Microsoft Power BI

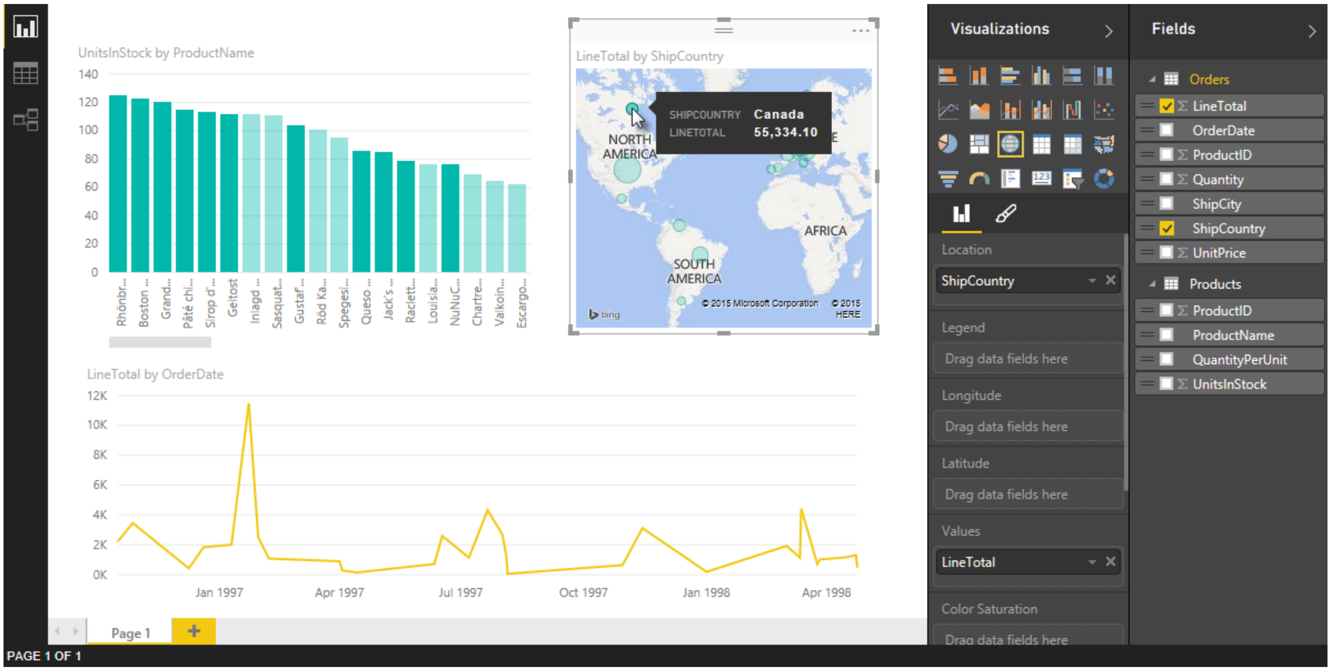
Tutorial: Analyzing sales data from Excel and an OData feed

With Power BI Desktop, you can connect to all sorts of different data sources, then combine and shape them in ways that facilitate making interesting, compelling data analysis and visualizations. In this tutorial, you'll learn how to combine data from two data sources.

It's common to have data spread across multiple data sources, such as product information in one database, and sales information in another. The techniques you'll learn in this document include an Excel workbook and an OData feed, but these techniques can be applied to other data sources too, like SQL Server queries, CSV files, or any data source in Power BI Desktop.

In this tutorial, you import data from Excel (it includes product information) and from an OData feed (which contains orders data). You'll perform transformation and aggregation steps, and combine data from both sources to produce a **Total Sales per Product and Year** report that includes interactive visualizations.

Here's what the final report will look like:



To follow the steps in this tutorial you need the Products workbook, which you can download: [click here to download **Products.xlsx**.]

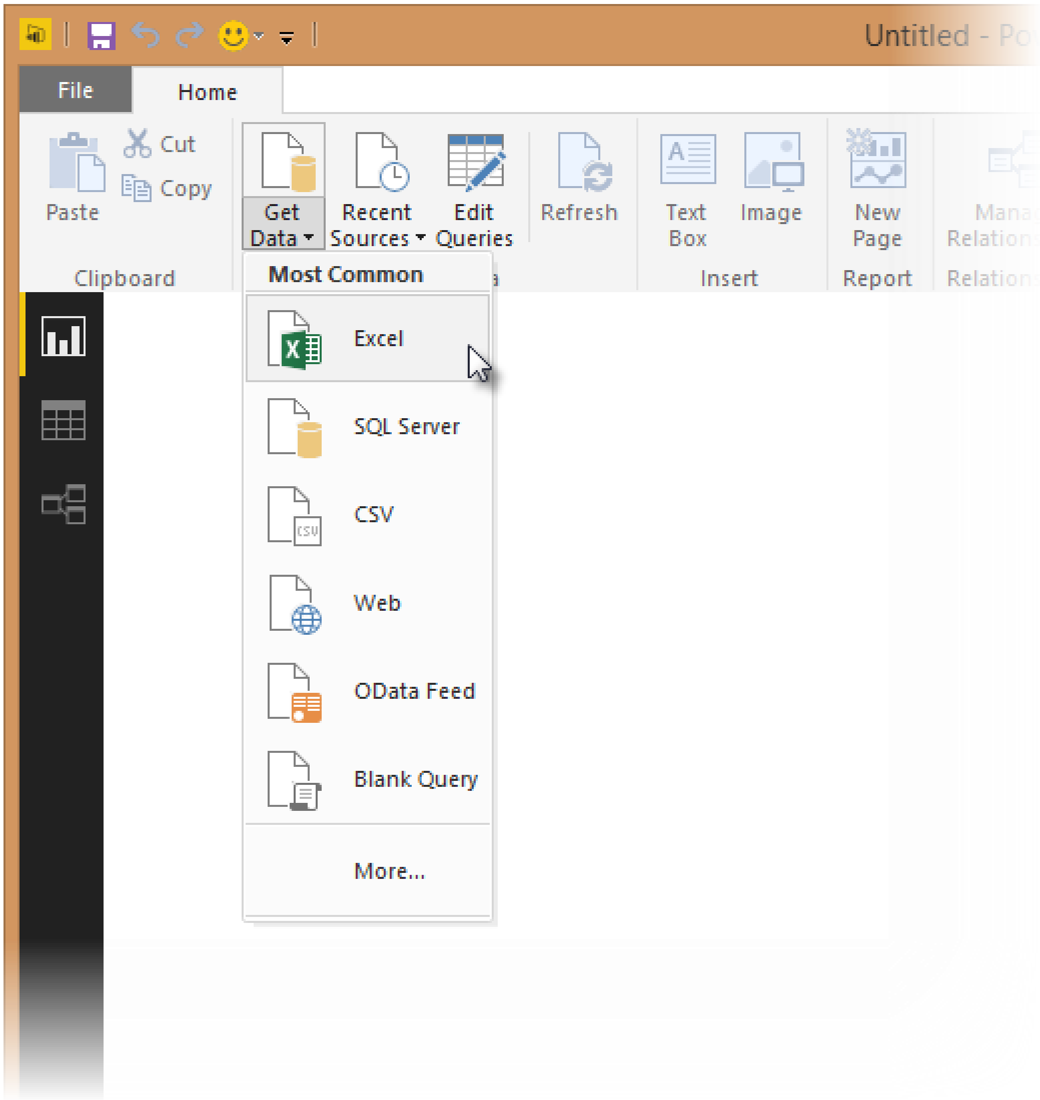
In the **Save As** dialog box, name the file **Products.xlsx**.

Task 1: Get product data from an Excel workbook

In this task, you import products from the Products.xlsx file into Power BI Desktop.

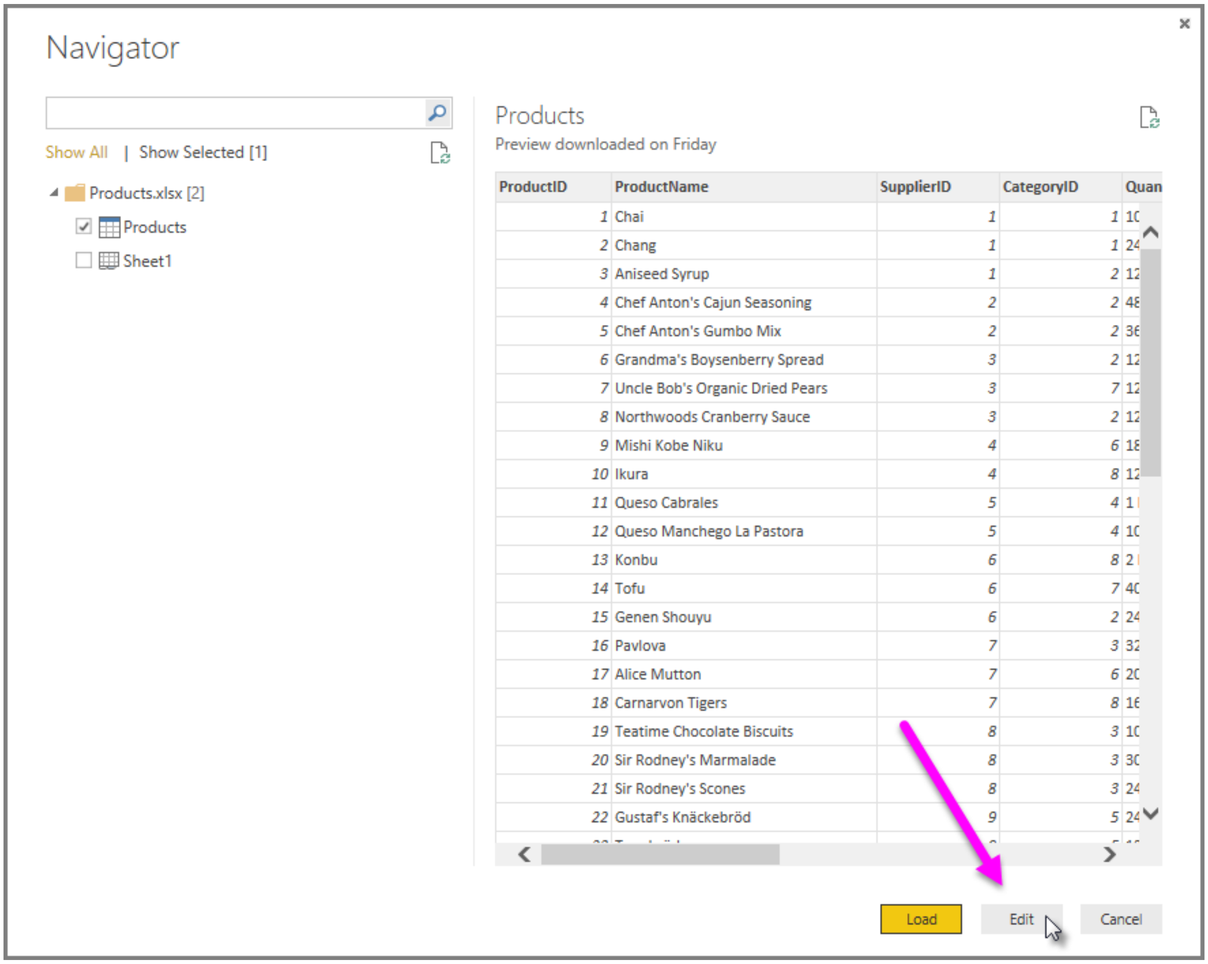
Step 1: Connect to an Excel workbook

1. Launch Power BI Desktop.
2. From the Home ribbon, select **Get Data**. Excel is one of the **Most Common** data connections, so you can select it directly from the **Get Data** menu.



3. If you select the Get Data button directly, you can also select **FIle > Excel** and select **Connect.**
4. In the **Open File** dialog box, select the **Products.xlsx** file.

5. In the **Navigator** pane, select the **Products** table and then select **Edit**.



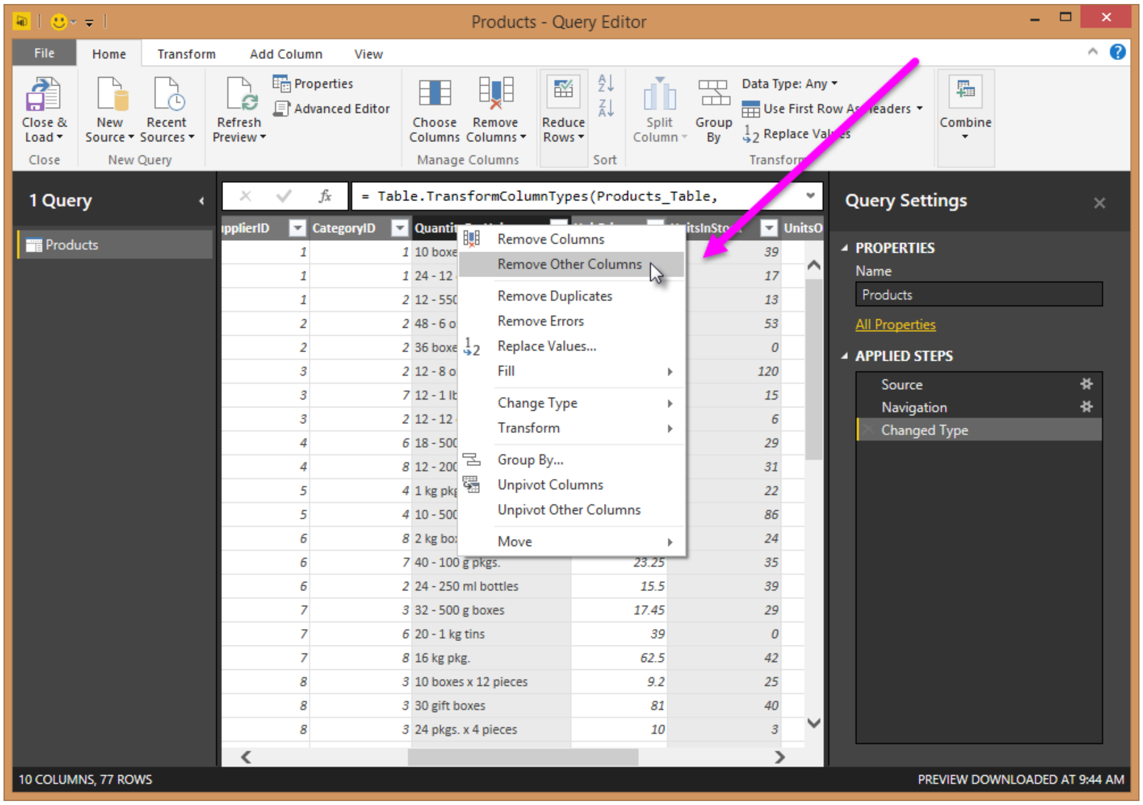
ProductID	ProductName	SupplierID	CategoryID	Quan
1	Chai	1	1	10
2	Chang	1	1	24
3	Aniseed Syrup	1	2	12
4	Chef Anton's Cajun Seasoning	2	2	48
5	Chef Anton's Gumbo Mix	2	2	36
6	Grandma's Boysenberry Spread	3	2	12
7	Uncle Bob's Organic Dried Pears	3	7	12
8	Northwoods Cranberry Sauce	3	2	12
9	Mishi Kobe Niku	4	6	18
10	Ikura	4	8	12
11	Queso Cabrales	5	4	1
12	Queso Manchego La Pastora	5	4	10
13	Konbu	6	8	2
14	Tofu	6	7	40
15	Genen Shouyu	6	2	24
16	Pavlova	7	3	32
17	Alice Mutton	7	6	20
18	Carnarvon Tigers	7	8	16
19	Teatime Chocolate Biscuits	8	3	10
20	Sir Rodney's Marmalade	8	3	30
21	Sir Rodney's Scones	8	3	24
22	Gustaf's Knäckebröd	9	5	24

Step 2: Remove other columns to only display columns of interest

In this step you remove all columns except **ProductID**, **ProductName**, **UnitsInStock**, and **QuantityPerUnit**. In Power BI Desktop, there are often a few ways to accomplish the same task. For example, many buttons in the ribbon can also be achieved by using the right-click menu on a column or a cell.

Power BI Desktop includes Query Editor, which is where you shape and transform your data connections. Query Editor opens automatically when you select **Edit** from **Navigator**. You can also open the Query Editor by selecting **Edit Queries** from the **Home** ribbon in Power BI Desktop. The following steps are performed in Query Editor.

1. In Query Editor, select the **ProductID**, **ProductName**, **QuantityPerUnit**, and **UnitsInStock** columns (use **Ctrl+Click** to select more than one column, or **Shift+Click** to select columns that are beside each other).
2. Select **Remove Columns** > **Remove Other Columns** from the ribbon, or right-click on a column header and click **Remove Other Columns**.



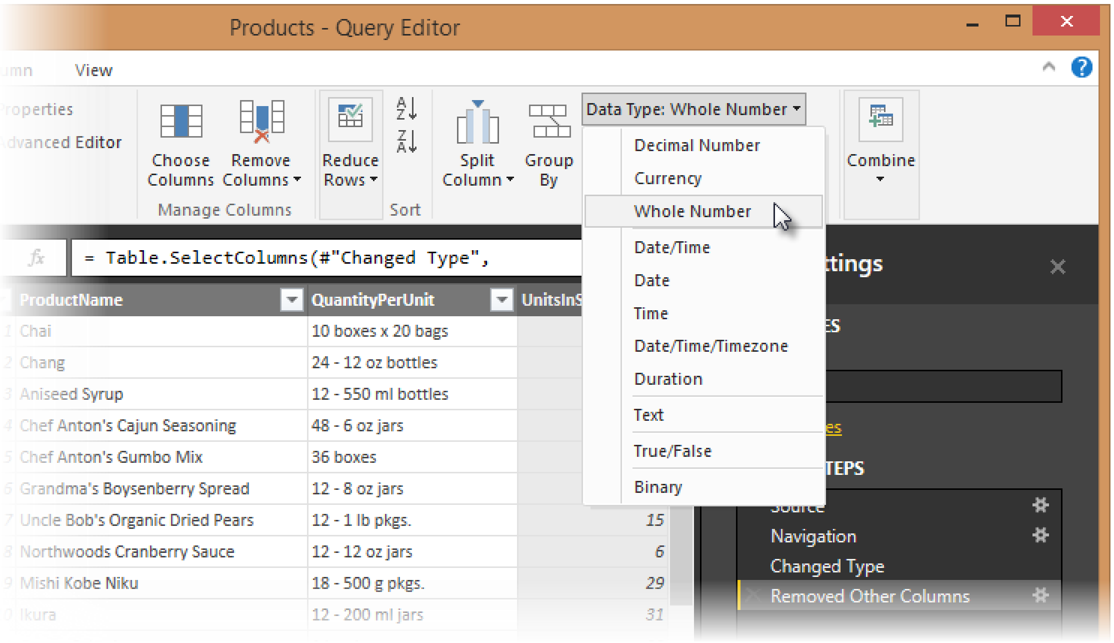
```
= Table.TransformColumnTypes(Products_Table,
```

upplierID	CategoryID	Quanti		itsInSto	UnitsO
1	1	10 boxe		39	
1	1	24 - 12		17	
1	2	12 - 550		13	
2	2	48 - 6 o		53	
2	2	36 boxe		0	
3	2	12 - 8 o		120	
3	7	12 - 1 lb		15	
3	2	12 - 12		6	
4	6	18 - 500		29	
4	8	12 - 200		31	
5	4	1 kg pkg		22	
5	4	10 - 500		86	
6	8	2 kg box		24	
6	7	40 - 100 g pkgs.	23.25	35	
6	2	24 - 250 ml bottles	15.5	39	
7	3	32 - 500 g boxes	17.45	29	
7	6	20 - 1 kg tins	39	0	
7	8	16 kg pkg.	62.5	42	
8	3	10 boxes x 12 pieces	9.2	25	
8	3	30 gift boxes	81	40	
8	3	24 pkgs. x 4 pieces	10	3	

Step 3: Change the data type of the UnitsInStock column

When Query Editor connects to data, it reviews each field and to determine the best data type. For the Excel workbook, products in stock will always be a whole number, so in this step you confirm the **UnitsInStock** column's datatype is Whole Number.

1. Select the **UnitsInStock** column.
2. Select the **Data Type** drop-down button in the **Home** ribbon.
3. If not already a Whole Number, select **Whole Number** for data type from the drop down (the **Data Type:** button also displays the data type for the current selection).



```
= Table.SelectColumns(#"Changed Type",
```

	ProductName	QuantityPerUnit	UnitsIn
1	Chai	10 boxes x 20 bags	
2	Chang	24 - 12 oz bottles	
3	Aniseed Syrup	12 - 550 ml bottles	
4	Chef Anton's Cajun Seasoning	48 - 6 oz jars	
5	Chef Anton's Gumbo Mix	36 boxes	
6	Grandma's Boysenberry Spread	12 - 8 oz jars	
7	Uncle Bob's Organic Dried Pears	12 - 1 lb pkgs.	15
8	Northwoods Cranberry Sauce	12 - 12 oz jars	6
9	Mishi Kobe Niku	18 - 500 g pkgs.	29
10	Ikura	12 - 200 ml jars	31

Power BI Desktop steps created

As you perform query activities in Query Editor, query steps are created and listed in the **Query Settings** pane, in the **Applied Steps** list. Each query step has a corresponding formula, also known as the "M" language. For more information about the “M” formula language, see [Learn about Power BI formulas](#).

Task	Query step	Formula
Connect to an Excel workbook	Source	Source{[Name="Products"]}[Data]
Promote the first row to table column headers	FirstRowAsHeader	[Table.PromoteHeaders](#) (Products)
Remove other columns to only display columns of interest	RemovedOtherColumns	[Table.SelectColumns](#) (FirstRowAsHeader,{"ProductID", "ProductName", "QuantityPerUnit", "UnitsInStock"})
Change datatype	Changed Type	Table.TransformColumnTypes(#"Removed Other Columns",{{"UnitsInStock", Int64.Type}})

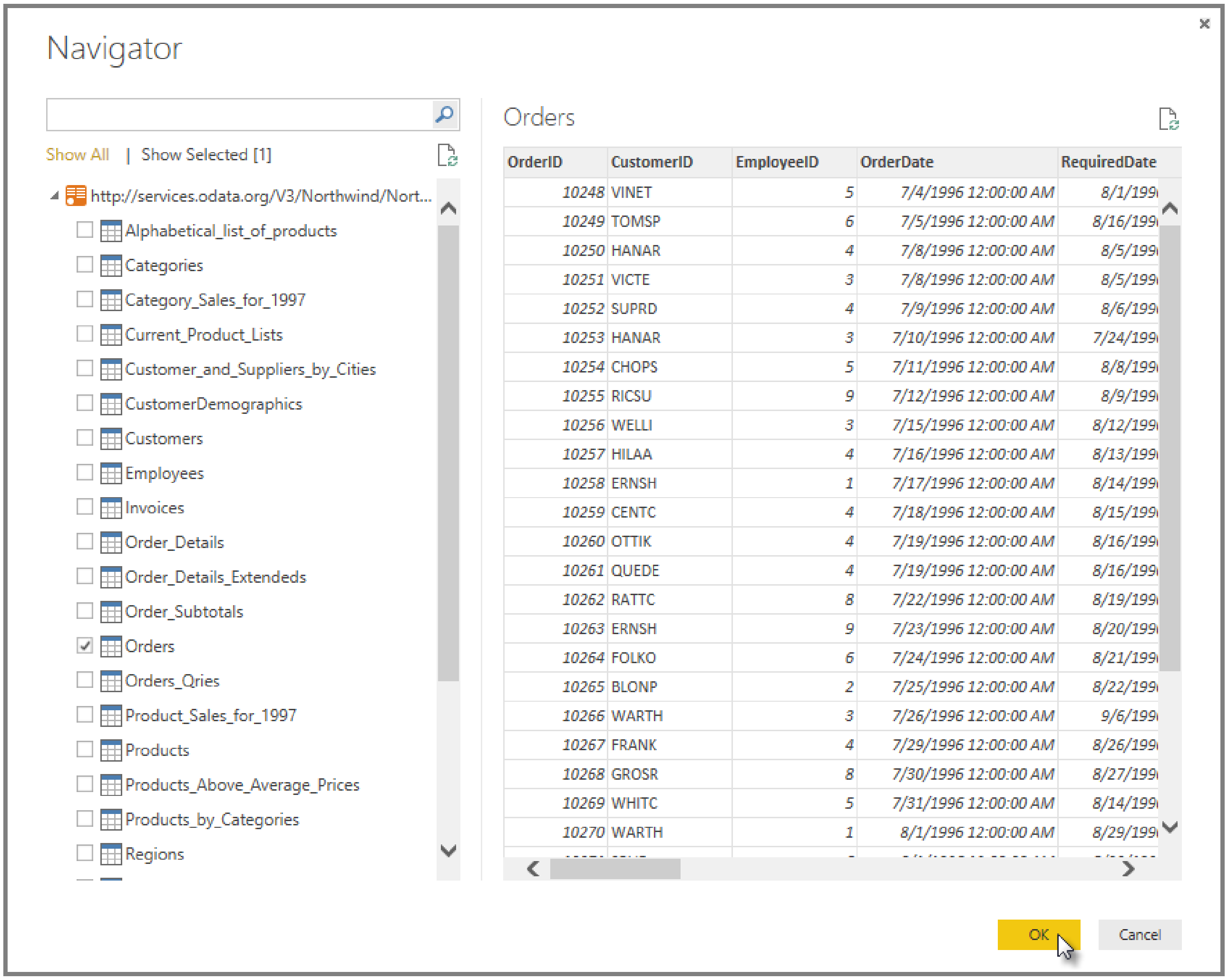
Task 2: Import order data from an OData feed

In this task, you'll bring in order data. This step represents connecting to a sales system. You import data into Power BI Desktop from the sample Northwind OData feed at the following URL, which you can copy (and then paste) in the steps below: http://services.odata.org/V3/Northwind/Northwind.svc/

Step 1: Connect to an OData feed

1. From the **Home** ribbon tab in Query Editor, select **Get Data.**
2. Browse to the **OData Feed** data source.
3. In the **OData Feed** dialog box, paste the **URL** for the Northwind OData feed.
4. Select **OK**.

5. In the **Navigator** pane, select the **Orders** table, and then select **Edit**.



http://services.odata.org/V3/Northwind/Nort...

- □ Alphabetical_list_of_products
- □ Categories
- □ Category_Sales_for_1997
- □ Current_Product_Lists
- □ Customer_and_Suppliers_by_Cities
- □ CustomerDemographics
- □ Customers
- □ Employees
- □ Invoices
- □ Order_Details
- □ Order_Details_Extendeds
- □ Order_Subtotals
- ☑ Orders
- □ Orders_Qries
- □ Product_Sales_for_1997
- □ Products
- □ Products_Above_Average_Prices
- □ Products_by_Categories
- □ Regions

Orders

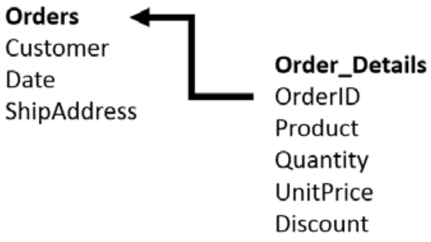
OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate
10248	VINET	5	7/4/1996 12:00:00 AM	8/1/199
10249	TOMSP	6	7/5/1996 12:00:00 AM	8/16/199
10250	HANAR	4	7/8/1996 12:00:00 AM	8/5/199
10251	VICTE	3	7/8/1996 12:00:00 AM	8/5/199
10252	SUPRD	4	7/9/1996 12:00:00 AM	8/6/199
10253	HANAR	3	7/10/1996 12:00:00 AM	7/24/199
10254	CHOPS	5	7/11/1996 12:00:00 AM	8/8/199
10255	RICSU	9	7/12/1996 12:00:00 AM	8/9/199
10256	WELLI	3	7/15/1996 12:00:00 AM	8/12/199
10257	HILAA	4	7/16/1996 12:00:00 AM	8/13/199
10258	ERNSH	1	7/17/1996 12:00:00 AM	8/14/199
10259	CENTC	4	7/18/1996 12:00:00 AM	8/15/199
10260	OTTIK	4	7/19/1996 12:00:00 AM	8/16/199
10261	QUEDE	4	7/19/1996 12:00:00 AM	8/16/199
10262	RATTC	8	7/22/1996 12:00:00 AM	8/19/199
10263	ERNSH	9	7/23/1996 12:00:00 AM	8/20/199
10264	FOLKO	6	7/24/1996 12:00:00 AM	8/21/199
10265	BLONP	2	7/25/1996 12:00:00 AM	8/22/199
10266	WARTH	3	7/26/1996 12:00:00 AM	9/6/199
10267	FRANK	4	7/29/1996 12:00:00 AM	8/26/199
10268	GROSR	8	7/30/1996 12:00:00 AM	8/27/199
10269	WHITC	5	7/31/1996 12:00:00 AM	8/14/199
10270	WARTH	1	8/1/1996 12:00:00 AM	8/29/199

Note You can click a table name, without selecting the checkbox, to see a preview.

Step 2: Expand the Order_Details table

The **Orders** table contains a reference to a **Details** table, which contains the individual products that were included in each Order. When you connect to data sources with multiples tables (such as a relational database) you can use these references to build up your query.

In this step, you expand the **Order_Details** table that is related to the **Orders** table, to combine the **ProductID**, **UnitPrice**, and **Quantity** columns from **Order_Details** into the **Orders** table. This is a representation of the data in these tables:

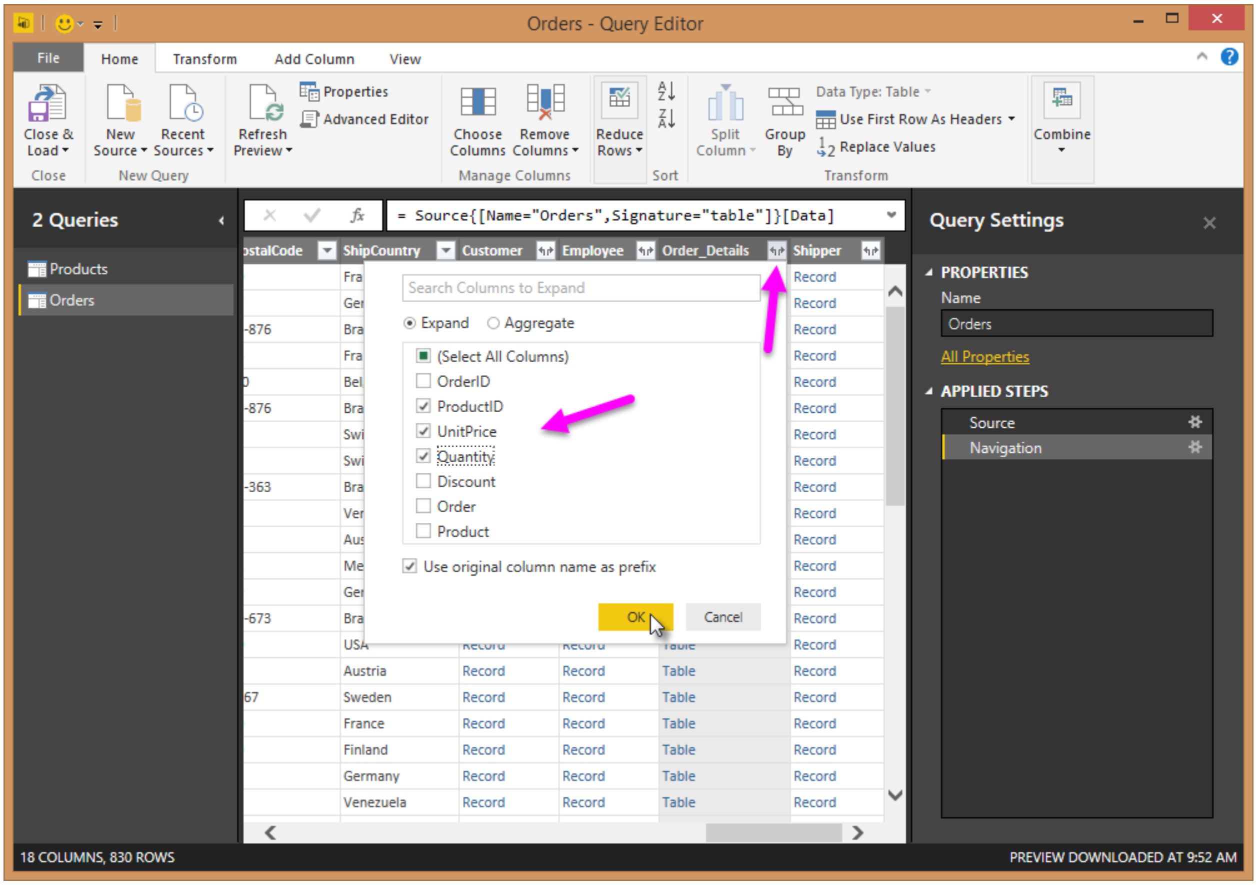


The **Expand** operation combines columns from a related table into a subject table. When the query runs, rows from the related table (**Order_Details**) are combined into rows from the subject table (**Orders**).

After you expand the **Order_Details** table, three new columns and additional rows are added to the **Orders** table, one for each row in the nested or related table.

1. In the **Query View**, scroll to the **Order_Details** column.
2. In the **Order_Details** column, select the expand icon (↔).
3. In the **Expand** drop-down:
   a. Select **(Select All Columns)** to clear all columns.
   b. Select **ProductID**, **UnitPrice**, and **Quantity**.

c. Click **OK**.



```
= Source{[Name="Orders",Signature="table"]}[Data]
```

Name: Orders

Search Columns to Expand

● Expand ○ Aggregate

- ■ (Select All Columns)
- ☐ OrderID
- ☑ ProductID
- ☑ UnitPrice
- ☑ Quantity
- ☐ Discount
- ☐ Order
- ☐ Product

☑ Use original column name as prefix

18 COLUMNS, 830 ROWS

PREVIEW DOWNLOADED AT 9:52 AM

Step 3: Remove other columns to only display columns of interest

In this step you remove all columns except **OrderDate, ShipCity**, **ShipCountry**, **Order_Details.ProductID**, **Order_Details.UnitPrice**, and **Order_Details.Quantity** columns. In the previous task, you used **Remove Other Columns**. For this task, you remove selected columns.

1. In the **Query View**, select all columns by completing a. and b.:
   a. Click the first column (**OrderID**).

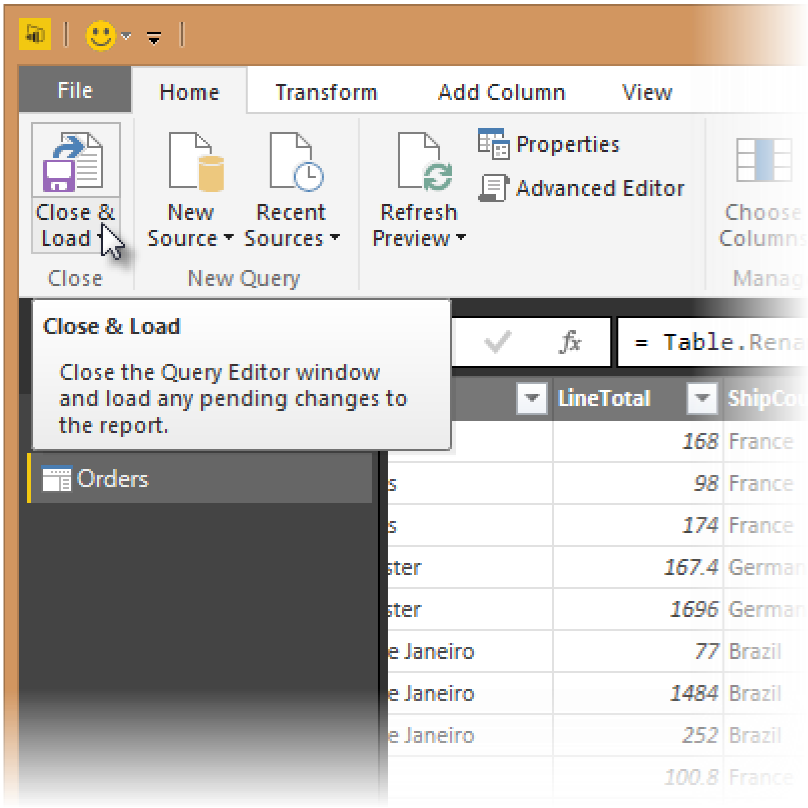
    b. Shift+Click the last column (**Shipper**).
    c. Now that all columns are selected, use Ctrl+Click to unselect the following columns: **OrderDate**, **ShipCity**, **ShipCountry**, **Order_Details.ProductID**, **Order_Details.UnitPrice**, and **Order_Details.Quantity**.
2. Now that only the columns we want to remove are selected, right-click on any selected column header and click **Remove Columns**.

Step 4: Calculate the line total for each Order_Details row

Power BI Desktop lets you to create calculations based on the columns you are importing, so you can enrich the data that you connect to. In this step, you create a **Custom Column** to calculate the line total for each **Order_Details** row.

Calculate the line total for each **Order_Details** row:

1. In the **Add Column** ribbon tab, click **Add Custom Column**.



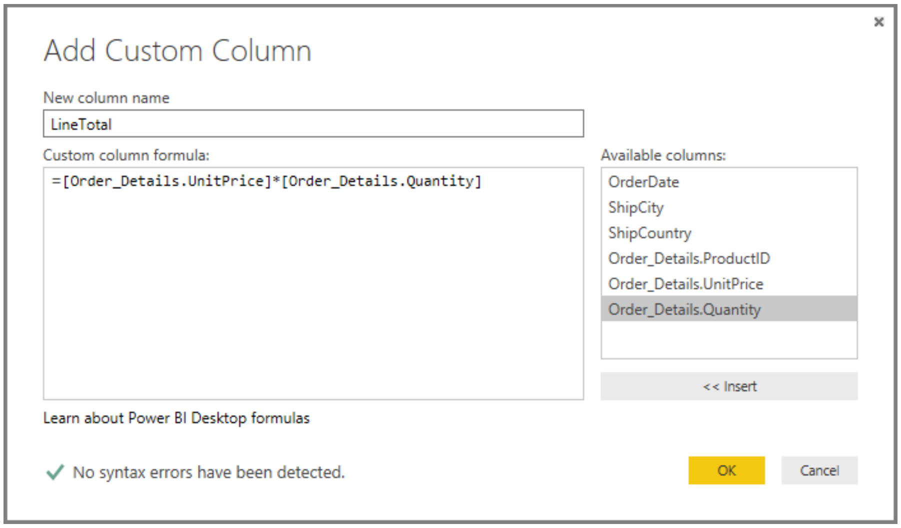
Close & Load

Close the Query Editor window and load any pending changes to the report.

```
= Table.Rena
```

	LineTotal	ShipCo
	168	France
s	98	France
s	174	France
ster	167.4	German
ster	1696	German
e Janeiro	77	Brazil
e Janeiro	1484	Brazil
e Janeiro	252	Brazil
	100.8	France

2. In the **Add Custom Column** dialog box, in the **Custom Column Formula** textbox, enter **[Order_Details.UnitPrice] * [Order_Details.Quantity]**.
3. In the **New column name** textbox, enter **LineTotal**.



Add Custom Column

New column name

LineTotal

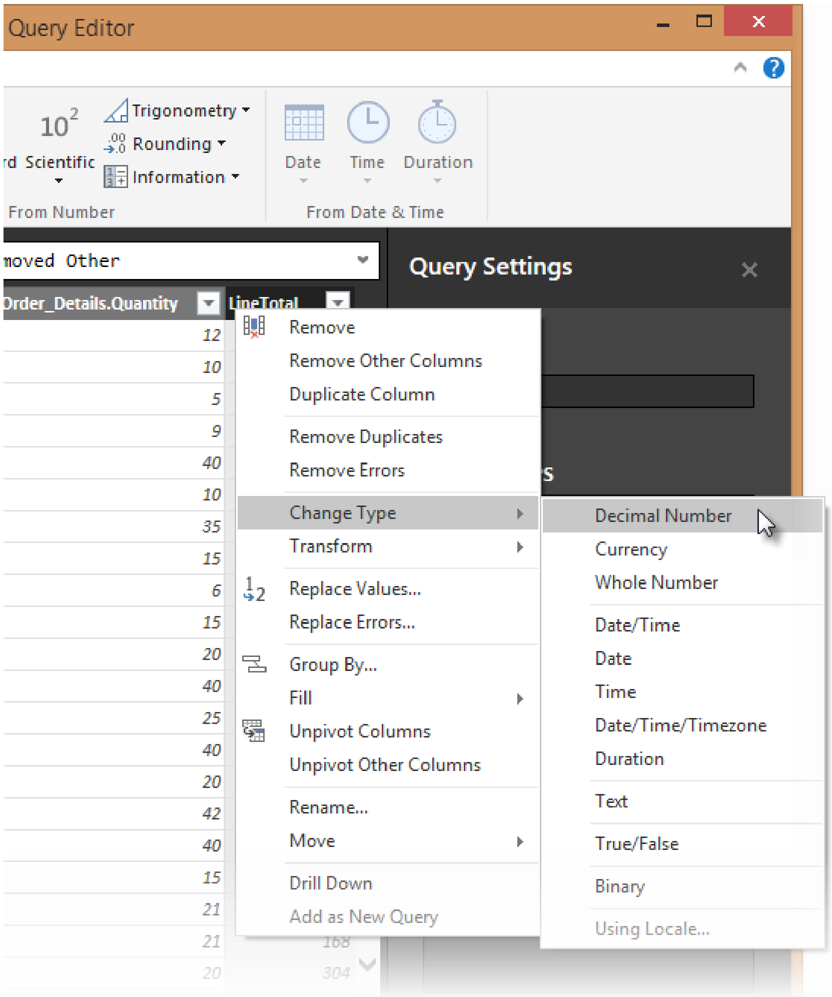
Custom column formula:

```
=[Order_Details.UnitPrice]*[Order_Details.Quantity]
```

4. Click **OK**.

Step 5: Set the datatype of the LineTotal field

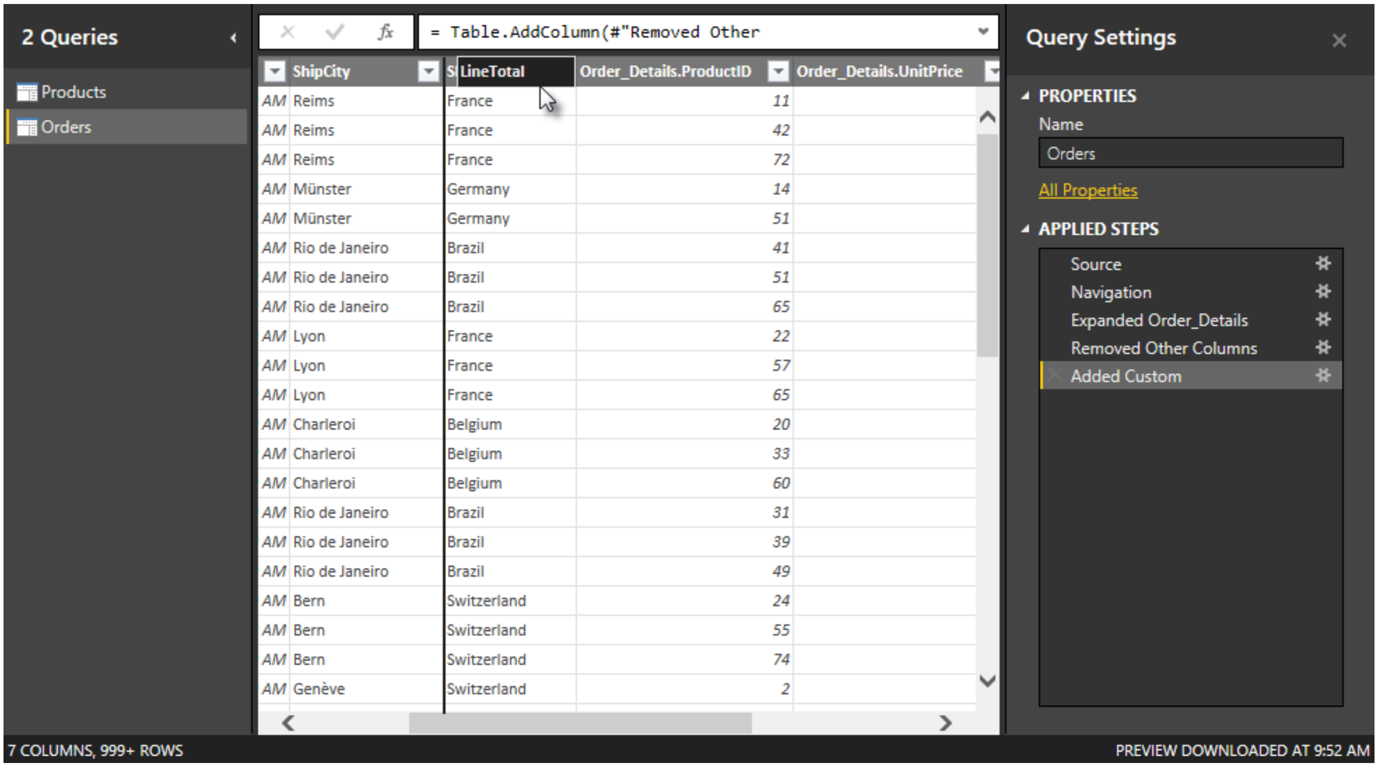
1. Right click the **LineTotal** column.
2. Select **Change Type** and choose **Decimal Number.**



Step 6: Rename and reorder columns in the query

In this step you finish making the model easy to work with when creating reports, by renaming the final columns and changing their order.

1. In **Query Editor**, drag the **LineTotal** column to the left, after **ShipCountry**.



```
= Table.AddColumn(#"Removed Other
```

Name: Orders

ShipCity	S LineTotal	Order_Details.ProductID	Order_Details.UnitPrice
Reims	France	11	
Reims	France	42	
Reims	France	72	
Münster	Germany	14	
Münster	Germany	51	
Rio de Janeiro	Brazil	41	
Rio de Janeiro	Brazil	51	
Rio de Janeiro	Brazil	65	
Lyon	France	22	
Lyon	France	57	
Lyon	France	65	
Charleroi	Belgium	20	
Charleroi	Belgium	33	
Charleroi	Belgium	60	
Rio de Janeiro	Brazil	31	
Rio de Janeiro	Brazil	39	
Rio de Janeiro	Brazil	49	
Bern	Switzerland	24	
Bern	Switzerland	55	
Bern	Switzerland	74	
Genève	Switzerland	2	

7 COLUMNS, 999+ ROWS

PREVIEW DOWNLOADED AT 9:52 AM

2. Remove the *Order_Details.* prefix from the **Order_Details.ProductID**, **Order_Details.UnitPrice** and **Order_Details.Quantity** columns, by double-clicking on each column header, and then deleting that text from the column name.

Power BI Desktop steps created

As you perform query activities in Query Editor, query steps are created and listed in the **Query Settings** pane, in the **Applied Steps** list. Each query step has a corresponding Power Query formula, also known as the "M" language. For more information about this formula language, see Learn about Power BI formulas.

Task	Query step	Formula
Connect to an OData feed	Source	Source{[Name="Orders"]}[Data]
Expand the Order_Details table	Expand Order_Details	Table.ExpandTableColumn (Orders, "Order_Details", {"ProductID", "UnitPrice", "Quantity"}, {"Order_Details.ProductID", "Order_Details.UnitPrice", "Order_Details.Quantity"})
Remove other columns to only display columns of interest	RemovedColumns	Table.RemoveColumns (#"Expand Order_Details",{"OrderID", "CustomerID", "EmployeeID", "RequiredDate", "ShippedDate", "ShipVia", "Freight", "ShipName", "ShipAddress", "ShipCity", "ShipRegion", "ShipPostalCode", "ShipCountry", "Customer", "Employee", "Shipper"})
Calculate the line total for each Order_Details row	InsertedColumns	Table.AddColumn (RemovedColumns, "Custom", each [Order_Details.UnitPrice] * [Order_Details.Quantity])

Task 3: Combine the Products and Total Sales queries

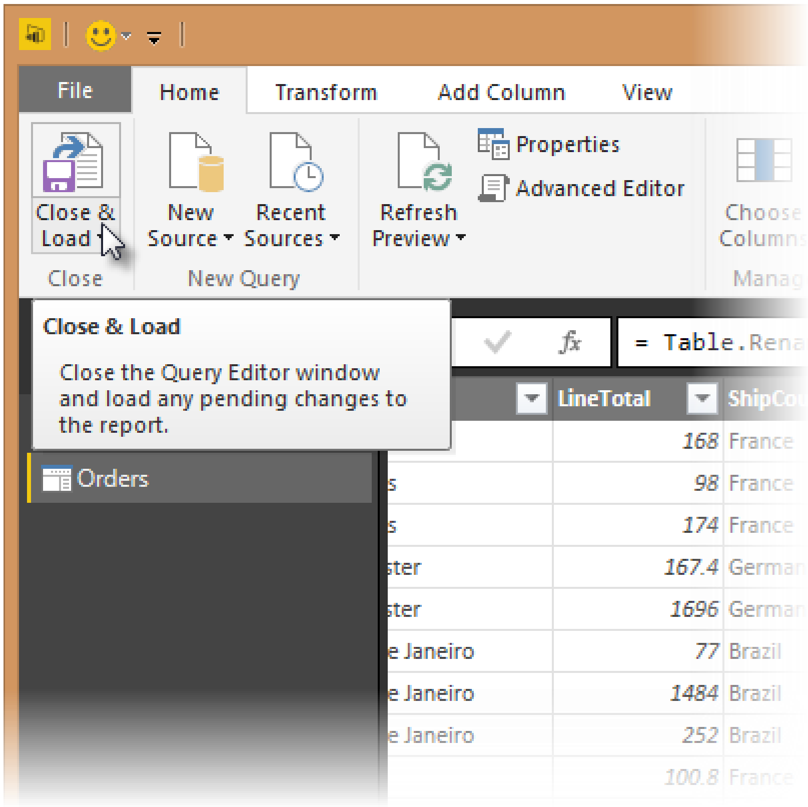
Power BI Desktop does not require you to combine queries to report on them. Instead, you can create **Relationships** between datasets. These relationships can be created on any column that is common to your datasets. For more information see Create and manage relationships.

In this tutorial, we have Orders and Products data that share a common 'ProductID' field, so we need to ensure there's a relationship between them in the model we're using with Power BI Desktop. Simply specify in Power BI Desktop that the columns from each table are related (i.e. columns that have the same values). Power BI Desktop works out the direction and cardinality of the relationship for you. In some cases, it will even detect the relationships automatically.

In this task, you confirm that a relationship is established in Power BI Desktop between the **Products** and **Total Sales** queries.

Step 1: Confirm the relationship between Products and Total Sales

1. First, we need to load the model that we created in Query Editor into Power BI Desktop. From the **Home** ribbon of Query Editor, select **Close & Load**.



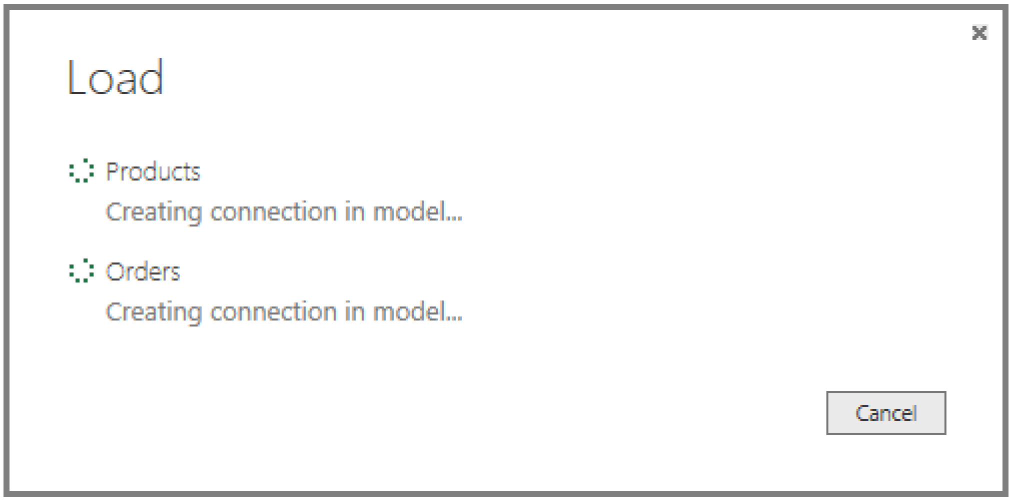
Close & Load

Close the Query Editor window and load any pending changes to the report.

```
= Table.Rena
```

	LineTotal	ShipCo
	168	France
s	98	France
s	174	France
ster	167.4	German
ster	1696	German
e Janeiro	77	Brazil
e Janeiro	1484	Brazil
e Janeiro	252	Brazil
	100.8	France

2. Power BI Desktop loads the data from the two queries.



Load

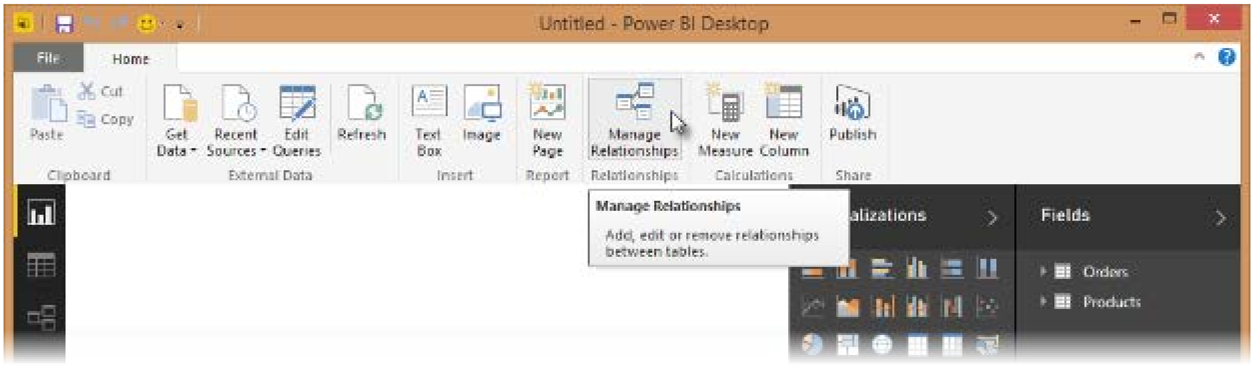
Products

Creating connection in model...

Orders

Creating connection in model...

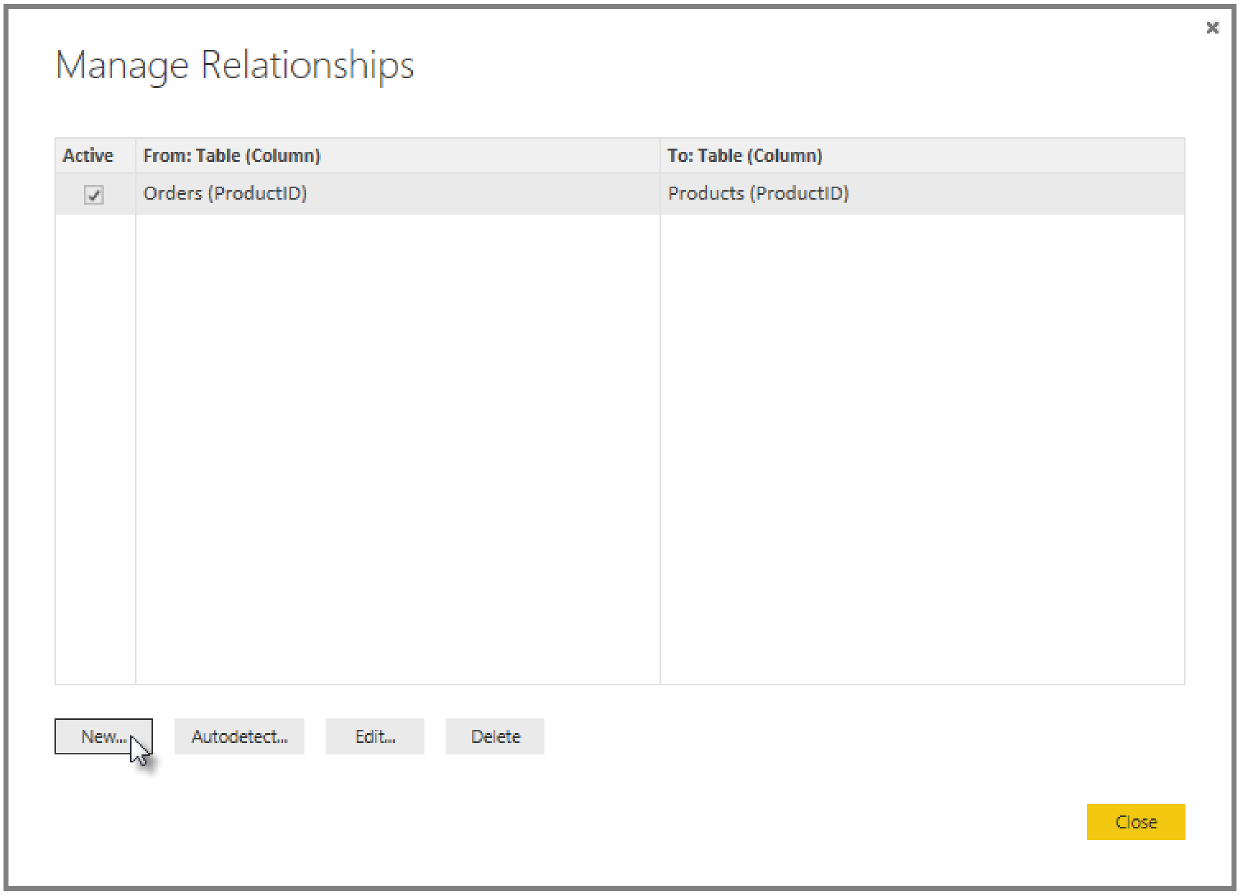
3. Once the data is loaded, select the **Manage Relationships** button **Home** ribbon.



Manage Relationships

Add, edit or remove relationships between tables.

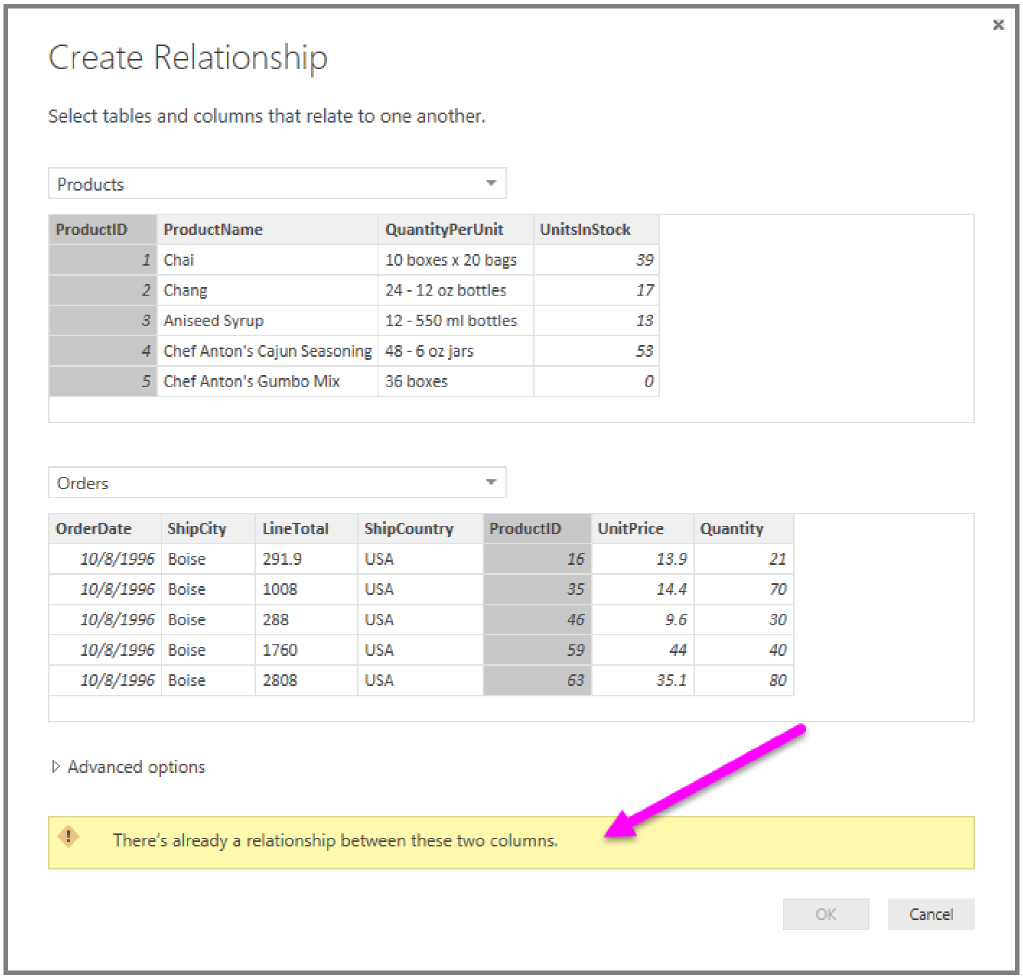
4. Select the **New…** button



Manage Relationships

Active	From: Table (Column)	To: Table (Column)
☑	Orders (ProductID)	Products (ProductID)

5. When we attempt to create the relationship, we see that one already exists! As shown in the **Create Relationship** dialog (by the shaded columns), the **ProductsID** fields in each query already have an established relationship.



Create Relationship

Select tables and columns that relate to one another.

Products

ProductID	ProductName	QuantityPerUnit	UnitsInStock
1	Chai	10 boxes x 20 bags	39
2	Chang	24 - 12 oz bottles	17
3	Aniseed Syrup	12 - 550 ml bottles	13
4	Chef Anton's Cajun Seasoning	48 - 6 oz jars	53
5	Chef Anton's Gumbo Mix	36 boxes	0

Orders

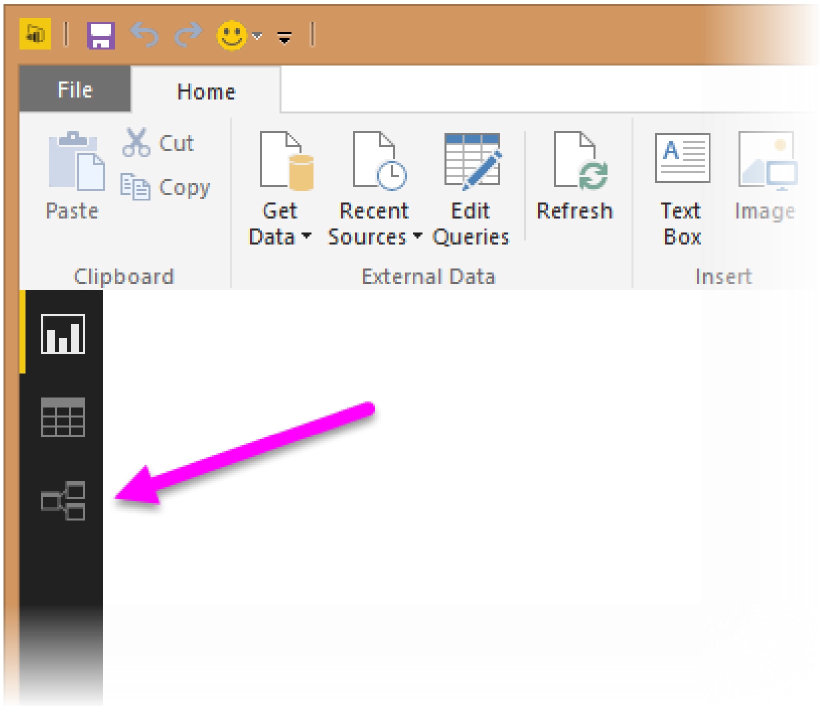
OrderDate	ShipCity	LineTotal	ShipCountry	ProductID	UnitPrice	Quantity
10/8/1996	Boise	291.9	USA	16	13.9	21
10/8/1996	Boise	1008	USA	35	14.4	70
10/8/1996	Boise	288	USA	46	9.6	30
10/8/1996	Boise	1760	USA	59	44	40
10/8/1996	Boise	2808	USA	63	35.1	80

▷ Advanced options

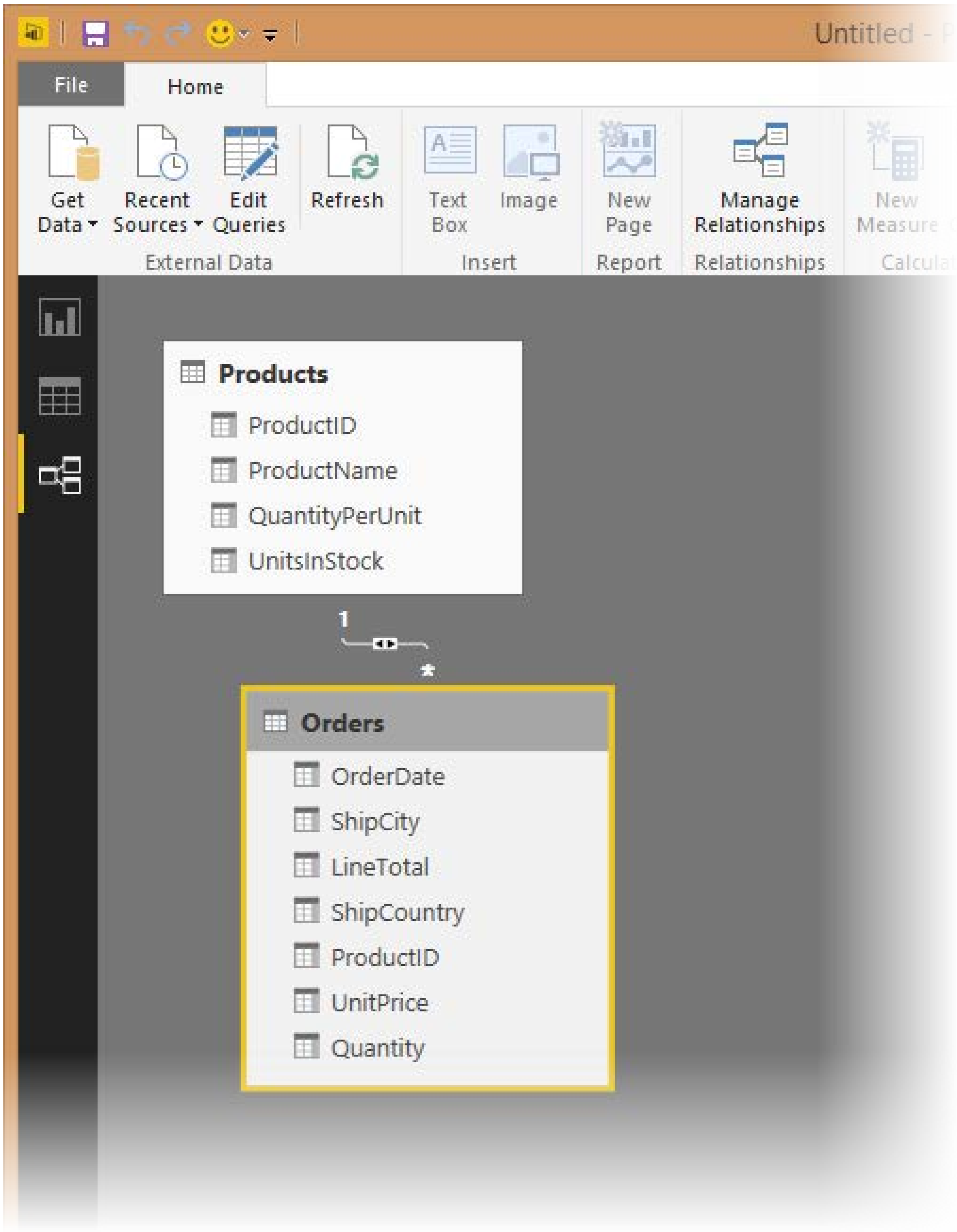
There's already a relationship between these two columns.

OK Cancel

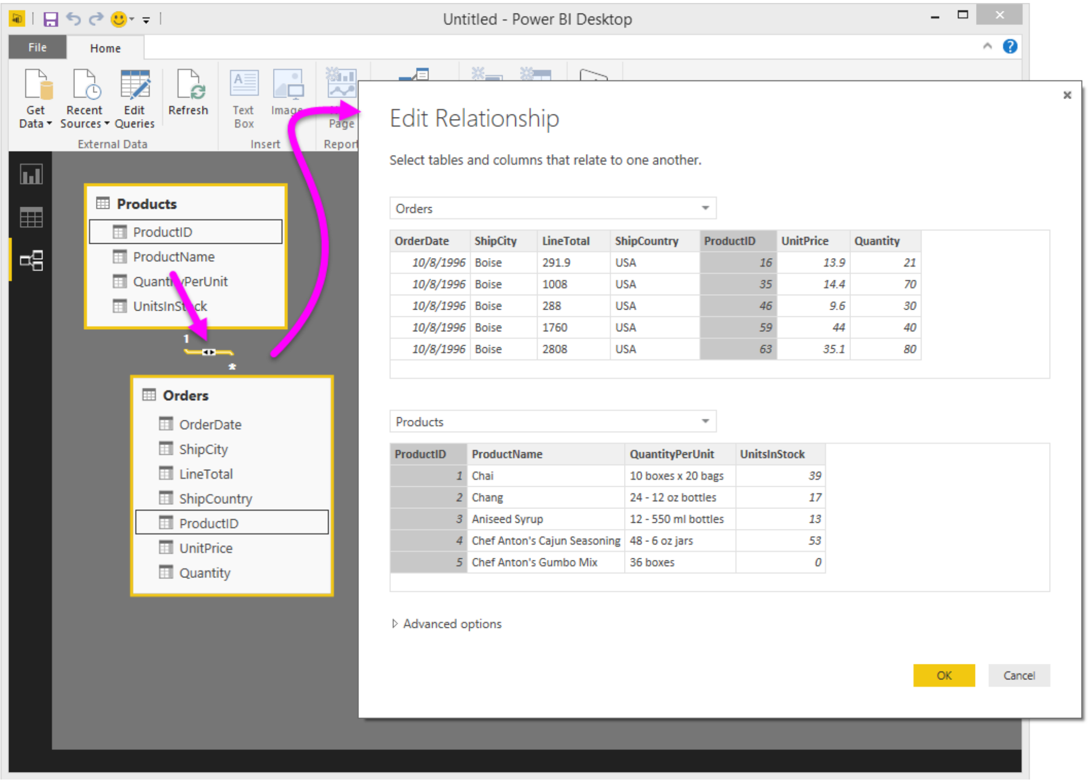
6. Select **Cancel**, and then select **Relationship** view in Power BI Desktop.



7. We see the following, which visualizes the relationship between the queries.



7. When you double-click the arrow on the line that connects the to queries, an **Edit Relationship** dialog appears.



Edit Relationship

Select tables and columns that relate to one another.

Orders

OrderDate	ShipCity	LineTotal	ShipCountry	ProductID	UnitPrice	Quantity
10/8/1996	Boise	291.9	USA	16	13.9	21
10/8/1996	Boise	1008	USA	35	14.4	70
10/8/1996	Boise	288	USA	46	9.6	30
10/8/1996	Boise	1760	USA	59	44	40
10/8/1996	Boise	2808	USA	63	35.1	80

Products

ProductID	ProductName	QuantityPerUnit	UnitsInStock
1	Chai	10 boxes x 20 bags	39
2	Chang	24 - 12 oz bottles	17
3	Aniseed Syrup	12 - 550 ml bottles	13
4	Chef Anton's Cajun Seasoning	48 - 6 oz jars	53
5	Chef Anton's Gumbo Mix	36 boxes	0

8. No need to make any changes, so we'll just select **Cancel** to close the **Edit Relationship** dialog.

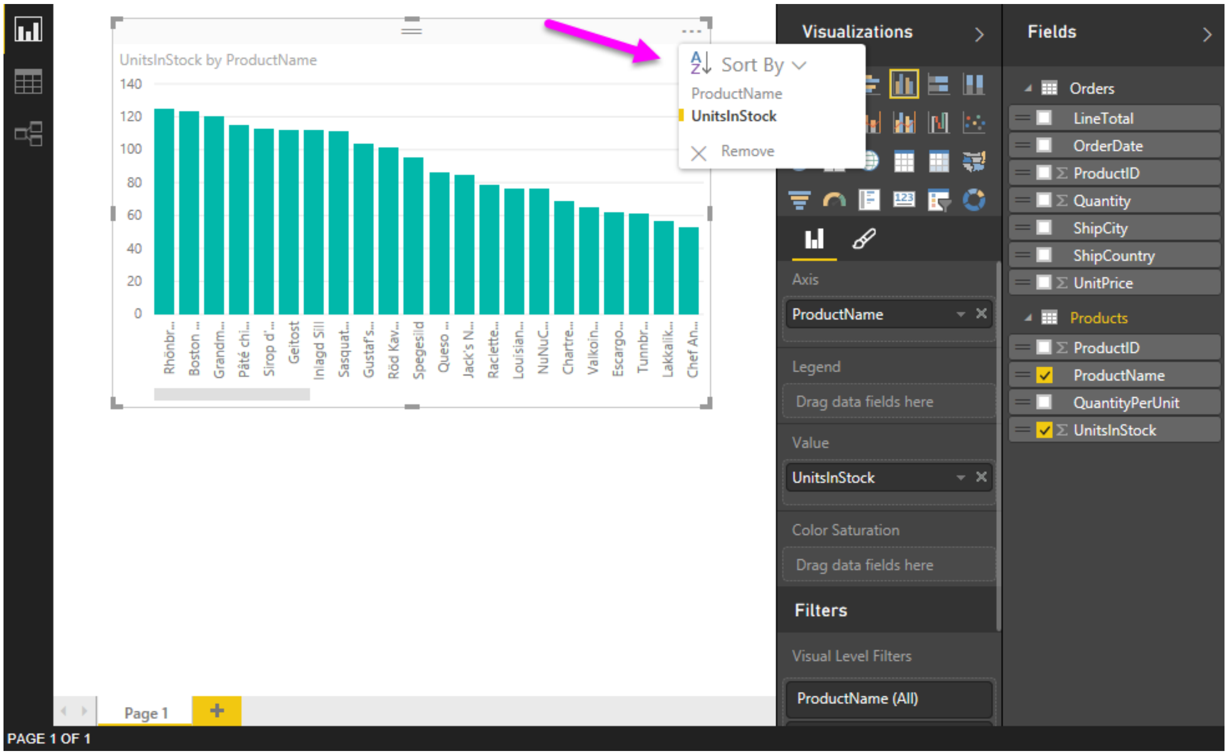
Task 4: Build visuals using your data

Power BI Desktop lets you create a variety of visualizations to gain insights from your data. You can build reports with multiple pages and each page can have multiple visuals. You can interact with your visualizations to help analyze and understand your data. For more information about editing reports, see Edit a Report.

In this task, you create a report based on the data previously loaded. You use the Fields pane to select the columns from which you create the visualizations.

Step 1: Create charts showing Units in Stock by Product and Total Sales by Year

1. Drag **UnitsInStock** from the Field pane (the Fields pane is along the right of the screen) onto a blank space on the canvas. A Table visualization is created. Next, drag ProductName to the Axis box, found in the bottom half of the Visualizations pane. Then we then select **Sort By > UnitsInStock** using the skittles in the top right corer of the visualization.

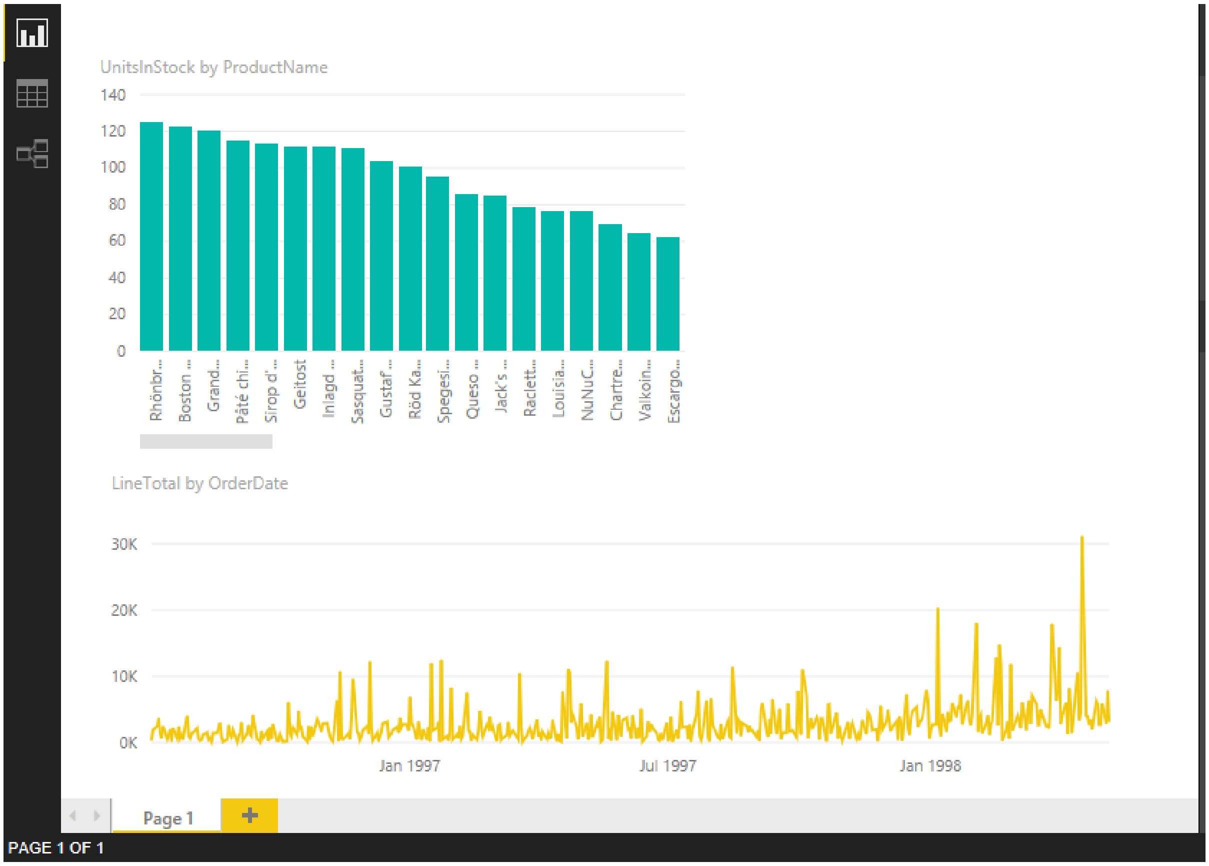


Axis: ProductName

Value: UnitsInStock

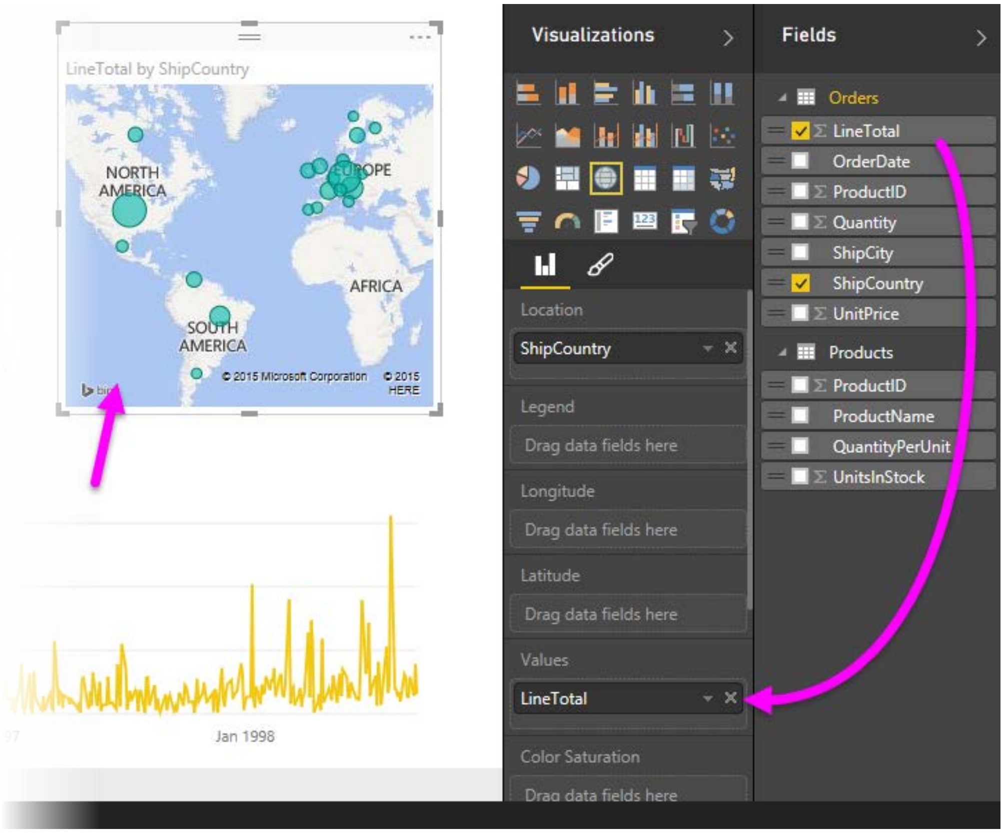
Visual Level Filters: ProductName (All)

2. Drag **OrderDate** to the canvas beneath the first chart, then drag LineTotal (again, from the Fields pane) onto the visual, then select Line Chart. The following visualization is created.



3. Next, drag **ShipCountry** to a space on the canvas in the top right. Because you selected a geographic field, a map was created automatically. Now drag **LineTotal** to the **Values** field;

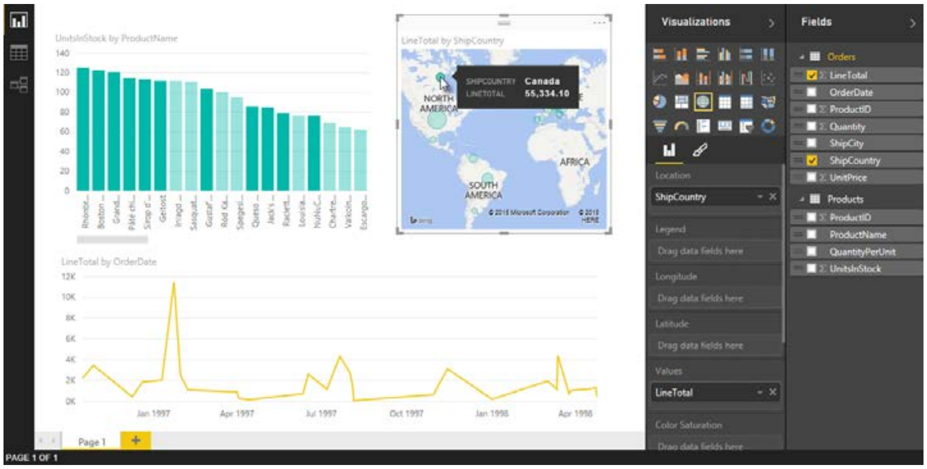
the circles on the map for each country are now relative in size to the **LineTotal** for orders shipped to that country.



Step 2: Interact with your report visuals to analyze further

Power BI Desktop lets you interact with visuals that cross-highlight and filter each other to uncover further trends. For more detail see [Filtering and Highlighting in Reports]()

1. Click on the light blue circle centered in **Canada.** Note how the other visuals are filtered to show Stock (**ShipCountry**) and Total Orders (**LineTotal**) just for Canada.



Complete Sales Analysis Report

After you perform all these steps, you will have a Sales Report that combines data from Products.xlsx file and Northwind OData feed. The report shows visuals that help analyze sales information from different countries. You can download a completed Power BI Desktop file for this tutorial here.