

자습서 - 초보자를 위한 스크립팅

Qlik Sense®

February 2023

Copyright © 1993-2023 QlikTech International AB. 무단 전재 및 복제를 금합니다.



| | |
|----------------------------|-----------|
| 1 자습서 시작! | 4 |
| 1.1 학습 내용 | 4 |
| 1.2 학습 대상 | 4 |
| 1.3 패키지 콘텐츠 | 4 |
| 1.4 이 자습서의 단원 | 4 |
| 1.5 추가 자료 및 리소스 | 5 |
| 2 데이터 로드 편집기에서 스크립팅 | 6 |
| 2.1 데이터 로드 편집기 | 6 |
| 2.2 스크립트 편집기 | 6 |
| 명령 및 함수에 대한 구문 도움말 액세스 | 7 |
| 스크립트에 주석 추가 | 7 |
| 코드 들여쓰기 | 7 |
| 준비된 테스트 스크립트 삽입 | 7 |
| 3 LOAD 및 SELECT 문 | 8 |
| 4 데이터 선택 및 로드 | 9 |
| 5 필드 이름 바꾸기 | 18 |
| 6 데이터 줄이기 | 20 |
| 7 데이터 변환 | 26 |
| 7.1 Resident LOAD | 26 |
| 7.2 선행 LOAD | 30 |
| 8 연결 | 32 |
| 8.1 자동 연결 | 32 |
| 8.2 강제 연결 | 35 |
| 8.3 연결 방지 | 36 |
| 9 순환 참조 | 38 |
| 9.1 순환 참조의 해결 | 39 |
| 10 가상 키 | 41 |
| 10.1 가상 키 해결 | 43 |
| 11 앱에서 데이터 사용 | 45 |
| 11.1 차트 추가 | 45 |
| 11.2 차원 및 측정값 추가 | 45 |
| 차원 생성 및 추가 | 46 |
| 측정값 생성 및 추가 | 46 |
| 11.3 감사합니다. | 50 |

1 자습서 시작!

자습서에 오신 것을 환영합니다. 이 자습서에서는 Qlik Sense에서 기본적인 스크립팅을 수행하는 방법에 대해 소개합니다.

Qlik Sense에서 앱에 시각화를 만들려면 먼저 데이터를 로드해야 합니다. 로드 스크립트 사용 방법을 알면 앱에 데이터를 로드하는 경우 데이터를 준비하고 조작할 수 있습니다.

데이터 관리자 또는 데이터 로드 편집기를 사용하여 데이터를 로드할 수 있습니다. 데이터 로드 스크립트를 만들고 편집하고 실행하려면 데이터 로드 편집기를 사용합니다.

1.1 학습 내용

이 자습서를 완료한 후에는 스크립트를 사용한 데이터 로드, 스크립트 편집 및 데이터 변환 작업에 익숙해져야 합니다.

1.2 학습 대상

Qlik Sense의 기본 사항에 익숙해야 합니다. 즉, 앱 및 시각화를 만들 수 있어야 합니다.

데이터 로드 편집기에 액세스하고 Qlik Sense Enterprise on Windows에서 데이터를 로드할 수 있어야 합니다.

1.3 패키지 콘텐츠

다운로드한 zip 패키지에는 자습서를 완료하는 데 필요한 다음 데이터 파일이 포함되어 있습니다.

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

또한 패키지에는 *Scripting Tutorial* 앱의 복사본이 포함되어 있습니다. 허브로 앱을 업로드할 수 있습니다.

학습 효과를 최대화하려면 자습서에 설명된 대로 직접 앱을 작성하는 것이 좋습니다. 또한 작업할 데이터를 로드하기 위해 자습서에 설명된 대로 데이터 파일을 업로드하고 연결해야 합니다.

문제가 있는 경우 앱을 통해 문제를 해결할 수 있습니다. 각 단원과 관련된 스크립트 세그먼트를 표시했습니다.

1.4 이 자습서의 단원

이 자습서를 완료하려면 Qlik Sense를 사용하는 환경에 따라 3-4시간이 걸립니다. 항목은 순서대로 완료되도록 설계되었습니다. 그러나 중단했다가 언제든지 다시 돌아올 수 있습니다. 다행히도 테스트는 없습니다.

- 데이터 로드에 대한 소개
- LOAD 및 SELECT 문
- 데이터 선택 및 로드
- 필드 이름 바꾸기
- 데이터 줄이기
- 데이터 변환
- 연결
- 순환 참조
- 가상 키
- 앱에서 데이터 사용

1.5 추가 자료 및 리소스

- [Qlik](#)에서는 보다 자세한 정보를 알아볼 수 있도록 다양한 리소스를 제공합니다.
- [Qlik 온라인 도움말](#)을 사용할 수 있습니다.
- [Qlik Continuous Classroom](#)에서 교육(무료 온라인 과정 포함)이 제공됩니다.
- [Qlik Community](#)에서 토론 포럼, 블로그 등을 찾을 수 있습니다.

2 데이터 로드 편집기에서 스크립팅

Qlik Sense는 데이터 로드 편집기에서 관리되는 데이터 로드 스크립트를 사용하여 다양한 데이터 소스에서 데이터에 연결하고 데이터를 검색합니다. 데이터 파일(예: Excel 파일 또는 .csv 파일)이 데이터 소스가 될 수 있습니다. 데이터베이스(예: Google BigQuery 또는 Salesforce 데이터베이스)도 데이터 소스가 될 수 있습니다.

데이터 관리자를 사용하여 데이터를 로드할 수도 있지만, 데이터 로드 스크립트를 만들고 편집하고 실행하려면 데이터 로드 편집기를 사용합니다.

로드할 필드와 테이블은 스크립트에서 지정합니다. 스크립팅은 데이터 소스에서 로드할 데이터를 지정하는 데 사용되는 경우가 많습니다. 스크립트 문을 사용하여 데이터 구조를 편집할 수도 있습니다.

데이터 로드 중에 Qlik Sense는 다른 테이블에서 공통 필드(키 필드)를 식별하여 데이터를 연결합니다. 앱 내 데이터의 결과 데이터 구조는 데이터 모델 뷰어에서 모니터링할 수 있습니다. 필드의 이름 변경을 통해 테이블 사이에 다른 연결을 구축하여 데이터 구조를 변경할 수 있습니다.

데이터는 Qlik Sense에 로드된 후 앱에 저장됩니다.

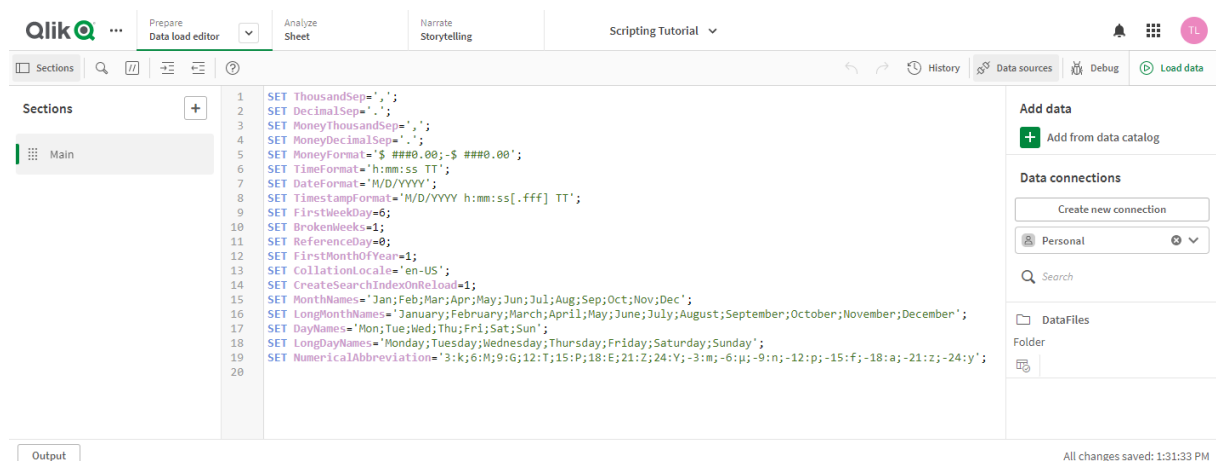
2.1 데이터 로드 편집기

데이터 로드 편집기에서 데이터를 로드하는 스크립트를 만들 수 있습니다. 편집기는 Qlik Sense의 드롭다운 메뉴에서 사용할 수 있습니다.

데이터 로드 편집기를 열면 화면 중간에 스크립트 편집기가 나타납니다. 스크립트 섹션이 왼쪽 메뉴에 탭으로 표시됩니다. Qlik Sense는 자동으로 기본 섹션을 만듭니다. 데이터 연결이 오른쪽 메뉴에 표시됩니다.

스크립트는 Qlik Sense 스크립트 구문을 사용하여 작성해야 합니다. Qlik Sense 구문 키워드는 파란색으로 강조 표시됩니다.

데이터 로드 편집기



2.2 스크립트 편집기

편집기에는 로드 스크립트를 개발하는 데 도움을 줄 수 있는 여러 가지 기능이 포함되어 있습니다.

명령 및 함수에 대한 구문 도움말 액세스

여러 가지 방법으로 Qlik Sense 구문 키워드에 대한 구문 도움말에 액세스할 수 있습니다.

도움말 포털 액세스

두 가지 방법으로 Qlik Sense 도움말 포털의 상세 도움말에 액세스할 수 있습니다.

- 도구 모음에서 (?)을 클릭하여 구문 도움말 모드를 시작합니다. 구문 도움말 모드에서 구문 키워드 (파란색과 밑줄로 표시)를 클릭하여 구문 도움말에 액세스할 수 있습니다.
- 키워드 내부 또는 끝에 커서를 놓고 Ctrl+H를 누릅니다.



구문 도움말 모드에서는 스크립트를 편집할 수 없습니다.

자동 완성 기능 사용

Qlik Sense 스크립트 키워드를 입력하기 시작하면 일치하는 키워드 중에서 선택할 수 있는 자동 완성 목록이 표시됩니다. 입력을 계속하면 목록이 좁혀지고 제안 구문 및 매개 변수가 포함된 템플릿에서 선택할 수 있습니다. 도구 설명에서 문 또는 함수의 도움말 포털 설명에 대한 링크뿐만 아니라 파라미터 및 추가 문을 포함하여 함수의 구문을 표시합니다.



또한 키워드 목록을 표시하는 키보드 바로 가기(Ctrl+Space)와 도구 설명을 표시하는 키보드 바로 가기(Ctrl+Shift+Space)를 사용할 수 있습니다.

스크립트에 주석 추가

스크립트 코드에 주석을 삽입하거나, 주석 표시를 사용하여 스크립트 코드의 일부를 비활성화할 수 있습니다. 한 줄에서 ///(슬래시 두 개)의 오른쪽에 이어지는 모든 텍스트는 주석으로 간주되며 스크립트를 실행할 때 실행되지 않습니다.

데이터 로드 편집기 도구 모음에는 코드를 주석 처리하거나 주석 처리를 제거할 수 있는 바로 가기가 포함되어 있습니다. [//]을 클릭하거나 Ctrl + K를 누르면 코드를 주석 처리하거나 주석 처리를 제거할 수 있습니다.

코드 들여쓰기

코드를 들여쓰면 가독성을 높일 수 있습니다. [Tab]을 클릭하여 텍스트를 들여쓰거나(들여쓰기 증가), [Shift+Tab]을 클릭하여 텍스트를 내어씹습니다(들여쓰기 감소).

준비된 테스트 스크립트 삽입

이 기능을 사용하면 일련의 인라인 데이터 필드를 로드하는 준비된 테스트 스크립트를 삽입할 수 있습니다. 이 기능을 사용하면 테스트용 데이터 셋을 신속하게 생성할 수 있습니다. 테스트 스크립트를 삽입하려면 Ctrl + 00을 누릅니다.

3 LOAD 및 SELECT 문

LOAD 및 SELECT 문을 사용하여 데이터를 Qlik Sense에 로드할 수 있습니다. 이러한 각 문은 내부 테이블을 생성합니다. LOAD는 파일에서 데이터를 로드하는 데 사용되고 SELECT는 데이터베이스에서 데이터를 로드하는 데 사용됩니다.

이 자습서에서는 파일의 데이터를 사용하므로 LOAD 문을 사용합니다.

또한 선행 LOAD를 사용하여 로드된 데이터의 내용을 편집할 수도 있습니다. 예를 들어 필드 이름을 바꾸려면 LOAD 문을 사용해야 합니다. SELECT 문은 필드 이름을 변경할 수 없습니다.

Qlik Sense에 데이터를 로드할 때 다음과 같은 규칙이 적용됩니다.

- Qlik Sense는 LOAD 또는 SELECT 문으로 생성된 테이블을 구분하지 않습니다. 이는 여러 테이블이 로드될 때 테이블이 LOAD 또는 SELECT 문 또는 이 둘의 조합으로 로드되어도 상관이 없음을 의미합니다.
- 문이나 데이터베이스에 있는 원래 테이블 내의 필드 순서는 Qlik Sense 논리에서 중요하지 않습니다.
- 필드 이름은 대/소문자를 구분하며 데이터 테이블 간에 연결을 설정하는 데 사용됩니다. 따라서 원하는 데이터 모델을 연결하기 위해 로드 스크립트에서 필드 이름을 변경해야 합니다.

4 데이터 선택 및 로드

Microsoft Excel 등의 파일 또는 기타 지원되는 파일 형식에서 데이터를 로드하는 작업은 데이터 로드 편집기에서 데이터 선택 대화 상자를 사용하여 간편하게 수행할 수 있습니다.

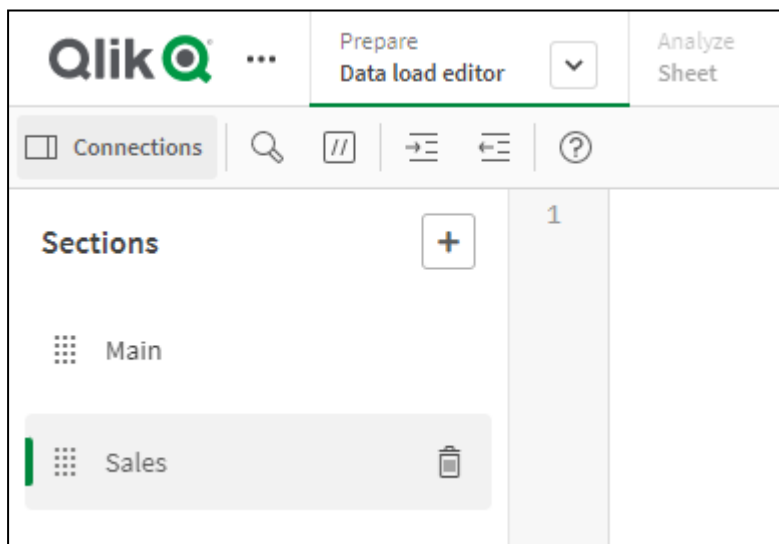
다음과 같이 하십시오.

1. Qlik Sense를 엽니다.
2. 새 앱을 만듭니다.
3. 앱 이름을 *Scripting Tutorial*로 지정한 다음 **만들기**를 클릭합니다.
4. 앱을 엽니다.



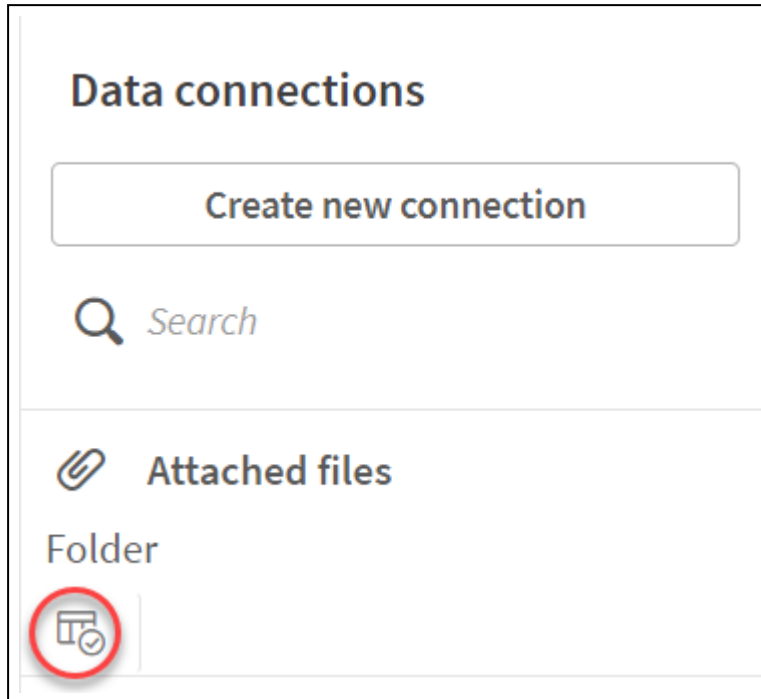
처음으로 데이터를 앱으로 로드하기 전에 **데이터 추가**를 사용하여 파일에서 쉽게 데이터를 로드할 수 있는 옵션이 있습니다. 하지만 이 자습서에서 스크립트를 보기를 원하므로 데이터 로드 편집기를 사용할 예정입니다.

5. 상단 도구 모음의 드롭다운 메뉴에서 데이터 로드 편집기를 엽니다.
6. 왼쪽 메뉴에서 **+**을 클릭하여 이름이 *Main*인 섹션 아래에 새 스크립트 섹션을 추가합니다.
둘 이상의 섹션을 사용하면 스크립트를 간편하게 정리할 수 있습니다. 스크립트 섹션은 데이터 로드 순으로 실행됩니다.
7. *Sales*를 입력하여 섹션에 이름을 지정합니다.
Scripting Tutorial 앱의 새 *Sales* 탭



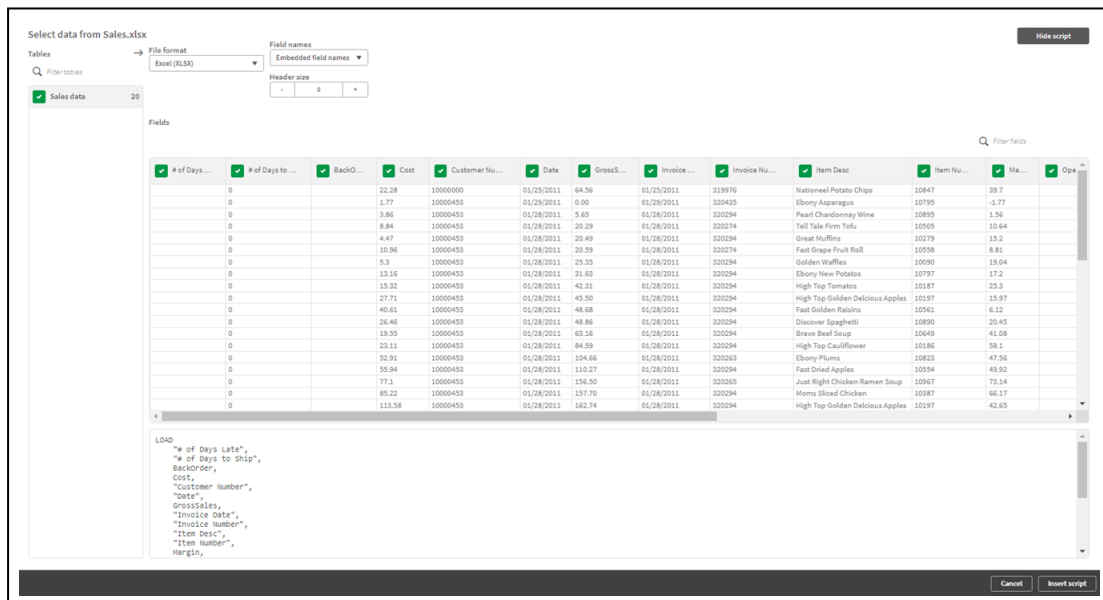
8. 오른쪽 메뉴의 **AttachedFiles**에서 **데이터 선택**을 클릭합니다.

데이터 선택 창



9. 업로드한 다음 *Sales.xlsx*를 선택합니다. 데이터 미리 보기 창이 열립니다.

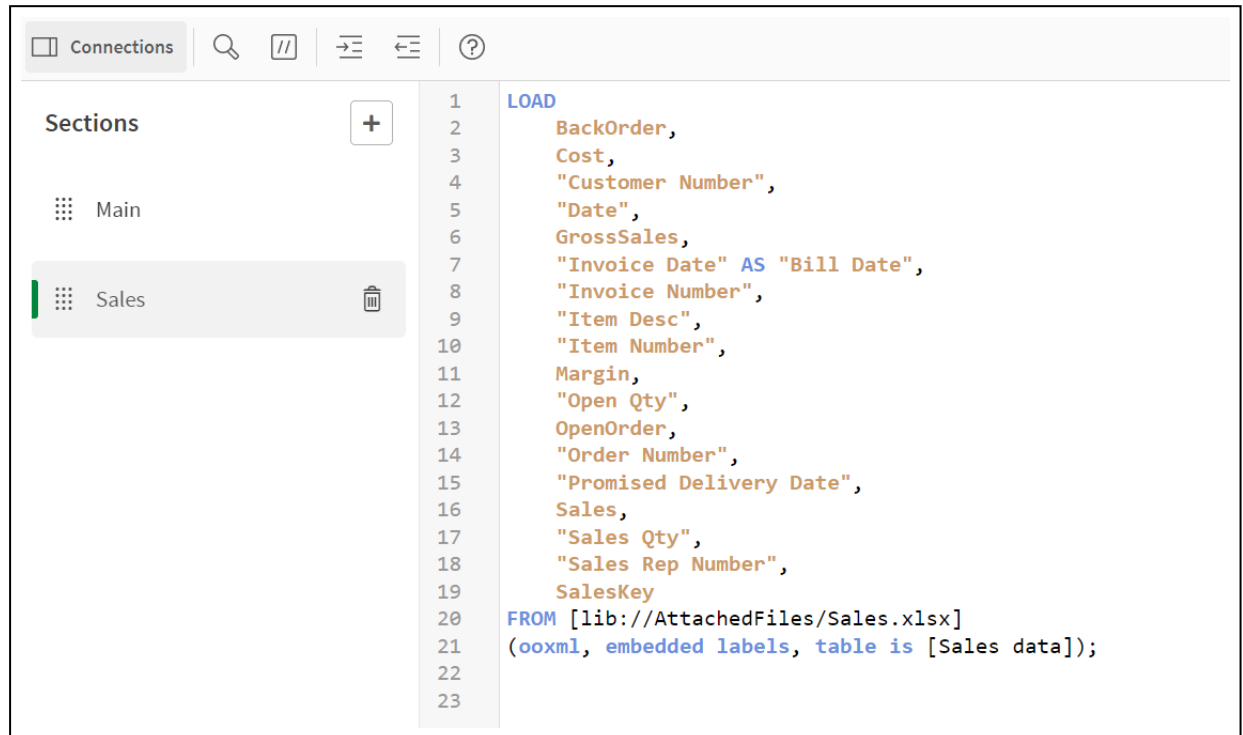
Sales 데이터 파일의 데이터 미리 보기 창



10. 필드 *# of Days Late* 및 *# of Days to Ship*을 선택 취소합니다. 전체 필드 이름을 보려면 필드 머리글을 클릭해야 할 수 있습니다.
11. 필드 필터링 검색 필드에서 *date*를 검색합니다.
12. 머리글 *Invoice Date*를 클릭하고 *Bill Date*를 입력하여 필드의 이름을 변경합니다.

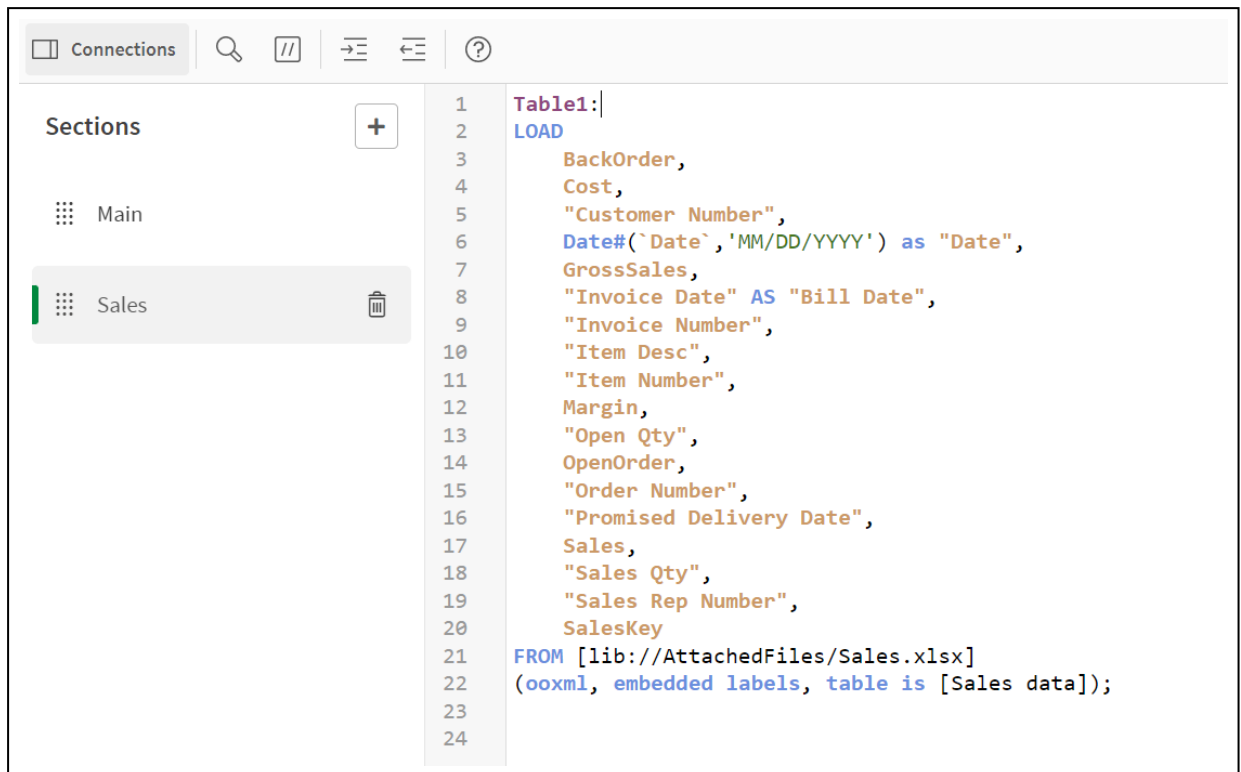
13. **스크립트 삽입**을 클릭합니다. 로드 스크립트가 스크립트 편집기의 *Sales* 섹션에 삽입됩니다. Qlik Sense는 공백이 포함된 필드 이름을 큰따옴표로 묶습니다. 스크립트는 다음과 같이 표시되어야 합니다.




Sales 탭의 로드 스크립트



14. *LOAD* 문 위에 다음 행을 추가하여 *Table1* 테이블의 이름을 지정합니다.
 Table1:
15. 이제 날짜가 정확하게 해석되도록 스크립트를 수정합니다. *Date* 필드를 다음으로 변경합니다.
 Date#(`Date`, 'MM/DD/YYYY') as "Date",
- 스크립트는 다음과 같이 표시되어야 합니다.

Sales 탭의 업데이트된 로드 스크립트





16. 오른쪽 위 모서리에서 **데이터 로드**를 클릭합니다.
스크립트 실행 진행률 창이 표시됩니다. 작업이 끝나면 오류 및 가상 키(없는 경우에도)에 대한 요약 정보가 표시됩니다.
17. **닫기**를 클릭합니다.
18. 상단 도구 모음의 드롭다운 메뉴에서 데이터 모델 뷰어를 엽니다. 를 클릭하면 데이터 모델 뷰어가 새 탭에서 열립니다.
19. 맨 위 메뉴에서  및 을 선택하여 이 자습서에서 사용되는 테이블 보기를 표시합니다. 테이블이 제대로 표시되지 않는 경우 기존 로드 스크립트를 제거하고 해당 스크립트를 다시 작성할 수 있습니다.

Sales 데이터에 대한 데이터 모델 뷰어의 테이블 보기

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

이제 *Dates*라는 다른 테이블을 로드하겠습니다. 테이블을 로드한 후 Qlik Sense는 *Date* 필드에서 *Sales* 테이블과 연결합니다.

20. **데이터 로드 편집기**를 엽니다.
21. 새 스크립트 섹션을 추가하려면 **+**을 클릭합니다.
22. 섹션 이름을 *Dates*로 지정합니다. 새 섹션 *Dates*가 *Sales* 아래에 배치되지 않은 경우 포인터를  위로 이동한 다음 섹션을 *Sales* 섹션 아래로 끌어 순서를 다시 정렬합니다.
23. 스크립트의 맨 위 행을 클릭하고 을 클릭합니다.
스크립트에 `//`이 추가되었는지 확인합니다.
24. `//` 뒤에 다음 텍스트를 추가합니다.
`Loading data from Dates.xlsx`

이제 스크립트의 맨 윗줄은 다음과 같습니다.

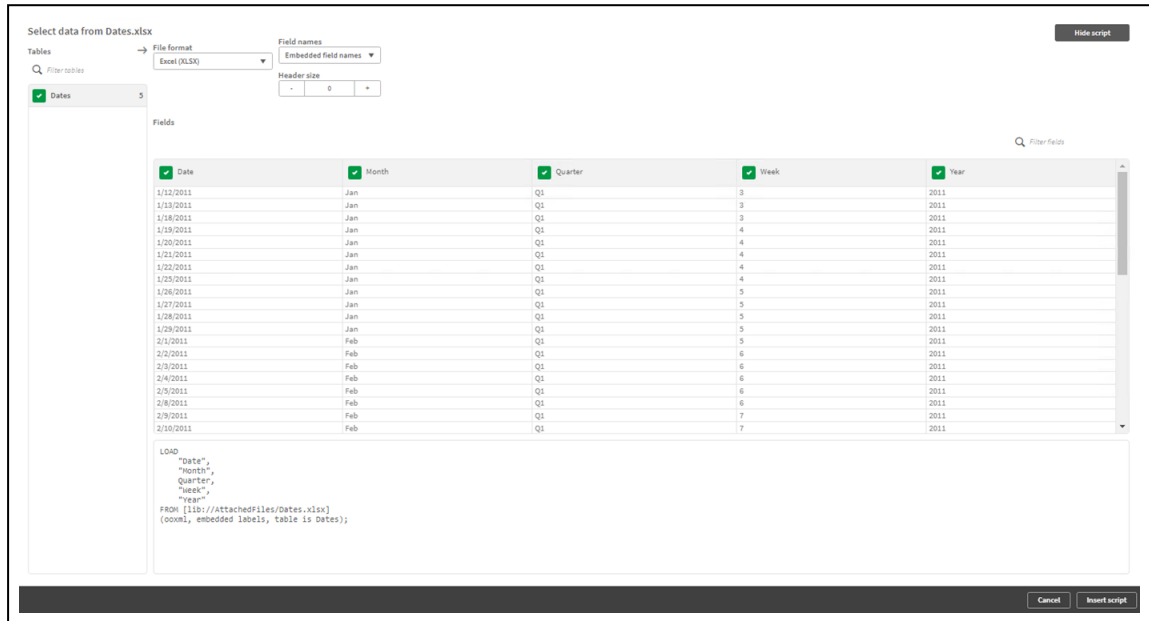
`// Loading data from Dates.xlsx`
25. 오른쪽 메뉴의 **AttachedFiles**에서 **데이터 선택**을 클릭합니다.



필드 이름 아래에서, 데이터를 로드할 때 테이블 필드의 이름을 포함하도록 포함된 필드 이름이 선택되어 있는지 확인합니다.

26. 업로드한 다음 *Dates.xlsx*를 선택합니다. 데이터 미리 보기 창이 열립니다.

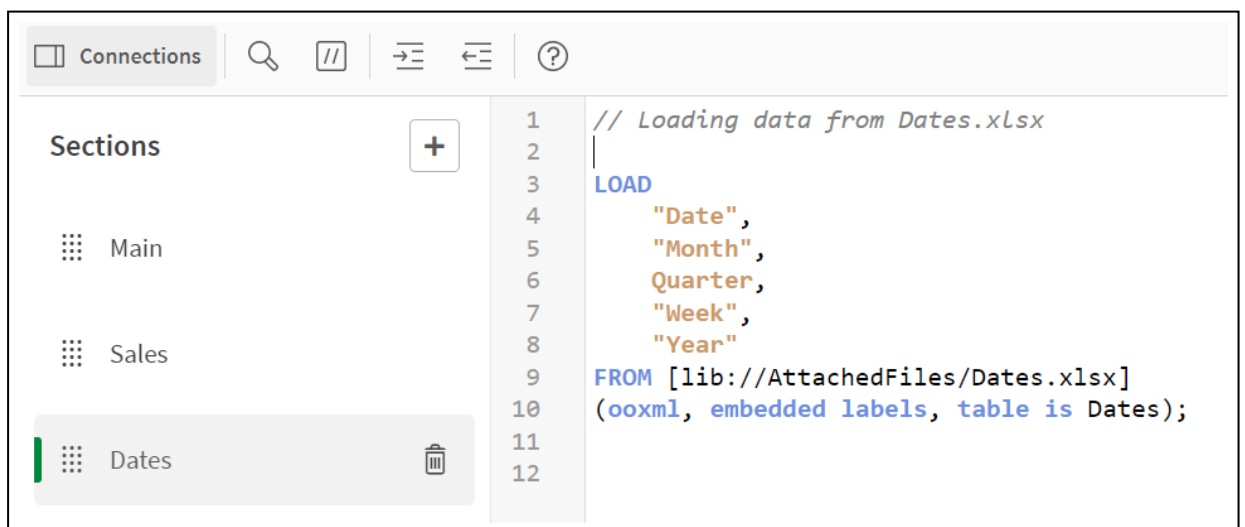
Dates 데이터 파일의 데이터 미리 보기 창



27. 스크립트 삽입을 클릭합니다.

스크립트는 다음과 같이 표시되어야 합니다.

Dates 탭의 로드 스크립트



28. `LOAD` 문 위의 행에 다음을 추가하여 `Table2` 테이블의 이름을 지정합니다.

`Table2:`

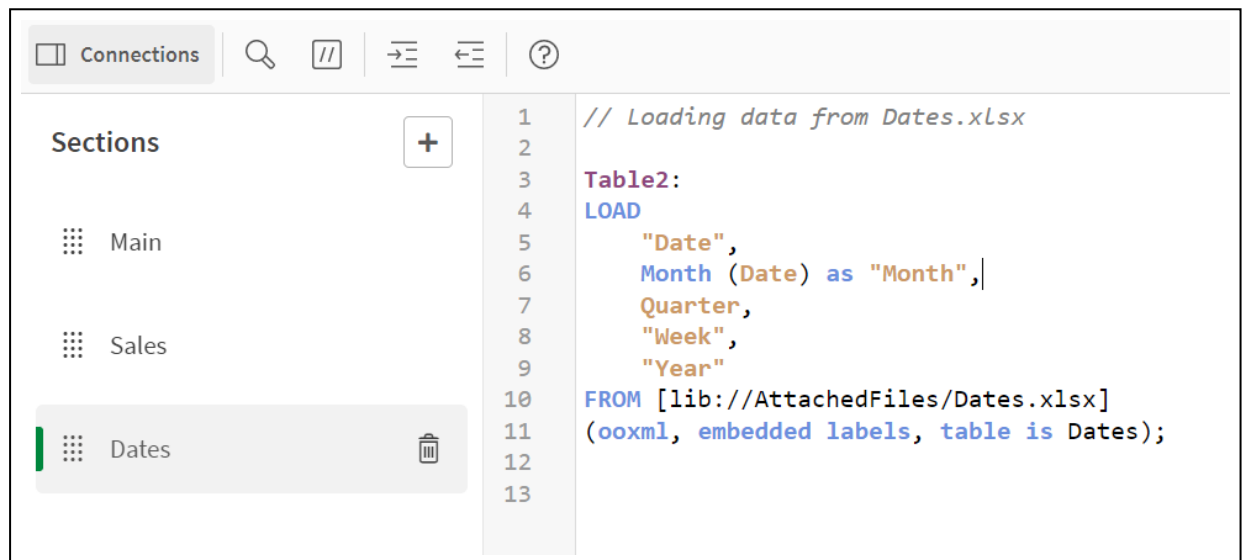
29. 파일 `Dates.xlsx`의 `Month` 열이 Qlik Sense으로 올바르게 해석되도록 하려면 `Date` 필드에 `Month` 함수를 적용해야 합니다.

월 필드를 다음으로 변경합니다.

`Month (Date) as "Month",`

스크립트는 다음과 같이 표시되어야 합니다.

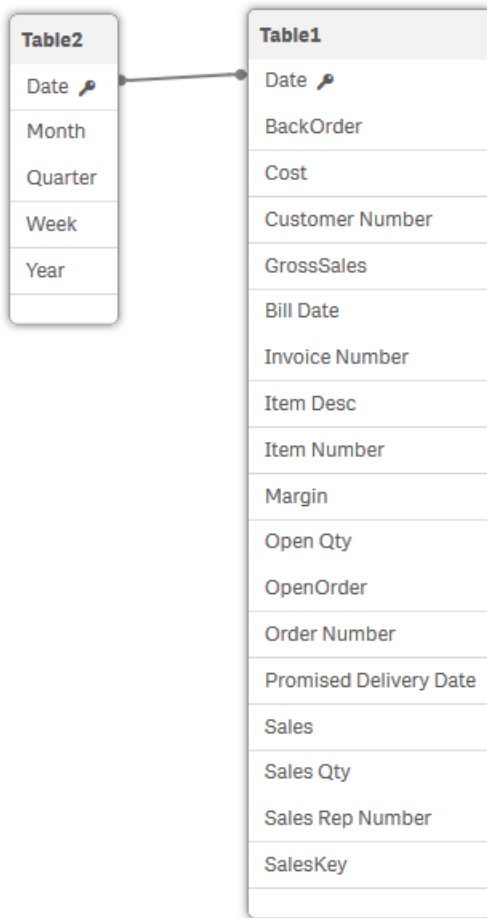
Dates 탭의 업데이트된 로드 스크립트



이제 파일 `Dates.xlsx`에서 선택된 데이터를 로드하기 위한 스크립트를 작성했습니다. 이제 데이터를 앱으로 로드하면 됩니다.

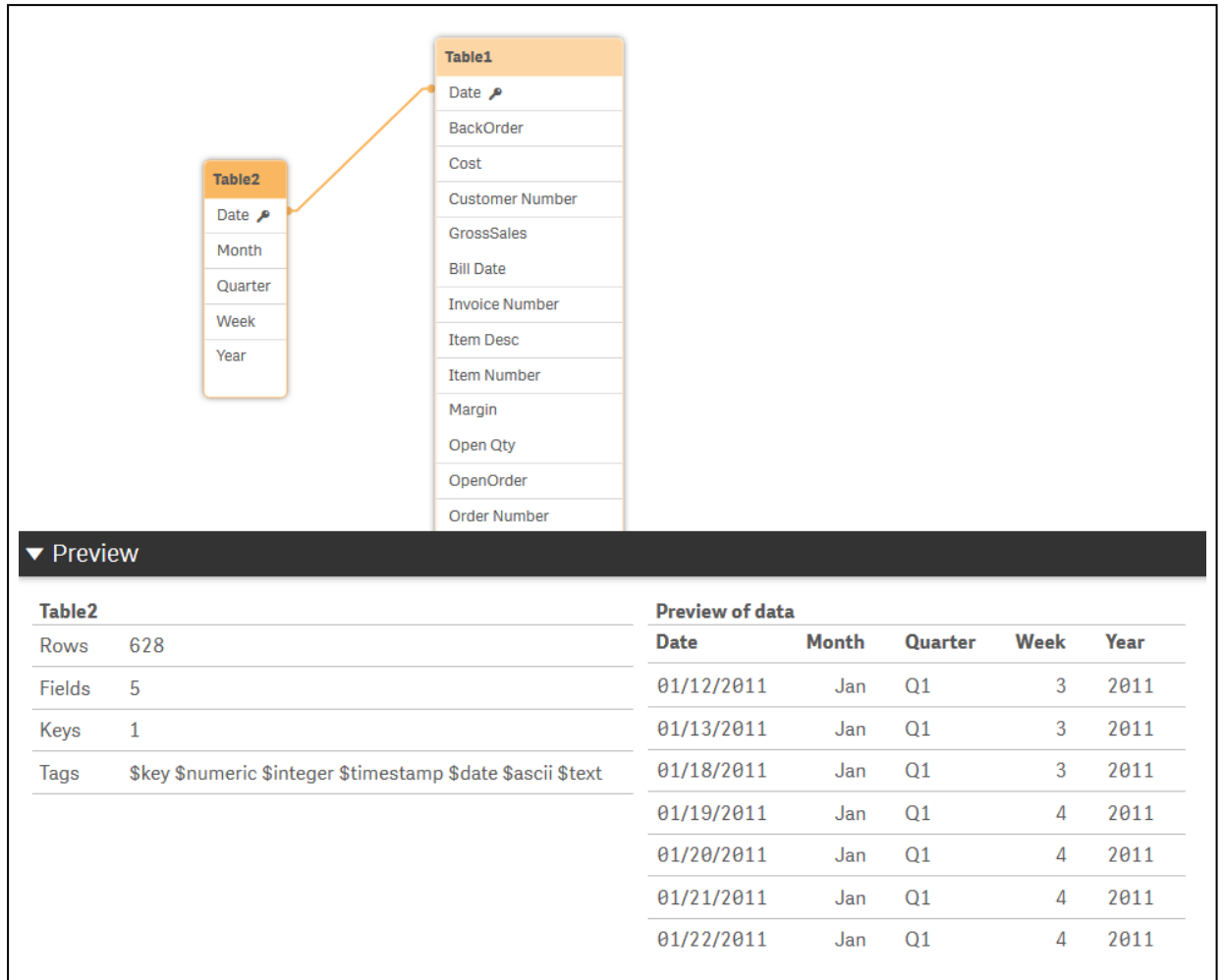
30. 오른쪽 위 모서리에서 **데이터 로드**를 클릭합니다.
데이터 로드를 클릭하면 데이터가 앱으로 로드되고 스크립트는 저장됩니다.
31. 스크립트 실행이 끝나면 **닫기**를 클릭합니다.
32. **데이터 모델 뷰어**를 엽니다.
 이제 두 테이블에서 이름이 `Date`인 두 필드 사이에 연결이 만들어졌음을 알 수 있습니다.

데이터 모델 뷰어의 테이블 보기



33. 왼쪽 아래 모서리에서 **미리 보기**를 클릭합니다. *Table2* 테이블의 이름을 클릭합니다.
- 그러면 해당 테이블에 대한 정보가 표시됩니다. **미리 보기** 필드에서 628개 행의 데이터가 내부 테이블 *Table2*로 로드되었음을 알 수 있습니다. 테이블의 필드를 클릭하면 해당 필드에 대한 정보가 표시됩니다.

데이터 모델 뷰어의 테이블 미리 보기



▼ Preview

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

이제 데이터를 앱의 시각화에 사용할 수 있습니다. 방법은 이 자습서의 후반부에 나와 있습니다.

5 필드 이름 바꾸기

이전 항목에서는 데이터 미리 보기 창에서 필드 이름을 바꾸는 방법에 대해 설명했습니다. 데이터 미리 보기 창의 제목에서 *Invoice Date*를 *Bill Date*로 변경했습니다. 로드 스크립트를 삽입하면 키워드 *AS*를 사용하여 필드 이름이 바뀌는 것을 확인할 수 있습니다.

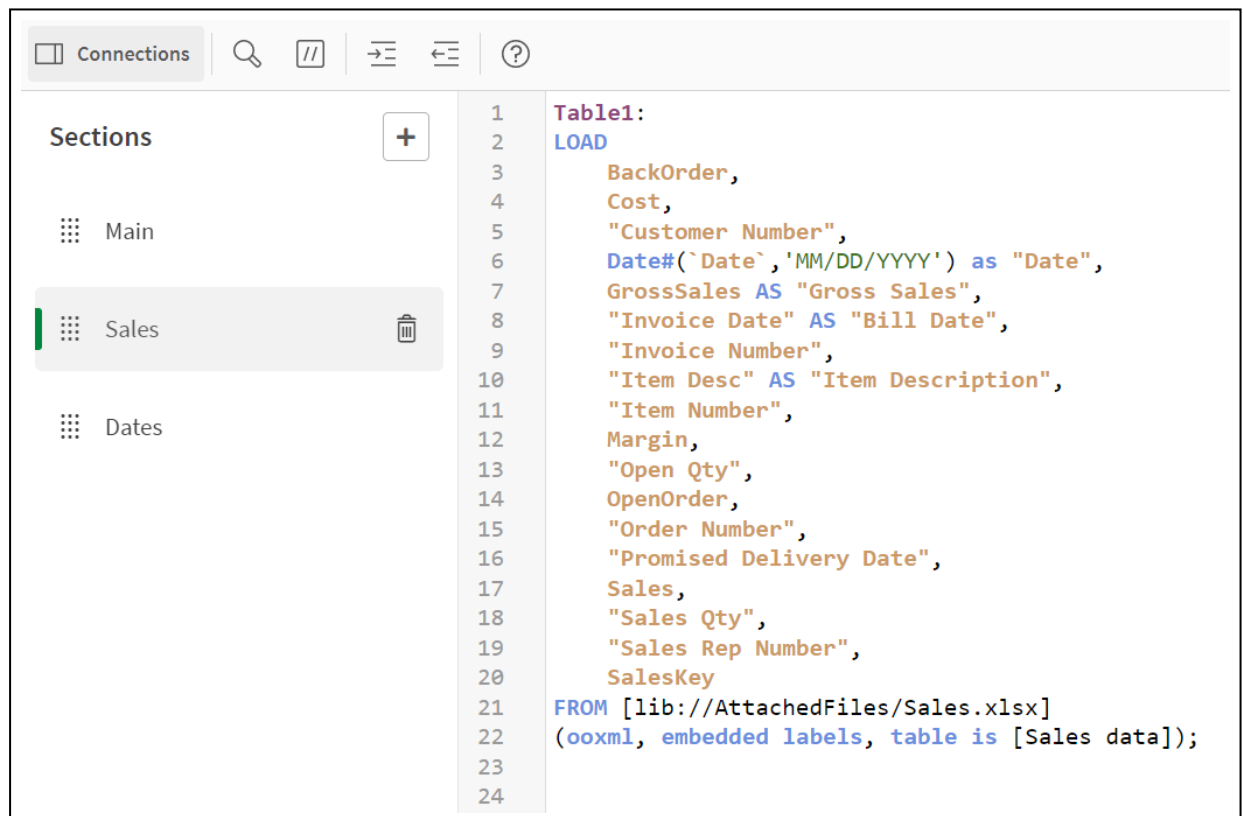
이 작업은 스크립트에서 직접 수행할 수도 있습니다.

다음과 같이 하십시오.

1. *Scripting Tutorial* 앱에서 **데이터 로드 편집기**를 엽니다.
2. **Sales** 탭을 클릭합니다.
3. 로드 스크립트에서 다음과 같이 변경합니다. 공백이 포함된 필드는 괄호로 묶어야 합니다.
 - i. *GrossSales*,를 다음으로 변경합니다.
`GrossSales AS "Gross Sales",`
 - ii. *"Item Desc"*,를 다음으로 변경합니다.
`"Item Desc" AS "Item Description",`

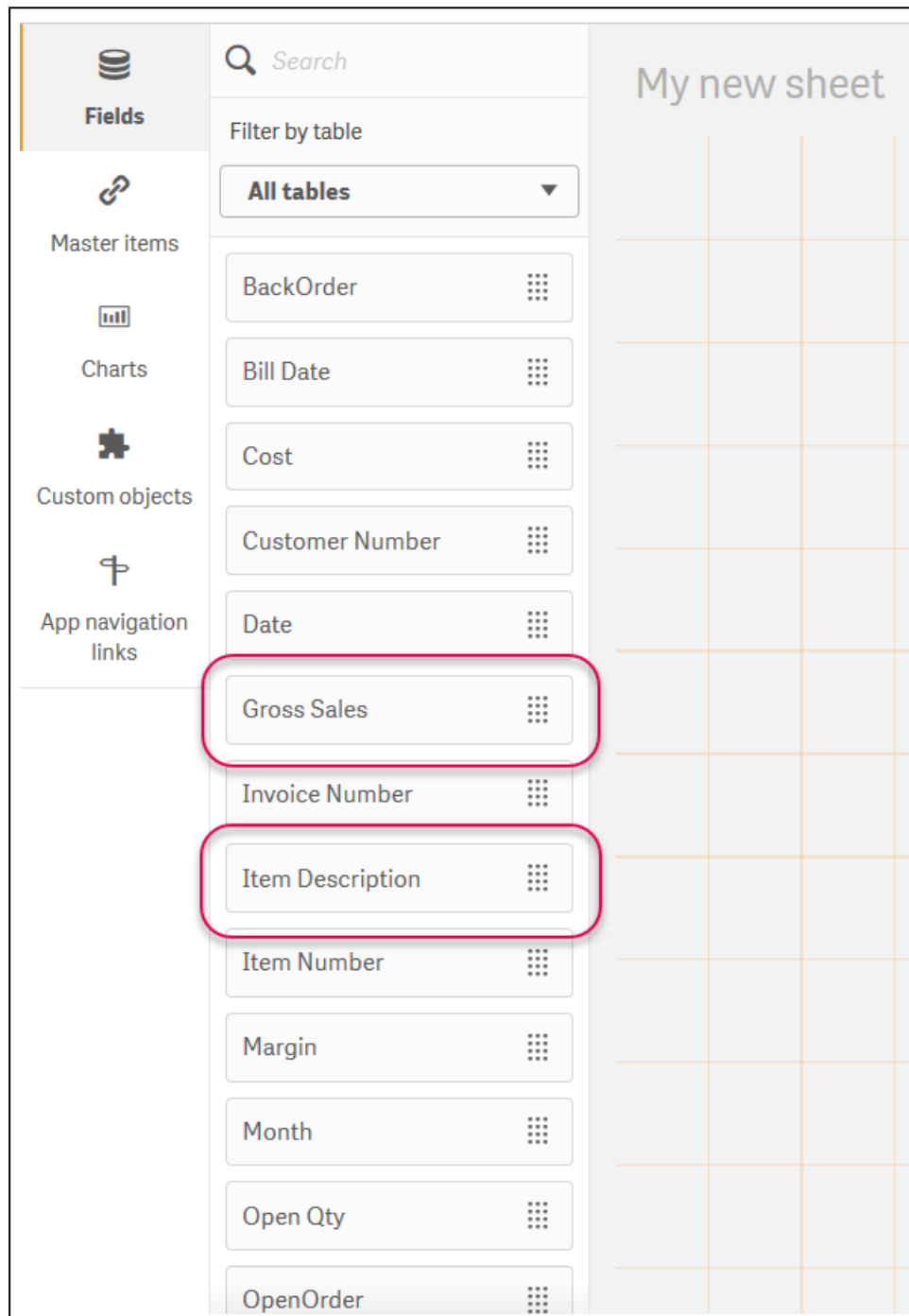
스크립트는 다음과 같이 표시되어야 합니다.

필드 이름을 바꾸는 스크립트가 있는 로드 스크립트 창



4. **데이터 로드**를 클릭합니다. 데이터가 로드됩니다.
5. **데이터 모델 뷰어**를 엽니다. 필드 이름이 변경되었는지 확인합니다.
6. 앱에서 모든 필드를 볼 수도 있습니다. 상위 도구 모음에서 **분석/시트** 탭을 클릭합니다. 앱이 시트 보기로 열립니다.
7. **시트 편집**을 클릭한 다음 자산 패널의 **필드**를 클릭합니다. 변경한 필드 이름을 볼 수 있습니다. 앱에서 만드는 시각화에서 이 필드를 사용할 수 있습니다.

분석 보기에서 이름이 변경된 필드



6 데이터 줄이기

Qlik Sense는 앱에 로드하는 데이터의 양을 줄일 수 있는 여러 가지 방법을 제공합니다. 예를 들어 파일 또는 데이터 커넥터에서 데이터를 필터링할 수 있습니다.

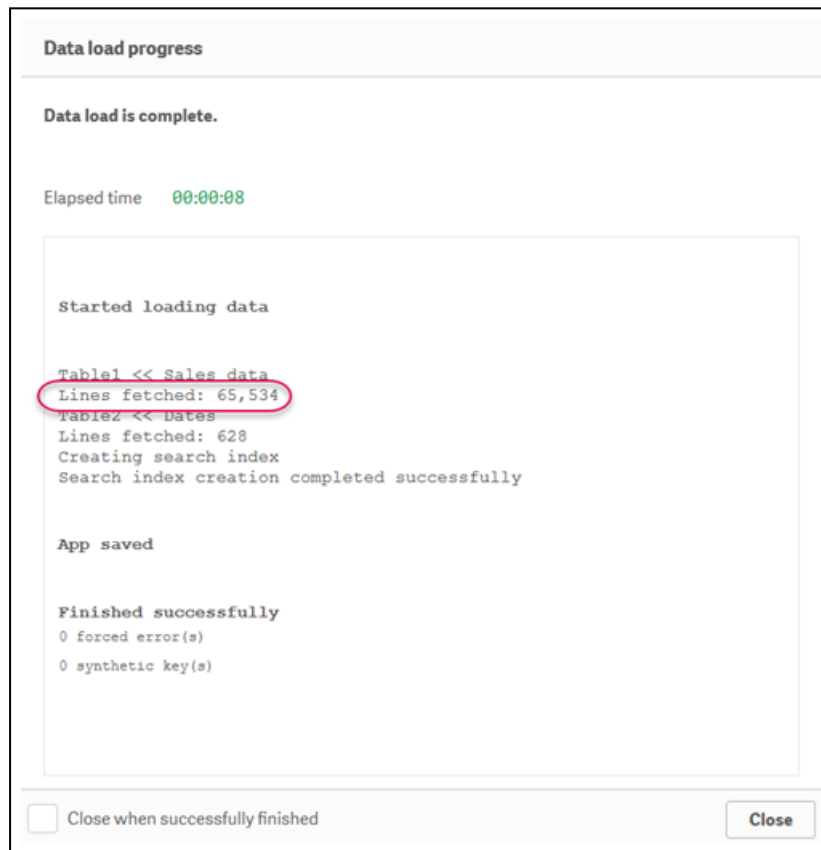
또한 로드 스크립트에서 직접 데이터를 줄일 수도 있습니다.

다음과 같이 하십시오.

1. *Scripting Tutorial* 앱에서 **데이터 로드 편집기**를 엽니다.
2. **데이터 로드**를 클릭합니다.

지금까지 작성한 로드 스크립트를 기반으로 Qlik Sense는 *Sales.xlsx* 데이터 파일에서 65,534줄을 *Table1*로 로드합니다. *Sales data*는 원래 *Sales.xlsx* 파일에서 테이블이 포함된 탭의 이름입니다.

데이터 로드 진행률 창

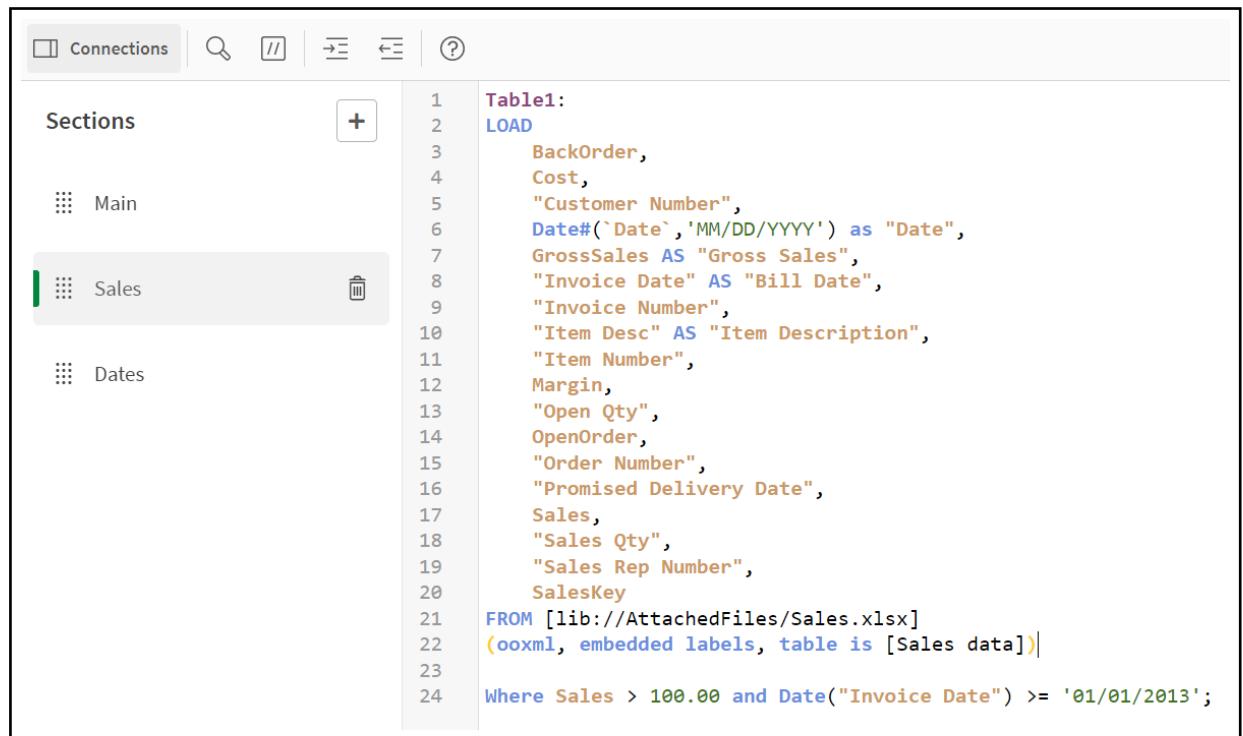


3. 데이터 로드 편집기에서 *Sales* 탭을 클릭합니다.
4. 다음 줄 끝 부분의 세미콜론을 삭제합니다.
`(ooxml, embedded labels, table is [Sales data]);`
5. 로드 스크립트의 끝 부분에 다음 줄을 추가합니다.
`where Sales > 100.00 and Date("Invoice Date") >= '01/01/2013';`

이는 Qlik Sense가 판매량이 \$100.00보다 큰 데이터만 로드하도록 합니다. 또한 Date 함수를 사용하여 날짜가 2013년 1월 1일 이후인 데이터를 로드하도록 합니다.

스크립트는 다음과 같이 표시되어야 합니다.

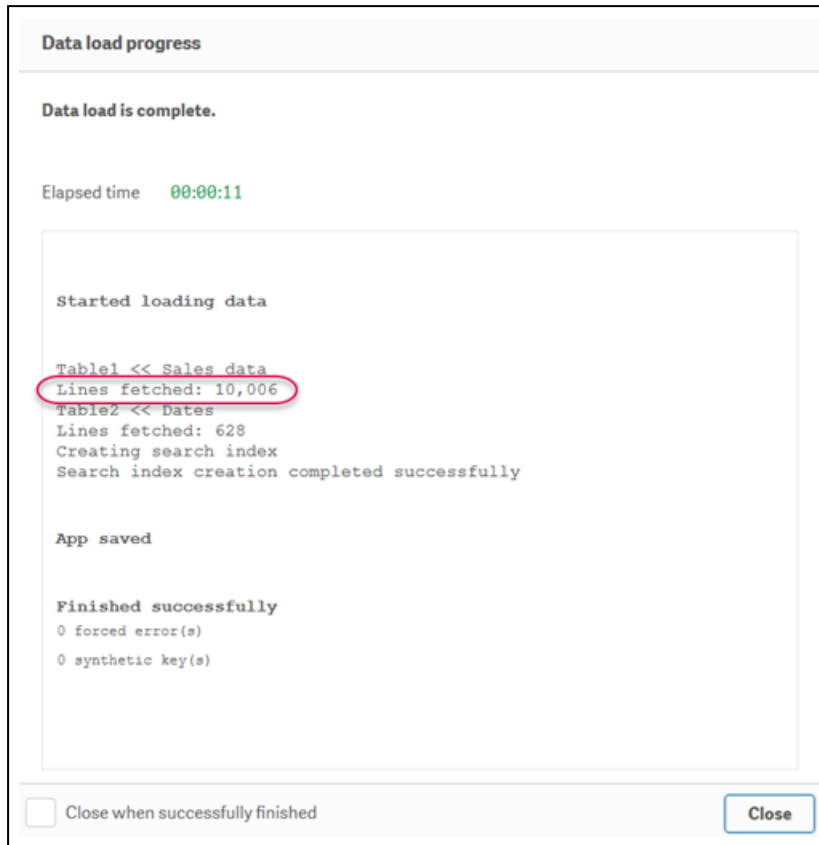
로드된 데이터 양을 줄이는 스크립트가 포함된 로드 스크립트 창



6. 데이터 로드를 클릭합니다.

업데이트된 로드 스크립트를 기반으로 이제 Qlik Sense는 *Sales.xlsx* 데이터 파일에서 더 적은 줄을 로드합니다.

데이터 로드가 감소한 데이터 로드 진행률 창



7. 앱의 테이블에 데이터를 추가하면 사용자가 만든 조건에 부합하는 데이터만 로드된 것을 확인할 수 있습니다.

Bill Date 및 *Sales* 필드가 포함된 테이블

My new sheet

Click to add title

| Bill Date | Sales |
|------------|--------|
| 01/01/2013 | 100.47 |
| 01/01/2013 | 106.93 |
| 01/01/2013 | 107.72 |
| 01/01/2013 | 108.88 |
| 01/01/2013 | 113.94 |
| 01/01/2013 | 118.89 |
| 01/01/2013 | 122.13 |
| 01/01/2013 | 123.6 |
| 01/01/2013 | 123.79 |
| 01/01/2013 | 124.32 |

Sales 필드를 차원으로 추가했습니다. 이는 *Sales* 값이 개별적으로 표시되도록 하기 위해서입니다. *Sales*를 측정값으로 추가할 경우 값은 날짜별로 집계됩니다.

일반적으로 *Sales*를 측정값으로 추가합니다. 측정값으로 추가하는 경우 열에 숫자 서식을 적용하여 값을 통화 금액(예: 달러)으로 표시하는 옵션이 있습니다.

Sales 측정값에 적용된 숫자 서식

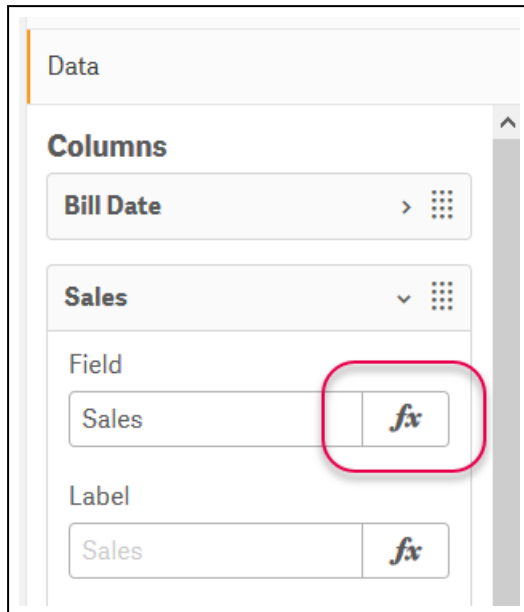
The screenshot shows the configuration panel for a measure named 'Sales'. The 'Expression' field is set to 'Sum(Sales)'. The 'Label' field is also set to 'Sum(Sales)'. The 'Number formatting' dropdown is set to 'Money'. The 'Format pattern' field displays '\$ ###0.00;- \$ ###0.00'. An 'Example' section at the bottom shows the formatted result as '\$ 1000.12'.

하지만 *Sales*를 차원으로 사용하고 있기 때문에 다른 접근 방식이 필요합니다. 이 경우 차트 표현식을 사용합니다. 이 자습서에서는 차트 함수에 대해 자세히 다루고 있지 않지만 간단한 예제를 살펴볼 수 있는 좋은 기회입니다.

*Sales*를 차원으로 사용하면 위 테이블에서 볼 수 있듯이 값이 숫자로 표시됩니다.

이를 수정하려면 **fx**를 클릭한 다음 *Money* 함수를 사용하여 차트 표현식 편집기 *Sales* 필드를 열면 됩니다.

식 편집기 열기



식 편집기에서 다음을 입력합니다.

=Money(sales)

식 편집기



이제 필드의 값이 통화 단위로 표시됩니다. 해당 단위(이 경우 달러)는 로드 스크립트의 **Main** 섹션에 지정됩니다.

Bill Date 및 *Sales* 필드가 포함된 테이블. 이제 *Sales*가 달러로 표시됨

My new sheet

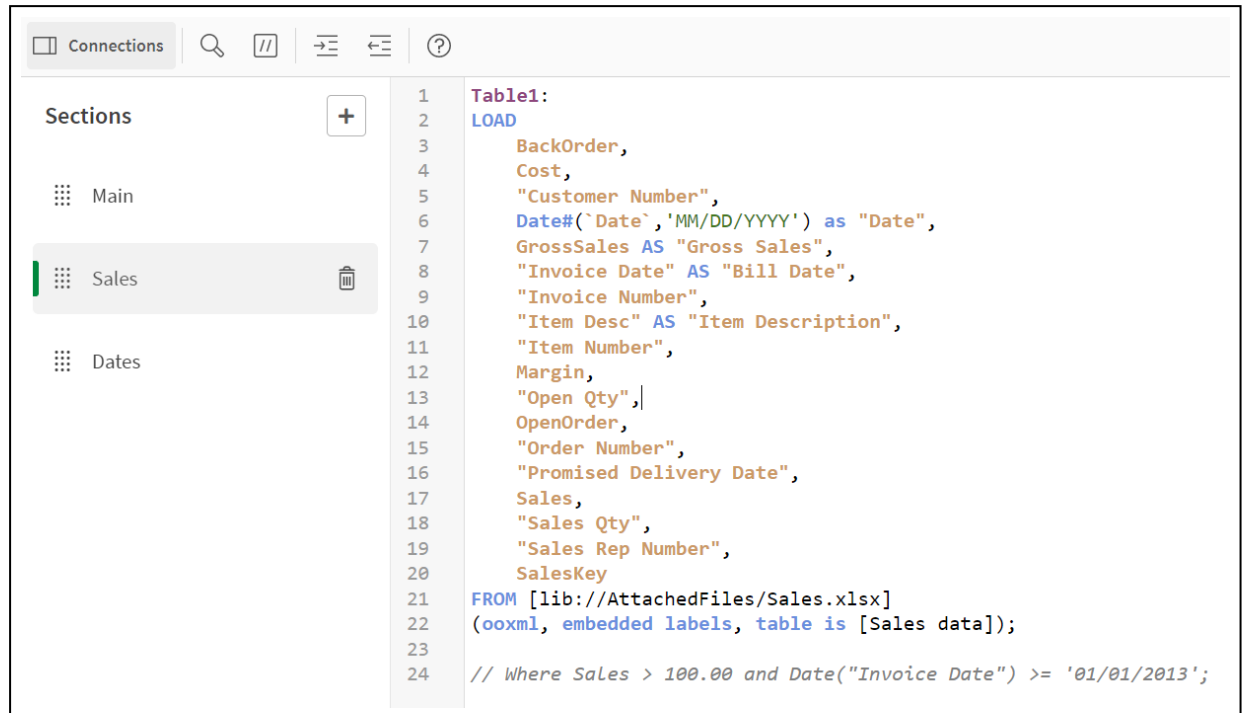
Click to add title

| Bill Date | Money(Sales) |
|------------|--------------|
| 01/01/2015 | \$100.47 |
| 01/01/2015 | \$108.83 |
| 01/01/2015 | \$107.72 |
| 01/01/2015 | \$108.88 |
| 01/01/2015 | \$115.94 |
| 01/01/2015 | \$118.89 |
| 01/01/2015 | \$122.13 |
| 01/01/2015 | \$123.60 |
| 01/01/2015 | \$123.73 |
| 01/01/2015 | \$124.32 |

- 이제 이 예제를 완료했습니다. Where 문을 주석 처리합니다. 처음 LOAD 문의 끝 부분에 세미콜론을 추가해야 합니다.

스크립트는 다음과 같이 표시되어야 합니다.

Where 문이 주석 처리된 로드 스크립트 창



7 데이터 변환

데이터 로드 편집기에서 다양한 기술을 사용하여 데이터를 변환하고 조작할 수 있습니다.

데이터 조작의 장점 중 하나는 테이블에서 몇 개의 선택한 열만 로드하는 등, 파일에서 데이터의 하위 집합만 로드하도록 선택하여 더 효과적으로 데이터를 처리할 수 있다는 점입니다. 또한 데이터를 여러 번 로드하여 원시 데이터를 여러 개의 새 논리 테이블로 분할할 수도 있습니다. 둘 이상의 소스에서 데이터를 로드하고 Qlik Sense에서 테이블 하나로 병합하는 것도 가능합니다.

이 항목에서는 Resident LOAD 및 Preceding LOAD를 사용하여 몇 가지 기본 데이터 변환을 수행합니다.

7.1 Resident LOAD

LOAD 문에서 Resident 소스 한정자를 사용하면 이전에 로드한 테이블에서 데이터를 로드할 수 있습니다. 날짜 또는 숫자 값 처리 등과 같이 Qlik Sense 함수를 사용하는 옵션이 없는 SELECT 문으로 로드한 데이터에서 계산을 수행하려는 경우에도 유용합니다.

이 예제에서는 *Sales_Buckets*라는 새 테이블을 만든 다음 Resident LOAD를 사용하여 *Table1*을 로드합니다. *Sales_Buckets* 테이블에서 *quantity_threshold*라는 변수를 만든 다음 Where 문을 사용하여 해당 임계값을 충족하는 데이터만 로드합니다.

1. *Scripting Tutorial* 앱에서 **데이터 로드 편집기**를 엽니다.
2. *Sales* 탭을 클릭합니다.
3. 스크립트의 끝 부분에 다음을 추가합니다.

```
SET quantity_threshold = 12000;
```

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

스크립트는 다음과 같이 표시되어야 합니다.

*Sales_Buckets*라는 새 테이블을 만드는 스크립트가 있는 로드 스크립트 창

The screenshot shows the Qlik Sense script editor interface. On the left, the 'Sections' pane lists 'Main', 'Sales' (selected), and 'Dates'. The main editor area displays a script with line numbers 1 through 32. The script defines a table named 'Table1' and creates a new table named 'Sales_Buckets' based on the data from 'Table1' where the quantity is greater than a threshold of 12000.

```

1 Table1:
2 LOAD
3     BackOrder,
4     Cost,
5     "Customer Number",
6     Date#(`Date`, 'MM/DD/YYYY') as "Date",
7     GrossSales AS "Gross Sales",
8     "Invoice Date" AS "Bill Date",
9     "Invoice Number",
10    "Item Desc" AS "Item Description",
11    "Item Number",
12    Margin,
13    "Open Qty",
14    OpenOrder,
15    "Order Number",
16    "Promised Delivery Date",
17    Sales,
18    "Sales Qty",|
19    "Sales Rep Number",
20    SalesKey
21 FROM [lib://AttachedFiles/Sales.xlsx]
22 (ooxml, embedded labels, table is [Sales data]);
23
24 SET quantity_threshold = 12000;
25
26 Sales_Buckets:
27 LOAD
28     "Sales Qty" as "High_Quantity",
29     "Item Description" as "Item",
30     "Customer Number" as "Customer"
31 Resident Table1
32 Where ("Sales Qty" > $(quantity_threshold));
  
```

4. 데이터 로드를 클릭합니다.
5. 데이터 모델 뷰어를 엽니다. 지정한 필드 및 설정한 임계값에 따라 로드된 데이터가 있는 *Sales_Buckets*라는 새 테이블이 생성되었음을 확인할 수 있습니다.

데이터 모델 뷰어의 Sales_Buckets 테이블

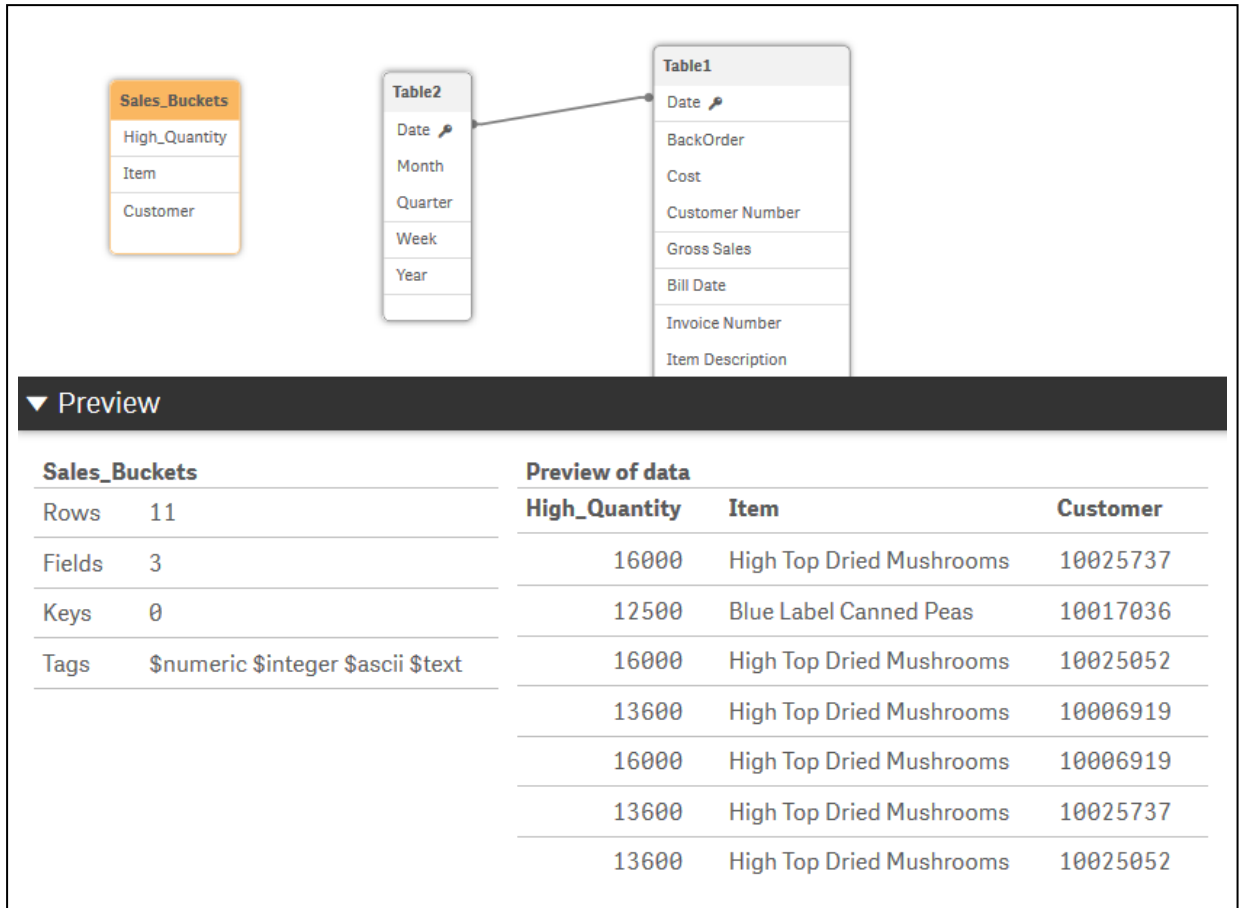


Table1

- Date
- BackOrder
- Cost
- Customer Number
- Gross Sales
- Bill Date
- Invoice Number
- Item Description

Table2

- Date
- Month
- Quarter
- Week
- Year

Sales_Buckets

- High_Quantity
- Item
- Customer

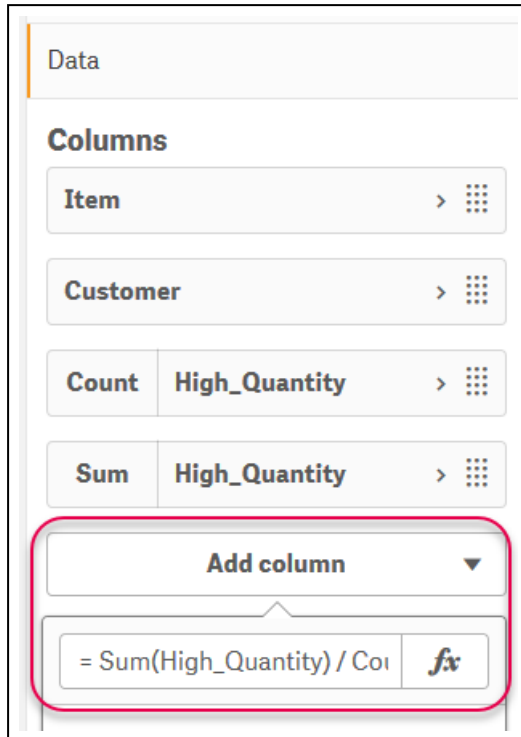
▼ Preview

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

6. 앱에서 테이블에 데이터를 추가합니다. *Item* 및 *Customer*를 차원으로 추가합니다. *High-Quantity*를 *Count*에 대해 집계된 측정값으로 추가한 다음 *Sum*에 대해 집계된 측정값으로 다시 추가합니다. 그런 후 다음 수식을 사용하여 새 열을 측정값으로 추가합니다.

= Sum(High_Quantity) / Count(High_Quantity)

수식을 사용한 새 측정값



테이블에 따르면 *Customer 10025737*은 *High Top Dried Mushrooms*에 대한 4건의 대량 주문을 했으며 평균 수량은 14,800개입니다. 필드의 데이터에 대해 정렬을 수행하려면 **완료**를 클릭하여 **편집** 모드를 종료합니다.

대량 주문을 한 고객을 보여주는 테이블

My new sheet

Click to add title

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

7. 이제 이 예제를 완료했습니다. *quantity_threshold* 변수 및 *Sales_Buckets* 테이블에 대한 스크립트를 주석 처리합니다.
- 이제 스크립트 끝 부분은 다음과 같습니다.

주석 처리된 스크립트

```

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

```

7.2 선행 LOAD

선행 Load를 사용하여 한 번에 데이터를 로드할 수 있도록 변환을 수행하고 필터를 적용할 수 있습니다. 일반적으로 사용하는 From 또는 Resident 등의 소스 한정자를 지정하지 않고 기본적으로 아래의 LOAD 또는 SELECT 문에서 로드하는 LOAD 문입니다. 이런 방식으로 개수에 관계없이 LOAD 문을 누적할 수 있습니다. 맨 아래의 문을 처음으로 평가한 다음 그 위의 문을 평가하는 방식으로 맨 위의 문이 평가될 때까지 계속 진행됩니다.

이 자습서 초반에 설명했듯이 LOAD 및 SELECT 문을 사용하여 Qlik Sense에 데이터를 로드할 수 있습니다. 위 문은 각각 내부 테이블을 생성합니다. LOAD는 파일 또는 인라인 테이블에서 데이터를 로드하는 데 사용되고 SELECT는 데이터베이스에서 데이터를 로드하는 데 사용됩니다. 이 자습서에서는 파일의 데이터를 사용했습니다. 이 예제에서는 인라인 테이블을 사용합니다. 그러나 선행 Load를 SELECT 문 위에 사용하여 데이터를 조작할 수 있습니다. 기본 사항은 여기에서 확인하는 것처럼 LOAD를 사용할 때와 동일합니다.

이 예는 이 자습서에서 로드 중인 데이터와 관계가 없습니다. 따라서 선행 Load가 어떤 것인지에 대한 예를 보여 주기 위해서만 사용됩니다. 데이터 로드 편집기에서 *Transactions*라는 인라인 테이블을 만듭니다. 선행 LOAD에서 날짜 해석이 수행되고 *transaction_date*라는 새 필드가 생성됩니다. 이 필드는 *sale_date* 필드에서 생성됩니다.

1. 새 앱을 만들고 *ReformatDate*라고 지정합니다.
2. 데이터 로드 편집기를 연 다음 *TransactionData*라는 새 탭을 만듭니다.
3. 다음 스크립트를 추가합니다.

```

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

```

스크립트는 다음과 같이 표시되어야 합니다.

선행 Load가 있는 로드 스크립트

Sections

+

Main

TransactionData

```

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

```

- 데이터 로드를 클릭합니다.
- 데이터 모델 뷰어를 엽니다. *Transactions* 테이블을 선택하고 확장합니다. 선행 Load 문에서 *로 지정한 대로 모든 필드가 로드된 것을 확인할 수 있습니다. *transaction_date*라는 새 필드가 생성되었습니다. 필드에는 서식이 재지정된 날짜가 있습니다.

데이터 모델 뷰어의 새 필드 *transaction_date*

Transactions

transaction_id

sale_date

transaction_amount

transaction_quantity

customer_id

size

color_code

transaction_date

▼ Preview

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

8 연결

연결은 두 테이블을 하나로 결합하는 연산입니다.

두 테이블은 각 개별 열 이름에 대해 하나의 열을 사용하여 하나를 다른 하나의 위에 누적하여 서로 추가됩니다. 데이터는 변경되지 않으며 결과 테이블에는 원래의 두 테이블을 합한 것과 동일한 수의 레코드가 포함됩니다. 여러 개의 연결 연산을 연속으로 수행할 수 있으며, 이런 경우 결과 테이블은 셋 이상의 테이블에서 컨케이네이트됩니다.

8.1 자동 연결

로드한 두 개 이상의 테이블에서 필드 이름과 필드의 수가 완전히 동일한 경우 Qlik Sense는 자동으로 여러 문의 내용을 한 테이블로 연결합니다.

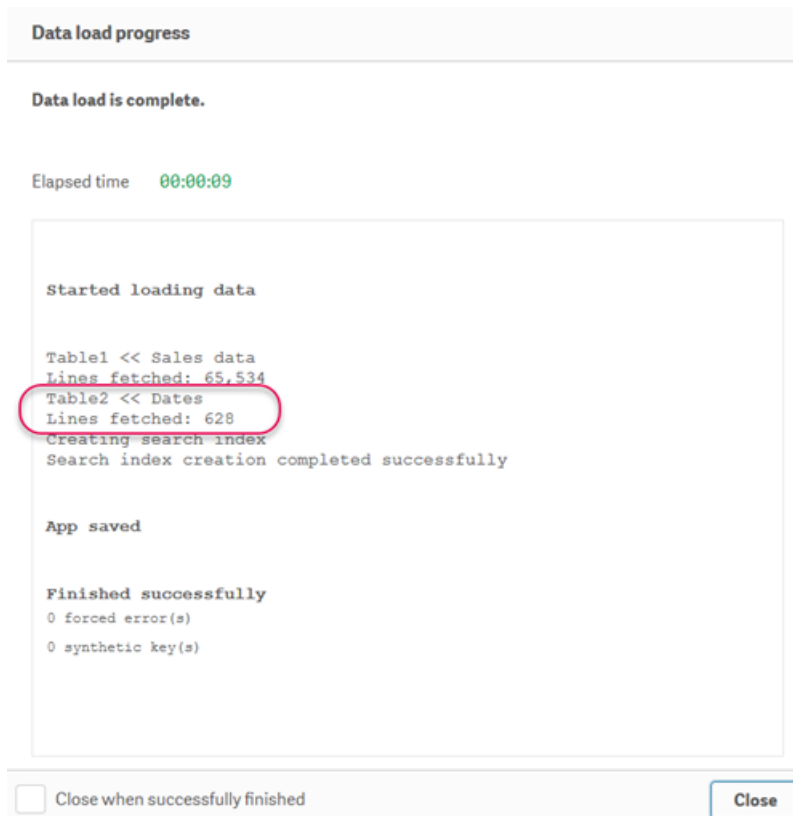
자동 연결을 수행하기 위해서는 필드의 수와 이름이 완전히 동일해야 합니다. 두 LOAD 문의 순서는 상관 없지만 테이블에는 첫 번째로 로드할 테이블의 이름이 지정됩니다.

다음과 같이 하십시오.

1. *Scripting Tutorial* 앱에서 **데이터 로드 편집기**를 엽니다.
2. **Dates** 탭을 클릭합니다.
3. **데이터 로드**를 클릭합니다.

지금까지 작성한 로드 스크립트를 기반으로 Qlik Sense는 *Dates.xlsx* 데이터 파일에서 628줄을 *Table2*로 로드합니다.

데이터 로드 진행률 창



4. *Dates* 섹션 스크립트의 새 줄에서 *Table2*의 LOAD 문을 복사하여 붙여넣습니다. 이렇게 하면 데이터가 두 번 로드됩니다. 두 번째 테이블 이름을 *Table2a*로 지정합니다. 기존 스크립트를 삭제하고 다음을 복사하여 붙여넣을 수도 있습니다.

Table2:

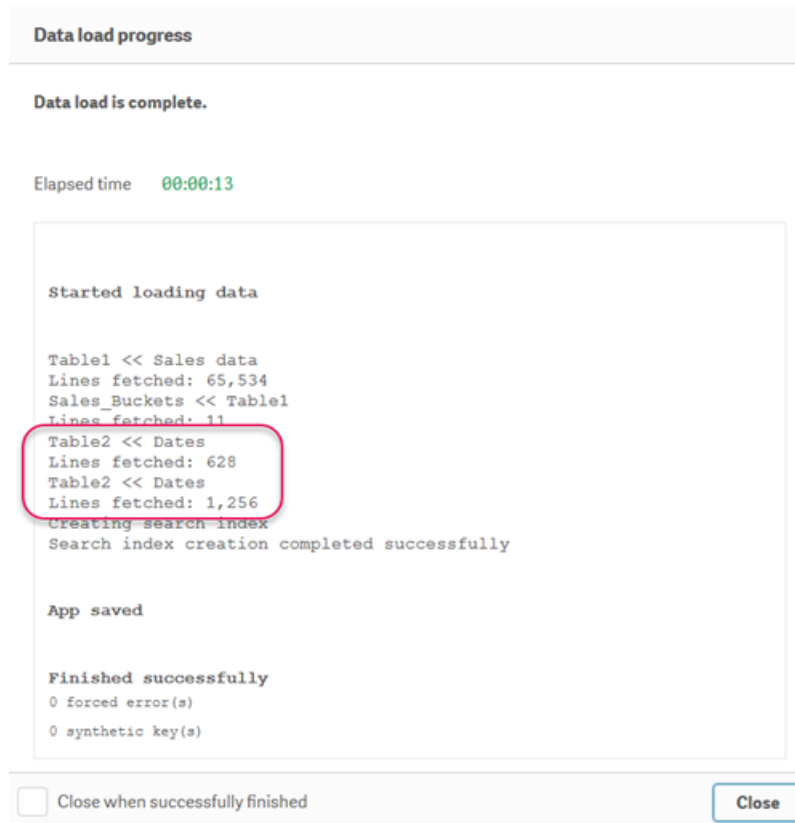
```
LOAD
    "Date",
    Month (Date) as "Month",
    Quarter,
    "Week",
    "Year"
FROM [lib://AttachedFiles/Dates.xlsx]
(ooxml, embedded labels, table is Dates);

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

5. **데이터 로드**를 클릭합니다.

Qlik Sense는 *Table2*를 로드한 다음 *Table2a*를 로드하지 않습니다. 대신 *Table2a*에 *Table2*와 동일한 필드 이름과 필드 수가 있다고 인식합니다. *Table2a*의 데이터를 *Table2*에 추가한 다음 테이블 *Table2a*를 삭제합니다. 그 결과 이제 *Table2*가 1,256줄입니다.

데이터 로드 진행률 창의 연결

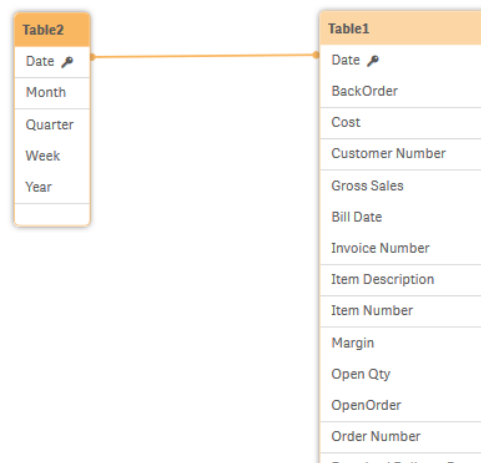


6. 데이터 모델 뷰어를 엽니다.

7. 미리 보기 표시를 클릭합니다.

*Table2*만 생성됩니다. *Table2*을 선택합니다. 테이블의 행은 256개입니다.

Table2를 보여 주는 데이터 모델 뷰어



▼ Preview

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

8.2 강제 연결

두 개 이상의 테이블이 완전하게 동일한 필드 집합을 가지지 않는 경우에도 Qlik Sense에서 두 테이블을 강제로 연결할 수 있습니다. 이를 위해서는 스크립트에 **Concatenate** 접두사를 사용하여 테이블을 다른 명명된 테이블이나 최근에 생성한 테이블과 컨케이트네이트하면 됩니다.

다음과 같이 하십시오.

1. Table2a의 LOAD 문을 편집하여 **Concatenate**를 추가하고 **Week**를 주석 처리합니다. 이제 스크립트는 다음과 같습니다.

Table2a:

```
Concatenate LOAD
    "Date",
    Month (Date) as "Month",
    Quarter,
    // "Week",
    "Year"
FROM [lib://AttachedFiles/Dates.xlsx]
```

```
(ooxml, embedded labels, table is Dates);
```

*Week*를 주석 처리하여 테이블이 동일하지 않도록 했습니다.

2. 데이터 로드를 클릭합니다.

3. 데이터 모델 뷰어를 엽니다.

이제 *Table2a*가 생성되지 않았음을 알 수 있습니다.

4. 데이터 모델 뷰어에서 *Table2*를 클릭한 다음 **미리 보기**를 클릭합니다.

테이블에 *Date*, *Month*, *Quarter*, *Week* 및 *Year* 필드가 있습니다. *Week* 필드는 *Table2*에서 로드되었으므로 계속 표시됩니다.

5. *Table2*에서 *Week*를 클릭합니다. 필드에 null이 아닌 값 수가 628로 미리 보기에 표시됩니다. 그러나 다른 필드를 클릭하면 null이 아닌 값 수가 1256으로 표시됩니다. *Week*는 *Table2*에서 한 번만 로드되었습니다. 값 또는 레코드 수는 *Table2* 및 *Table2a*의 레코드 수 합계입니다.

8.3 연결 방지

로드한 두 개 이상의 테이블에서 필드 이름과 필드의 수가 완전히 동일한 경우 Qlik Sense는 자동으로 여러 문의 내용을 한 테이블로 연결합니다. NoConcatenate 문을 사용하면 이를 방지할 수 있습니다. 그러면 연관된 LOAD 또는 SELECT 문을 사용하여 로드된 테이블이 기존 테이블과 컨케트네이트되지 않습니다.

다음과 같이 하십시오.

1. 두 테이블의 내용을 완전히 분리할 수 있도록 하려면 *Table2a*의 LOAD 문에 NoConcatenate를 추가하고 Qlik Sense가 일치하는 필드를 기반으로 가상 키를 만들지 않도록 필드 이름을 바꿉니다. 두 테이블이 동일한 필드를 갖도록 *Table2*의 *Week*에 대한 주석 처리를 제거합니다.

이제 스크립트는 다음과 같습니다.

Table2:

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

Table2a:

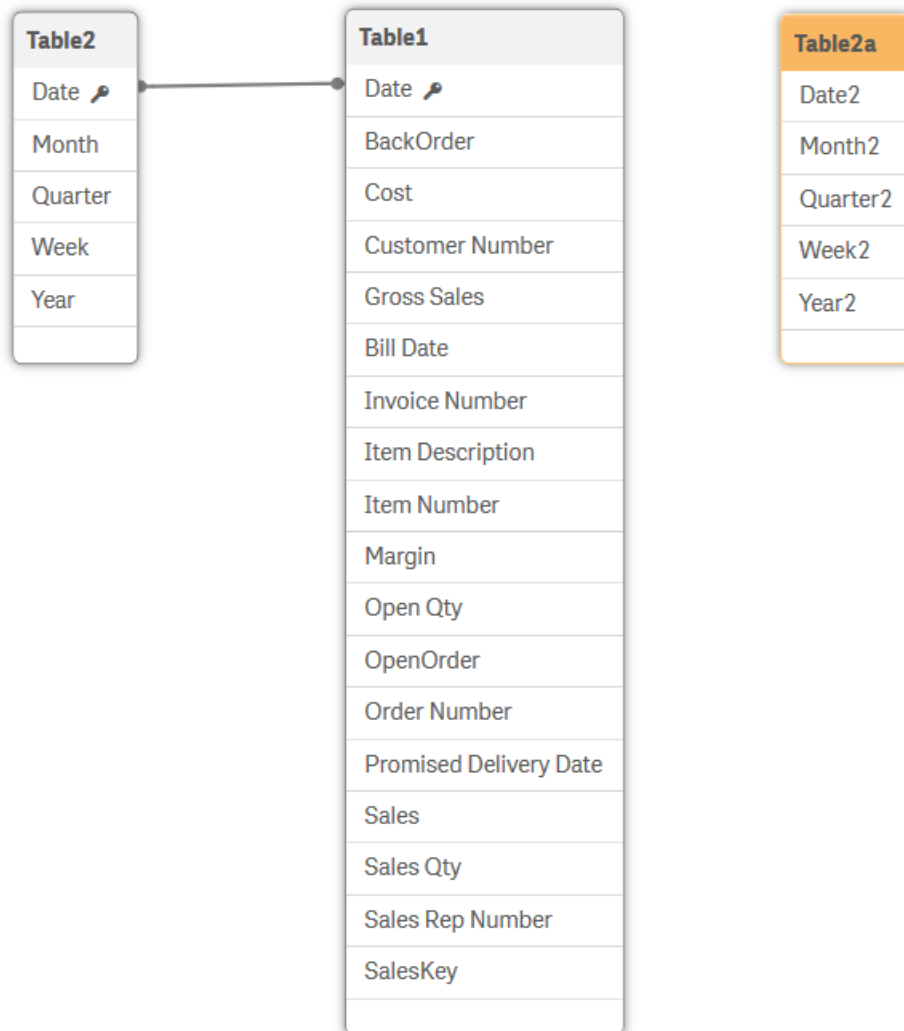
```
NoConcatenate LOAD
    "Date" as "Date2",
    Month (Date) as "Month2",
    Quarter as "Quarter2",
    "week" as "week2",
    "Year" as "Year2"
FROM [lib://AttachedFiles/Dates.xlsx]
(ooxml, embedded labels, table is Dates);
```

2. 데이터 로드를 클릭합니다.

3. 데이터 모델 뷰어를 엽니다.

이제 두 테이블이 완전히 분리되었음을 알 수 있습니다.

Table2 및 Table 2a를 보여 주는 데이터 모델 뷰어



- 이제 연결에 대한 데모가 완료되었으므로 *Table2a*가 더 이상 필요하지 않습니다. LOAD 문에서 *Table2a*에 대한 모든 행을 삭제한 다음 **데이터 로드**를 클릭합니다.

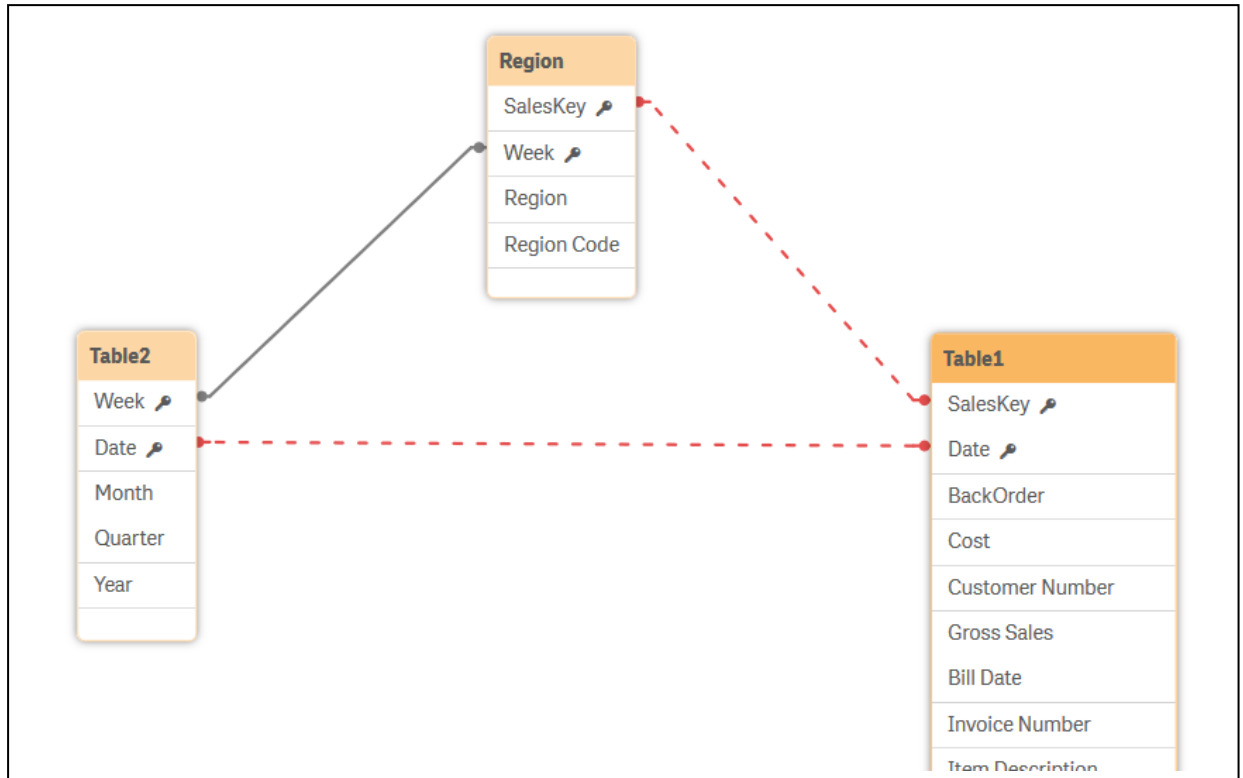
9 순환 참조

데이터 구조에 순환 참조(루프)가 있는 경우 두 필드 간 연결 경로가 둘 이상인 것처럼 테이블이 연결됩니다.

다음과 같이 하십시오.

1. *Scripting Tutorial* 앱에서 **데이터 로드 편집기**를 엽니다.
2. 새 스크립트 섹션을 추가하려면 **+**을 클릭합니다.
3. 섹션 이름을 *Region*으로 지정합니다.
4. 오른쪽 메뉴의 **AttachedFiles**에서 **데이터 선택**을 클릭합니다.
5. 업로드한 다음 *Region.txt*를 선택합니다. 데이터 미리 보기 창이 열립니다.
6. 모든 필드를 선택하고 **필드 이름** 아래의 **포함된 필드 이름**이 데이터를 로드할 때 테이블 필드의 이름을 포함하도록 선택되어 있는지 확인합니다.
7. **스크립트 삽입**을 클릭합니다.
8. **데이터 로드**를 클릭합니다.
이번에는 데이터 로드 시 문제가 발생한 것으로 나타납니다. 순환 참조가 만들어졌습니다. **데이터 로드 진행률** 창에는 로드 중에 순환 참조가 발견되었다는 오류 메시지가 표시됩니다. 하지만 로드는 완료되고 앱은 저장됩니다.
9. **데이터 모델 뷰어**를 엽니다.
테이블 간의 연결을 쉽게 볼 수 있도록 테이블을 끌어서 재구성할 수 있습니다.

순환 참조를 보여 주는 데이터 모델 뷰어



빨간색 점선은 순환 참조가 만들어졌음을 나타냅니다. 순환 참조로 인해 데이터 해석 시 모호성이 발생할 수 있으므로 피해야 합니다.

9.1 순환 참조의 해결

순환 참조가 발생한 원인을 파악할 수 있도록 데이터 모델 뷰어에서 테이블을 자세히 살펴보기로 하겠습니다.

위 스크린샷에서 Table1과 Table2를 살펴보면 공통적으로 Date 필드가 있는 것을 확인할 수 있습니다. 또한 Table1과 Region에 공통으로 SalesKey 필드가 있는 것을 알 수 있습니다. 마지막으로 Table2 및 Region에는 공통으로 Week 필드가 있습니다. 이것은 루프, 즉 순환 참조가 만들어졌음을 의미합니다. 이로 인해 이후의 데이터 분석에서 문제가 발생할 수 있으므로 제거하도록 하겠습니다.

이 문제를 해결하는 가장 쉬운 방법은 필드 중 하나의 이름을 변경하거나 제거하는 것입니다. 여기서는 앱에 필요하지 않는 일부 데이터를 로드했으므로 제거해보겠습니다.

다음과 같이 하십시오.

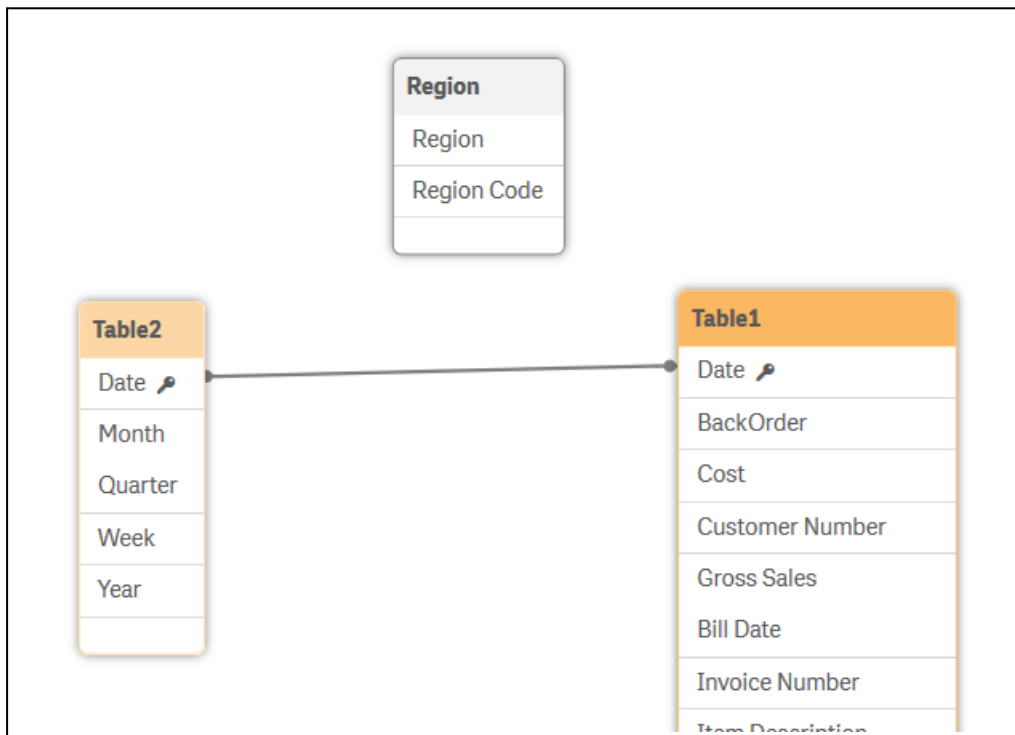
1. 데이터 로드 편집기를 엽니다.
2. Region 섹션을 클릭하고 LOAD 문에서 다음 두 행을 삭제합니다.
"week",
SalesKey
3. 또한 "Region Code" 뒤에서 쉼표도 제거되었는지 확인합니다.

이제 스크립트는 다음과 같습니다.

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

4. 데이터 로드를 클릭합니다.
5. 데이터 모델 뷰어를 엽니다.
*Region*에 대한 원치 않는 참조가 제거되었습니다.

순환 참조가 제거되었음을 보여 주는 데이터 모델 뷰어



10 가상 키

두 개 이상의 내부 테이블에 두 개 이상의 공통적인 필드가 있는 경우 이는 복합 키 관계를 의미합니다. Qlik Sense에서는 가상 키를 통해 이를 처리합니다. 이 키는 복합 키에서 발생하는 모든 조합을 나타내는 익명 필드입니다.

복합 키 수가 증가하면 메모리 사용량이 증가하고 성능에 영향을 미칠 수 있습니다. 이는 데이터 양, 테이블 구조 및 기타 요인에 따라 달라질 수도 있습니다. 서로 종속되어 있는 여러 개의 가상 키가 있는 경우 이를 제거하는 것이 좋습니다.

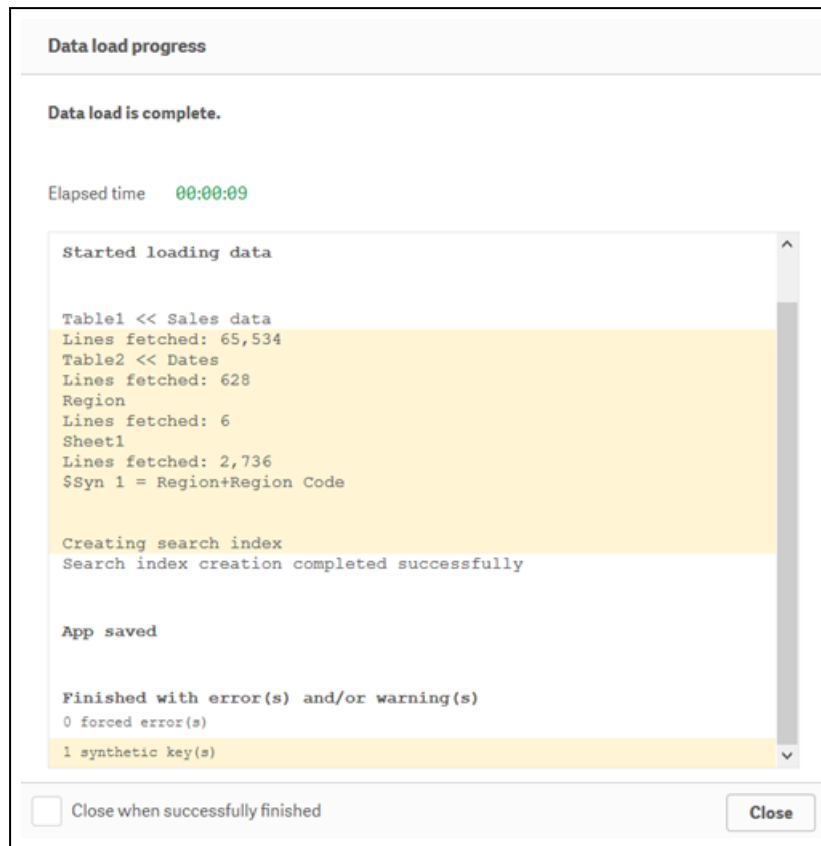
이제 마지막 데이터 집합을 로드할 차례입니다.

다음과 같이 하십시오.

1. *Scripting Tutorial* 앱에서 **데이터 로드 편집기**를 엽니다.
2. 새 스크립트 섹션을 추가하려면 **+**을 클릭합니다.
3. 섹션 이름을 *Customers*로 지정합니다.
4. 오른쪽 메뉴의 **AttachedFiles**에서 **데이터 선택**을 클릭합니다.
5. 업로드한 다음 *Customers.xlsx*를 선택합니다. 데이터 미리 보기 창이 열립니다.
6. *Select Sheet1*
7. **스크립트 삽입**을 클릭합니다.
8. **데이터 로드**를 클릭합니다.

이제 데이터 로드 진행률 창에서 가상 키가 만들어졌음을 알 수 있습니다.

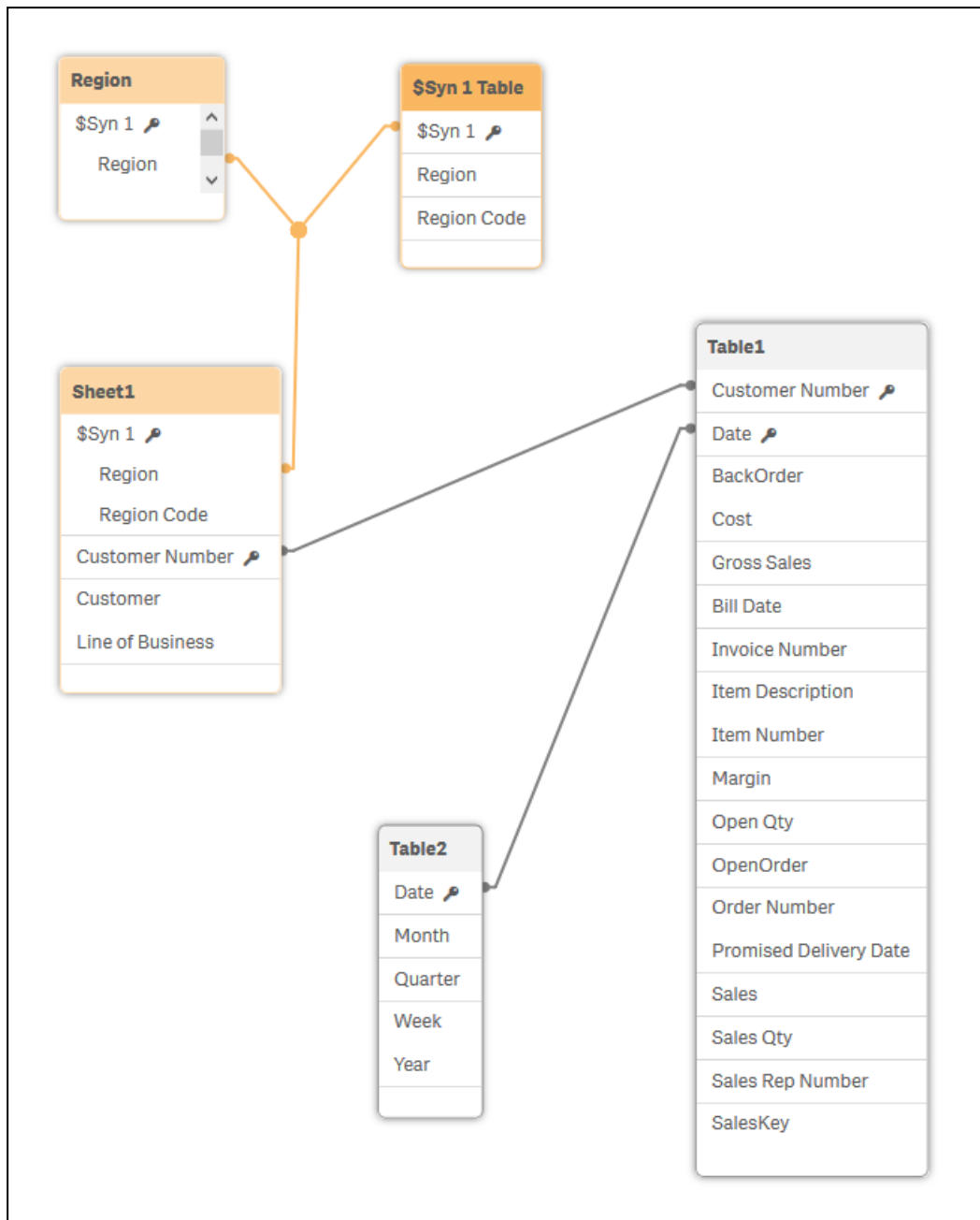
가상 키 경고가 있는 데이터 로드 진행률 창



9. 데이터 모델 뷰어를 엽니다.

새 테이블 *\$Syn 1 Table*이 만들어졌으면 가상 키가 만들어진 것을 알 수 있습니다. 여기에는 연결된 테이블 *Sheet1* 및 *Region*이 공통으로 가지고 있는 모든 필드(*Region* 및 *Region code*)가 포함되어 있습니다. 이 경우 연결이 약간 혼동되거나 오해의 소지가 있을 수 있으므로 그대로 두는 것은 바람직하지 않습니다.

가상 키를 보여 주는 데이터 모델 뷰어



10.1 가상 키 해결

가상 키를 없애는 가장 쉬운 방법은 테이블에서 하나 이상의 필드의 이름을 변경하는 것입니다. 이 작업은 데이터를 로드할 때 완료할 수 있습니다. 이제 가상 키를 제거하는 방법을 단계별로 알아보겠습니다.

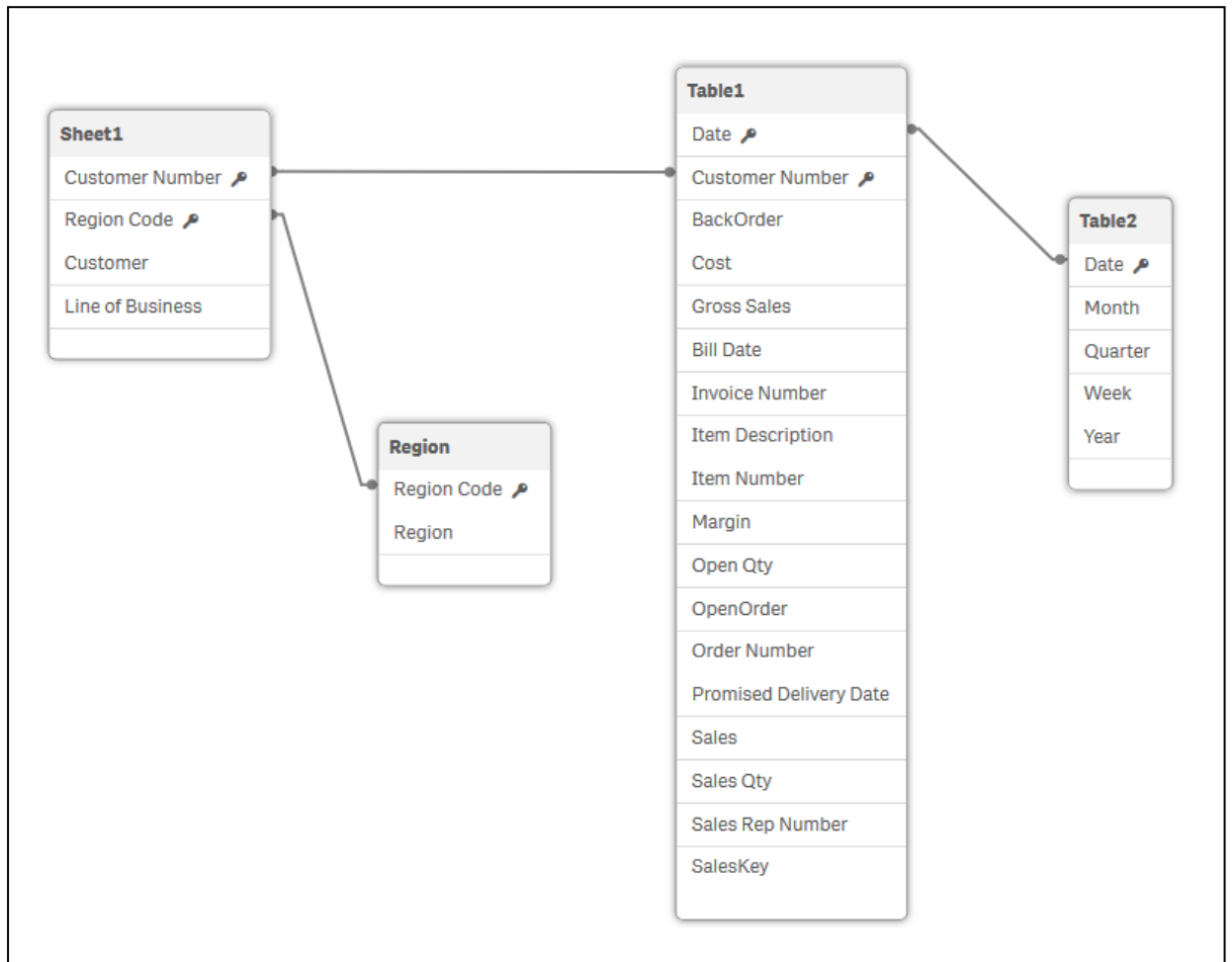
다음과 같이 하십시오.

1. 데이터 로드 편집기를 엽니다.
2. *Customers* 섹션을 클릭하고 **LOAD** 문에서 다음 행을 삭제합니다.

Region,

3. **데이터 로드**를 클릭합니다.
4. **데이터 모델 뷰어**를 엽니다.
가상 키가 제거되었습니다.

가상 키가 제거되었음을 보여 주는 데이터 모델 뷰어



11 앱에서 데이터 사용

이 자습서를 끝내기 위해 이제 로드된 데이터를 앱의 시각화에 배치해야 합니다.

11.1 차트 추가

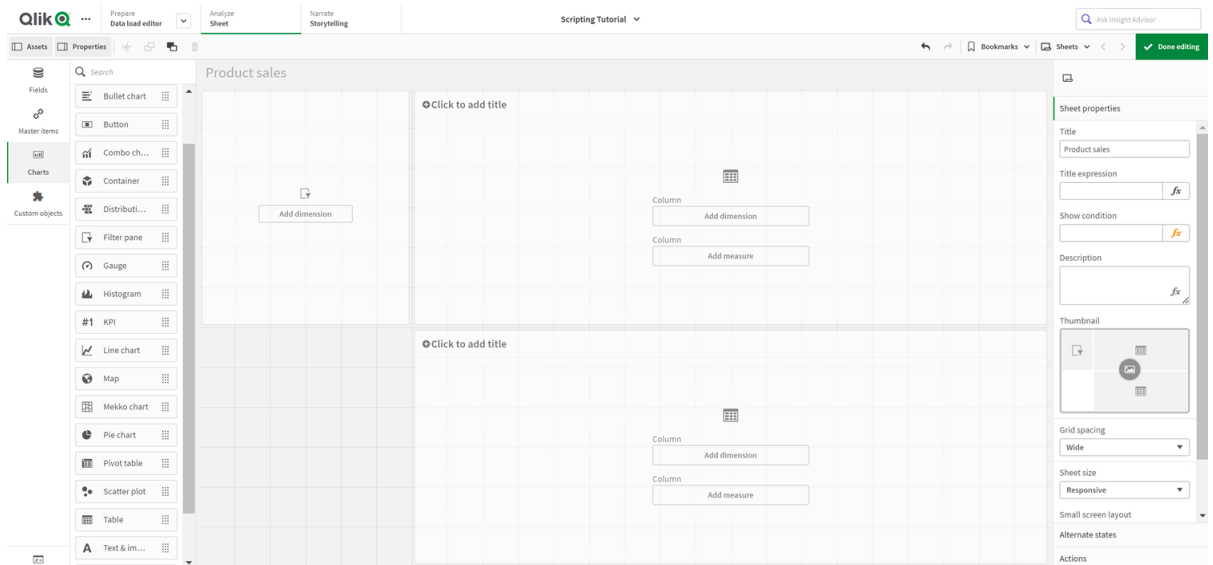
이제 데이터가 로드되었으면 데이터를 사용하여 몇 가지 차트를 만들 수 있습니다. 차트는 흔히 시각화라고도 합니다. 필요한 차원과 측정값을 추가할 때까지는 차트가 완료된 것이 아닙니다. 차트를 추가한 후 차원과 측정값을 계속 추가하여 시작할 수 있습니다.

다음과 같이 하십시오.

1. *Scripting Tutorial* 앱에서 새 시트 만들기
2. **차트** 탭에서 시트로 필터 창을 끌어서 놓고 핸들을 사용하여 3셀 폭, 4셀 높이가 되도록 크기를 조정합니다. 시트의 왼쪽 위 모서리에 배치합니다.
3. 막대형 차트를 오른쪽 위 모서리로 끌어서 놓고 시트 측면까지 늘여서 5셀 크기와 폭으로 만듭니다.
4. 꺾은선형 차트를 남은 공간으로 끌어서 놓습니다.

시트의 아이콘은 추가된 차트의 종류를 나타냅니다. 이제 차트에 차원과 측정값을 추가하여 시각화로 완료할 수 있습니다.

빈 차트가 있는 Qlik Sense 시트




11.2 차원 및 측정값 추가

다음 단계는 차원과 측정값을 추가하는 것입니다. 왼쪽 필터 창 위에 시간 차원을 추가하여 시작합니다. 필터 창은 공간을 절약하는 데 도움이 됩니다. *Year*, *Quarter*, *Month* 및 *Week*에 대해 각각 필터 창을 하나씩 갖는 대신, 같은 용도의 필터 창을 하나만 사용할 수 있습니다.

차원 생성 및 추가

다음과 같이 하십시오.


1. 왼쪽 자산 패널 맨 위에서  을 클릭하여 필드를 엽니다. 여기서 데이터 로드 편집기에서 로드한 모든 테이블의 모든 필드를 찾을 수 있습니다.
2. 목록 맨 아래로 스크롤하여 **Year** 필드를 클릭합니다. 왼쪽 위 필터 창의 가운데로 끌어서 놓습니다.
3. 같은 방법으로 **Quarter**, **Month** 및 **Week**를 필터 창에 추가합니다.
이제 **Year**, **Quarter**, **Month** 및 **Week**의 4개 차원이 있는 필터 창을 만들었습니다.

측정값 생성 및 추가

대부분의 시각화에는 차원과 측정값이 모두 필요합니다. 측정값은 집계 표현식의 결과로, 많은 경우에 **Sum**, **Max**, **Min**, **Avg**(평균) 또는 **Count** 등의 일반 함수입니다.


막대형 차트에 지역별 판매량이 표시됩니다.

다음과 같이 하십시오.

1. 필드  를 클릭합니다.
2. **Region** 필드를 클릭하고 막대형 차트 영역의 가운데로 끌어서 놓습니다.
3. **"Region"** 추가를 클릭합니다.
4. **Sales** 필드를 클릭하고 막대형 차트 영역의 가운데로 끌어서 놓습니다.
5. **측정값으로 추가 > Sum(Sales)**를 클릭합니다.
6. 맨 오른쪽 패널 창에서 **모양**을 클릭한 후 **프레젠테이션**을 클릭합니다. **가로**를 선택합니다.
이제 막대가 가로로 표시됩니다.
7. 맨 오른쪽 속성 패널에서 **정렬**을 클릭합니다.
정렬 순서가 표시됩니다.
8. 차원이 **Region**(차원 값, 사전순) 대신 **Sum([Sales])(측정값)**로 정렬되도록 **Sum([Sales])**를 **Region** 위로 끌어서 놓습니다.
막대형 차트가 완성되고 서로 다른 지역의 판매 결과가 표시됩니다. 이 차트가 기본 막대형 차트입니다. 속성 패널(오른쪽)에서 다양한 옵션을 사용하여 이 차트를 개선할 수 있습니다. 한 가지 개선 가능성을 보여주기 위해 단순한 제목보다 더 많은 것이 포함된 제목 영역을 사용하겠습니다.
9. 막대형 차트의 제목 필드에 다음을 붙여넣습니다.
`= 'Total sales: $' & Round(Sum(Sales)/1000000, 0.1) & 'M'`
10. Enter를 누릅니다.

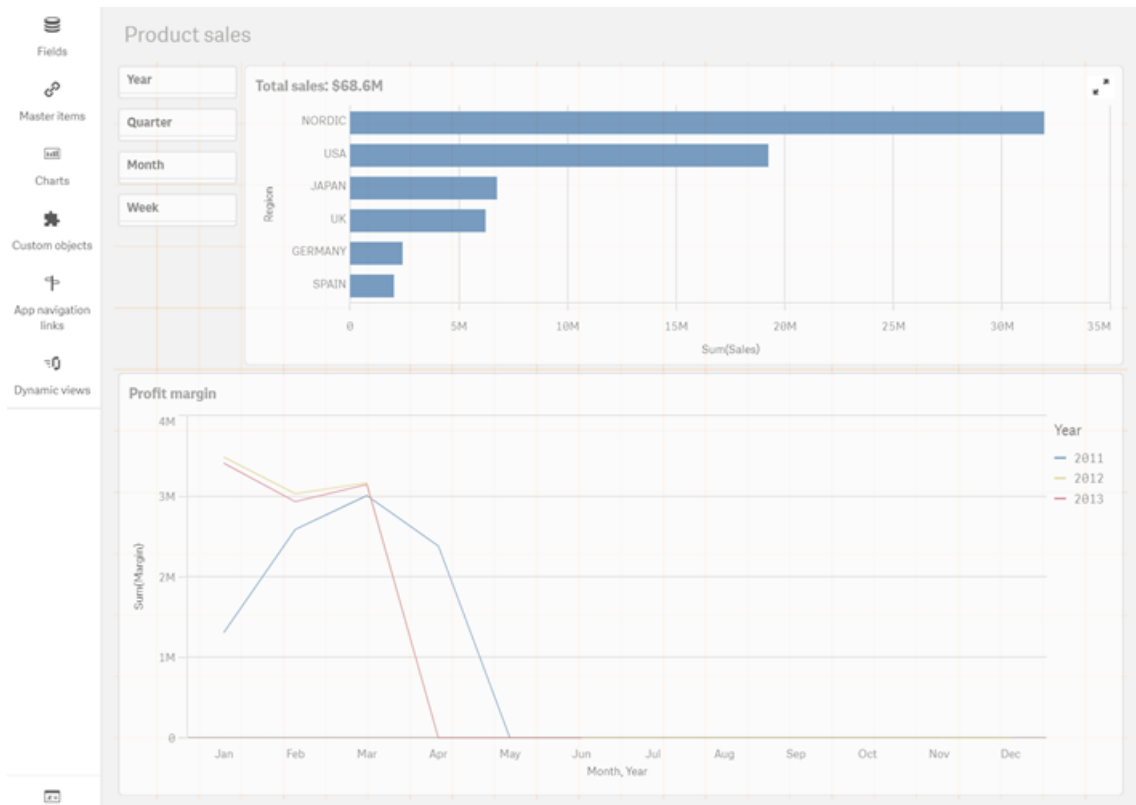
이 시트의 최종 시각화 결과는 꺾은선형 차트입니다.

다음과 같이 하십시오.

1. 필드  를 클릭합니다.
2. **Month** 필드를 클릭하고 꺾은선형 차트 영역의 가운데로 끌어서 놓습니다.

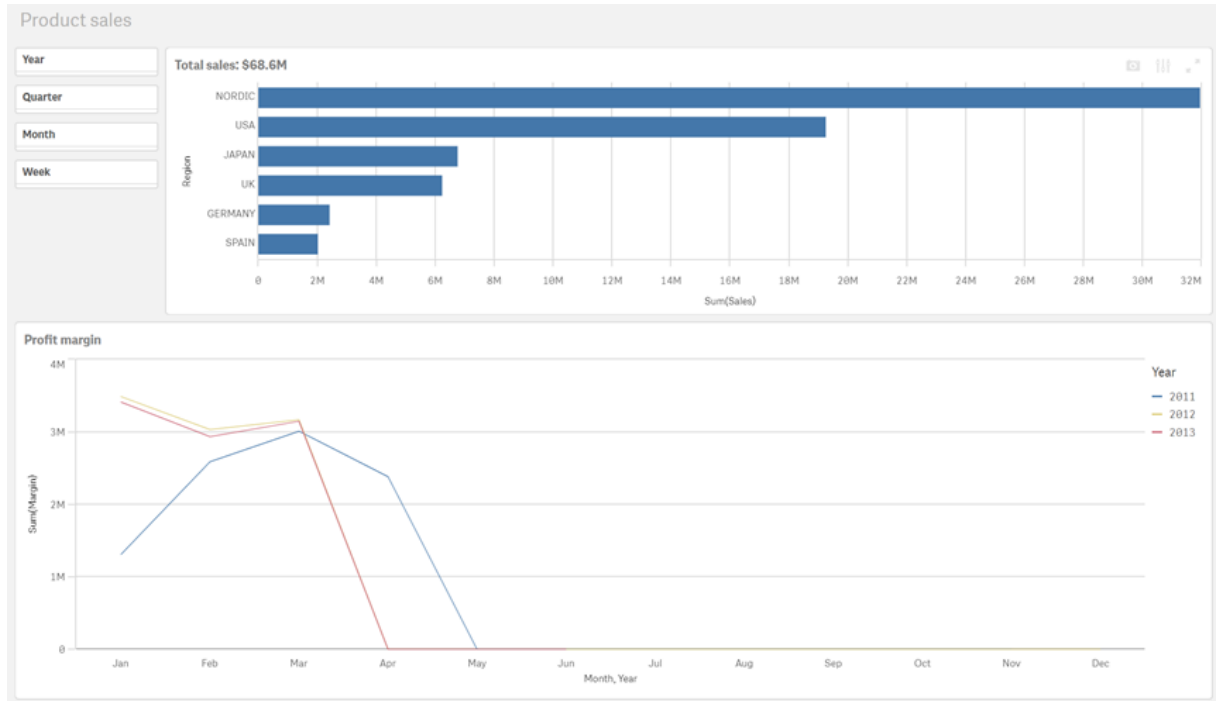
3. "Month" 추가를 클릭합니다.
4. Year 필드를 클릭하고 꺾은선형 차트 영역의 가운데로 끌어서 놓습니다.
5. "Year" 추가를 클릭합니다.
6. Margin 필드를 클릭하고 꺾은선형 차트 영역의 가운데로 끌어서 놓습니다.
7. 측정값으로 추가 > Sum(Margin)을 클릭합니다.
8. 꺾은선형 차트 맨 위에 제목으로 Profit margin을 추가합니다.

데이터가 있는 차트



9. 시트 편집을 중지합니다.
- 이제 시트가 완성되었으며 시트 주변을 클릭하고 시트 내용과 상호 작용을 시작할 수 있습니다.

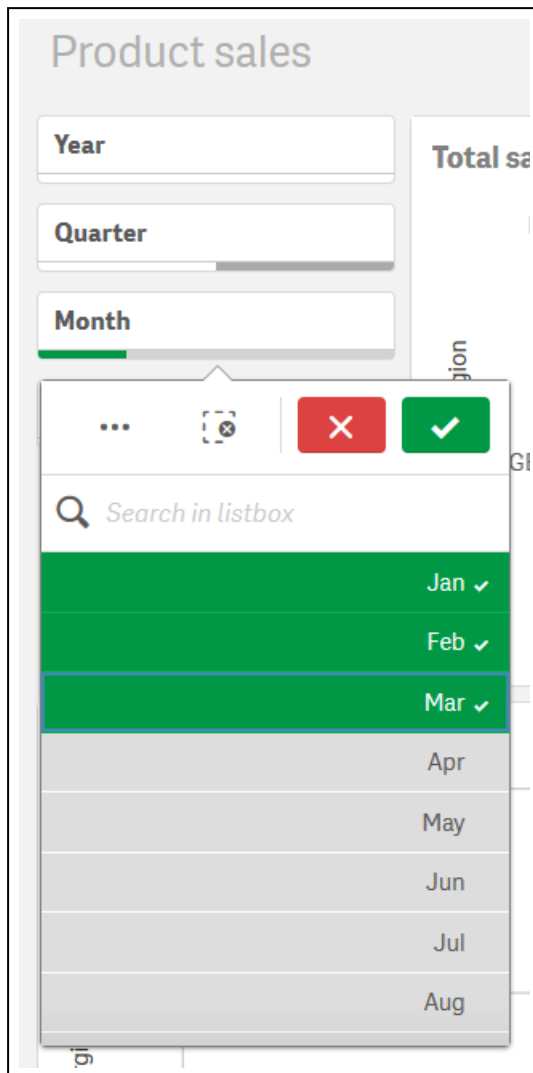
완료된 시트



원본 *Sales.xlsx* 파일에서 월별 판매 데이터의 양을 제한했기 때문에 각 연도의 3월 말 이후로는 차트에 제공되는 데이터가 거의 없습니다. 각 연도의 처음 3개월만 비교하도록 필터 창에서 선택할 수 있습니다.

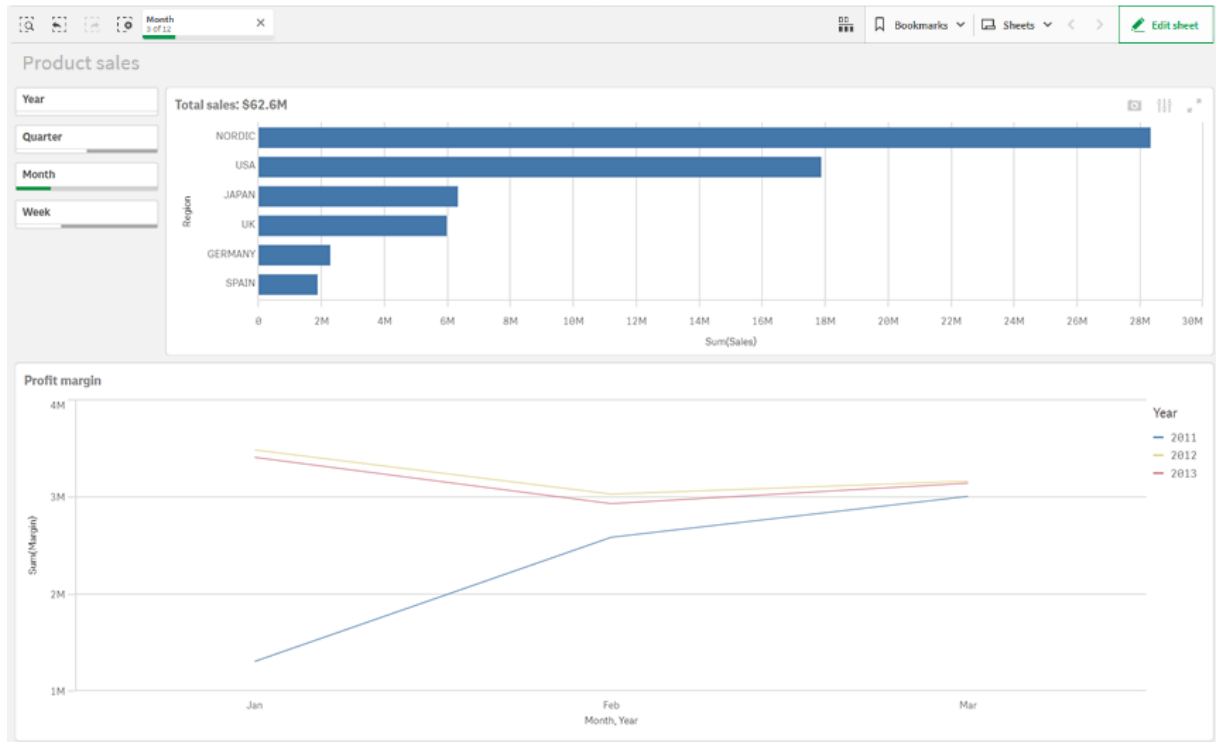
10. 필터 상자에서 *Month* 필드를 클릭한 다음 *Jan*, *Feb* 및 *Mar*을 선택합니다.

필터 상자



11. 필터 창을 닫습니다. *Profit margin* 차트에 각 연도의 처음 3개월에 대한 데이터가 표시됩니다.

선택 내용을 기반으로 업데이트된 Profit margin 차트



11.3 감사합니다.

이제 이 자습서가 끝났습니다. Qlik Sense에서 스크립트를 작성하는 데 필요한 기본 지식을 얻으셨기 바랍니다. 웹 사이트를 방문하면 앱에 대한 더욱 많은 영감을 얻을 수 있습니다.