

Combine data from multiple data sources (Power Query)

Note: Power Query is known as **Get & Transform** in Excel 2016. Information provided here applies to both. To learn more, see [Get & Transform in Excel 2016](#).

In this tutorial, you'll use Power Query's Query Editor to import data from a local Excel file that contains product information, and from an OData feed that contains product order information. You perform transformation and aggregation steps, and combine data from both sources to produce a **Total Sales per Product and Year** report.

In order to perform this tutorial, you need the [Products and Orders](#) workbook. In the **Save As** dialog box, name the file **Products and Orders.xlsx**.

In this tutorial

Task 1: [Import products into an Excel workbook](#)

Step 1: [Connect to an Excel workbook](#)

Step 2: [Promote the first row to table column headers](#)

Step 3: [Remove other columns to only display columns of interest](#)

[Power Query steps created](#)

Step 4: [Import a products query](#)

Task 2: [Import order data from an OData feed](#)

Step 1: [Connect to an OData feed](#)

Step 2: [Expand an Order_Details table](#)

[Expand the Order_Details Table link](#)

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

Remove selected columns

Step 4: Calculate the line total for each Order_Details row

Calculate the line total for each Order_Details row

Step 5: Transform an OrderDate year column

Step 6: Group rows by ProductID and Year

Step 7: Rename a query

Final query results

Power Query steps created

Step 8: Disable query download into an Excel workbook

Disable a query download

Task 3: Combine the Products and Total Sales queries

Step 1: Merge ProductID into a Total Sales query

Step 2: Expand a merge column

Expand the NewColumn table link

Power Query steps created

Step 3: Load a Total Sales per Product query into an Excel Data Model

Load the Total Sales per Product query into the Excel Data Model

Final Total Sales per Product query

Task 1: Import products into an Excel workbook

In this task, you import products from the [Products and Orders.xlsx](#) file into an Excel workbook.


Step 1: Connect to an Excel workbook

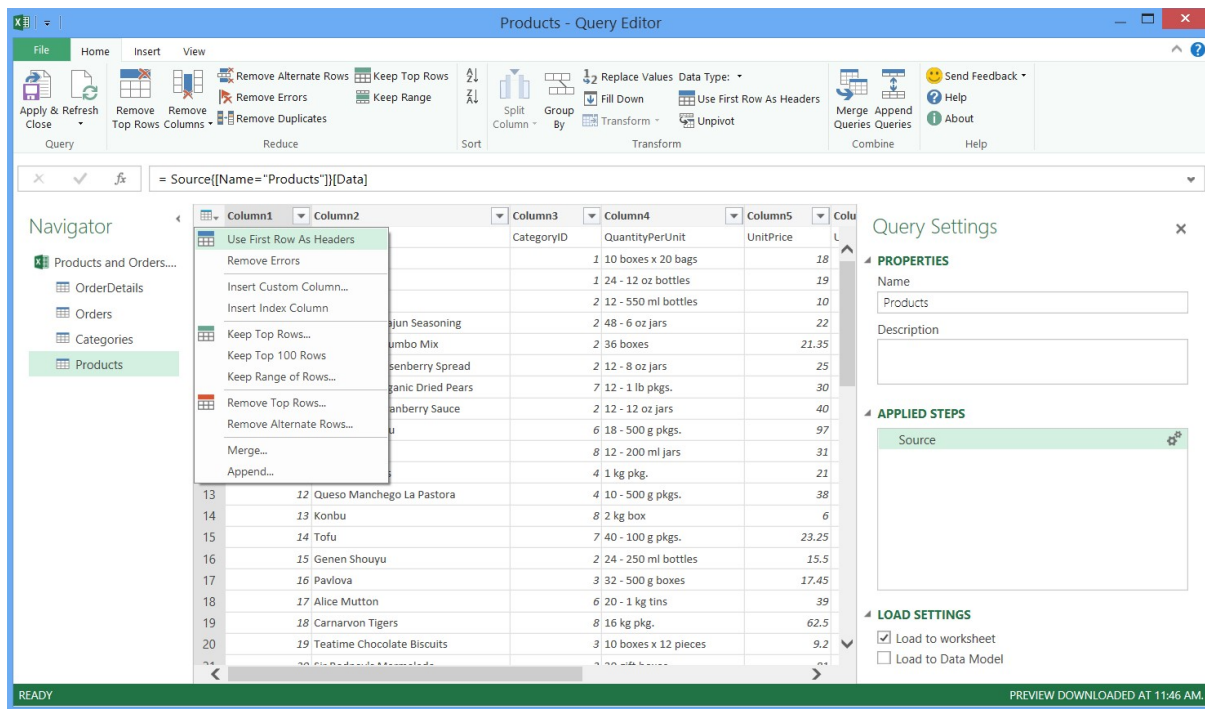
1. Create an Excel workbook.
2. In the **POWER QUERY** ribbon tab, click **From File** > **From Excel**.
3. In the **Excel** browse dialog box, browse for or type the Products and Orders.xlsx path to import or link to a file.
4. In the **Navigator** pane, double click the **Products** worksheet or click **Products** and click **Edit Query**. When you edit a query, or connect to a new data source, the **Query Editor** window appears.

Note: For a very quick video on how to display **Query Editor**, see the end of this article.

Step 2: Promote the first row to table column headers

In the **Query Preview** grid, the first row of the table does not contain the table column names. To promote the first row to table column headers:

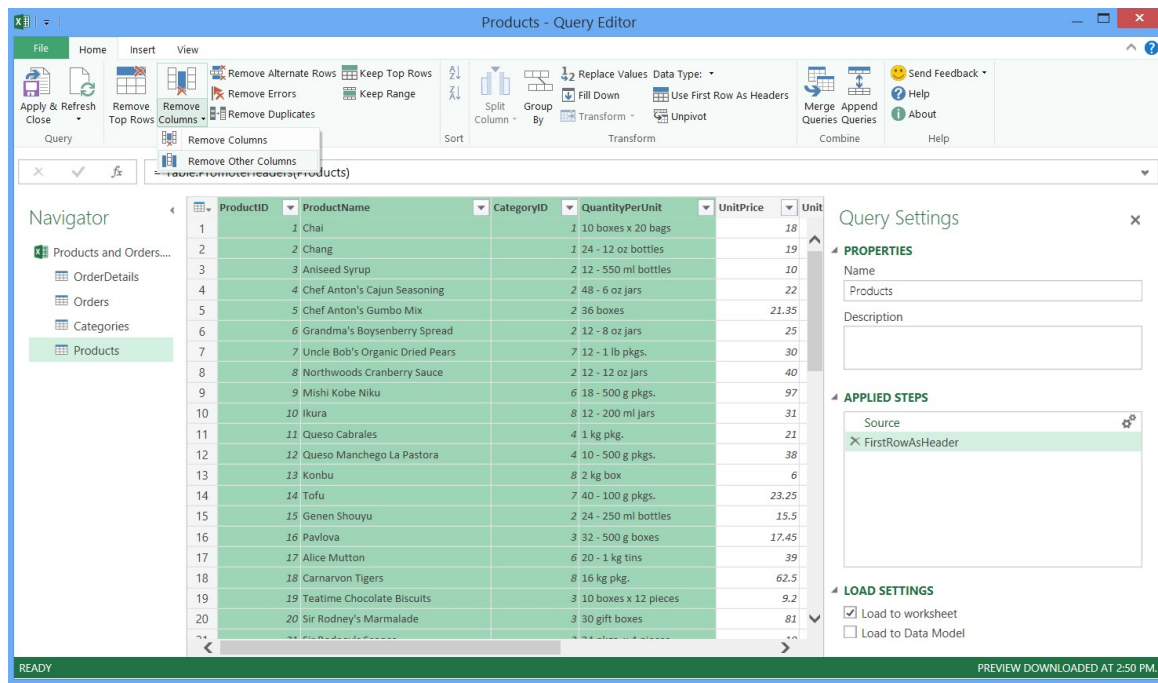
1. Click the table icon () in the top-left corner of the data preview.
2. Click **Use First Row as Headers**.



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

In this step you remove all columns except **ProductID**, **ProductName**, **CategoryID**, and **QuantityPerUnit**.

1. In the **Query Preview** grid, select the **ProductID**, **ProductName**, **CategoryID**, and **QuantityPerUnit** columns (use Ctrl+Click or Shift+Click).
2. In the **Query Editor** ribbon, click **Remove Columns** > **Remove Other Columns** or right-click on a column header, and click **Remove Other Columns**.



Power Query steps created

As you perform query activities in Power Query, 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 the Power Query formula language, see [Learn about Power Query 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	RemovedOtherColumns	Table.SelectColumns

Task	Query step	Formula
display columns of interest		(FirstRowAsHeader,{"ProductID", "ProductName", "CategoryID", "QuantityPerUnit"})

Step 4: Import a products query

In this step, you import the **Products** query into your Excel workbook.

1. In the **Query Editor** ribbon, click **Apply & Close**. The results appear in a new Excel worksheet.

[Top of Page](#)

Task 2: Import order data from an OData feed

In this task, you import data into your Excel workbook from the sample Northwind OData feed at <http://services.odata.org/Northwind/Northwind.svc>.

Step 1: Connect to an OData feed

1. In the **POWER QUERY** ribbon tab, click **From Other Sources** > **From OData Feed**.
2. In the **OData Feed** dialog box, enter the **URL** for the Northwind OData feed.
3. Click **OK**.
4. In the **Navigator** pane, double click the **Orders** table or click **Orders** and click **Edit**.

Note: When you hover your mouse over a table, you will see a table preview fly out.

The screenshot shows the Power Query interface. On the right, the **Navigator** pane displays a list of tables from the OData feed `http://services.odata.org/Northwind/No...`. The **Orders** table is selected and highlighted in green. Below the list are buttons for **Edit**, **Load**, and **Cancel**.

On the left, a preview window for the **Orders** table is shown. It contains a table with the following data:

OrderID	CustomerID	EmployeeID	OrderDate	Re
10248	VINET	5	7/4/1996 12:00:00 AM	
10249	TOMSP	6	7/5/1996 12:00:00 AM	
10250	HANAR	4	7/8/1996 12:00:00 AM	
10251	VICTE	3	7/8/1996 12:00:00 AM	
10252	SUPRD	4	7/9/1996 12:00:00 AM	
10253	HANAR	3	7/10/1996 12:00:00 AM	
10254	CHOPS	5	7/11/1996 12:00:00 AM	
10255	RICSU	9	7/12/1996 12:00:00 AM	
10256	WELLI	3	7/15/1996 12:00:00 AM	

Below the table, the **Columns [18]** are listed: `OrderID, CustomerID, EmployeeID, OrderDate, RequiredDate, ShippedDate, ShipVia, Freight, ShipName, ShipAddress, ShipCity, ShipRegion, ShipPostalCode, ShipCountry, Customer, Employee, Order_Details, Shipper`.


Step 2: Expand an Order_Details table

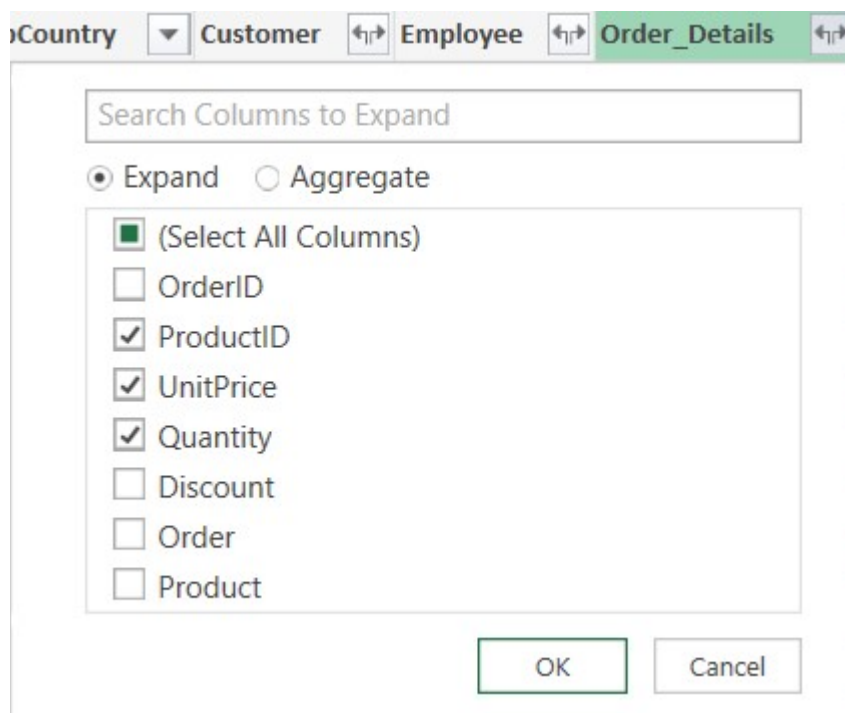
In this step, you expand the **Order_Details** table that is related to the **Orders** table, to combine the **ProductID**, **UnitPrice**, and **Quantity** columns from **Order_Details** into the **Orders** table. The **Expand** operation combines columns from a related table into a subject table. When the query runs, rows from the related table (**Order_Details**) are combined into rows from the subject table (**Orders**).

In Power Query, a column containing a link to a related table has an **Entry** link or **Table** link. An **Entry** link navigates to a single related record, and represents a one-to-one relationship with a subject table. A **Table** link navigates to a related table, and represents a one-to-many relationship with a subject table. A link represents navigation properties in a data source within a relational model. For an OData feed, navigation properties represent an entity with a foreign key association. In a database, such as SQL Server, navigation properties represent foreign key relationships in the database.

Expand the Order_Details Table link

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 Preview** pane, scroll to the **Order_Details** column.
2. In the **Order_Details** column, click the expand icon ().
3. In the **Expand** drop-down:
 - a. Click **(Select All Columns)** to clear all columns.
 - b. Click **ProductID**, **UnitPrice**, and **Quantity**.
 - c. Click **OK**.



Note: In Power Query, you can expand tables linked from to a column, and also the ability to perform aggregate operations on the columns of the linked table before expanding the data in the subject table. For more information about how to perform aggregate operations, see [Aggregate data from a column](#).

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

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


Remove selected columns

1. In the **Query Preview** pane, select all columns:
 - a. Click the first column (**OrderID**).
 - b. Shift+Click the last column (**Shipper**).
 - c. Ctrl+Click the **OrderDate**, **Order_Details.ProductID**, **Order_Details.UnitPrice**, and **Order_Details.Quantity** columns.
2. Right-click on a selected column header, and click **Remove Columns**.

Step 4: Calculate the line total for each Order_Details row

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 **Query Preview** pane, click the table icon () at the top-left corner of the preview.
2. Click **Insert Column > Custom**.
3. In the **Insert Custom Column** dialog box, in the **Custom Column Formula** textbox, enter **[Order_Details.UnitPrice] * [Order_Details.Quantity]**.

4. In the **New column name** textbox, enter **Line Total**.
5. Click **OK**.

Insert Custom Column

New column name
Line Total

Custom column formula:
= [Order_Details.UnitPrice] * [Order_Details.Quantity]

Available columns:
OrderID
CustomerID
EmployeeID
OrderDate
RequiredDate
ShippedDate
ShipVia
Freight

<< Insert

[Learn about Power Query formulas](#)

✓ No syntax errors have been detected.

OK Cancel

Step 5: Transform an OrderDate year column

In this step, you transform the **OrderDate** column to render the order date year.

1. In the **Preview** grid, right-click the **OrderDate** column, and click **Transform** > **Year**.
2. Rename the **OrderDate** column to **Year**:
 - a. Double-Click the **OrderDate** column, and enter **Year** or
 - b. Right-Click on the **OrderDate** column, click **Rename**, and enter **Year**.

Step 6: Group rows by ProductID and Year

1. In the **Query Preview** grid, select **Year** and **Order_Details.ProductID**.
2. Right-Click one of the headers, and click **Group By**.
3. In the **Group By** dialog box:
 - a. In the **New column name** textbox, enter **Total Sales**.
 - b. In the **Operation** drop-down, select **Sum**.
 - c. In the **Column** drop-down, select **Line Total**.
4. Click **OK**.

Group By...

Specify the columns to group by.

Group by +

Year -

Order_Details.ProductID -

New column name Operation Column +

Total Sales Sum Line Total -

OK Cancel

Step 7: Rename a query

Before you import the sales data into Excel, name the query **Total Sales**:

1. In the **Query Settings** pane, in the **Name** text box enter **Total Sales**.

Final query results

After you perform each step, you will have a Total Sales query over the Northwind OData feed.

	Year	Order_Details.ProductID	Total Sales
1	1996	11	1814.4
2	1996	42	400.4
3	1996	72	7263
4	1996	14	1581
5	1996	51	6911.2
6	1996	41	877.8
7	1996	65	2604
8	1996	22	100.8
9	1996	57	2074.8
10	1996	20	6868.8
11	1996	33	394

Power Query steps created

As you perform query activities in Power Query, 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 the Power Query formula language, see [Learn about Power Query 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"},

Task	Query step	Formula
		{"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])
Transform the OrderDate column to render the year	RenamedColumns	Table.RenameColumns (InsertedCustom,{{"Custom", "Line Total"}})
	TransformedColumn	Table.TransformColumns (RenamedColumns,{{"OrderDate", Date.Year}})
	RenamedColumns1	Table.RenameColumns

Task	Query step	Formula
		(TransformedColumn,{{"OrderDate", "Year"}})
Group rows by ProductID and Year	GroupedRows	Table.Group (RenamedColumns1, {"Year", "Order_Details.ProductID"}, {{"Total Sales", each List.Sum([Line Total]), type number}})

Step 8: Disable query download into an Excel workbook

Since the **Total Sales** query does not represent the final **Total Sales per Product and Year** report, you disable query download into the Excel workbook. When the **Load to Worksheet** option is **Off** in the **Queries settings** pane, the data result of this query is not downloaded, but the query can still be combined with other queries in order to build the desired result. You learn how to combine this query with the Products query in the next task.

Disable a query download

1. In the **Query Settings** pane, uncheck **Load to worksheet**.
2. In the **Query Editor** ribbon, click **Apply & Close**. In the **Workbook Queries** pane, the **Total Sales** query displays **Load is disabled**.

Workbook Queries



2 queries

 Products

Last updated on Thursday.

77 rows loaded.

 Total Sales

Last updated at 3:07 PM.

Load is disabled.

[Top of Page](#)

Task 3: Combine the Products and Total Sales queries

Power Query enables you to combine multiple queries, by merging or appending them. The **Merge** operation is performed on any Power Query query with a tabular shape, independent of the data source that the data comes from. For more information about combining data sources, see [Combine multiple queries](#).

In this task, you combine the **Products and Total Sales** queries by using a **Merge** and **Expand** query step.

Step 1: Merge ProductID into a Total Sales query

1. In the Excel workbook, navigate to the **Products** query on **Sheet2**.
2. In the **QUERY** ribbon tab, click **Merge**.
3. In the **Merge** dialog box, select **Products** as the primary table, and select **Total Sales** as the second or related query to merge. **Total Sales** will become a new expandable column.

4. To match **Total Sales** to **Products** by **ProductID**, select the **ProductID** column from the **Products** table, and the **Order_Details.ProductID** column from the **Total Sales** table.
5. In the **Privacy Levels** dialog box:
 - a. Select **Organizational** for your privacy isolation level for both data sources.
 - b. Click **Save**.
6. Click **OK**.

Security Note: Privacy Levels prevent a user from inadvertently combining data from multiple data sources, which might be private or organizational. Depending on the query, a user could inadvertently send data from the private data source to another data source that might be malicious. Power Query analyzes each data source and classifies it into the defined level of privacy: Public, Organizational, and Private. For more information about Privacy Levels, see [Privacy Levels](#).

Merge

Select tables and matching columns to create a merged table.

Products

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

Total Sales

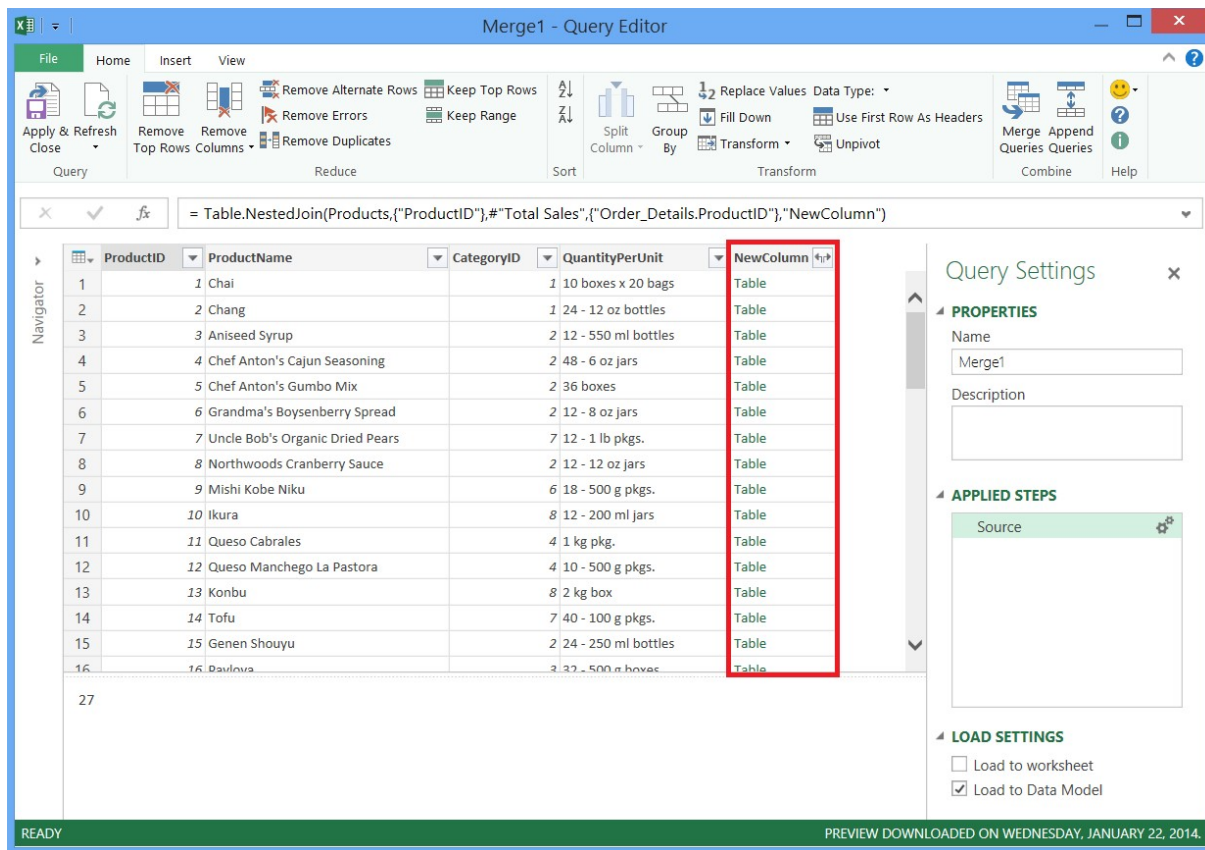
Year	Order_Details.ProductID	Total Sales
1996	11	1814.4
1996	42	400.4
1996	72	7263
1996	14	1581
1996	51	6911.2

☐ Only include matching rows

☒ The selection has matched 77 out of the first 77 rows.

OK
Cancel


After you click **OK**, the **Merge** operation creates a query. The query result contains all columns from the primary table (**Products**), and a single column containing a navigation link to the related table (**Total Sales**). An **Expand** operation adds new columns into the primary or subject table from the related table.



Step 2: Expand a merge column

In this step, you expand the merge column with the name **NewColumn** to create two new columns in the **Products** query: **Year** and **Total Sales**.

Expand the NewColumn table link

1. In the **Query Preview** grid, click the **NewColumn** expand icon ().
2. In the **Expand** drop-down:
 - a. Click **(Select All Columns)** to clear all columns.
 - b. Click **Year** and **Total Sales**.
 - c. Click **OK**.

- Rename these two columns to **Year** and **Total Sales**.
- Sort Descending** by **Total Sales** to find out which products and in which years the products got the highest volume of sales.
- Rename** the query to **Total Sales per Product**.

The screenshot shows the Power Query Editor window titled "Total Sales per Product - Query Editor". The ribbon includes tabs for File, Home, Insert, and View. The Home tab is active, showing options for Query (Apply & Refresh, Close), Reduce (Remove, Remove Top Rows, Remove Columns, Remove Errors, Remove Duplicates), Sort (Sort, Split Column, Group By), Transform (Replace Values, Fill Down, Transform, Unpivot), and Combine (Merge, Append, Queries, Queries). The formula bar shows the query formula: `= Table.Sort(RenamedColumns,{"Total Sales", Order.Descending})`. The data table has columns: ProductID, ProductName, CategoryID, QuantityPerUnit, Year, and Total Sales. The data is sorted by Total Sales in descending order. The right-hand pane shows the Query Settings for "Total Sales per Product", including Properties (Name, Description) and Applied Steps (Source, Expand NewColumn, RenamedColumns, SortedRows). The Load Settings section shows options for "Load to worksheet" and "Load to Data Model".

ProductID	ProductName	CategoryID	QuantityPerUnit	Year	Total Sales
1	38 Côte de Blaye		1 12 - 75 cl bottles	1998	682
2	38 Côte de Blaye		1 12 - 75 cl bottles	1997	5196
3	29 Thüringer Rostbratwurst		6 50 bags x 30 sausgs.	1998	39365
4	59 Raclette Courdavault		4 5 kg pkg.	1997	375
5	29 Thüringer Rostbratwurst		6 50 bags x 30 sausgs.	1997	36194
6	56 Gnocchi di nonna Alice		5 24 - 250 g pkgs.	1997	3475
7	38 Côte de Blaye		1 12 - 75 cl bottles	1996	295
8	59 Raclette Courdavault		4 5 kg pkg.	1998	282
9	51 Manjimup Dried Apples		7 50 - 300 g pkgs.	1997	2606
10	62 Tarte au sucre		3 48 pies	1997	2267
11	60 Camembert Pierrot		4 15 - 300 g rounds	1997	217
12	17 Alice Mutton		6 20 - 1 kg tins	1997	1971
13	60 Camembert Pierrot		4 15 - 300 g rounds	1998	182
14	62 Tarte au sucre		3 48 pies	1998	1730
15	18 Carnarvon Tigers		8 16 kg pkg.	1997	172
16	69 Gudbrandsdalsost		4 10 kg pkg.	1997	151
17	28 Rössle Sauerkraut		7 25 - 825 g cans	1997	146
18	7 Uncle Bob's Organic Dried Pears		7 12 - 1 lb pkgs.	1998	124
19	29 Thüringer Rostbratwurst		6 50 bags x 30 sausgs.	1996	121
20	51 Manjimup Dried Apples		7 50 - 300 g pkgs.	1998	117

Power Query steps created

As you perform **Merge** query activities in Power Query, 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 the Power Query formula language, see [Learn about Power Query formulas](#).

Task	Query step	Formula
------	------------	---------

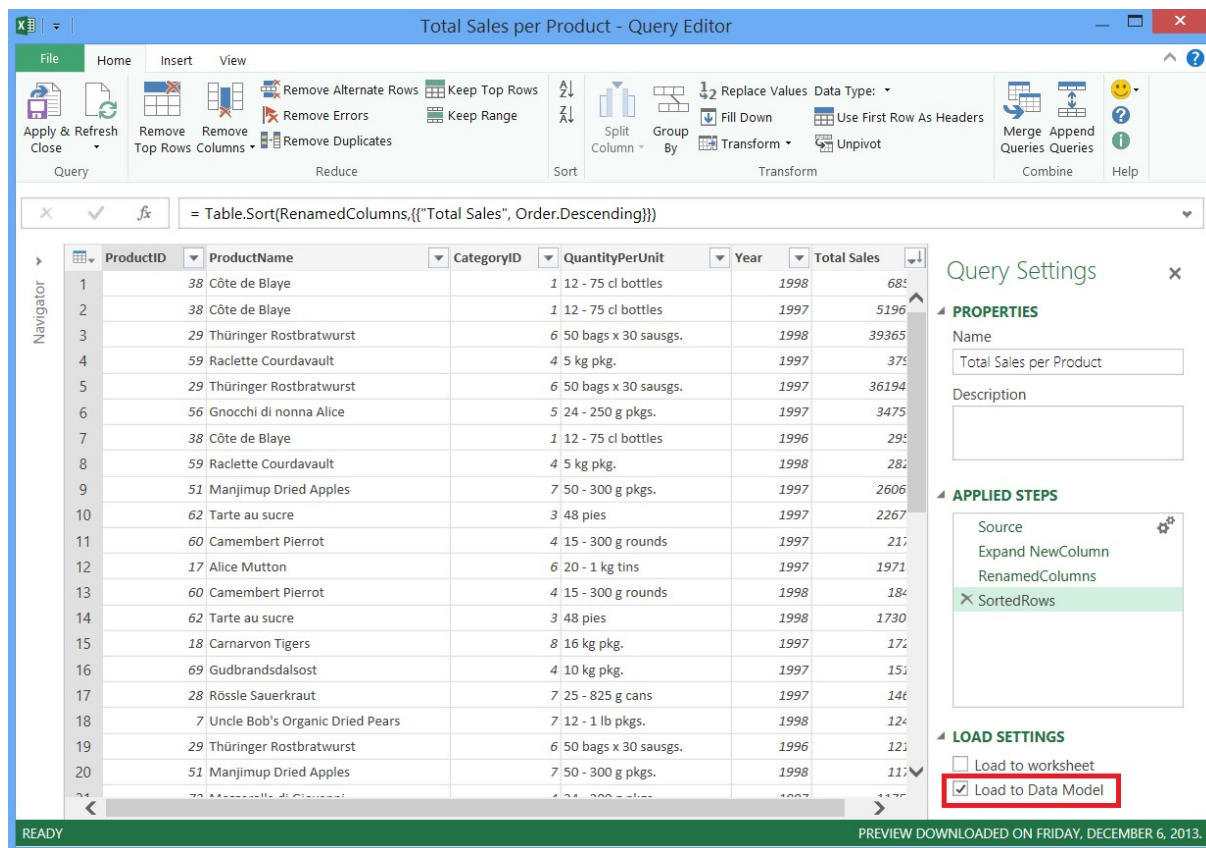
Task	Query step	Formula
Merge ProductID into the Total Sales query	Source (data source for Merge operation)	Table.NestedJoin (Products,{"ProductID"},#"Total Sales", {"Order_Details.ProductID"},"NewColumn")
Expand a merge column	ExpandNewColumn	Table.ExpandTableColumn (Source, "NewColumn", {"Year", "Total Sales"}, {"NewColumn.Year", "NewColumn.Total Sales"})
	RenamedColumns	Table.RenameColumns (#"Expand NewColumn", {"NewColumn.Year", "Year"}, {"NewColumn.Total Sales", "Total Sales"})
	SortedRows	Table.Sort (RenamedColumns,{"Total Sales", Order.Descending}))

Step 3: Load a Total Sales per Product query into an Excel Data Model

In this step, you disable the **Load to Worksheet** option and load a query into the **Excel Data Model**, in order to build a report connected to the query result. In addition to loading query results into an Excel worksheet, Power Query enables you to load a query result into an **Excel Data Model**. After you load data into the **Excel Data Model**, you can use Power Pivot and Power View to further data analysis.

Load the Total Sales per Product query into the Excel Data Model

1. In the **Query Settings** pane, uncheck **Load to worksheet** and check **Load to data model**.
2. To load the query into the **Excel Data Model**, click **Apply & Close**.

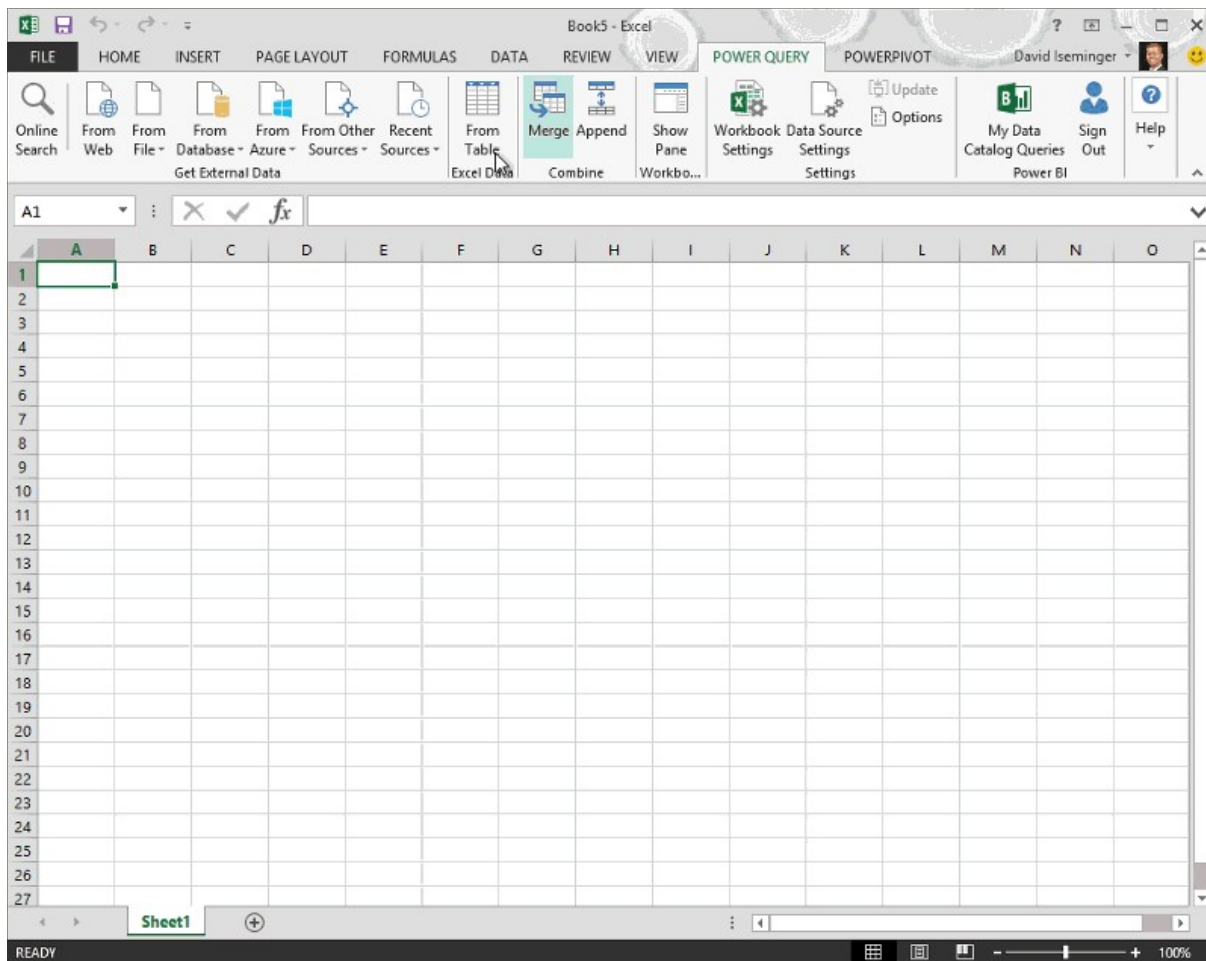


Final Total Sales per Product query

After you perform each step, you will have a **Total Sales per Product** query that combines data from the Products and Orders.xlsx file and Northwind OData feed. This query can be applied to a Power Pivot model. In addition, changes to the query in Power Query modify and refresh the resulting table in the Power Pivot model.

Note: The **Query Editor** only appears when you load, edit, or create a new query using **Power Query**. The following video shows the **Query Editor** window appearing after editing a query from an Excel workbook. To view the **Query Editor** without

loading or editing an existing workbook query, from the **Get External Data** section in the **Power Query** ribbon tab, select **From Other Sources > Blank Query**. The following video shows one way to display the **Query Editor**.



[Top of Page](#)

Was this information helpful?

Yes

No