesem Beispiel erstellen wir Kennzahlen aus der gleichen internen Tabelle, was bedeutet, wir können sie im gleichen Diagramm verwenden. Zum Beispiel können Sie ein Diagramm mit SubTotal und OrderQty als Kennzahlen erstellen.

```
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,
```

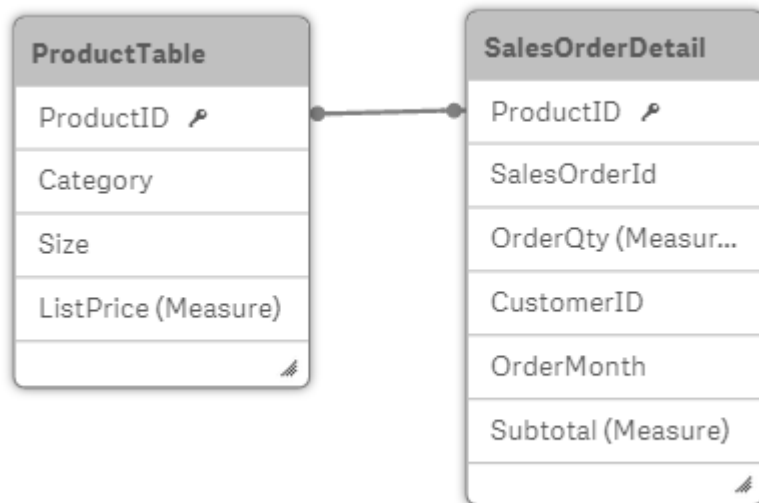
```
OrderDate,
NATIVE('month([OrderDate])') as OrderMonth,
NATIVE('year([OrderDate])') as OrderYear
MEASURE
SubTotal,
TaxAmt,
TotalDue,
OrderQty
DETAIL
DueDate,
ShipDate,
CreditCardApprovalCode,
PersonID,
StoreID,
AccountNumber,
rowguid,
ModifiedDate
FROM AW2012.Sales.SalesOrderDetail
JOIN AW2012.Sales.SalesOrderHeader
ON (AW2012.Sales.SalesOrderDetail.SalesOrderID =
    AW2012.Sales.SalesOrderHeader.SalesOrderID)
JOIN AW2012.Sales.Customer
ON(AW2012.Sales.Customer.CustomerID =
    AW2012.Sales.SalesOrderHeader.CustomerID);
```

Verwenden von Unterabfragen in Direct Discovery

Wenn die Kardinalität des Schlüsselfelds, das die Tabellen verbindet, hoch ist, d. h., wenn es eine große Anzahl von distinkten Werten enthält, kann eine Auswahl in Qlik Sense einen sehr großen SQL-Befehl zur Folge haben, weil die **WHERE key_field IN**-Bedingung eine große Anzahl von Werten umfasst. In diesem Fall besteht eine mögliche Lösung darin, Qlik Sense stattdessen mehrere Unterabfragen generieren zu lassen.

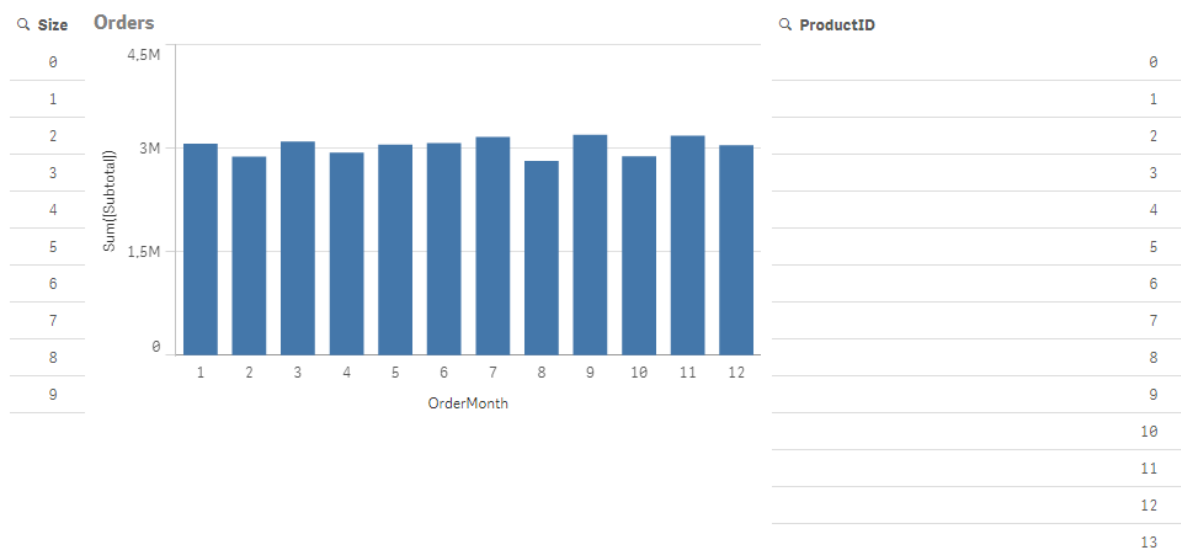
Wir veranschaulichen dies anhand einer Produkttabelle (ProductTable), die über eine Produkt-ID (SalesOrderDetail) mit einer Auftragsstabelle verlinkt ist (ProductID), wobei beide Tabellen im Direct Discovery-Modus sind.

4 Laden von Daten mit dem Datenladeskript



Wir erstellen ein Diagramm mit OrderMonth als Dimension und Sum(Subtotal) als Kennzahl und verwenden eine Filterbox zur Auswahl von Size.

Orders



Szenario 1: niedrige Kardinalität

In diesem Szenario enthält die Produktabelle eine geringe Anzahl an distinkten Produkten, 266. Wenn wir eine Auswahl in Size treffen, generiert Direct Discovery einen SQL-Befehl zur Lieferung der Daten und verwendet dafür eine **WHERE ProductID IN**-Bedingung, die Produkt-IDs in der ausgewählten Größe enthält, in diesem Fall 19 Produkte.

4 Laden von Daten mit dem Datenladeskript

Orders



Das generierte SQL sieht wie folgt aus:

```
SELECT ProductID, month([OrderDate]), SUM(OrderQty), SUM(SubTotal)
FROM SalesTable
WHERE ProductID IN ( 12, 51, 67, 81, 89, 92, 100, 108, 142, 150, 151, 162, 191, 206, 220, 222, 251,
254)
GROUP BY ProductID, month([OrderDate])
```

Szenario 2: Verwenden von Unterabfragen

Wenn dasselbe Beispiel eine große Anzahl an distinkten Produkten enthält, zum Beispiel 20.000, lässt sich durch die Auswahl eines Dimensionsfilters, zum Beispiel Size, ein SQL-Befehl mit einer **WHERE ProductID IN**-Bedingung generieren, die Tausende Produkt-IDs umfasst. Der daraus resultierende Befehl könnte aufgrund von Speicher- oder Leistungsbeschränkungen zu groß für die Verarbeitung durch die Datenquelle sein.

Die Lösung besteht darin, von Qlik Sense Unterabfragen erstellen zu lassen. Dazu muss **DirectEnableSubquery** auf "true" gesetzt werden. Der generierte SQL-Befehl könnte dann wie folgt aussehen:

```
SELECT ProductID, month([OrderDate]), SUM(OrderQty), SUM(SubTotal)
FROM SalesTable
WHERE ProductID IN
( SELECT DISTINCT "Aw2012"."dbo"."PRODUCT"."PRODUCTID" WHERE "Aw2012"."dbo"."PRODUCT"."SIZE" IN (3) )
GROUP BY ProductID, month([OrderDate])
```

4 Laden von Daten mit dem Datenladeskript

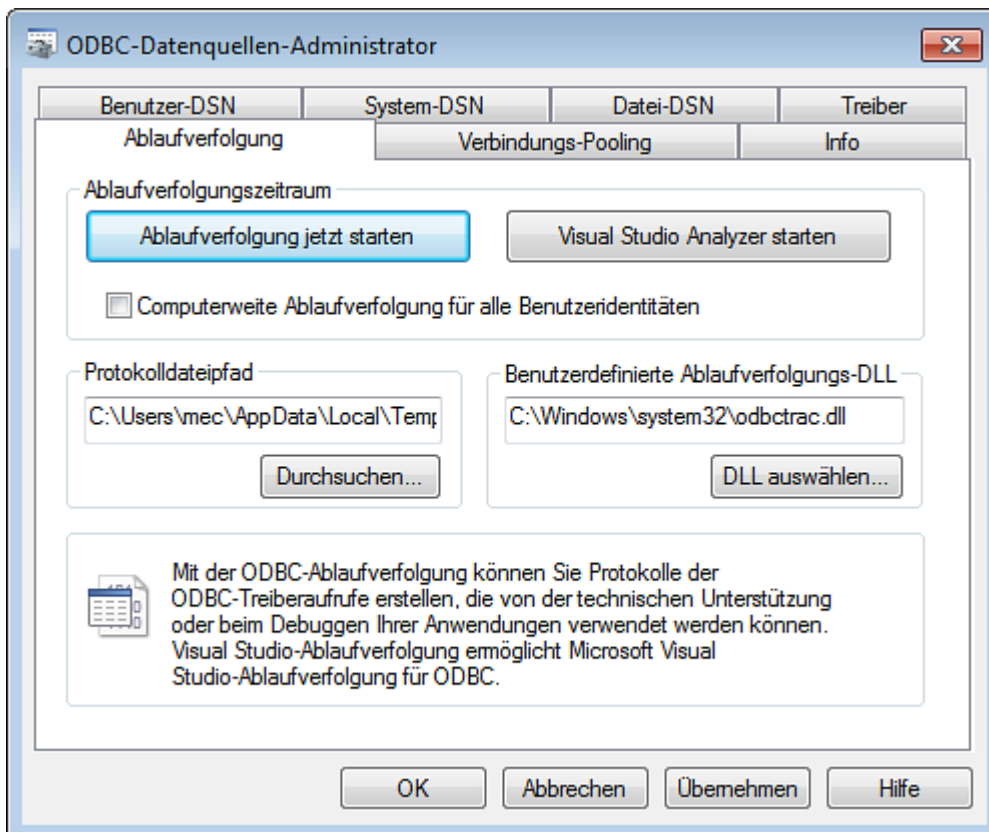
Die Größe der **WHERE ProductID IN**-Bedingung ist nun nicht mehr von der Anzahl der Schlüssel als Resultat der Auswahl abhängig.

Die folgenden Einschränkungen gelten bei der Verwendung von Unterabfragen:

- Die Syntax der Unterabfrage wird nur aufgerufen, wenn Sie Daten auswählen, die das Filtern eines Diagramms nach Daten einer anderen Tabelle umfassen.
- Die Datenmenge innerhalb der Schlüssel ist der begrenzende Faktor, nicht die Schlüsselanzahl.
- Unterabfragen werden nur aufgerufen, wenn alle beteiligten Tabellen im Direct Discovery-Modus sind. Wenn Sie das Diagramm nach den Daten einer Tabelle im In-Memory-Modus filtern, wird eine **IN**-Bedingung generiert.

Protokollieren von Zugriff auf Direct Discovery

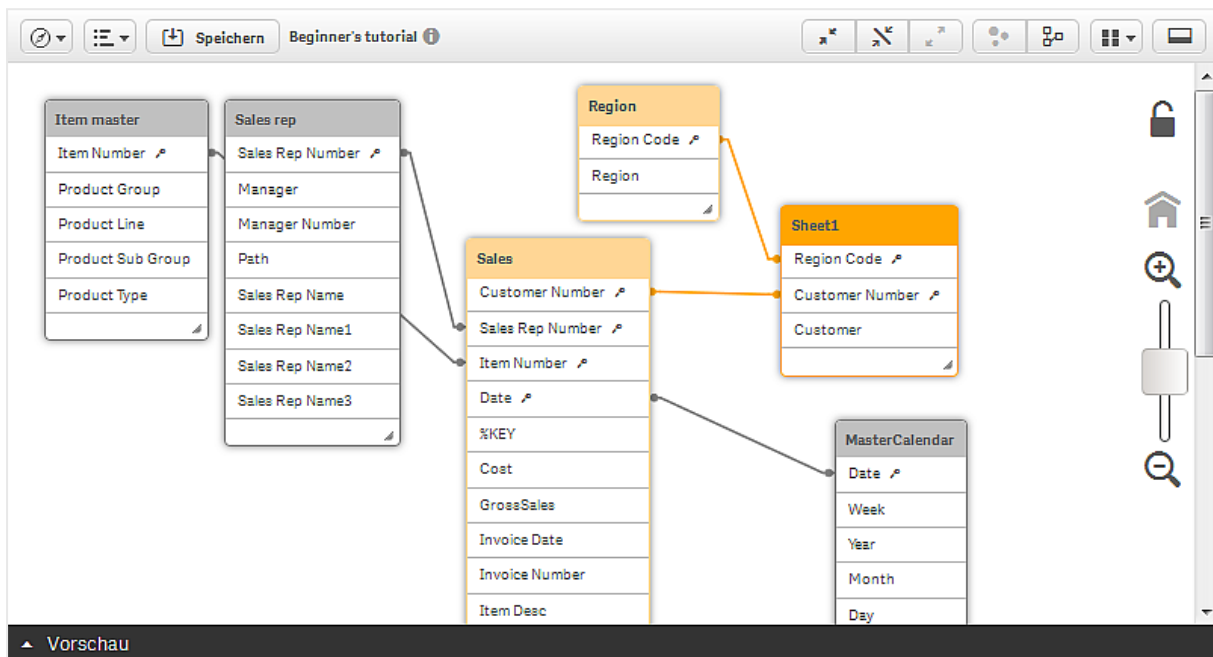
An die Datenquelle weitergegebene Direct DiscoverySQL-Befehle können in den trace-Dateien der Datenbankverbindung protokolliert werden. Bei einer Standard-ODBC-Verbindung wird die Ablaufverfolgung mit dem **ODBC Data Source Administrator** gestartet:






Die resultierende trace-Datei gibt die durch die Benutzerauswahlen und Interaktionen erzeugten SQL-Befehle an.

5 Anzeigen des Datenmodells

In der Datenmodellansicht erhalten Sie eine Übersicht über die Datenstruktur der App. Sie können eine Vorschau der Daten der Tabellen und Felder im Datenmodell anzeigen. Sie können Dimensionen und Kennzahlen auch ohne Vorlage anlegen.























Jede Datentabelle wird im Datenmodell durch eine Box dargestellt, wobei der Tabellename der Titel ist und alle Felder in der Tabelle aufgelistet werden. Tabellenzuordnungen werden mit Linien angezeigt, wobei eine gepunktete Linie auf einen Zirkelbezug verweist. Wenn Sie eine Tabelle oder ein Feld auswählen, erkennen Sie durch die Hervorhebung von Verknüpfungen sofort, welche Felder und Tabellen im Zusammenhang stehen.

Sie können die Zoom-Einstellung verändern, indem Sie auf ,  klicken oder den Schieberegler verwenden. Klicken Sie auf , um die Zoomeinstellung auf 1:1 zurückzusetzen.

5.1 Symbolleiste

Im Datenmodell weist die Symbolleiste oben auf dem Bildschirm folgende Tools auf:



	<p>Navigationsmenü mit folgenden Optionen:</p> <ul style="list-style-type: none">  App-Übersicht  Datenmanager  Dateneditor  Hub öffnen
---	---


	Menü mit folgenden Optionen:  Daten hinzufügen  Hilfe  Info
Speichern	Speichern Sie die Änderungen.
	Klicken Sie zum Ein- oder Ausblenden der App-Details auf das Infosymbol.
	Blenden Sie alle Tabellen komplett aus, damit nur der Tabellename angezeigt wird.
	Verringern Sie die Größe aller Tabellen, damit der Tabellename und alle Felder mit Verknüpfungen zu anderen Tabellen angezeigt werden.
	Blenden Sie alle Tabellen ein, damit alle Felder angezeigt werden.
	Interne Tabellenansicht – das Qlik Sense-Datenmodell einschließlich der synthetischen Felder.
	Datenquellen – das Datenmodell der Quelldatentabellen.
	Layout-Menü mit folgenden Optionen:  Raster-Layout  Auto-Layout  Layout wiederherstellen
	Öffnen und Schließen des Fensters Vorschau.




5.2 Verschieben und Anpassen der Größe von Tabellen im Datenmodell

Verschieben von Tabellen

Sie können Tabellen durch Ziehen verschieben. Die Tabellenpositionen werden gespeichert, wenn die App gespeichert wird.

Sie können das Tabellenlayout (Position und Größe) sperren, indem Sie im rechten Bereich auf  klicken. Klicken Sie zum Entsperrern des Tabellenlayouts auf .

Sie können das Layout auch automatisch anordnen, indem Sie die Optionen unter  in der Symbolleiste verwenden:

-  **Raster-Layout** – die Tabellen werden in einem Raster angeordnet.
-  **Auto-Layout** – die Tabellen werden an das Fenster angepasst.
-  **Layout wiederherstellen** - das Layout wird auf den letzten Status zurückgesetzt, der in der Datenmodellansicht geöffnet war.

Anpassen der Größe von Tabellen

Sie können die Anzeigegröße von Tabellen mithilfe des Pfeils rechts unten an einer Tabelle anpassen. Die Anzeigegröße wird nicht gespeichert, wenn die App gespeichert wird.

Sie können auch die Optionen für die automatische Anpassung der Anzeigegröße verwenden:

✖ **Nächste Ebene komplett ausblenden** – blendet alle Tabellen komplett aus, damit nur der Tabellename angezeigt wird.

↗ **Verlinkte Felder anzeigen** – verringert die Größe aller Tabellen, damit der Tabellename und alle Felder mit Verknüpfungen zu anderen Tabellen angezeigt werden.


✔ **Nächste Ebene komplett einblenden** – blendet alle Tabellen ein, damit alle Felder in den Tabellen angezeigt werden.

5.3 Vorschau von Tabellen und Feldern im Datenmodell


Im Datenmodell können Sie eine Vorschau jeder Datentabelle unten auf dem Bildschirm anzeigen. Mithilfe der Vorschau können Sie schnell den Inhalt einer Tabelle oder eines Feldes prüfen. Wenn Sie ein Feld auswählen, können Sie der App schnell Dimensionen und Kennzahlen hinzufügen.

Außerdem werden im Vorschaubereich die Metadaten der ausgewählten Tabelle bzw. des Felds angezeigt.

Sie können den Vorschaubereich wie folgt ein- oder ausblenden:

- Klicken Sie in der Symbolleiste auf .
- Klicken Sie auf **Preview**.



Direct Discovery-Daten werden in der Vorschau  nicht angezeigt.

Vorschau einer Tabelle anzeigen

Gehen Sie folgendermaßen vor:

- Klicken Sie im Datenmodell auf eine Tabellenüberschrift.

Der Vorschaubereich mit Feldern und Werten der ausgewählten Tabelle wird angezeigt.

Item master		Vorschau der Daten				
Item Number	Product Group	Product Line	Product Sub Group	Product Type		
10001	Beverages	Drink	Juice	Pure Juice Beverages		
10002	Beverages	Drink	Flavored Drinks	Drinks		
10003	Beverages	Drink	Flavored Drinks	Drinks		
10004	Beverages	Drink	Soda	Carbonated Beverages		
10005	Beverages	Drink	Soda	Carbonated Beverages		
10006	Beverages	Drink	Soda	Carbonated Beverages		

Vorschau eines Feldes anzeigen

Gehen Sie folgendermaßen vor:

- Klicken Sie im Datenmodell auf ein Tabellenfeld.

Der Vorschaubereich wird mit dem ausgewählten Feld und seinen Werten sowie Metadaten angezeigt. Sie können das Feld auch als Master-Dimension oder -Kennzahl hinzufügen.

Product Line		Vorschau der Daten				
Dichte	100%	Item Number	Product Group	Product Line	Product Sub Group	Product Type
Anteil der Werte	100%	10001	Beverages	Drink	Juice	Pure Juice
Hat Duplikate	wahr	10002	Beverages	Drink	Flavored Drinks	Drinks
Gesamt: Distinkte Werte	2	10003	Beverages	Drink	Flavored Drinks	Drinks
Anzeigen: Distinkte Werte	2	10004	Beverages	Drink	Soda	Carbonate
Nicht NULL-Werte	827	10005	Beverages	Drink	Soda	Carbonate
Tags:	\$ascii \$text	10006	Beverages	Drink	Soda	Carbonate

- Die **Dichte** (Density) ist die Zahl der Datensätze, die in diesem Feld einen Wert haben (d. h. nicht NULL sind) im Verhältnis zur Gesamtzahl der Datensätze der Tabelle.
- Die **Menge der Werte** (Subset Ratio) ist die Zahl der distinkten Werte des Feldes innerhalb der Tabelle im Verhältnis zur Gesamtzahl der Werte des Feldes in anderen Tabellen des Datenmodells. Dies ist nur für Schlüsselfelder relevant.
- Wenn das Feld mit **[Perfect key]** markiert ist, enthält jede Zeile einen einzigartigen Schlüsselwert.

5.4 Erstellen einer Master-Dimension aus dem Datenmodell

Wenn Sie mit einer nicht veröffentlichten App arbeiten, können Sie Master-Dimensionen erstellen, die wiederverwendet werden können. Benutzer einer veröffentlichten App haben Zugriff auf die Master-Dimensionen in der Bibliothek, können diese jedoch nicht bearbeiten. Das Datenmodell ist in einer veröffentlichten App nicht verfügbar.

Gehen Sie folgendermaßen vor:

1. Wählen Sie im Datenmodell ein Feld aus und öffnen Sie das Fenster für die **Vorschau**.
2. Klicken Sie auf **Als Dimension hinzufügen**.
Der Dialog **Neue Dimensionen erstellen** wird mit dem ausgewählten Feld angezeigt. Der Name des ausgewählten Feldes wird auch als Standardname der Dimension verwendet.
3. Sie können den Namen ändern und optional eine Beschreibung und Tags hinzufügen.
4. Klicken Sie auf **Neue Dimension**.
5. Klicken Sie auf **Erledigt**, um den Dialog zu schließen.

Die Dimension wird jetzt unter der Dialogseite "Master-Elemente" des Extras-Fensters gespeichert.



Sie können schnell verschiedene Dimensionen als Master-Elemente hinzufügen, indem Sie nach dem Hinzufügen der einzelnen Dimensionen auf **Dimension hinzufügen** klicken. Klicken Sie auf **Erledigt** wenn Sie fertig sind.



Direct Discovery-Tabellen sind im Datenmodell durch  gekennzeichnet.

5.5 Erstellen einer Master-Kennzahl aus dem Datenmodell

Wenn Sie mit einer nicht veröffentlichten App arbeiten, können Sie Master-Kennzahlen erstellen, die wiederverwendet werden können. Benutzer einer veröffentlichten App haben Zugriff auf die Master-Kennzahlen in der Bibliothek, können diese jedoch nicht bearbeiten.

Gehen Sie folgendermaßen vor:

1. Wählen Sie im Datenmodell ein Feld aus und öffnen Sie das Fenster für die **Vorschau**.
2. Klicken Sie auf **Als Kennzahl hinzufügen**.
Der Dialog **Neue Kennzahl erstellen** wird mit dem ausgewählten Feld angezeigt. Der Name des ausgewählten Felds wird auch als Standardname der Kennzahl verwendet.
3. Geben Sie eine Formel für die Kennzahl ein.
4. Sie können den Namen ändern und optional eine Beschreibung und Tags hinzufügen.
5. Klicken Sie auf **Erstellen**.

Die Kennzahl wird jetzt unter der Dialogseite "Master-Elemente" des Extras-Fensters gespeichert.

6 Beste Praktiken für Datenmodelle

In diesem Abschnitt werden abhängig von der Datenstruktur und vom gewünschten Datenmodell unterschiedliche Möglichkeiten zum Laden von Daten in die Qlik Sense-App beschrieben.

6.1 Umwandeln von Datenspalten in Zeilen

Meine Daten sehen wahrscheinlich wie folgt aus und ich möchte die Umsatzzahlen in einem separaten Feld anzeigen:

Year	Q1	Q2	Q3	Q4
2013	34	54	53	52
2014	47	56	65	67
2015	57	56	63	71

Empfohlene Vorgehensweise

Verwenden Sie beim Laden der Tabelle den **Crosstable**-Zusatz.

Das Ergebnis sieht folgendermaßen aus:

Year	Quarter	Sales
2013	Q1	34
2013	Q2	54
2013	Q3	53
2013	Q4	52
2014	Q1	47
...

6.2 Umwandeln von Datenzeilen in Felder

Ich habe eine generische Tabelle mit drei Feldern, die mit diesem Beispiel vergleichbar sind, und ich möchte jedes Attribut als separate Tabelle haben:

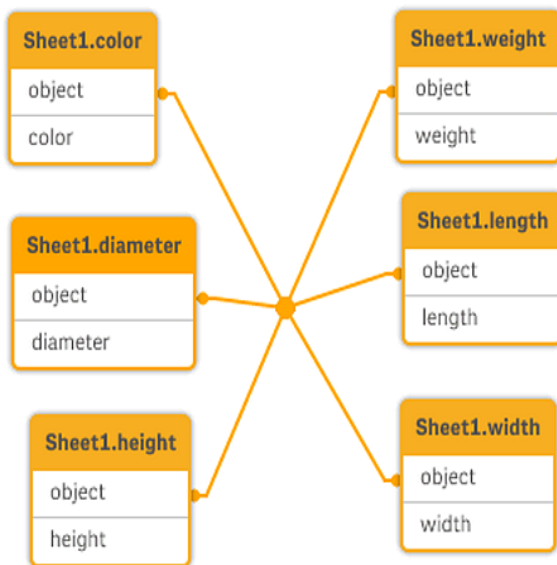
Object	Attribute	Value
ball	color	red
ball	diameter	25
ball	weight	3

Object	Attribute	Value
box	color	56
box	height	30
box	length	20
box	width	25

Empfohlene Vorgehensweise

Erstellen Sie ein generisches Datenmodell unter Verwendung des Load-Zusatzes **Generisch**.

Sie erhalten ein Datenmodell, das folgendermaßen aussieht:



6.3 Laden von Daten, die in hierarchischen Ebenen organisiert sind, z. B. in einem Organisationsschema.

Meine Daten sind in einer Tabelle mit benachbarten Knoten gespeichert, die folgendermaßen aussieht:

NodeID	ParentNodeID	Title
1	-	General manager
2	1	Country manager
3	2	Region manager

Empfohlene Vorgehensweise

Laden Sie die Daten mit dem Hierarchy-Zusatz, um eine Tabelle mit aufgeschlüsselten Ebenen zu erstellen:

NodeID	ParentNodeID	Title	Level1	Level2	Level3
1	-	General manager	General manager	-	-
2	1	Country manager	General manager	Country manager	-
3	2	Region manager	General manager	Country manager	Region manager

6.4 Laden von ausschließlich neuen oder aktualisierten Datensätzen aus einer großen Datenbank

Ich habe eine Datenbank mit einer großen Anzahl von Datensätzen und möchte es vermeiden, die gesamte Datenbank neu zu laden, um die Daten in meiner App zu aktualisieren. Ich möchte lediglich neue oder aktualisierte Datensätze laden und gelöschte Datensätze aus der Datenbank entfernen.

Empfohlene Vorgehensweise

Implementieren Sie eine Lösung mit inkrementellem Laden unter Verwendung von QVD-Dateien.

6.5 Zusammenführen von Daten aus zwei Tabellen mit einem gemeinsamen Feld

Qlik Sense verknüpft Tabellen mit einem gemeinsamen Feld automatisch, aber ich möchte selbst bestimmen, wie die Tabellen zusammengeführt werden.

Empfohlene Vorgehensweise : Join / Keep

Sie können zwei Tabellen mit den Zusätzen **Join** oder **Keep** in einer einzigen Tabelle zusammenführen.

Empfohlene Vorgehensweise : Mapping

Eine Alternative zum Zusammenlegen von Tabellen ist das Mapping. Dabei wird die Suche nach verknüpften Werten in einer Mapping-Tabelle automatisiert. Dadurch können Sie die zu ladende Datenmenge reduzieren.

6.6 Einordnung eines diskreten Werts in ein Intervall

Ich habe eine Tabelle aus diskreten numerischen Werten (Event) und möchte sie in ein oder mehrere Intervalle einordnen (Start und End).

Time	Event	Comment	Start	End	Order
00:00	0	Start of shift 1			
01:18	1	Line stop	01:00	03:35	A
02:23	2	Line restart 50%	02:30	07:58	B
04:15	3	Line speed 100%	03:04	10:27	C
08:00	4	Start of shift 2	07:23	11:43	D
11:43	5	End of production			

Empfohlene Vorgehensweise

Verwenden Sie den **IntervalMatch**-Zusatz, um das Feld Time mit dem durch Start und End definierten Intervall zu verknüpfen.

Sollte das Intervall nicht ausdrücklich mit Start und Ende definiert sein, sondern nur mit einem Zeitstempel wie in der nachfolgenden Tabelle, dann müssen Sie eine Intervalltabelle erstellen.

Currency	Change Date	Rate
EUR		8.59
EUR	28/01/2013	8.69
EUR	15/02/2013	8.45
USD		6.50
USD	10/01/2013	6.56
USD	03/02/2013	6.30

6.7 Handhabung von inkonsistenten Feldwerten

Meine Daten enthalten Feldwerte, die in verschiedenen Tabellen unterschiedliche Bezeichnungen aufweisen. Beispielsweise enthält eine Tabelle den Wert US im Feld „Land“, wohingegen eine andere Tabelle den Wert United States enthält. Diese Situation verhindert Verknüpfungen.

Table 1		Table 2	
Country	Region	Country	Population
US	Maryland	United States	304
US	Idaho	Japan	128
US	New York	Brazil	192
US	California	China	1333

Empfohlene Vorgehensweise

Führen Sie mit Hilfe einer Mapping-Tabelle eine Datenpflege durch. Dabei werden Feldwerte verglichen und korrekte Verknüpfungen ermöglicht.

6.8 Laden räumlicher Daten zur Visualisierung von Daten mit einer Karte

Ich habe Daten, die ich mit Hilfe einer Karte visualisieren möchte, z. B. Umsatzdaten pro Land oder pro Geschäft. Zur Verwendung der Kartenvisualisierung muss ich Bereichs- oder Punktdaten laden.

Empfohlene Vorgehensweise

Sie können Bereichs- oder Punktdaten, die mit Ihren Datenwertstandorten übereinstimmen, aus einer KML- oder einer Excel-Datei laden. Zusätzlich müssen Sie die eigentliche Hintergrundkarte laden.

6.9 Laden von neuen und aktualisierten Datensätzen mit inkrementellem Laden

Wenn Ihre App große Datenmengen aus Datenbankquellen enthält, die kontinuierlich aktualisiert werden, dann kann das erneute Laden des gesamten Datensatzes zeitaufwändig sein. Beim inkrementellen Laden werden nur neue oder geänderte Datensätze aus der Datenbank geladen. Alle übrigen Daten sollten bereits in der App verfügbar sein. Dies erreichen Sie durch das inkrementelle Laden mit QVD-Dateien.

Der Prozess läuft folgendermaßen ab:

1. Laden Sie neue oder aktualisierte Daten aus der Datenbankquelltable.
Dieser Prozess dauert lange, es wird jedoch nur eine begrenzte Zahl von Datensätzen geladen.
2. Laden Sie bereits in der App vorhandene Daten aus der QVD-Datei.
Es werden viele Datensätze geladen, der Prozess geht jedoch viel schneller.
3. Erstellen Sie eine neue QVD-Datei.
Das ist die Datei, die Sie beim nächsten inkrementellen Laden verwenden werden.
4. Diese Schritte werden für alle Tabellen wiederholt.

Die folgenden Beispiele zeigen Fälle, in denen inkrementelles Laden verwendet wird. Wenn es die Struktur der Quelldatenbank und die Vorgehensweise erfordern, können jedoch komplexere Lösungen erforderlich sein.

- Nur angehängte Datensätze (typischerweise für Protokolldateien verwendet)
- Nur eingefügte Datensätze (keine Aktualisierung oder Löschung)
- Neu eingefügte und geänderte Datensätze (keine Löschung)
- Neu eingefügte, geänderte und gelöschte Datensätze

Sie können QVD-Dateien im optimierten Modus oder im Standardmodus einlesen. (Je nach Komplexität der Operation ergibt sich die ausgewählte Methode automatisch aus der Qlik Sense-Engine.) Der optimierte Modus ist ungefähr 10-mal schneller als der Standardmodus oder etwa 100-mal schneller als der herkömmliche Ladevorgang der Datenbank.

Nur angehängte Datensätze

Das einfachste Beispiel für einen solchen Fall ist eine Protokolldatei, bei der neue Datensätze einfach hinten angehängt und niemals gelöscht werden. Folgende Bedingungen müssen erfüllt sein:

- Die Datenbank muss eine Protokolldatei sein (oder eine andere Datei, in der Datensätze angehängt und nicht eingefügt oder gelöscht werden). Die Protokolldatei muss sich in einer Textdatei befinden (ODBC-, OLE DB- und andere Datenbanken werden nicht unterstützt).
- Qlik Sense speichert die Anzahl der bereits vorher geladenen Datensätze und liest nur die neuen Datensätze am Ende der Datei ein.

Beispiel:

```
Buffer (Incremental) Load * From LogFile.txt (ansi, txt, delimiter is '\t', embedded labels);
```

Nur eingefügte Datensätze (keine Aktualisierung oder Löschung)

Werden die Daten in der Datenquelle anders als in einer einfachen Log-Datei angehängt, reicht der Ansatz "Nur angehängte Datensätze" nicht aus. Das Problem kann dennoch mit ein klein wenig mehr Aufwand gelöst werden. Folgende Bedingungen müssen erfüllt sein:

- Die Datenquelle kann eine beliebige sein.
- Qlik Sense liest nur die Datensätze ein, die nach der letzten Ausführung des Skripts neu angelegt wurden.
- Es existiert ein Feld, das eine ModificationTime oder ähnliches enthält und anhand dessen Qlik Sense erkennt, welche Datensätze neu sind.

Beispiel:

```
QV_Table:
SQL SELECT PrimaryKey, X, Y FROM DB_TABLE
WHERE ModificationTime >= #$(LastExecTime)#
AND ModificationTime < #$(BeginningThisExecTime)#;
```

```
Concatenate LOAD PrimaryKey, X, Y FROM File.QVD;
STORE QV_Table INTO File.QVD;
```

Die Hash-Zeichen in der SQL WHERE-Bedingung definieren den Anfang und das Ende eines Datums. Im Datenbank-Handbuch finden Sie die korrekte Datums-Syntax für Ihre Datenbank.

Neu eingefügte und geänderte Datensätze (keine Löschung)

In diesem Fall wurden bestehende Datensätze seit der letzten Ausführung des Skripts geändert. Folgende Bedingungen müssen erfüllt sein:

- Die Datenquelle kann eine beliebige sein.
- Qlik Sense liest nur die Datensätze ein, die nach der letzten Ausführung des Skripts neu angelegt oder geändert wurden.
- Es existiert ein Feld, das eine ModificationTime oder ähnliches enthält und anhand dessen Qlik

Sense erkennt, welche Datensätze neu sind.

- Ein Primärschlüssel wird benötigt, anhand dessen Qlik Sense geänderte Datensätze in der QVD-Datei erkennt.
- Diese Lösung bewirkt einen Wechsel zum Lese-Standardmodus für die QVD-Datei (statt des optimierten Modus), was bedeutend schneller ist als das Laden der gesamten Datenbank.

Beispiel:

```
QV_Table:
SQL SELECT PrimaryKey, X, Y FROM DB_TABLE
WHERE ModificationTime >= #$(LastExecTime)#;

Concatenate LOAD PrimaryKey, X, Y FROM File.QVD
WHERE NOT Exists(PrimaryKey);

STORE QV_Table INTO File.QVD;
```

Neu eingefügte, geänderte und gelöschte Datensätze

Dies ist der komplizierteste Fall, denn hier wurden seit der letzten Ausführung des Skripts zusätzlich Datensätze gelöscht. Folgende Bedingungen müssen erfüllt sein:

- Die Datenquelle kann eine beliebige sein.
- Qlik Sense liest nur die Datensätze ein, die nach der letzten Ausführung des Skripts neu angelegt oder geändert wurden.
- Qlik Sense entfernt Datensätze, die nach der letzten Ausführung des Skripts aus der Datenbank gelöscht wurden.
- Es existiert ein Feld, das eine ModificationTime oder ähnliches enthält und anhand dessen Qlik Sense erkennt, welche Datensätze neu sind.
- Ein Primärschlüssel wird benötigt, anhand dessen Qlik Sense geänderte Datensätze in der QVD-Datei erkennt.
- Diese Lösung bewirkt einen Wechsel zum Lese-Standardmodus für die QVD-Datei (statt des optimierten Modus), was bedeutend schneller ist als das Laden der gesamten Datenbank.

Beispiel:

```
Let ThisExecTime = Now( );

QV_Table:
SQL SELECT PrimaryKey, X, Y FROM DB_TABLE
WHERE ModificationTime >= #$(LastExecTime)#
AND ModificationTime < #$(ThisExecTime)#;

Concatenate LOAD PrimaryKey, X, Y FROM File.QVD
WHERE NOT EXISTS(PrimaryKey);

Inner Join SQL SELECT PrimaryKey FROM DB_TABLE;

If ScriptErrorCount = 0 then
STORE QV_Table INTO File.QVD;
Let LastExecTime = ThisExecTime;
```

End If

6.10 Kombinieren von Tabellen mit Join und Keep

Bei der Verknüpfung werden zwei Tabellen zu einer Tabelle kombiniert. Dabei bilden jeweils zwei Datensätze aus den Ursprungstabellen, meist zwei Datensätze, die einen Wert in einem Feld gemeinsam haben, einen neuen Datensatz in der entstehenden Tabelle. In Qlik Sense können Verknüpfungen im Skript definiert werden. Dabei werden programminterne Tabellen erzeugt.

Durch den **join**-Zusatz im Skript ist es möglich, Tabellen beim Einlesen durch Joins zusammenzuschließen. In diesem Fall speichert Qlik Sense nur eine einzelne interne Tabelle als Ergebnis des Joins. Dies ist in einigen Situationen erforderlich, jedoch bestehen auch Nachteile.

- Die internen Tabellen werden fast immer größer und Qlik Sense arbeitet entsprechend langsamer.
- Einige Informationen gehen möglicherweise verloren, z. B. die Häufigkeit (Anzahl der Datensätze) in der zugrunde liegenden Tabelle.

Der **Keep**-Zusatz bewirkt, dass vor dem Speichern in Qlik Sense die beiden Tabellen verglichen werden und dass aufgrund dieses Vergleichs in einer oder in beiden Tabellen bestimmte Datensätze wegfallen. Dadurch kann in den meisten Fällen auf einen Join verzichtet werden.



In diesem Handbuch wird der Begriff „Join“ nur für Joins gebraucht, die vor dem Speichern der Tabellen entstehen. Die Verknüpfung von Tabellen, nachdem diese angelegt wurden, ist im Grunde genommen aber auch nichts anderes als ein Join.

Joins innerhalb eines SQL SELECT-Befehls

Bei einigen ODBC-Treibern ist es möglich, Joins durch **SELECT**-Befehle herzustellen. Dies hat im Großen und Ganzen dieselbe Wirkung wie ein **Join**-Zusatz.

Die meisten ODBC-Treiber sind jedoch nicht in der Lage, einen vollständigen (bidirektionalen) Outer Join herzustellen. Sie können nur linke oder rechte Outer Joins herstellen. Ein rechter (bzw. linker) Join funktioniert nur, wenn in der rechten (bzw. linken) Tabelle der entsprechende Schlüssel existiert. Ein vollständiger Outer Join enthält jede Kombination. Qlik Sense stellt automatisch vollständige Outer Joins her.

Joins direkt in **SELECT**-Befehlen zu erstellen, ist ungleich schwieriger, als die Verknüpfungen in Qlik Sense zu nutzen.

Beispiel:

```
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;
```

Dieser **SELECT**-Befehl schließt eine Tabelle mit Aufträgen und eine Tabelle mit Auftragsdaten durch einen Join zusammen. Es handelt sich um einen rechten Join, d. h. alle Datensätze aus der Tabelle *OrderDetails* sind enthalten, auch diejenigen mit einer *OrderID*, die in der Tabelle *Orders* nicht vorkommt. Datensätze, die in der Tabelle *Orders*, nicht aber in der Tabelle *OrderDetails* vorkommen, sind nicht enthalten.

Join

Am einfachsten erstellen Sie Joins durch einen **Join**-Zusatz im Skript, mit dem die interne Tabelle mit einer anderen benannten Tabelle oder mit der zuletzt erstellten Tabelle zusammengefügt wird. Dieser kombiniert die Tabelle mit einer bereits erstellten Tabelle durch einen Outer Join, d. h. in der entstehenden Tabelle sind alle möglichen Kombinationen von Datensätzen enthalten.

Beispiel:

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

Die entstehende Tabelle enthält die Felder a, b, c und d. Die Anzahl der Datensätze hängt von den Feldwerten der beiden Tabellen ab.



*Die Namen der Felder, über die der Join gemacht wird, müssen exakt übereinstimmen. Der Join kann über beliebig viele Felder gemacht werden. Meistens haben die Tabellen nur ein oder wenige Felder gemeinsam. Haben die Tabellen gar kein Feld gemeinsam, so ergibt der Join das karthesische Produkt der Tabellen. Es ist möglich, dass die Tabellen in allen Feldnamen übereinstimmen, dies ergibt jedoch im Allgemeinen keinen Sinn. Sofern im **Join**-Befehl mit **Join**-Zusatz kein anderer Tabellename einer zuvor geladenen Tabelle angegeben ist, bezieht er sich jeweils auf die direkt zuvor angelegte Tabelle. Die Reihenfolge der Befehle ist daher nicht beliebig.*

Keep

Der ausdrückliche Zusatz **Join** im Datenladeskript bewirkt eine vollständige Zusammenfügung der beiden Tabellen, d. h. die beiden Tabellen werden zu einer zusammengeschlossen. Dies führt oft zu sehr umfangreichen Tabellen. Einer der großen Vorteile von Qlik Sense besteht aber gerade darin, dass automatisch Verknüpfungen zwischen Tabellen hergestellt werden, ohne dass ein Join gemacht wird. Dadurch wird Speicherplatz gespart, die Zugriffszeiten verkürzen sich, und das System behält eine hohe Flexibilität. Durch den **Keep**-Zusatz reduziert sich die Zahl der Fälle, in denen Joins benötigt werden, nochmals.

Der Zusatz **Keep** zwischen zwei **LOAD**- oder **SELECT**-Befehlen bewirkt, dass vor dem Speichern in Qlik Sense die beiden Tabellen verglichen werden und aufgrund dieses Vergleichs in einer oder in beiden Tabellen bestimmte Datensätze wegfallen. Dem **Keep**-Zusatz muss stets einer der Zusätze **Inner**, **Left** oder **Right** vorangehen. Die Auswahl der Datensätze in den Tabellen erfolgt nach denselben Regeln wie bei dem entsprechenden Join. Die Tabellen werden jedoch nicht zusammengeschlossen, sondern als zwei Tabellen unter verschiedenen Namen in Qlik Sense gespeichert.

Inner

Vor **Join** und **Keep** kann im Datenladeskript der Zusatz **Inner** gestellt werden.

Vor **Join** bewirkt er, dass ein Inner Join hergestellt wird. Die dadurch entstehende Tabelle enthält nur die Datensätze, zu denen in der jeweils anderen Tabelle ein passender Datensatz vorhanden ist.

Vor einem **Keep**-Zusatz bewirkt "inner", dass von beiden Tabellen nur die Datensätze in Qlik Sense gespeichert werden, für die in der jeweils anderen Tabelle ein passender Datensatz vorhanden ist.

Beispiel:

In diesen Beispielen werden die Quelltabellen Table1 und Table2 verwendet:

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

Quelltabellen Beispiele Inner

Inner Join

Zunächst wird für die Tabellen ein **Inner Join** durchgeführt. Daraus ergibt sich VTable mit lediglich einer einzigen Zeile, der einzige Datensatz, der in beiden Tabellen vorhanden ist, wobei die Daten von beiden Tabellen kombiniert werden.

VTable:
SELECT * from Table1;
inner join SELECT * from Table2;

VTable		
A	B	C
1	aa	xx

Beispiel Inner Join

Inner Keep

Wird stattdessen ein **Inner Keep** durchgeführt, bleiben weiterhin zwei Tabellen bestehen. Die beiden Tabellen sind über das gemeinsame Feld A miteinander verknüpft.

VTab1:
SELECT * from Table1;
VTab2:
inner keep SELECT * from Table2;

VTab1		VTab2	
A	B	A	C
1	aa	1	xx

Beispiel Inner Keep

Left

Vor **Join** und **Keep** kann im Datenladeskript der Zusatz **left** gestellt werden.

Vor einem **Join**-Zusatz bewirkt "left", dass ein Left Join hergestellt wird. Die dadurch entstehende Tabelle enthält die Datensätze der ersten Tabelle ergänzt mit den passenden Sätzen der zweiten Tabelle, sofern vorhanden.

Vor einem **Keep**-Zusatz bewirkt "left", dass von der zweiten Tabelle nur die Datensätze in Qlik Sense gespeichert werden, für die ein passender Datensatz in der ersten Tabelle existiert.

Beispiel:

In diesen Beispielen werden die Quelltabellen Table1 und Table2 verwendet:

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

Quelltabellen Beispiele Left

Zunächst wird für die Tabellen ein **Left Join** durchgeführt, wodurch eine VTable mit allen Zeilen aus Table1 entsteht, die mit den passenden Feldern in Table2 kombiniert wird.

VTable:

```
SELECT * from Table1;
left join SELECT * from Table2;
```

VTable		
A	B	C
1	aa	xx
2	cc	—
3	ee	—

Beispiel Left Join

Wird stattdessen ein **Left Keep** durchgeführt, bleiben weiterhin zwei Tabellen bestehen. Die beiden Tabellen sind über das gemeinsame Feld A miteinander verknüpft.

VTab1:

```
SELECT * from Table1;
```

VTab2:

left keep SELECT * from Table2;

VTab1	
A	B
1	aa
2	cc
3	ee

VTab2	
A	C
1	xx

Beispiel Left Keep

Right

Vor **Join** und **Keep** kann im Datenladeskript der Zusatz **right** gestellt werden.

Vor einem **Join**-Zusatz bewirkt "right", dass ein Right Join hergestellt wird. Die dadurch entstehende Tabelle enthält die Datensätze der zweiten Tabelle, ergänzt durch die passenden Sätze der ersten Tabelle, sofern vorhanden.

Vor einem **Keep**-Zusatz bewirkt "right", dass von der ersten Tabelle nur die Datensätze in Qlik Sense gespeichert werden, für die ein passender Datensatz in der ersten Tabelle existiert.

Beispiel:

In diesen Beispielen werden die Quelltabellen Table1 und Table2 verwendet:

Table1	
A	B
1	aa
2	cc
3	ee

Table2	
A	C
1	xx
4	yy

Quelltabellen Beispiele Right

Zunächst wird für die Tabellen ein **Right Join** durchgeführt, wodurch eine VTable mit allen Zeilen aus Table2 entsteht, die mit den passenden Feldern in Table1 kombiniert wird.

VTable:

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

VTable		
A	B	C
1	aa	xx
4	-	yy

Beispiel Right Join

Wird stattdessen ein **Left Keep** durchgeführt, bleiben weiterhin zwei Tabellen bestehen. Die beiden Tabellen sind über das gemeinsame Feld A miteinander verknüpft.

VTab1:

```
SELECT * from Table1;
VTab2:
right keep SELECT * from Table2;
```

VTab1		VTab2	
A	B	A	C
1	aa	1	xx
		4	yy

Beispiel Right Keep

6.11 Mapping als Alternative zu Joining

Der Zusatz **Join** bietet in Qlik Sense eine leistungsstarke Funktion, die es ermöglicht, mehrere Datentabellen in einem Datenmodell zu kombinieren. Ein Nachteil davon ist jedoch, dass die kombinierten Tabellen sehr groß werden können, worunter möglicherweise die Performance leidet. Wenn Sie einen einzelnen Wert aus einer anderen Tabelle aufrufen müssen, ist Mapping eine Alternative zu **Join**. Dies kann Ihnen das Laden überflüssiger Daten ersparen, die Ihre Berechnungen nur unnötig verlangsamen und potenziell zu Berechnungsfehlern führen können, da sich durch Join-Befehle die Anzahl der Datensätze in den Tabellen ändern kann.

Eine Mapping-Tabelle besteht aus zwei Spalten, einem Vergleichsfeld (Eingabe) und einem Mapping-Feldwert (Ausgabe).

Diesem Beispiel liegt eine Tabelle mit Bestellungen (Orders) zugrunde und wir möchten erfahren, in welchem Land die jeweiligen Kunden beheimatet sind. Diese Information ist in der Kundentabelle (Customers) gespeichert.

Orders-Datentabelle

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-Datentabelle

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

Zum Abrufen des Landes (Country) eines Kunden ist eine Mapping-Tabelle erforderlich, die folgendermaßen aussieht:

CustomerID	Country
1	Spain
2	Italy
3	Germany
4	France

Die Mapping-Tabelle mit dem Namen MapCustomerIDtoCountry ist im Skript folgendermaßen definiert:

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

Im nächsten Schritt wird das Mapping mithilfe der Funktion **ApplyMap** beim Laden der Tabelle mit den Bestellungen übernommen:

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

Anhand des dritten Parameters der Funktion **ApplyMap** wird der Wert festgelegt, der ausgegeben wird, wenn der Wert in der Mapping-Tabelle nicht gefunden wird, wie in diesem Fall **Null()**.

Die sich ergebende Tabelle sieht folgendermaßen aus:

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

6.12 Arbeiten mit Kreuztabellen

Als Kreuztabelle wird ein oft verwendeter Tabellentyp bezeichnet, bei dem eine Matrix von Werten zwischen zwei senkrecht aufeinander stehenden Wertelisten steht.

Pivotierung einer Kreuztabelle mit einer Spalte aufheben

Manchmal stehen vor der Kreuztabelle noch mehrere Spalten, die als normale Felder zu betrachten sind. In diesem Fall gibt es eine Spalte, Year, und eine Umsatzdatenmatrix pro Monat.

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

Wird diese Tabelle auf die übliche Weise in Qlik Sense eingelesen, ergeben sich ein Feld mit der Bezeichnung *Year* und je ein Feld für jeden der aufgeführten Monate. Dies ist allerdings ungünstig. Sinnvoller ist es, wenn drei Felder entstehen:

- In diesem Fall ist die Spalte *Year* in der vorstehenden Tabelle grün markiert.
- Das Attribut-Feld – in diesem Fall die Monatsnamen Jan - Jun – ist gelb markiert. Dieses Feld kann entsprechend *Month* benannt werden.
- Die Datenmatrixwerte sind blau gekennzeichnet. In diesem Fall stehen sie für die Umsatzzahlen, die entsprechend *Sales* benannt werden können.

Dies erreichen Sie, indem Sie den Zusatz **crosstable** vor dem entsprechenden Befehl **LOAD** oder **SELECT** einfügen.

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

Damit wird die folgende Tabelle in Qlik Sense erzeugt:

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

Pivotierung einer Kreuztabelle mit zwei Spalten aufheben

Hier befinden sich zwei Spalten vor der Wertematrix.

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

Die Zahl dieser Spalten wird als dritter Parameter zum Zusatz **crosstable** eingefügt:

```
crosstable (Month, sales, 2) LOAD * from ex2.xlsx;
Qlik Sense zeigt dann folgendes Ergebnis an:
```

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

6.13 Generische Datenbanken

Eine generische Datenbank ist eine Tabelle, in der die Feldnamen in einer Spalte und die zugehörigen Werte in einer weiteren Spalte stehen. Generische Datenbanken werden häufig für Attribute verschiedener Objekte verwendet.

Bitte betrachten Sie das Beispiel GenericTable unten. Dies ist eine generische Datenbank, in der zwei Objekte beschrieben werden: ein Ball und eine Kiste. Einige Attribute, wie Farbe oder Gewicht, sind für beide Objekte angegeben. Andere Eigenschaften, wie Durchmesser, Höhe, Länge oder Breite, sind nur für ein Objekt verfügbar.

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

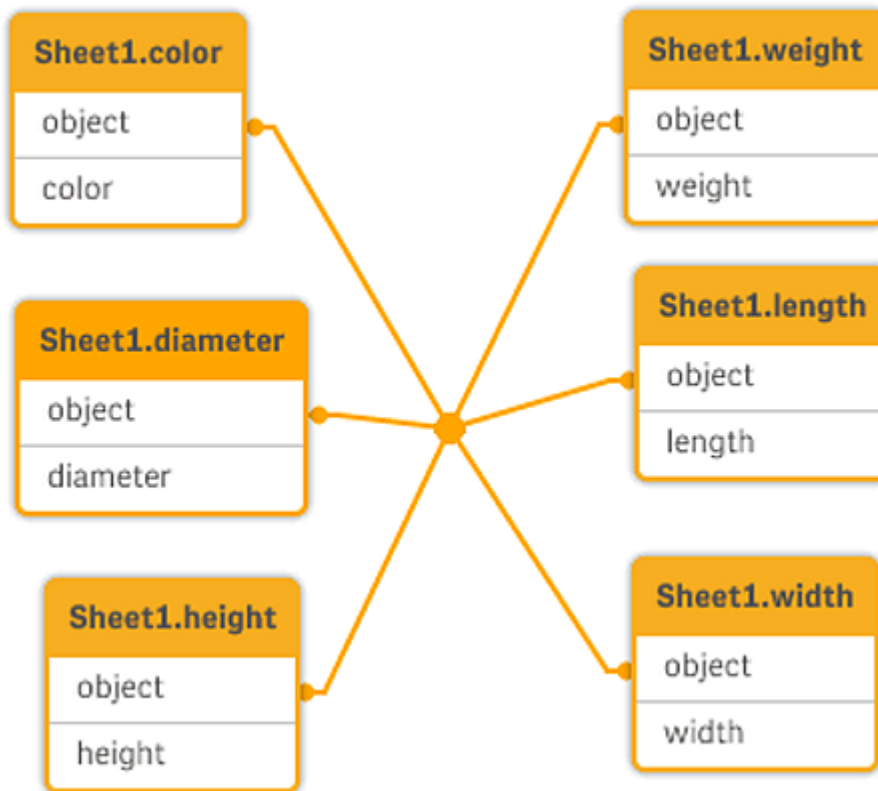
Einerseits wäre es ungünstig, für jedes Attribut eine eigene Spalte anzulegen, weil einige Eigenschaften für bestimmte Objekte nicht von Bedeutung sind.

Aus Gründen der Übersicht sollten Längen, Farben und Gewichte in der Anzeige aber nicht vermischt werden.

Wenn die Datenbank mithilfe der Standardmethode in Qlik Sense geladen wird und die Daten in einer Tabelle dargestellt werden, sieht dies folgendermaßen aus:

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

Wird die Tabelle jedoch als generische Datenbank geladen, werden die Informationen der zweiten und dritten Spalte auf mehrere Tabellen aufgeteilt, eine für jeden Wert der zweiten Spalte:



Die benötigte Syntax ist einfach:

Beispiel:

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

Generische Datenbanken können mit dem Befehl **LOAD** oder **SELECT** geladen werden.

6.14 Einordnung diskreter Werte in Intervalle

Der Zusatz **intervalmatch** vor den Befehlen **LOAD** oder **SELECT** dient dazu, diskrete numerische Werte in numerische Intervalle einzuordnen. Nachfolgend wird diese Funktion am Beispiel eines Produktionsprozesses erläutert.

Intervalmatch-Beispiel

Bitte betrachten Sie die zwei nachstehenden Tabellen. Die erste Tabelle zeigt Bearbeitungsstart und -ende verschiedener Aufträge. Die zweite Tabelle enthält punktuelle Ereignisse. Wie lässt sich nun ein Zusammenhang zwischen den Tabellen herstellen, in dem Sinn, dass erkennbar wird, welche Aufträge von welchen Ereignissen betroffen waren?

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

Zunächst werden die Tabellen eingelesen, dann werden die Werte des Feldes *Time* den durch *Start* und *End* definierten Zeitintervallen zugeordnet:

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

Sie können jetzt wie folgt eine Tabelle in Qlik Sense erstellen:

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:...

Jetzt ist offensichtlich, dass hauptsächlich Auftrag *A* vom Stopp der Produktionslinie betroffen war. Auftrag *B* und *C* wurden ebenfalls durch die verminderte Geschwindigkeit der Produktionslinie beeinträchtigt. Nur die Aufträge *C* und *D* wurden teilweise von *Shift 2* bearbeitet.

Beim Gebrauch von **intervalmatch** beachten Sie Folgendes:

- Vor dem Befehl **intervalmatch** muss das Feld mit den diskreten Werten (*Time* in unserem Beispiel) bereits in Qlik Sense eingelesen worden sein. Das Feld wird nicht durch **intervalmatch** eingelesen!

- Die Tabelle, die durch **intervalmatch LOAD** oder **SELECT** geladen wird, muss genau zwei Felder enthalten (*Start* und *End* in unserem Beispiel). Um Verknüpfungen mit anderen Feldern zu ermöglichen, muss das Feld mit den diskreten Werten zusammen mit anderen Feldern in einem separaten Befehl **LOAD** oder **SELECT** geladen werden (der erste Befehl **SELECT** in unserem Beispiel).
- Die Intervalle sind immer abgeschlossen. Das heißt, dass die Grenzwerte in den Intervallen enthalten sind. Nicht-numerische Werte als Intervallgrenzen führen dazu, dass das Intervall nicht berücksichtigt wird (undefiniert). Bei NULL-Werten als Intervallgrenzen ist das Intervall unbegrenzt (unendlich groß).
- Wenn sich die Intervalle überschneiden, werden die Werte jedem passenden Intervall zugeordnet.

Behebung von Problemen mit sich langsam verändernden Dimensionen mit der erweiterten Syntax von **intervalmatch**

Die Syntax des erweiterten Befehls **intervalmatch** dient dazu, ein häufig vorkommendes Problem zu lösen: sich langsam verändernde Dimensionen in den Datenquellen.

Beispiel:

```
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 wird nur benötigt, wenn die Daten aus einer externen Textdatei eingelesen werden, denn dort sind fehlende Werte als leere Strings statt als NULL-Werte definiert.

Das Einlesen von *IntervalTable* liefert folgende Tabelle:

6 Beste Praktiken für Datenmodelle

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

Der Befehl **nullasvalue** erlaubt das Mapping von NULL-Werten auf die gelisteten Felder.

Nun werden die Attributfelder *Key*, *FirstDate*, *LastDate* generiert, und zwar mithilfe der Befehle **previous** und **order by**. Danach wird die Tabelle *IntervalTable* gelöscht. Sie wird durch diese Schlüsseltable ersetzt.

Das Einlesen von *Transact* liefert folgende Tabelle:

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

Durch den Befehl **intervalmatch** mit dem vorausgehenden Zusatz **inner join** wird der Schlüssel durch einen neuen, synthetisch generierten Schlüssel ersetzt, der sich mit der Tabelle *Transact* verknüpft. Damit ergibt sich folgende Tabelle:

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

6.15 Erstellen eines Datumsintervalls aus einem einzigen Datum

Manchmal werden Zeitintervalle nicht explizit mit einem Start- und Endwert gespeichert. Stattdessen werden sie nur durch ein Feld impliziert – den Zeitstempel der letzten Änderung.

Ein Beispiel hierfür wäre die untere Tabelle, in der Währungskurse für mehrere Währungen angegeben werden. Jede Währungskursänderung befindet sich in einer eigenen Zeile; in jeder Zeile ist ein neuer Umrechnungskurs angegeben. Außerdem enthält die Tabelle Zeilen mit leerem Datum, die dem ursprünglichen Umrechnungskurs entsprechen, bevor die erste Änderung vorgenommen wurde.

Currency	Change Date	Rate
EUR		8.59
EUR	28/01/2013	8.69
EUR	15/02/2013	8.45
USD		6.50
USD	10/01/2013	6.56
USD	03/02/2013	6.30

Diese Tabelle legt eine Gruppe nicht überlappender Intervalle fest. Das Startdatum ist dabei "Change Date" und das Enddatum wird durch das darauf folgende Intervall festgelegt. Da das Enddatum nicht explizit in einer gesonderten Spalte gespeichert ist, muss diese Spalte erstellt werden, damit in der neuen Tabelle eine Liste mit Intervallen angegeben werden kann.

In diesem Skript wird die Tabelle In_Rates per Inline-Load erzeugt. Achten Sie darauf, dass die Daten in der Spalte Change Date dasselbe Format wie das lokale Datumsformat aufweisen.

```
In_Rates:
LOAD * Inline [
Currency,Change Date,Rate
EUR,,8.59
EUR,28/01/2013,8.69
EUR,15/02/2013,8.45
USD,,6.50
USD,10/01/2013,6.56
USD,03/02/2013,6.30
];
```

Gehen Sie folgendermaßen vor:

- Bestimmen Sie den Datumsbereich, in dem Sie arbeiten möchten. Der Start des Bereichs muss vor dem ersten Datum liegen, das in den Daten enthalten ist, und das Ende des Bereichs nach dem letzten Datum in den Daten liegen.

```
Let vBeginTime = Num('1/1/2013');
Let vEndTime = Num('1/3/2013');
Let vEpsilon = Pow(2,-27);
```
- Laden Sie die Quelldaten, aber ändern Sie leere Daten wie im vorherigen Stichpunkt beschrieben zum Beginn des Bereichs. Das Änderungsdatum sollte als "From Date" geladen werden.

6 Beste Praktiken für Datenmodelle

Sortieren Sie die Tabelle zunächst nach Currency und anschließend nach "From Date" in absteigender Reihenfolge, sodass die neuesten Daten oben angezeigt werden.

Tmp_Rates:

```
LOAD Currency, Rate,
      Date(If(IsNum([Change Date]), [Change Date], $(#vBeginTime))) as FromDate
Resident In_Rates;
```

- Führen Sie einen zweiten Lauf über die Daten aus, in dem der Wert "To Date" berechnet wird. Wenn sich die Währung des aktuellen Datensatzes von der des vorherigen unterscheidet, handelt es sich dabei um den ersten Datensatz einer neuen Währung (aber zugleich um das letzte Intervall). Deshalb sollten Sie das in Schritt 1 definierte Bereichsende verwenden. Handelt es sich dabei um dieselbe Währung, sollten Sie das "From Date" des vorhergehenden Datensatzes verwenden, eine geringe Zeitspanne abziehen und diesen Wert als "To Date" für den aktuellen Datensatz verwenden.

Rates:

```
LOAD Currency, Rate, FromDate,
      Date(If( Currency=Peek(Currency),
              Peek(FromDate) - $(#vEpsilon),
              $(#vEndTime)
            )) as ToDate
Resident Tmp_Rates
Order By Currency, FromDate Desc;
```

- Legen Sie die Tabelle und die temporäre Tabelle ab.
Drop Table Tmp_Rates;

Das nachfolgend aufgeführte Skript aktualisiert die Quelltable folgendermaßen:

Currency	Rate	FromDate	ToDate
EUR	8.45	15/02/2013	vEndTime
EUR	8.69	28/01/2013	14/02/2013 23:59:59
EUR	8.59	vBeginTime	28/01/2013 23:59.59
USD	6.30	03/02/2013	vEndTime
USD	6.56	10/01/2013	2/02/2013 23:59:59
USD	6.50	vBeginTime	9/01/2013 23:59.59

Bei der Ausführung des Skripts ergibt sich eine Tabelle, die die Intervalle richtig auflistet. Verwenden Sie zum Anzeigen der resultierenden Tabelle den **Vorschau**-Abschnitt des Datenmodells.

Preview of data

Currency	Rate	FromDate	ToDate
EUR	8.45	15/02/2013	01/03/2013
EUR	8.69	28/01/2013	14/02/2013
EUR	8.59	01/01/2013	27/01/2013
USD	6.30	03/02/2013	01/03/2013
USD	6.56	10/01/2013	02/02/2013
USD	6.50	01/01/2013	09/01/2013

Die Tabelle lässt sich anschließend mithilfe der **Intervalmatch**-Methoden zum Vergleich mit einem vorhandenen Datum einsetzen.

Beispiel:

Das vollständige Qlik Sense-Skript sieht folgendermaßen aus:

```
Let vBeginTime = Num('1/1/2013');
Let vEndTime = Num('1/3/2013');
Let vEpsilon = Pow(2,-27);

In_Rates:
LOAD * Inline [
Currency,Change Date,Rate
EUR,,8.59
EUR,28/01/2013,8.69
EUR,15/02/2013,8.45
USD,,6.50
USD,10/01/2013,6.56
USD,03/02/2013,6.30
];

Tmp_Rates:
LOAD Currency, Rate,
      Date(If(IsNum([Change Date]), [Change Date], $(#vBeginTime))) as FromDate
Resident In_Rates;

Rates:
LOAD Currency, Rate, FromDate,
      Date(If( Currency=Peek(Currency),
              Peek(FromDate) - $(#vEpsilon),
              $(#vEndTime)
            )) as ToDate
Resident Tmp_Rates
Order By Currency, FromDate Desc;

Drop Table Tmp_Rates;
```

6.16 Hierarchiedaten laden

Unregelmäßige Hierarchien mit n Ebenen gibt es u. a. häufig bei Daten, die eine geografische Ordnung oder die Organisationsstruktur eines Unternehmens repräsentieren. Solche Hierarchien werden zumeist in Form von Tabellen dargestellt, in der benachbarte Knoten einander zugeordnet werden. Jede Zeile der Tabelle repräsentiert einen Knoten und enthält eine Referenz zum direkt übergeordneten Knoten.

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

Jeder Knoten wird in dieser Tabelle durch eine Zeile repräsentiert, auch wenn der Knoten mehrere untergeordnete Knoten hat. Natürlich kann die Tabelle auch weitere Felder enthalten, etwa mit Attributen.

Eine Tabelle mit benachbarten Knoten ist optimal für die Wartung, aber schwierig für die tägliche Arbeit. Stattdessen werden in Abfragen und Analysen andere Repräsentationen verwendet. Im Allgemeinen werden dazu die Ebenen der Hierarchie so aufgeschlüsselt, dass für jede Ebene ein separates Feld entsteht. Diese Struktur lässt sich gut in einem Baum darstellen. Um die Ebenen einer Hierarchie auf diese Weise aufzuschlüsseln, kann der **hierarchy**-Zusatz im Datenladeskript verwendet werden.

Beispiel:

```
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

Ein Problem der erweiterten Knoten ist jedoch, dass die Felder für die Ebenen sich nicht besonders gut für Suche und Auswahl eignen, da bereits im Vorfeld bekannt sein muss, auf welcher Ebene das betreffende Element zu finden ist. Eine Vorfahrtabelle ist eine andere Darstellung, mit der sich dieses Problem lösen lässt. Dieses Problem wird mit einer dritten Darstellung gelöst, der Vorfahrtabelle, auch Brückentabelle genannt.

Eine Vorfahrtabelle enthält einen Datensatz für jede untergeordnete Vorfahrenbeziehung, die in den Daten vorhanden ist. Sie enthält Schlüssel und Namen für die untergeordneten und auch die übergeordneten Knoten. Das heißt, dass jeder Datensatz beschreibt, zu welchem Knoten ein bestimmter Knoten gehört. Um eine solche Darstellung zu erreichen, dient der Zusatz **hierarchybelongsto** im Datenladeskript.

6.17 Laden von Kartendaten

Zum Erstellen einer Kartenvisualisierung müssen Sie Punktdaten (Excel- oder KML-Datei) oder Bereichsdaten (KML-Datei) laden. Standardmäßig sind im Datenauswahl-Dialog alle Felder ausgewählt – auch, wenn sie keine Daten enthalten. Eine KML-Datei kann beispielsweise Bereichsdaten, aber keine Punktdaten enthalten. Es ist zwar in Ordnung, leere Felder zu laden. Beim Arbeiten mit der Kartenvisualisierung können leere Felder jedoch unpraktisch sein. Das Hinzufügen einer Dimension ohne Daten liefert kein visuelles Ergebnis und könnte Benutzer verwirren. Daher sollten Sie die Auswahlen aller leeren Datenfelder im Datenauswahl-Dialog deaktivieren, bevor Sie die Daten laden.



Die maximale Anzahl von Werten in einer Punktebenenkarte beträgt 3333.

Erstellen einer Karte aus Daten in einer KML-Datei

Normalerweise enthält eine KML-Datei Punktdaten, Bereichsdaten oder beides. Im folgenden Screenshot des Datenauswahl-Dialogs können Sie erkennen, dass die Auswahl des Felds *FloridaCounties.Point* deaktiviert wurde, weil das Feld keine Daten enthält. Durch das Deaktivieren des Felds laufen Sie nicht Gefahr, Kartendimensionen ohne Daten zu erstellen.

Daten aus FloridaCounties.kml auswählen Skriptvorschau anzeigen

Tabellen Dateiformat **KML**

Florida Counties K... 2 Felder filtern

<input checked="" type="checkbox"/> FloridaCounties.Name	<input type="checkbox"/> FloridaCounties.Point	<input checked="" type="checkbox"/> FloridaCounties.Area
Citrus		[[[-82.426628,28.694908],[-82.41
Brevard		[[[-80.493599,28.413055],[-80.48
Franklin		[[[-85.021421,29.823213],[-85.02
Broward		[[[-80.29704,26.334356],[-80.297
Hamilton		[[[-83.055898,30.619483],[-83.05
Jackson		[[[-85.432857,30.703505],[-85.43
Okeechobee		[[[-80.982471,27.238772],[-80.98
Monroe		[[[-80.894429,24.852241],[-80.89
Duval		[[[-81.316712,30.401676],[-81.31
Manatee		[[[-82.448493,27.388604],[-82.44

```

LOAD
  FloridaCounties.Name,
  FloridaCounties.Area
FROM [lib://Tutorial source/FloridaCounties.kml]
(kml, Table is [Florida Counties KML]);

```

Abbrechen Skript einfügen

Wenn Sie auf **Skript einfügen** klicken, wird das folgende Skript generiert:

```

LOAD
  FloridaCounties.Name,
  FloridaCounties.Area
FROM 'lib://data 7/FloridaCounties.kml'
(kml, Table is [Florida Counties KML]);

```

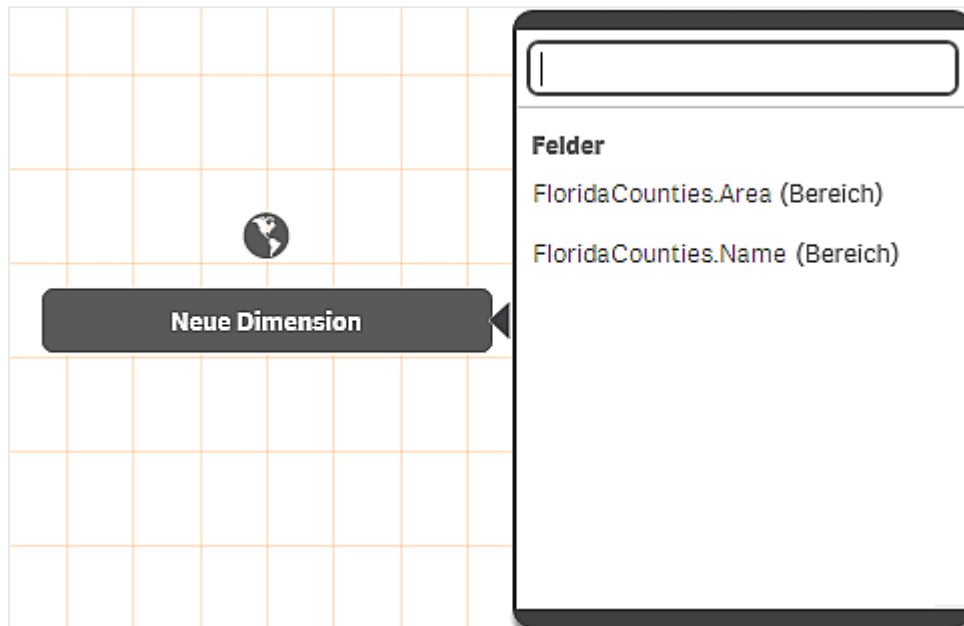
Wenn die Auswahl des leeren Felds nicht gelöscht worden wäre, hätte das Skript auch den folgenden String enthalten:

```

  FloridaCounties.Point,

```

Wenn Sie das Skript ausführen und dem Arbeitsblatt ein Kartendiagramm hinzufügen, können Sie entweder *FloridaCounties.Area (area)* oder *FloridaCounties.Name (area)* als Dimension hinzufügen.



Zwar erzeugen beide Felder dasselbe visuelle Ergebnis (eine Karte mit den Countys). Es gibt jedoch einen Unterschied, wenn Sie den Mauszeiger über ein County bewegen (bzw. bei einem Touchgerät tippen). Wenn Sie *FloridaCounties.Name (area)* auswählen, wird der Name des Countys als Popup-Hilfe angezeigt; wenn Sie *FloridaCounties.Area (area)* auswählen, werden die Bereichsdaten angezeigt. Der Name des Countys ist zweifellos interessanter. Außerdem dauert das Laden des Felds *FloridaCounties.Area* viel länger als beim Feld *FloridaCounties.Name*.

Optional können Sie auch eine Kennzahl hinzufügen und die Farbgebung nach Kennzahl verwenden, um die Differenzen der Kennzahlwerte zwischen den verschiedenen Countys widerzuspiegeln.



Wenn die KML-Datei weder Punkt- noch Bereichsdaten enthält, können Sie aus der Datei keine Daten laden. Wenn die KML-Datei beschädigt ist, wird eine Fehlermeldung angezeigt und Sie können die Daten nicht laden.

Erstellen einer Karte aus Punktdaten in einer Excel-Datei

Sie können eine Karte anhand von Punktdaten (Koordinaten) aus einer Excel-Datei erstellen.

Punktdatenformate

Punktdaten können aus Excel-Dateien gelesen werden. Zwei Formate werden unterstützt:

- Die Punktdaten werden in einer Spalte gespeichert. Jeder Punkt ist als Anordnung von x - und y -Koordinaten definiert: $[x, y]$. Bei geografischen Koordinaten entspricht dies $[Längengrad, Breitengrad]$. Wenn Sie dieses Format verwenden, müssen Sie das Punktdatenfeld mit `$geopoint`; kennzeichnen.

- Die Punktdaten werden in zwei Spalten gespeichert: eine für den Breitengrad und eine für den Längengrad. Die Funktion `geomakepoint()` erzeugt einen Punkt basierend auf diesem Format.

In den folgenden Beispielen nehmen wir an, dass die Dateien dieselben Daten zu den Standorten der Niederlassungen eines Unternehmens enthalten, jedoch in zwei unterschiedlichen Formaten.

Beispiel 1:

Die Excel-Datei enthält für jede Niederlassung Folgendes:

- Niederlassung (`office`)
- Adresse (`Location`)
- Anzahl der Mitarbeiter (`Employees`)

Das Ladeskript sieht beispielsweise folgendermaßen aus:

```
LOAD
    office,
    Location,
    Employees
FROM 'lib://Maps/offices.xls'
(biff, embedded labels, table is (Sheet1$));
```

Das Feld `Location` enthält die Punktdaten und muss mit `$geopoint;` gekennzeichnet werden, damit es als Punktdatenfeld erkannt wird. Fügen Sie nach dem letzten String im **LOAD**-Befehl folgenden String hinzu:

```
TAG FIELDS Location WITH $geopoint;
```

Das vollständige Skript sieht dann folgendermaßen aus:

```
LOAD
    office,
    Location,
    Employees
FROM 'lib://Maps/offices.xls'
(biff, embedded labels, table is (Sheet1$));
```

```
TAG FIELDS Location WITH $geopoint;
```

Führen Sie das Skript aus und erstellen Sie eine Kartenvisualisierung. Fügen Sie die Punktdimension zu Ihrer Karte hinzu.

Beispiel 2:

Die Excel-Datei enthält für jede Niederlassung Folgendes:

- Niederlassung (`office`)
- Breitengrad (`Latitude`)

- Längengrad (Longitude)
- Anzahl der Mitarbeiter (Employees)

Das Ladeskript sieht beispielsweise folgendermaßen aus:

```
LOAD
  Office,
  Latitude,
  Longitude,
  Employees
FROM 'lib://Maps/Offices.xls'
(biff, embedded labels, table is (Sheet1$));
```

Kombinieren Sie die Daten in den Feldern Latitude und Longitude, um ein neues Feld für die Punkte zu definieren.

Führen Sie das Skript aus und erstellen Sie eine Kartenvisualisierung. Fügen Sie die Punktdimension zu Ihrer Karte hinzu.

Sie können die Dimension Location im Skript erstellen, indem Sie über dem **LOAD**-Befehl folgenden String hinzufügen:

```
LOAD *, GeoMakePoint(Latitude, Longitude) as Location;
```

Die Funktion GeoMakePoint() verknüpft die Längen- und Breitengraddaten.

Das vollständige Skript sieht dann folgendermaßen aus:

```
LOAD *, GeoMakePoint(Latitude, Longitude) as Location;
LOAD
  Office,
  Latitude,
  Longitude,
  Employees
FROM 'lib://Maps/Offices.xls'
(biff, embedded labels, table is (Sheet1$));
```

Führen Sie das Skript aus und erstellen Sie eine Kartenvisualisierung. Fügen Sie die Punktdimension zu Ihrer Karte hinzu.

Anzahl der dargestellten Punkte

Aus Performancegründen ist die Zahl der Punkte, die angezeigt werden können, begrenzt. Durch das Vornehmen von Auswahlen können Sie die angezeigte Datenmenge reduzieren.

6.18 Datenpflege durch Mapping

In der Praxis werden Sie merken, dass Werte aus mehreren Tabellen oft unterschiedliche Bezeichnungen oder Schreibweisen haben, obwohl der Informationsgehalt derselbe ist. Das ist nicht nur unübersichtlich,

sondern auch deshalb ungünstig, weil in der Software inhaltlich korrekte Verknüpfungen nicht zustande kommen, wenn die Schreibweise nicht exakt übereinstimmt. Dieses Problem lässt sich durch Mapping beheben.

Mapping-Tabellen

Bei Tabellen, die durch **mapping load** oder **mapping select** geladen werden, erfolgt die Speicherung in einem separaten Bereich. Sie werden lediglich während der Ausführung des Skripts für das Mapping benötigt und separat gespeichert. Nach Ausführung des Skripts werden sie automatisch gelöscht.

Es gilt:

- Mapping-Tabellen bestehen aus zwei Spalten: Die erste enthält Vergleichswerte, die zweite die gewünschten Mapping-Werte.
- Die zwei Spalten müssen benannt werden, doch die Namen haben keine Relevanz. Die Spaltennamen müssen keine Verbindung zu Feldnamen in normalen internen Tabellen haben.

Mit Mapping-Tabellen arbeiten

Beim Laden von länderspezifischen Daten kommt es vor, dass verschiedene Schreibweisen für ein und dasselbe Land verwendet werden. In diesem Beispiel werden die U.S.A. auch als US, U.S. und United States bezeichnet.

Um eine einheitliche Schreibweise und die passenden Verknüpfungen herzustellen, nutzen Sie eine Mapping-Tabelle wie in der Abbildung.

Das Skript sieht nun folgendermaßen aus:

```
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);
```

Der Befehl **mapping** lädt die Datei *MappingTable.txt* als Mapping-Tabelle mit der Bezeichnung *CountryMap*.

Der Befehl **map** aktiviert das Mapping für das Feld *Country* anhand der bereits geladenen Mapping-Tabelle *CountryMap*.

Der Befehl **LOAD** lädt die Tabellen *CountryA* und *CountryB*. Beim Ladevorgang werden die beiden Tabellen automatisch zu einer zusammengefasst, da sie exakt dieselben Feldnamen enthalten. Dann werden die Werte des Feldes *Country* mit den Vergleichswerten in der ersten Spalte der Mapping-Tabelle verglichen. Die Werte US, U.S. und United States werden durch die in der zweiten Spalte der Tabelle angegebenen Werte ersetzt, etwa durch den Wert *USA*.

Das Mapping ist stets der letzte Schritt des Ladevorgangs, bevor das Feld in der Qlik Sense-Tabelle gespeichert wird. Bei einem typischen **LOAD**- oder **SELECT**-Befehl werden die Arbeitsschritte in folgender Reihenfolge ausgeführt:

1. Auswertung von Formeln
2. Umbenennen von Feldern mit as
3. Umbenennen von Feldern mit alias
4. Qualifizierung von Feldern
5. Daten-Mapping bei Übereinstimmung des Feldnamens

Das bedeutet, das Mapping geschieht nicht immer dann, wenn ein Feldname als Teil einer Formel auftaucht, sondern erst direkt vor dem Speichern des Werts in einer Qlik Sense-Tabelle.

Sollen nachfolgend geladene Werte nicht mehr gemappt werden, verwenden Sie den Befehl **unmap**. Er beendet das Mapping.

Mapping in Formeln erreichen Sie durch die Funktion **applymap**.

Um Teile von Werten zu mappen, benutzen Sie die Funktion **mapsubstring**.

7 Fehlerbehebung – Laden von Daten

In diesem Abschnitt werden Probleme beschrieben, die beim Laden und Modellieren von Daten in Qlik Sense auftreten können.

7.1 Datenverbindung funktioniert nicht mehr, nachdem SQL Server neu gestartet wird

Möglicher Grund

Wenn Sie eine Datenverbindung zu einem SQL Server herstellen und dann den SQL Server neu starten, funktioniert die Datenverbindung möglicherweise nicht mehr und Sie können keine Daten auswählen. Die Verbindung von Qlik Sense zum SQL Server wurde getrennt und konnte nicht wiederhergestellt werden.

Empfohlene Vorgehensweise

Qlik Sense:

Gehen Sie folgendermaßen vor:

- Schließen Sie die App und öffnen Sie sie erneut über den Hub.

Qlik Sense Desktop:

Gehen Sie folgendermaßen vor:

1. Schließen Sie alle Apps.
2. Starten Sie Qlik Sense Desktop neu

7.2 Warnung zu synthetischen Schlüsseln beim Laden von Daten

Wenn Sie mehrere Dateien geladen haben, erhalten Sie möglicherweise eine Warnung, dass nach dem Laden der Daten synthetische Schlüssel erstellt wurden.

Möglicher Grund

Wenn zwei Tabellen mehr als ein gemeinsames Feld enthalten, erstellt Qlik Sense einen synthetischen Schlüssel, um die Verlinkung aufzulösen.

Empfohlene Vorgehensweise

Ist die Verlinkung aussagekräftig, kann sie in vielen Fällen so verbleiben. Es ist jedoch ratsam, die Datenstruktur im Datenmodell zu überprüfen.

7.3 Warnung zu Zirkelbezügen beim Laden von Daten

Möglicher Grund

Wenn Sie mehr als zwei Tabellen geladen haben, können die Tabellen so verknüpft sein, dass mehr als ein Pfad von Verknüpfungen zwischen zwei Feldern besteht. Dann entsteht eine Schleife in der Datenstruktur.

Empfohlene Vorgehensweise

7.4 Datenauswahlprobleme mit einer OLE DB-Datenquelle

Möglicher Grund

Wenn Sie keine Daten von einer OLE DB-Datenverbindung auswählen können, prüfen Sie die Konfiguration der Verbindung.

Empfohlene Vorgehensweise

Gehen Sie folgendermaßen vor:

1. Vergewissern Sie sich, dass der Verbindungsstring korrekt ist.
2. Prüfen Sie, ob Sie ordnungsgemäße Login-Daten verwenden.

7.5 Zeichensatzprobleme mit Nicht-ANSI-codierten Datendateien

Möglicherweise treten in nicht-ANSI-codierten Datendateien beim Verwenden einer ODBC-Datenverbindung Probleme mit der Zeichencodierung auf.

Möglicher Grund

ODBC-Datenverbindungen bieten nicht volle Funktionalität für die Zeichensatzcodierung.

Empfohlene Vorgehensweise

Gehen Sie folgendermaßen vor:

- Importieren Sie die Datendateien, falls möglich, über eine Ordnerdatenverbindung, die weitere Optionen zur Verarbeitung von Zeichencodes unterstützt. Dies ist wahrscheinlich die beste Option, wenn Sie eine Microsoft Excel-Tabelle oder eine Textdatendatei laden.

7.6 Benutzerdefinierter Konnektor funktioniert nicht

Sie versuchen, im Dateneditor eine Datenverbindung zu einem benutzerdefinierten Konnektor eines Drittanbieters herzustellen, aber die Verbindung schlägt fehl oder eine vorhandene Verbindung wird als "unbekannt" gekennzeichnet.

Der benutzerdefinierte Konnektor ist nicht ordnungsgemäß installiert

Möglicher Grund

Der benutzerdefinierte Konnektor ist nicht gemäß den Installationsanweisungen installiert. Wenn eine App einen benutzerdefinierten Konnektor auf einer Site mit mehreren Knoten verwendet, muss der Konnektor auf allen Knoten installiert sein.

Empfohlene Vorgehensweise

Gehen Sie folgendermaßen vor:

- Vergewissern Sie sich, dass der Konnektor gemäß den Anweisungen auf allen Knoten der Site installiert ist.

Der benutzerdefinierte Konnektor ist nicht für Qlik Sense angepasst

Möglicher Grund

QlikView-Konnektoren müssen für Qlik Sense angepasst werden, um Daten auswählen zu können.

Empfohlene Vorgehensweise (wenn Sie den Konnektor mit dem QVX SDK selbst entwickelt haben)

Gehen Sie folgendermaßen vor:

- Sie müssen den Konnektor für Qlik Sense mit einer Schnittstelle zur Auswahl von Daten anpassen.

Empfohlene Vorgehensweise (wenn Ihnen der Konnektor bereitgestellt wurde)

Gehen Sie folgendermaßen vor:

- Wenden Sie sich an den Bereitsteller des Konnektors, um einen für Qlik Sense angepassten Konnektor zu erhalten.

7.7 Das Datenladeskript wird ohne Fehler ausgeführt, die Daten werden aber nicht geladen

Das Skript wird ohne Syntax- oder Ladefehler ausgeführt, aber die Daten werden nicht erwartungsgemäß geladen. Eine allgemeine Empfehlung ist, das Debuggen zu aktivieren, um das Skript schrittweise durchzugehen und die Ausführungsergebnisse zu prüfen. Hier nun aber auch noch einige häufige Fehlerursachen.

Ein Befehl ist nicht mit einem Semikolon abgeschlossen

Möglicher Grund

Sie haben vergessen, einen Befehl am Ende mit einem Semikolon abzuschließen.

Empfohlene Vorgehensweise

Gehen Sie folgendermaßen vor:

- Schließen Sie alle Befehle mit einem Semikolon ab.

Einfaches Anführungszeichen innerhalb eines Strings

Möglicher Grund

Ein String enthält ein einfaches Anführungszeichen, z. B. in einem SET-Variablen-Befehl.

Empfohlene Vorgehensweise

Gehen Sie folgendermaßen vor:

- Wenn ein String ein einfaches Anführungszeichen enthält, muss ein zusätzliches einfaches Anführungszeichen als Escapezeichen ergänzt werden.

7.8 Spalten werden bei Auswahl von Daten aus einer Datei mit festen Satzlängen nicht wie erwartet angeordnet

Möglicher Grund

Die Datei verwendet Tabulatorzeichen zum Auffüllen der Spalten. Üblicherweise richten sich die Feldüberschriften nicht an den erwarteten Daten aus, wenn Sie im Auswahldialog **Feldumbrüche** auswählen.

In diesem Fall ist das Tabulatorzeichen normalerweise mit einer Anzahl von Zeichen gleichgesetzt.

Empfohlene Vorgehensweise

Gehen Sie folgendermaßen vor:

1. Wählen Sie in **Feldnamen** die Option **Keine Feldnamen** aus.
2. Wählen Sie **Feldumbrüche** aus.
3. Erhöhen Sie die Einstellung für **Tab-Länge**, bis die Spalten auf die Überschriften ausgerichtet sind.
4. Fügen Sie Feldumbrüche ein, indem Sie auf die entsprechenden Spaltenpositionen klicken.

5. Wählen Sie **Datenvorschau** aus.
6. Wählen Sie in **Feldnamen** die Option **Eingebettete Feldnamen** aus.

Die Spalten sind jetzt ordnungsgemäß angeordnet und jedes Feld sollte den richtigen Feldnamen aufweisen.

7.9 Fehlermeldung "**Ungültiger Pfad**" beim Anhängen einer Datei

Möglicher Grund

Der Dateiname ist zu lang. Qlik Sense unterstützt nur Dateinamen bis maximal 171 Zeichen.

Empfohlene Vorgehensweise

Benennen Sie die Datei so um, dass der Name aus weniger als 172 Zeichen besteht.

7.10 Fehler beim Laden einer App, die aus einem QlikView-Dokument konvertiert wurde

Wenn Sie eine App neu laden, die aus einem QlikView-Dokument konvertiert worden ist, wird wegen der Unterschiede zwischen den beiden Produkten möglicherweise eine Fehlermeldung ausgegeben.

Im Skript werden absolute Dateipfadverweise verwendet

Möglicher Grund

Das Ladeskript verweist auf Dateien, die absolute Pfade verwenden, die jedoch im standardmäßigen Qlik Sense-Modus nicht unterstützt werden. Beispiele für Fehlermeldungen sind "Invalid Path" und "LOAD statement only works with lib:// paths in this script mode".

Empfohlene Vorgehensweise

Gehen Sie folgendermaßen vor:

- Ersetzen Sie alle Dateiverweise mit **lib://**-Verweisen zu Datenverbindungen in Qlik Sense.

Im Skript werden nicht unterstützte Funktionen oder Befehle verwendet

Möglicher Grund

Wenn beim Ausführen des Skripts im Dateneditor ein Syntaxfehler auftritt, so kann dieser mit der Verwendung nicht unterstützter QlikView-Skriptbefehle oder Funktionen in Qlik Sense zusammenhängen.

Empfohlene Vorgehensweise

Gehen Sie folgendermaßen vor:

- Entfernen Sie den unzulässigen Befehl oder ersetzen Sie ihn durch einen zulässigen.

7.11 Probleme bei der Verbindungsherstellung mit Microsoft Excel-Dateien über ODBC oder dem Laden solcher Dateien.

Möglicher Grund

Bei der Einrichtung einer ODBC-Datenverbindung mit einer Microsoft Excel-Datei oder dem Laden von Daten aus einer Microsoft Excel-Datei über eine ODBC-Datenverbindung treten möglicherweise Probleme auf. Das ist im Allgemeinen auf Probleme mit der ODBCDSN-Konfiguration in Windows oder auf Probleme mit den ODBC-Treibern zurückzuführen.

Empfohlene Vorgehensweise

Qlik Sense hat eine eigene Unterstützung für das Laden von Microsoft Excel-Dateien. Tauschen Sie, falls möglich, die ODBC-Datenverbindung gegen eine Ordnerdatenverbindung aus, die den Ordner verbindet, der die Microsoft Excel-Dateien enthält.

7.12 Das Anhängen einer Datei durch Ablegen in **Daten hinzufügen** funktioniert nicht

Sie versuchen eine Daten anzuhängen, indem Sie diese aus **Windows Explorer** ziehen und im Dialog **Daten hinzufügen** Dateien anhängen in Qlik Sense ablegen, allerdings wird die Datei nicht hochgeladen.

Möglicher Grund

Die Datei ist einem ZIP-Archiv gespeichert. In Qlik Sense können Sie keine individuellen Dateien aus einem ZIP-Archiv anhängen, auch wenn das Archiv in einem Ordner in **Windows Explorer** angezeigt wird.

Empfohlene Vorgehensweise

Extrahieren Sie die Dateien aus dem ZIP-Archiv, bevor Sie sie anhängen.