What is Power Query?

Power Query is a data transformation and data preparation engine. Power Query comes with a graphical interface for getting data from sources and a Power Query Editor for applying transformations. Because the engine is available in many products and services, the destination where the data will be stored depends on where Power Query was used. Using Power Query, you can perform the extract, transform, and load (ETL) processing of data.

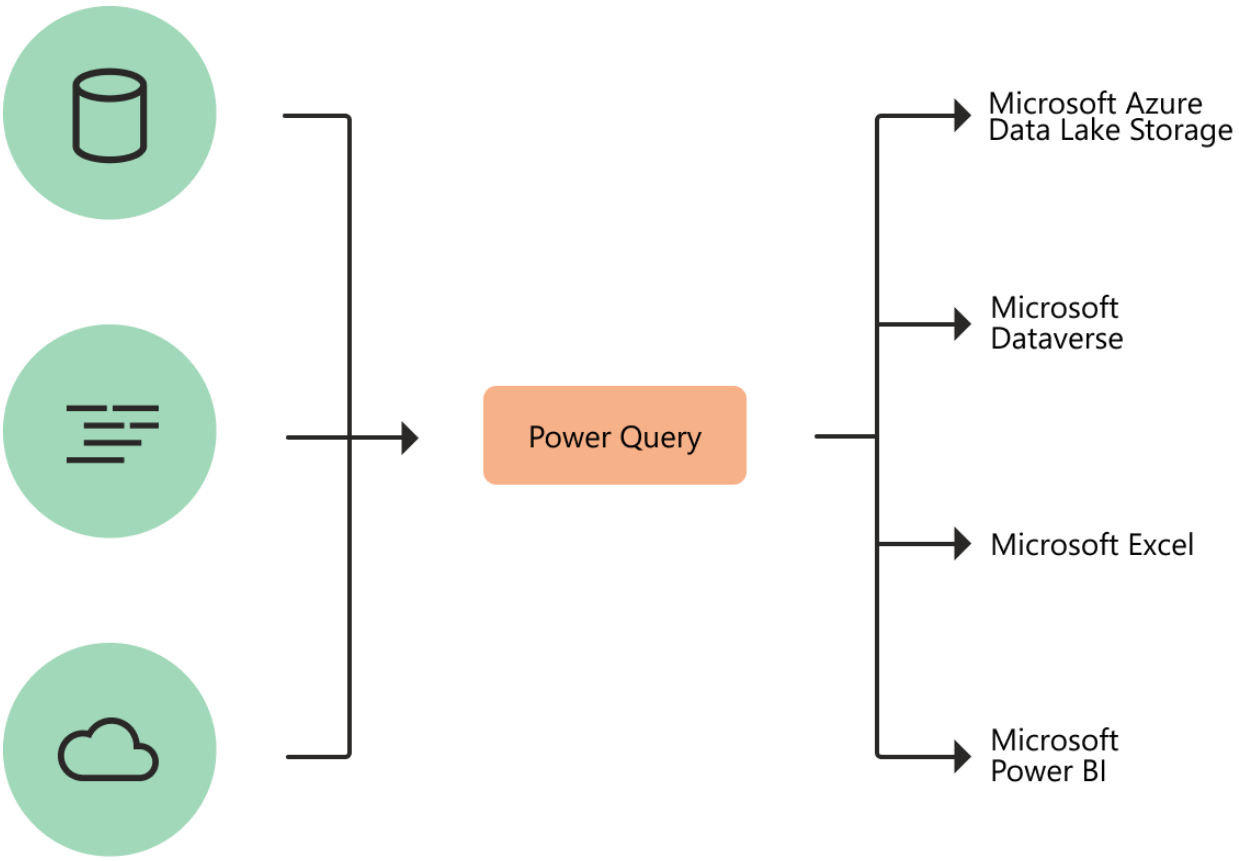


Diagram with symbolized data sources on the right, passing though Power query for transformation, and then going to various destinations, such as Azure Data Lake Storage, Dataverse, Microsoft Excel, or Power BI.

How Power Query helps with data acquisition

Business users spend up to 80 percent of their time on data preparation, which delays the work of analysis and decision-making. Several challenges contribute to this situation, and Power Query helps address many of them.

Existing challenge	How does Power Query help?
Finding and connecting to data is too difficult	Power Query enables connectivity to a wide range of data sources, including data of all sizes and shapes.
Experiences for data connectivity are too fragmented	Consistency of experience, and parity of query capabilities over all data sources.
Data often needs to be reshaped before consumption	Highly interactive and intuitive experience for rapidly and iteratively building queries over any data source, of any size.
Any shaping is one-off and not repeatable	When using Power Query to access and transform data, you define a repeatable process (query) that can be easily refreshed in the future to get up-to-date data. In the event that you need to modify the process or query to account for underlying data or schema changes, you can use the same interactive and intuitive experience you used when you initially defined the query.
Volume (data sizes), velocity (rate of change), and variety (breadth of data sources and data shapes)	Power Query offers the ability to work against a subset of the entire dataset to define the required data transformations, allowing you to easily filter down and transform your data to a manageable size. Power Query queries can be refreshed manually or by taking advantage of scheduled refresh capabilities in specific products (such as Power BI) or even programmatically (by using the Excel object model). Because Power Query provides connectivity to hundreds of data sources and over 350 different types of data transformations for each of these sources, you can work with data from any source and in any shape.

Power Query experiences

The Power Query user experience is provided through the Power Query Editor user interface. The goal of this interface is to help you apply the transformations you need simply by interacting with a user-friendly set of ribbons, menus, buttons, and other interactive components.

The Power Query Editor is the primary data preparation experience, where you can connect to a wide range of data sources and apply hundreds of different data transformations by previewing data and selecting transformations from the UI. These data transformation capabilities are common across all data sources, whatever the underlying data source limitations.

When you create a new transformation step by interacting with the components of the Power Query interface, Power Query automatically creates the M code required to do the transformation so you don't need to write any code.

Currently, two Power Query experiences are available:

- **Power Query Online**—Found in integrations such as Power BI dataflows, Microsoft Power Platform dataflows, Azure Data Factory wrangling dataflows, and many more that provide the experience through an online webpage.
- **Power Query for Desktop**—Found in integrations such as Power Query for Excel and Power BI Desktop.

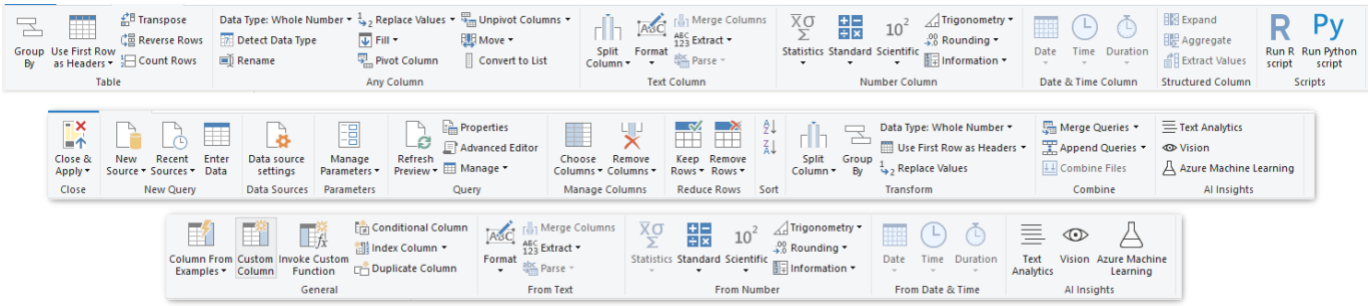
Note

Although two Power Query experiences exist, they both provide almost the same user experience in every scenario.

Transformations

The transformation engine in Power Query includes many prebuilt transformation functions that can be used through the graphical interface of the Power Query Editor. These transformations can be as simple as removing a column or filtering rows, or as common as using the first row as a table header. There are also advanced transformation options such as merge, append, group by, pivot, and unpivot.

All these transformations are made possible by choosing the transformation option in the menu, and then applying the options required for that transformation. The following illustration shows a few of the transformations available in Power Query Editor.



Dataflows

Power Query can be used in many products, such as Power BI and Excel. However, using Power Query within a product limits its usage to only that specific product. *Dataflows* are a product-agnostic service version of the Power Query experience that runs in the cloud. Using dataflows, you can get data and transform data in the same way, but instead of sending the output to Power BI or Excel, you can store

the output in other storage options such as Dataverse or Azure Data Lake Storage. This way, you can use the output of dataflows in other products and services.

Power Query M formula language

In any data transformation scenario, there are some transformations that can't be done in the best way by using the graphical editor. Some of these transformations might require special configurations and settings that the graphical interface doesn't currently support. The Power Query engine uses a scripting language behind the scenes for all Power Query transformations: the Power Query M formula language, also known as M.

The M language is the data transformation language of Power Query. Anything that happens in the query is ultimately written in M. If you want to do advanced transformations using the Power Query engine, you can use the Advanced Editor to access the script of the query and modify it as you want. If you find that the user interface functions and transformations won't perform the exact changes you need, use the Advanced Editor and the M language to fine-tune your functions and transformations.

Power Query MCopy

```
let
    Source = Exchange.Contents("xyz@contoso.com"),
    Mail1 = Source{[Name="Mail"]}[Data],
    #"Expanded Sender" = Table.ExpandRecordColumn(Mail1, "Sender", {"Name"},
{"Name"}),
    #"Filtered Rows" = Table.SelectRows(#"Expanded Sender", each ([HasAttachments] =
true)),
    #"Filtered Rows1" = Table.SelectRows(#"Filtered Rows", each ([Subject] = "sample
files for email PQ test") and ([Folder Path] = "\Inbox\")),
    #"Removed Other Columns" = Table.SelectColumns(#"Filtered
Rows1",{"Attachments"}),
    #"Expanded Attachments" = Table.ExpandTableColumn(#"Removed Other Columns",
"Attachments", {"Name", "AttachmentContent"}, {"Name", "AttachmentContent"}),
    #"Filtered Hidden Files1" = Table.SelectRows(#"Expanded Attachments", each
[Attributes]?[Hidden]? <> true),
    #"Invoke Custom Function1" = Table.AddColumn(#"Filtered Hidden Files1",
"Transform File from Mail", each #"Transform File from Mail"([AttachmentContent])),
    #"Removed Other Columns1" = Table.SelectColumns(#"Invoke Custom Function1",
{"Transform File from Mail"}),
    #"Expanded Table Column1" = Table.ExpandTableColumn(#"Removed Other Columns1",
"Transform File from Mail", Table.ColumnNames(#"Transform File from Mail"(#"Sample
File"))),
    #"Changed Type" = Table.TransformColumnTypes(#"Expanded Table
Column1",{{"Column1", type text}, {"Column2", type text}, {"Column3", type text},
{"Column4", type text}, {"Column5", type text}, {"Column6", type text}, {"Column7",
type text}, {"Column8", type text}, {"Column9", type text}, {"Column10", type text}})
in
    #"Changed Type"
```

More information: [Power Query M formula language]

Where can you use Power Query?

The following table lists Microsoft products and services where Power Query can be found.

Product	M engine[1]	Power Query Desktop[2]	Power Query Online[3]	Dataflows[4]
Excel for Windows	Yes	Yes	No	No
Excel for Mac	Yes	No	No	No
Power BI	Yes	Yes	Yes	Yes
Power Apps	Yes	No	Yes	Yes
Power Automate	Yes	No	Yes	No
Power BI Report Server	Yes	Yes	No	No
Azure Data Factory	Yes	No	Yes	Yes
SQL Server Integration Services	Yes	No	No	No
SQL Server Analysis Services	Yes	Yes	No	No
Dynamics 365 Customer Insights	Yes	No	Yes	Yes

[1]**M engine**	The underlying query execution engine that runs queries expressed in the Power Query formula language ("M").
[2]**Power Query Desktop**	The Power Query experience found in desktop applications.
[3]**Power Query Online**	The Power Query experience found in web browser applications.
[4]**Dataflows**	Power Query as a service that runs in the cloud and is product-agnostic. The stored result can be used in other applications as services.

The Power Query user interface

With Power Query, you can connect to many different data sources and transform the data into the shape you want.

In this article, you'll learn how to create queries with Power Query by discovering:

- How the "Get Data" experience works in Power Query.
- How to use and take advantage of the Power Query user interface.
- How to perform common transformations like grouping and merging data.

Examples in this article connect to and use the Northwind OData feed.

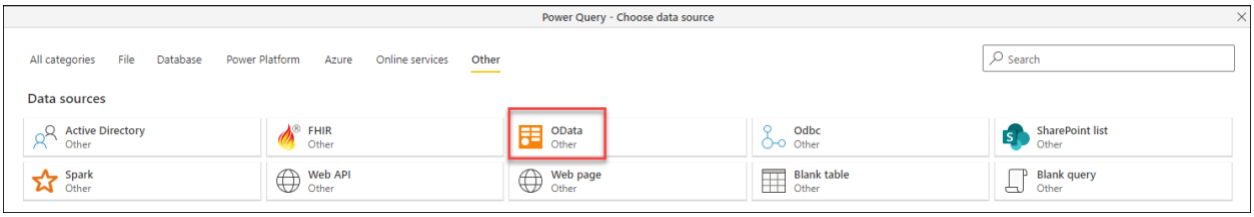
Copy
```
https://services.odata.org/V4/Northwind/Northwind.svc/
```

Connect to an OData feed

Note

To learn more about where to get data from each of the Microsoft products that include Power Query, go to **Where to get data**.

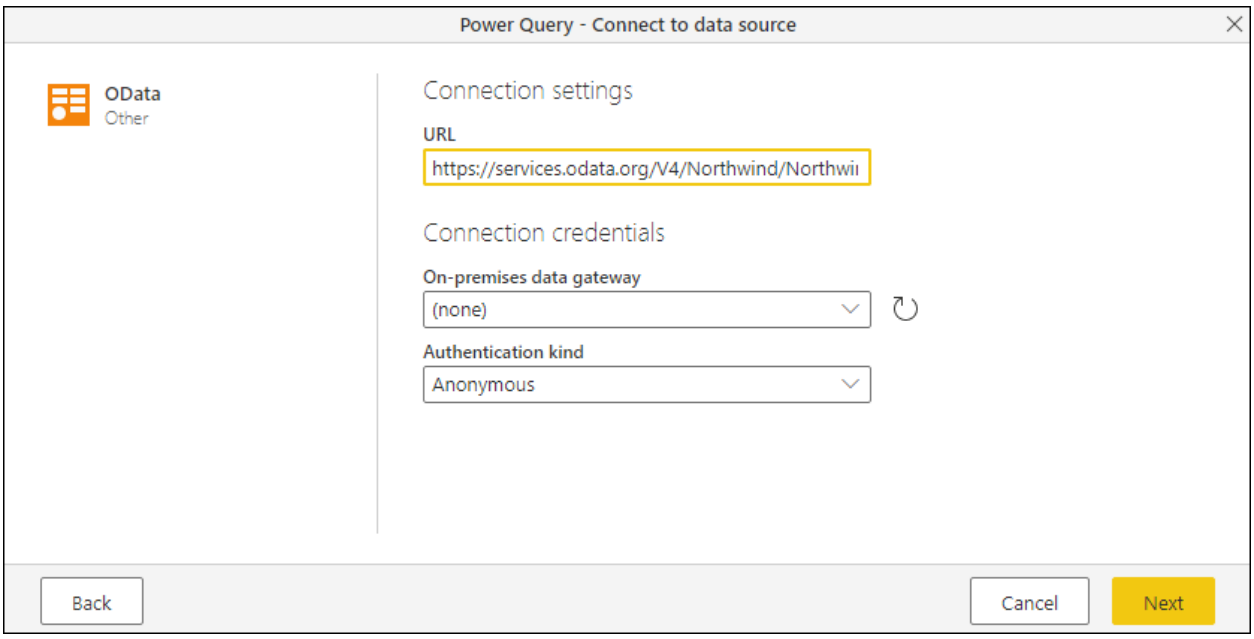
To start, locate the **OData** feed connector from the "Get Data" experience. You can select the **Other** category from the top, or search for **OData** in the search bar in the top-right corner.



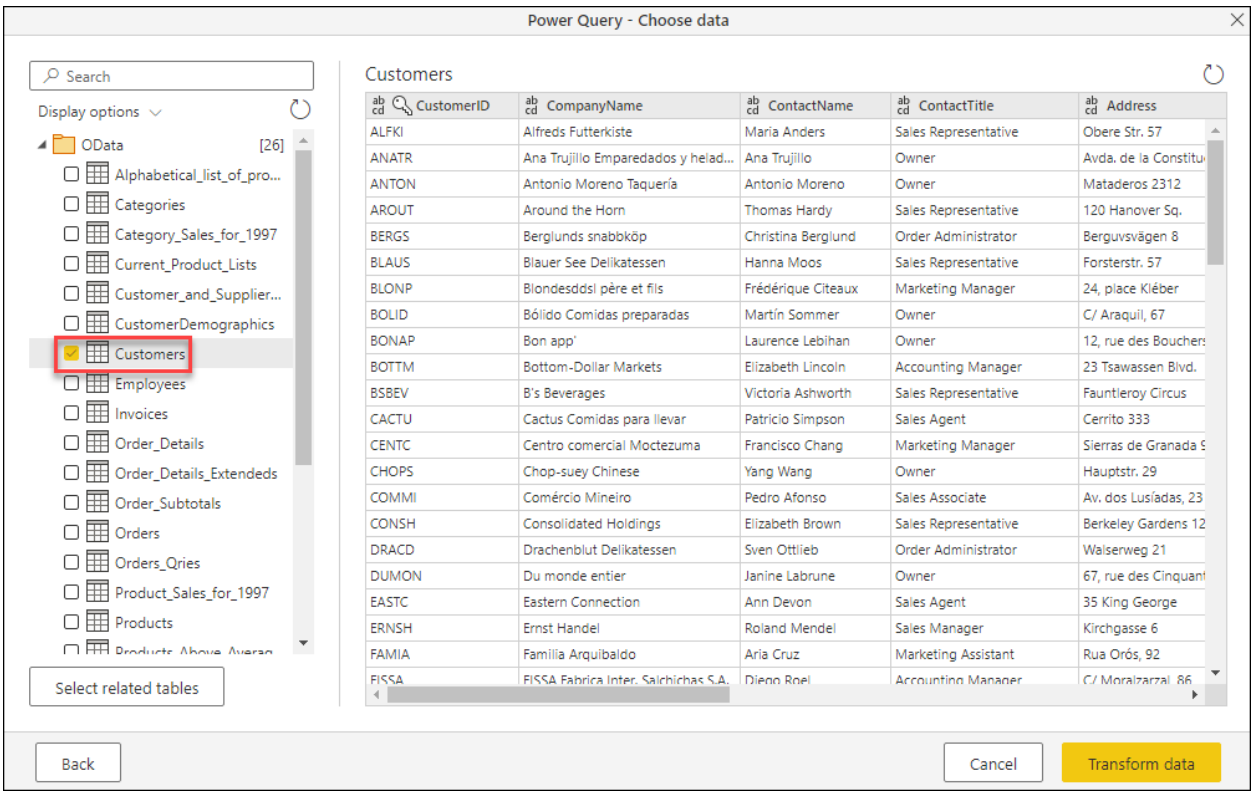
Once you select this connector, the screen displays the connection settings and credentials.

- For **URL**, enter the URL to the Northwind OData feed shown in the previous section.
- For **On-premises data gateway**, leave as none.
- For **Authentication kind**, leave as anonymous.

Select the **Next** button.



The **Navigator** now opens, where you select the tables you want to connect to from the data source. Select the **Customers** table to load a preview of the data, and then select **Transform data**.



The dialog then loads the data from the Customers table into the Power Query editor.

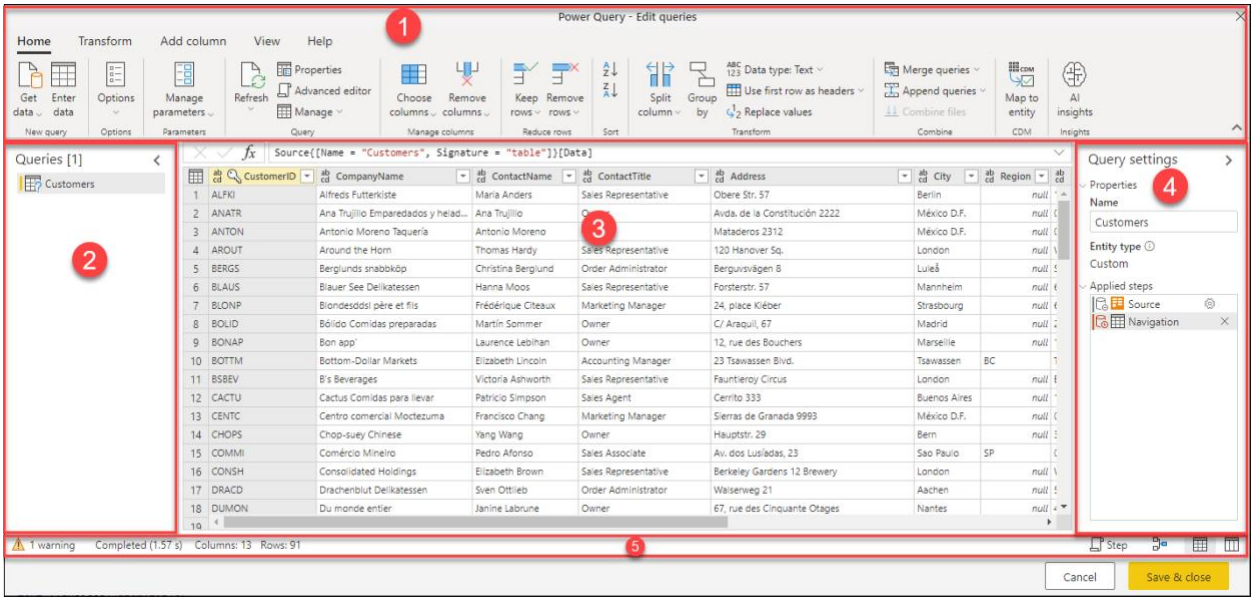
The above experience of connecting to your data, specifying the authentication method, and selecting the specific object or table to connect to is called the *get data experience* and is documented with further detail in the Getting data article.

Note

To learn more about the OData feed connector, go to **OData feed**.

The Power Query editor user experience

The Power Query editor represents the Power Query user interface, where you can add or modify queries, manage queries by grouping or adding descriptions to query steps, or visualize your queries and their structure with different views. The Power Query user interface has five distinct components.



1. **Ribbon**: the ribbon navigation experience, which provides multiple tabs to add transforms, select options for your query, and access different ribbon buttons to complete various tasks.
2. **Queries pane**: a view of all your available queries.
3. **Current view**: your main working view, that by default, displays a preview of the data for your query. You can also enable the diagram view along with the data preview view. You can also switch between the schema view and the data preview view while maintaining the diagram view.
4. **Query settings**: a view of the currently selected query with relevant information, such as query name, query steps, and various indicators.

5. **Status bar**: a bar displaying relevant important information about your query, such as execution time, total columns and rows, and processing status. This bar also contains buttons to change your current view.

Note

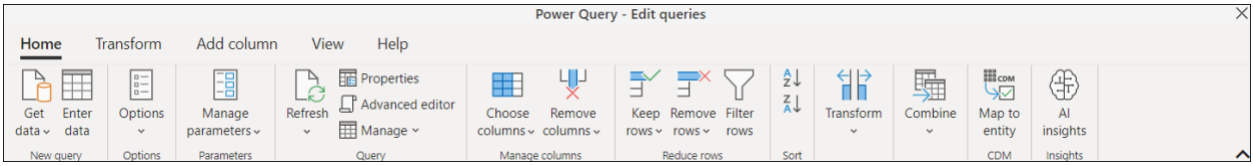
The schema and diagram view are currently only available in Power Query Online.

Using the Power Query editor

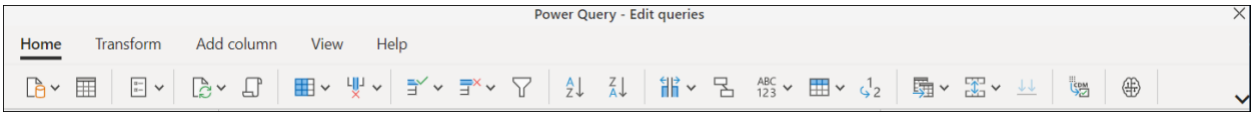
In this section, you'll begin transforming your data using Power Query. But before you start working on transforming the data, we'll discuss some of the UI panes that can be expanded or collapsed depending on their context. Selecting the appropriate panes lets you focus on the view that matters the most to you. We'll also discuss the different views that are available in the Power Query UI.

The ribbon

The ribbon is the component where you'll find most of the transforms and actions that you can do in the Power Query editor. It has multiple tabs, whose values depend on the product integration. Each of the tabs provides specific buttons and options, some of which might be redundant across the whole Power Query experience. These buttons and options provide you with easy access to the transforms and actions that you may need.



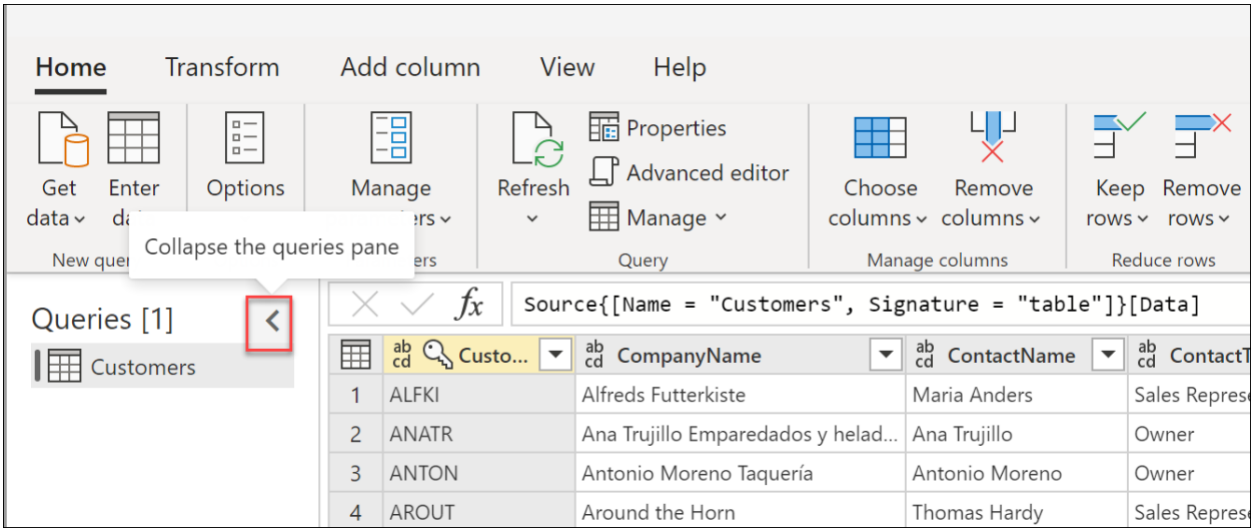
The Power Query interface is responsive and tries to adjust your screen resolution to show you the best experience. In scenarios where you'd like to use a compact version of the ribbon, there's also a collapse button at the bottom-right corner of the ribbon to help you switch to the compact ribbon.



You can switch back to the standard ribbon view by simply clicking on the expand icon at the bottom-right corner of the ribbon

Expand and collapse panes

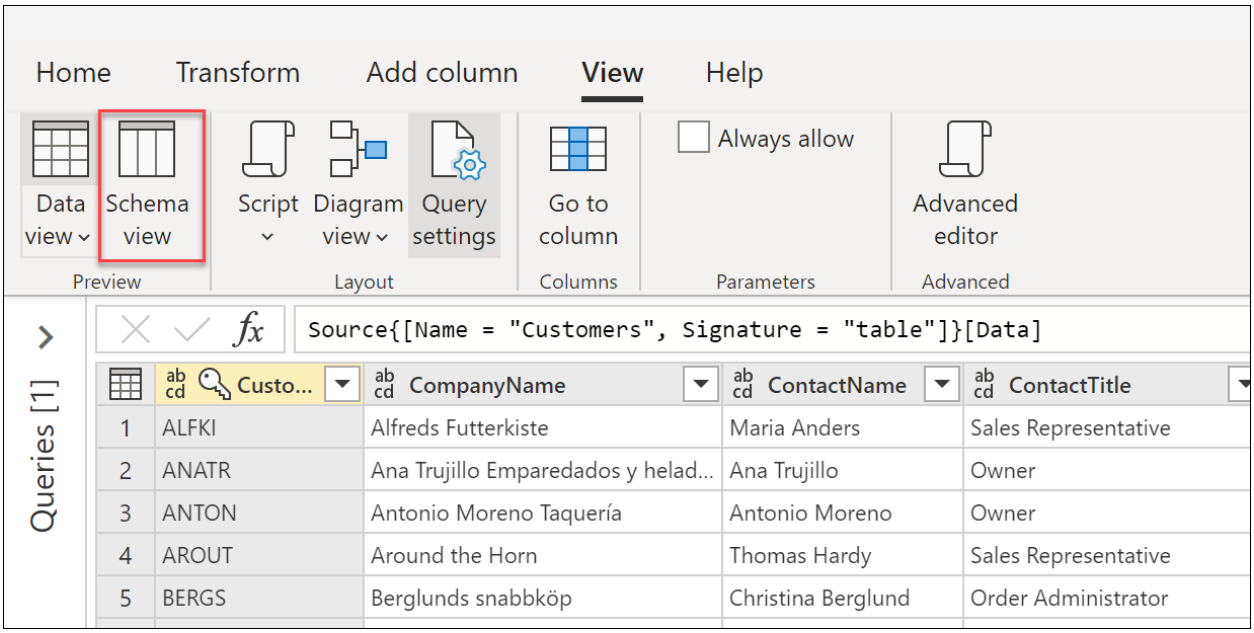
You'll notice that throughout the Power Query user interface there are icons that help you collapse or expand certain views or sections. For example, there's an icon on the top right-hand corner of the Queries pane that collapses the queries pane when selected, and expands the pane when selected again.



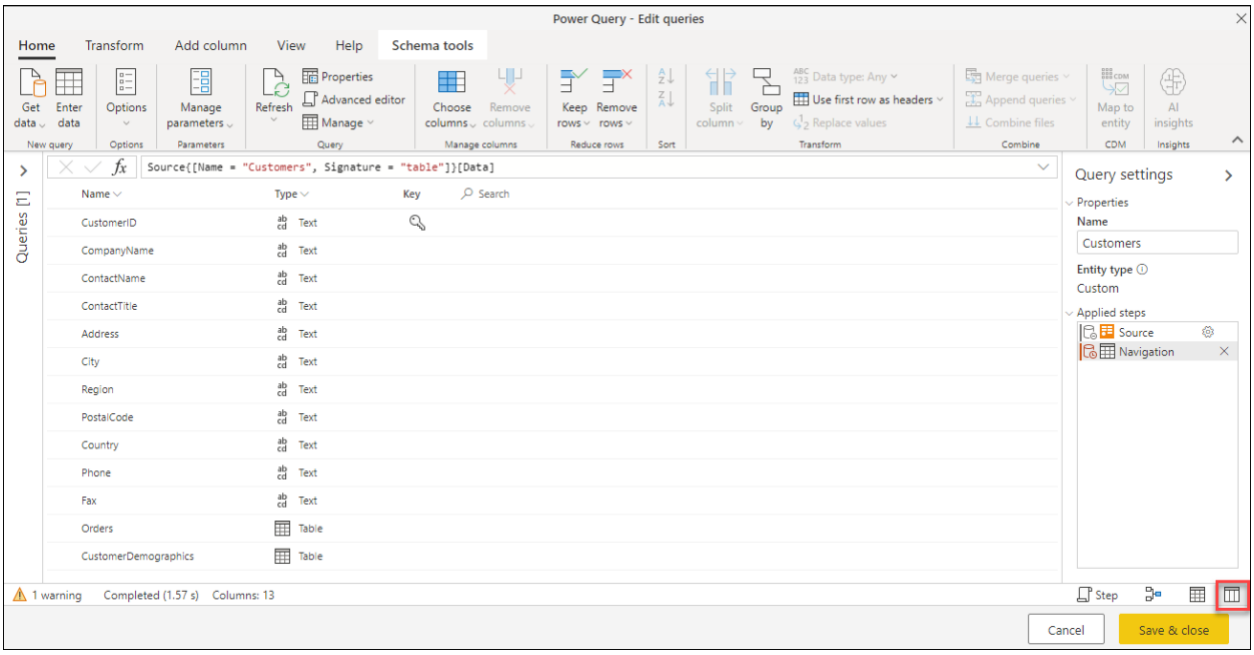
Switch between views

Apart from being able to collapse certain panes and sections in the Power Query user interface, you can also switch what views are displayed. To switch views, go to the **View** tab in the ribbon and you'll find the **Preview** and **Layout** groups, which control how the Power Query user interface will look.

You're encouraged to try all of these options to find the view and layout that you feel most comfortable working with. As an example, select **Schema view** from the ribbon.



The right side of the status bar also contains icons for the diagram, data, and schema views. You can use these icons to change between views. You can also use these icons to enable or disable the view of your choice.



What is schema view

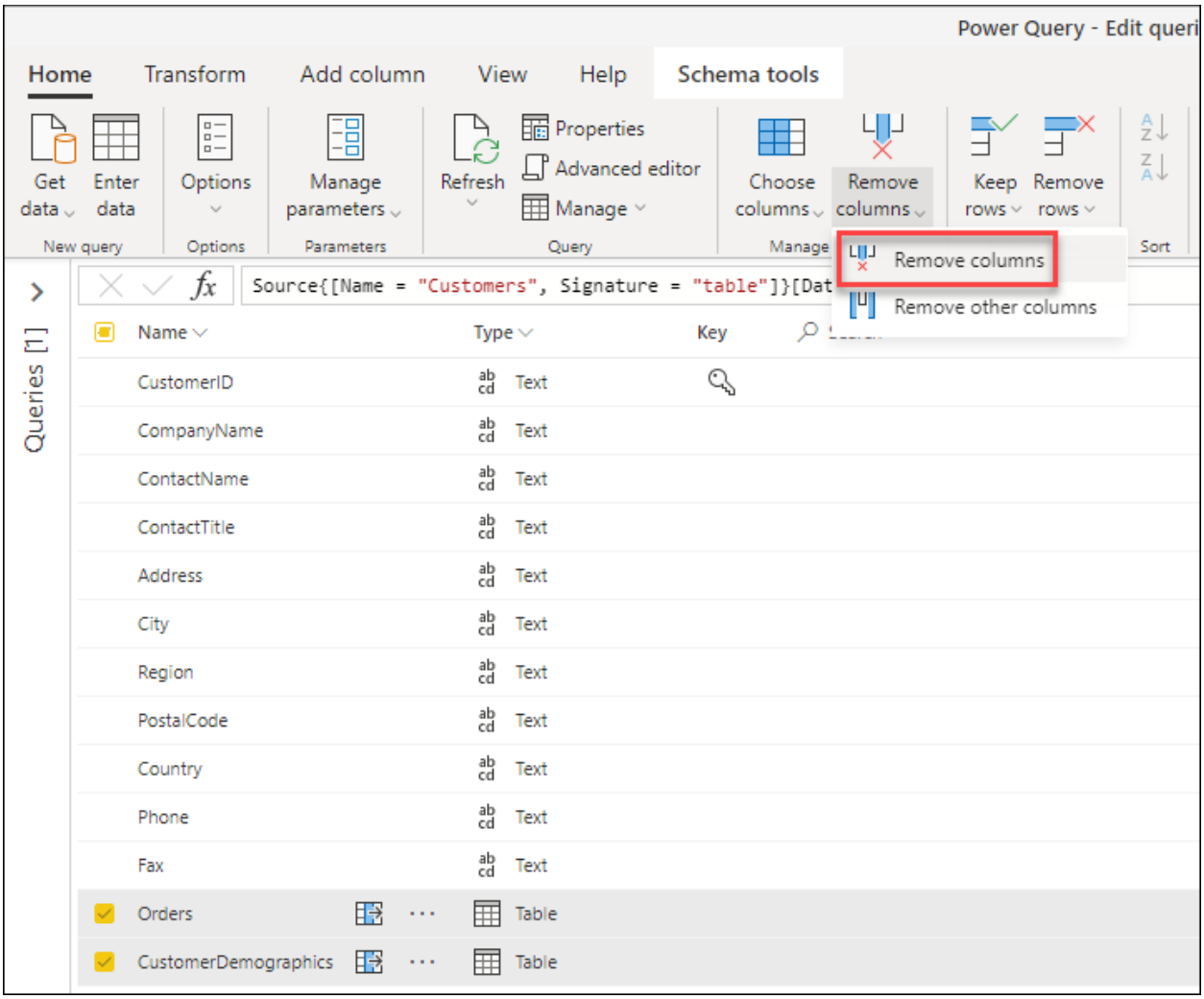
The schema view offers you a quick and straightforward way to interact only with the components of the schema for your table, such as the column names and data types. We recommend the schema view when you want to do schema-related actions, such as

removing columns, renaming columns, changing column data types, reordering columns, or duplicating columns.

Note

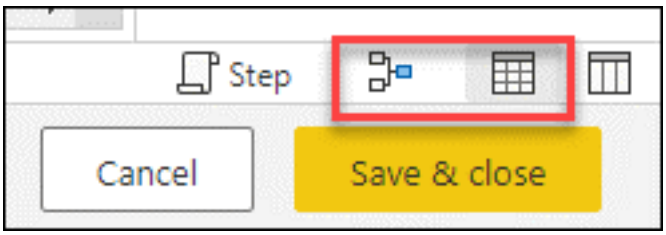
To learn more about schema view, go to **Using Schema view**.

For example, in schema view, select the check mark next to the **Orders** and **CustomerDemographics** columns, and from the ribbon select the **Remove columns** action. This selection applies a transformation to remove these columns from your data.

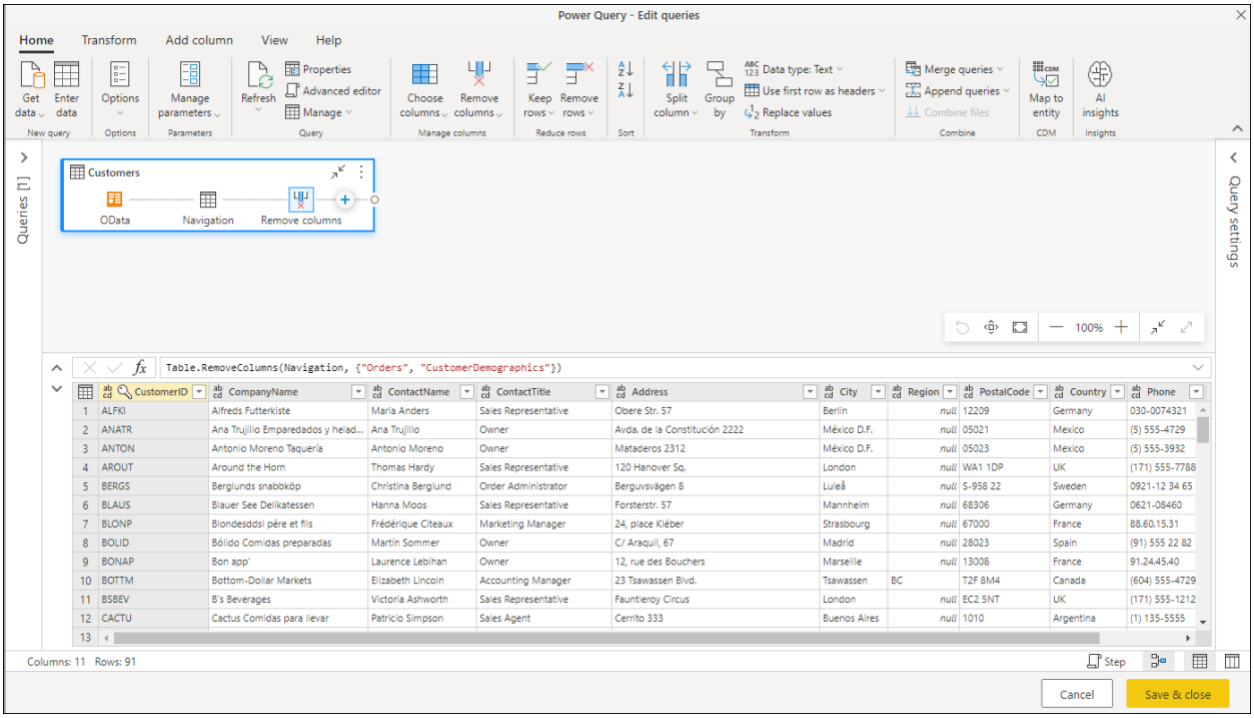


What is diagram view

You can now switch back to the data preview view and enable diagram view to use a more visual perspective of your data and query.



The diagram view helps you visualize how your query is structured and how it might interact with other queries in your project. Each step in your query has a distinct icon to help you recognize the transform that was used. There are also lines that connect steps to illustrate dependencies. Since both data preview view and diagram view are enabled, the diagram view displays on top of the data preview.

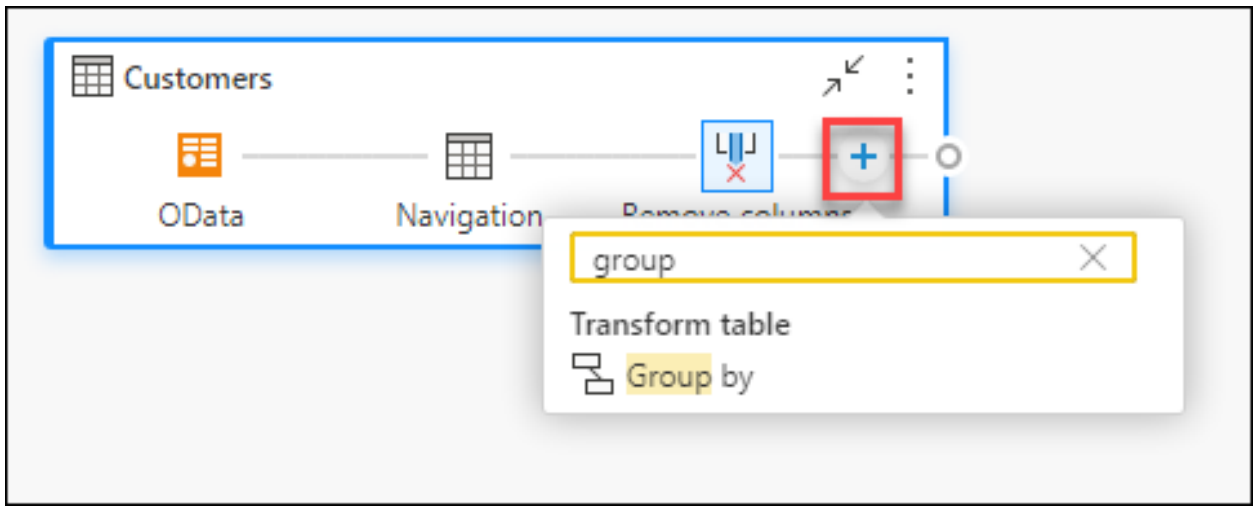


Note

To learn more about diagram view, go to **Diagram view**.

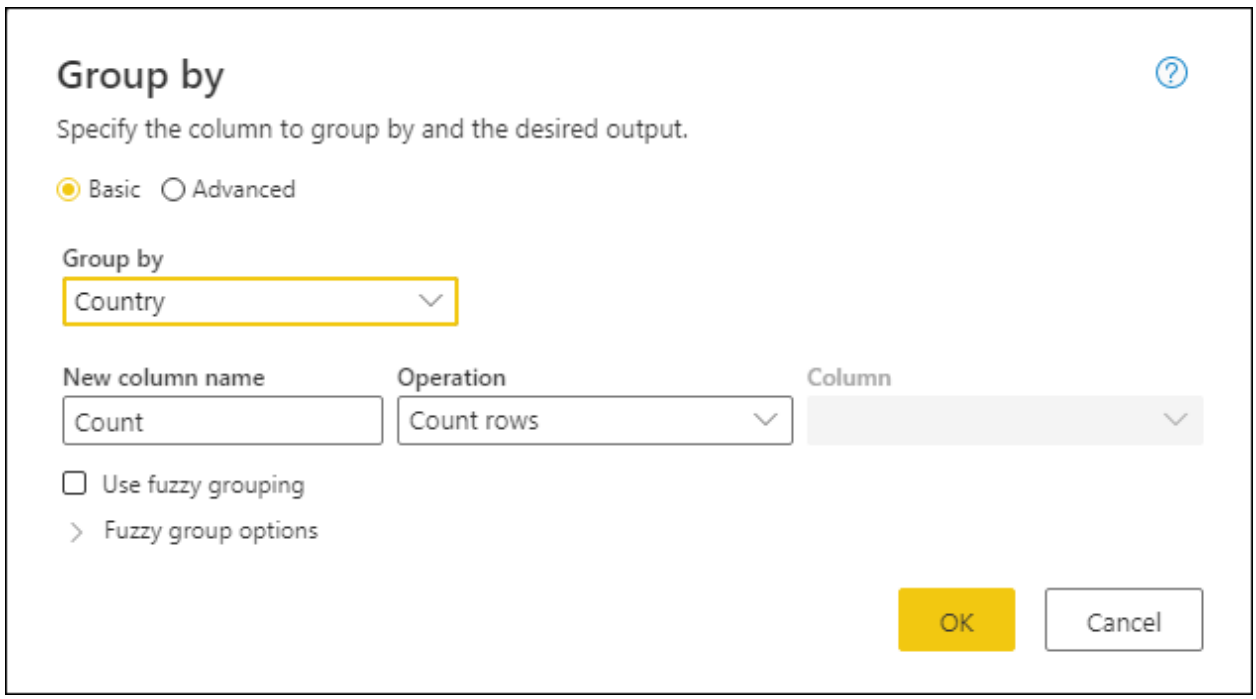
Begin transforming your data

With diagram view enabled, select the plus sign. You can search for a new transform to add to your query. Search for **Group by** and select the transform.



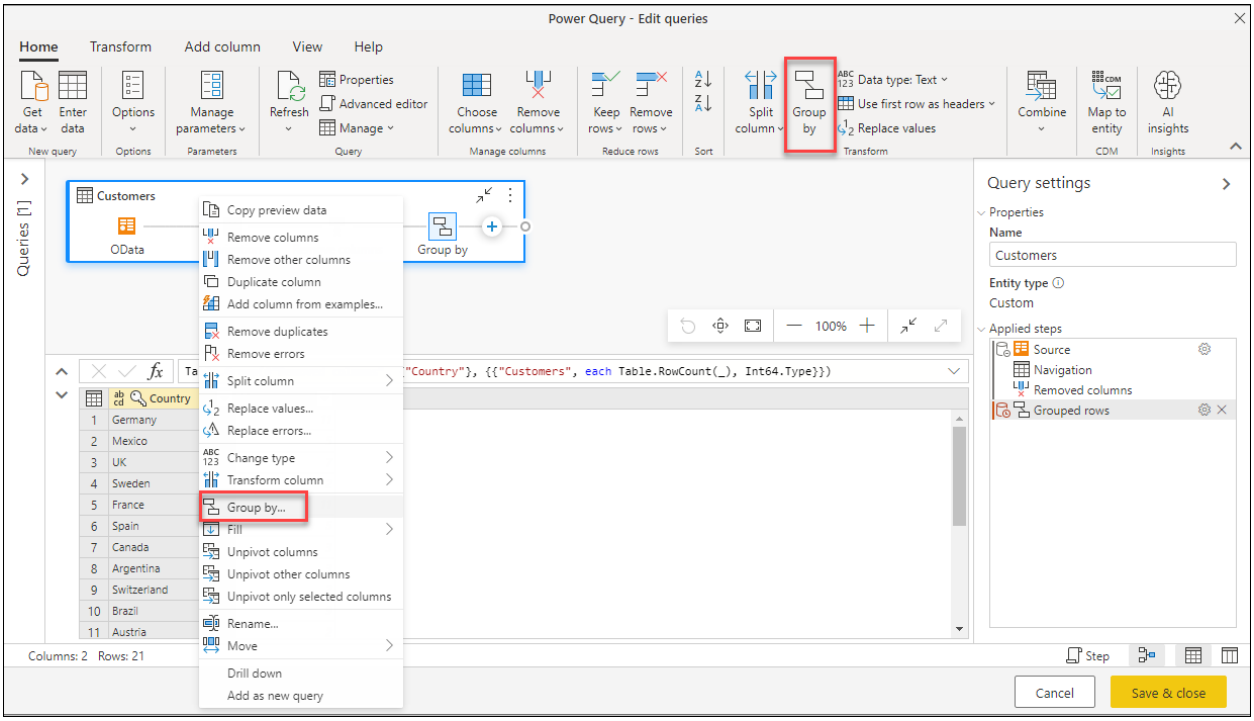
The **Group by** dialog then appears. You can set the **Group by** operation to group by the country and count the number of customer rows per country.

1. Keep the **Basic** radio button selected.
2. Select **Country** to group by.
3. Select **Customers** and **Count rows** as the column name and operation respectively.



Select **OK** to perform the operation. Your data preview refreshes to show the total number of customers by country.

An alternative way to launch the **Group by** dialog would be to use the **Group by** button in the ribbon or by right-clicking the **Country** column.

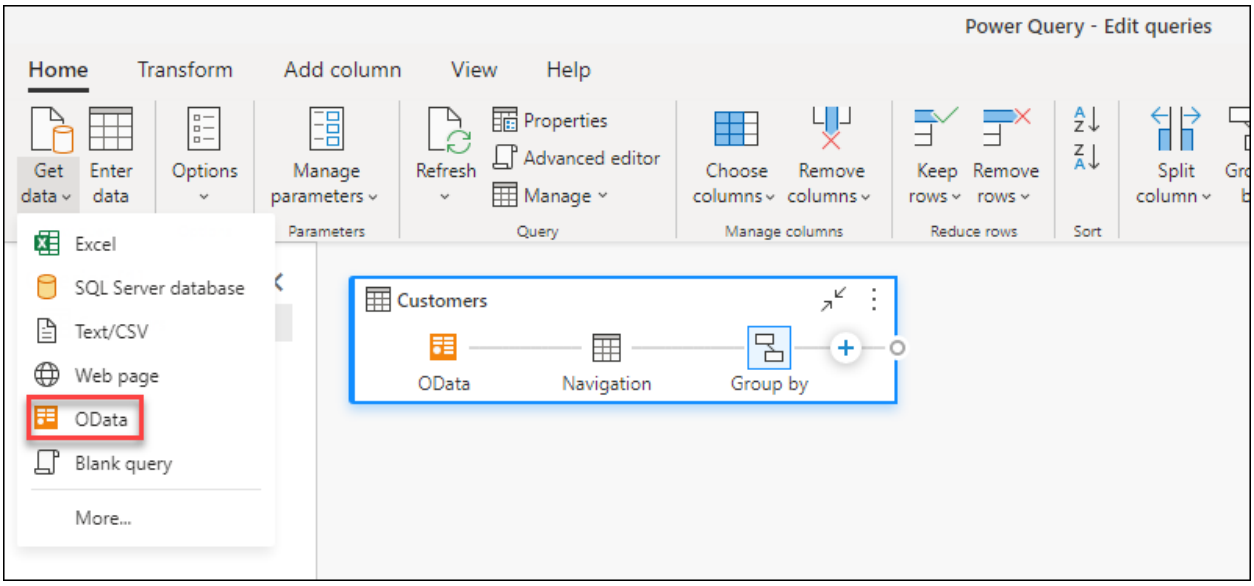


For convenience, transforms in Power Query can often be accessed from multiple places, so users can opt to use the experience they prefer.

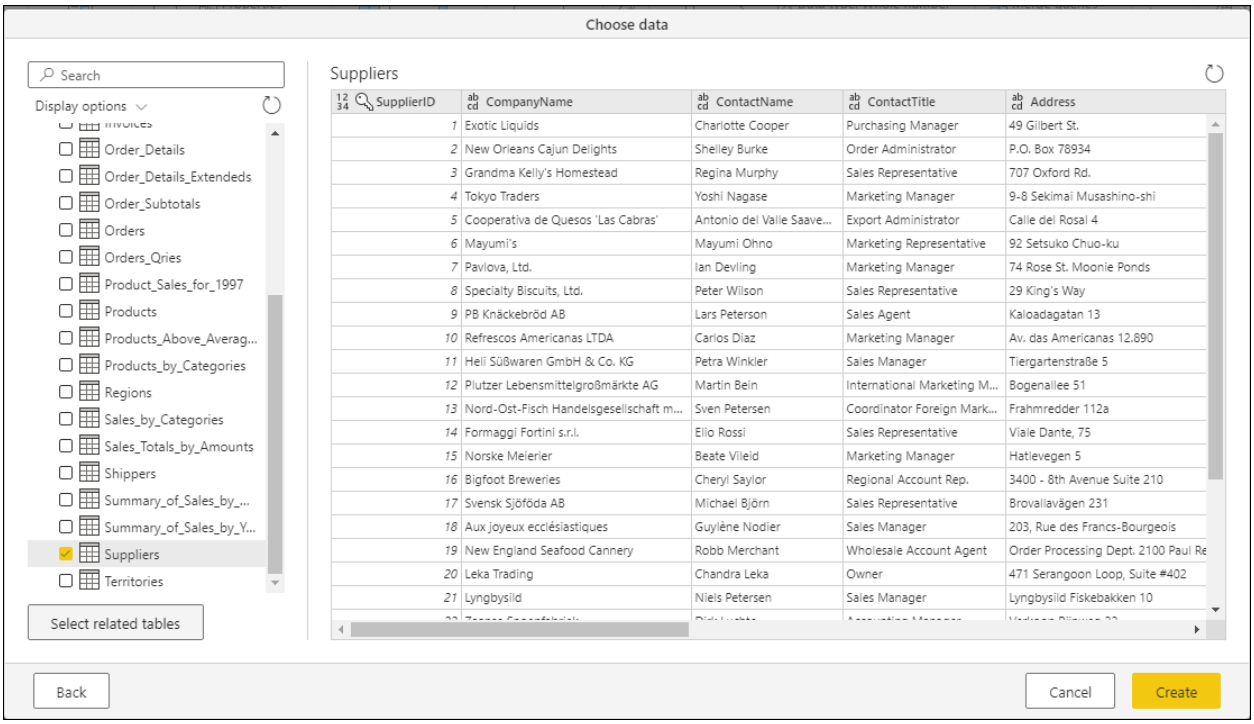
Adding a new query

Now that you have a query that provides the number of customers per country, you can add context to this data by finding the total number of suppliers for each territory.

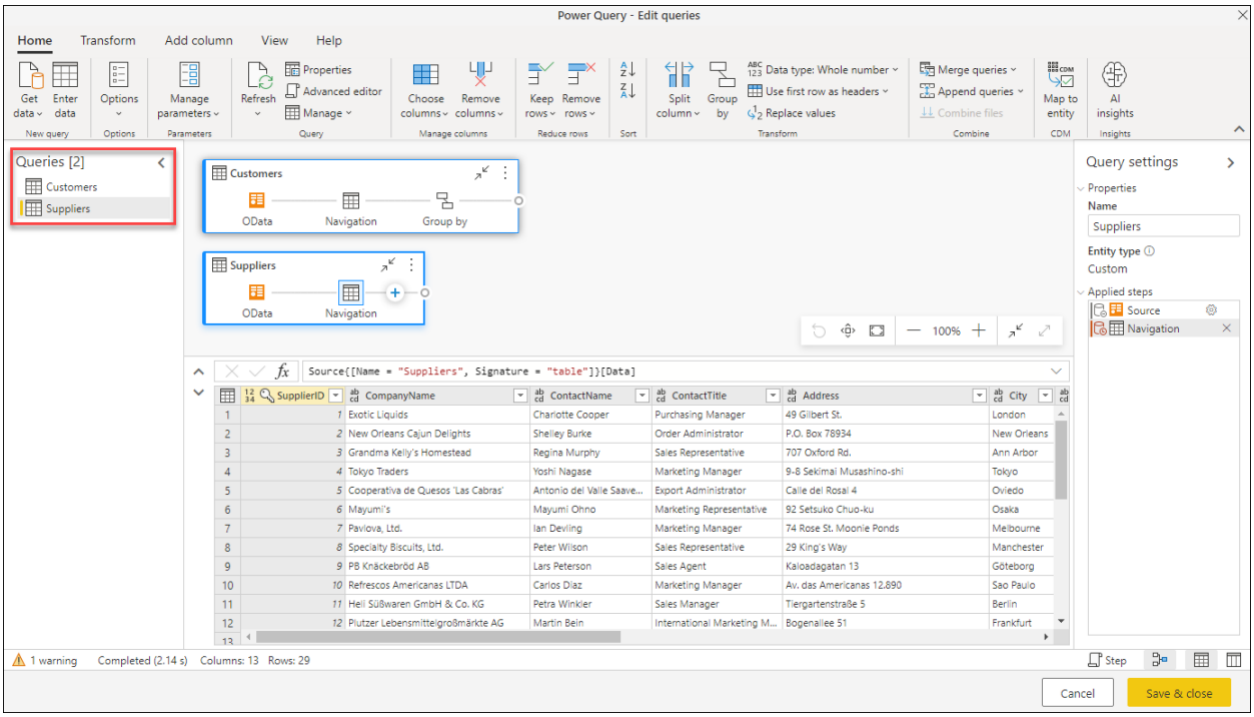
First, you'll need to add the **Suppliers** data. Select **Get Data** and from the drop-down menu, and then select **OData**.



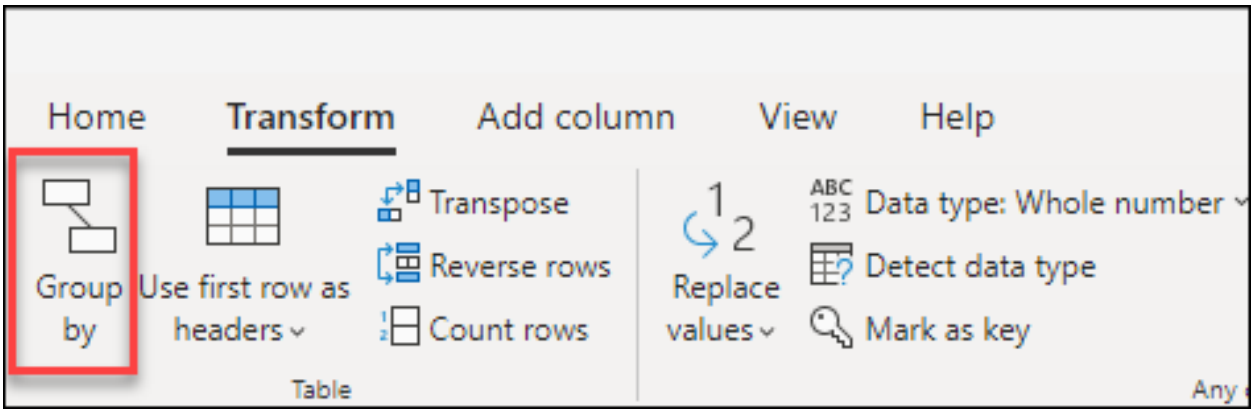
The OData connection experience reappears. Enter the connection settings as described in Connect to an OData feed to connect to the Northwind OData feed. In the **Navigator** experience, search for and select the **Suppliers** table.



Select **Create** to add the new query to the Power Query editor. The queries pane should now display both the **Customers** and the **Suppliers** query.

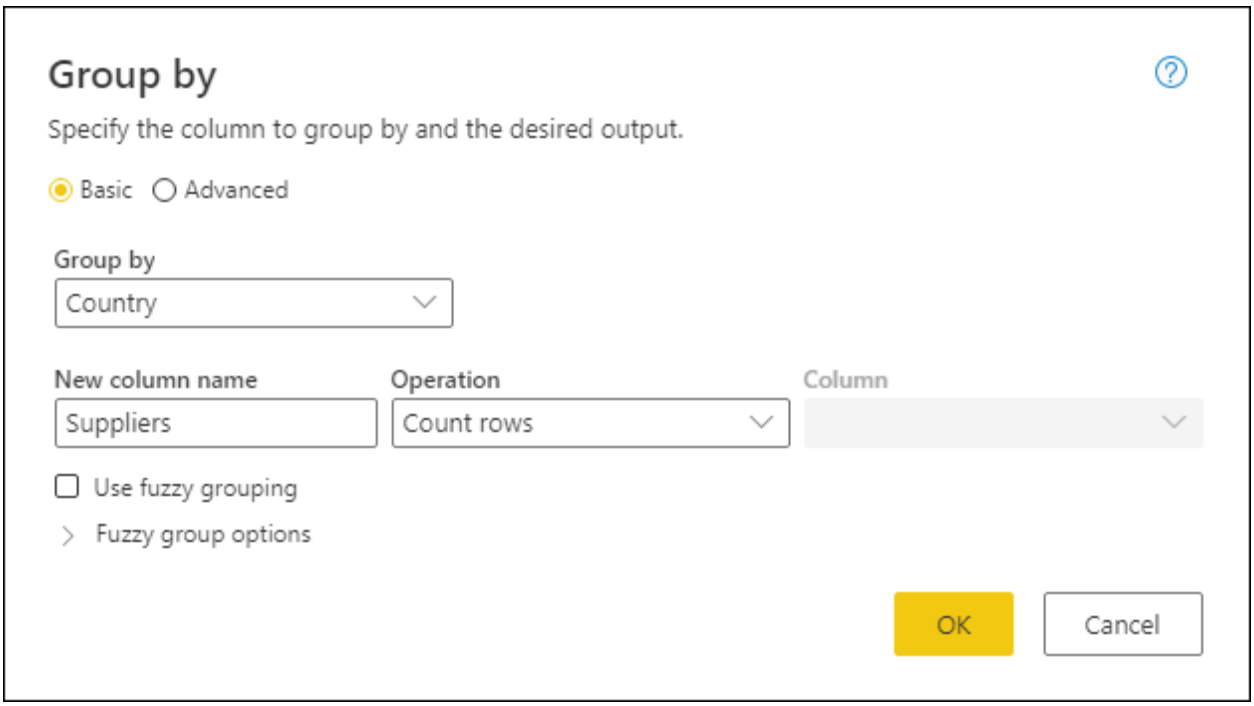


Open the **Group by** dialog again, this time by selecting the **Group by** button on the ribbon under the **Transform** tab.



In the **Group by** dialog, set the **Group by** operation to group by the country and count the number of supplier rows per country.

1. Keep the **Basic** radio button selected.
2. Select **Country** to group by.
3. Select **Suppliers** and **Count rows** as the column name and operation respectively.

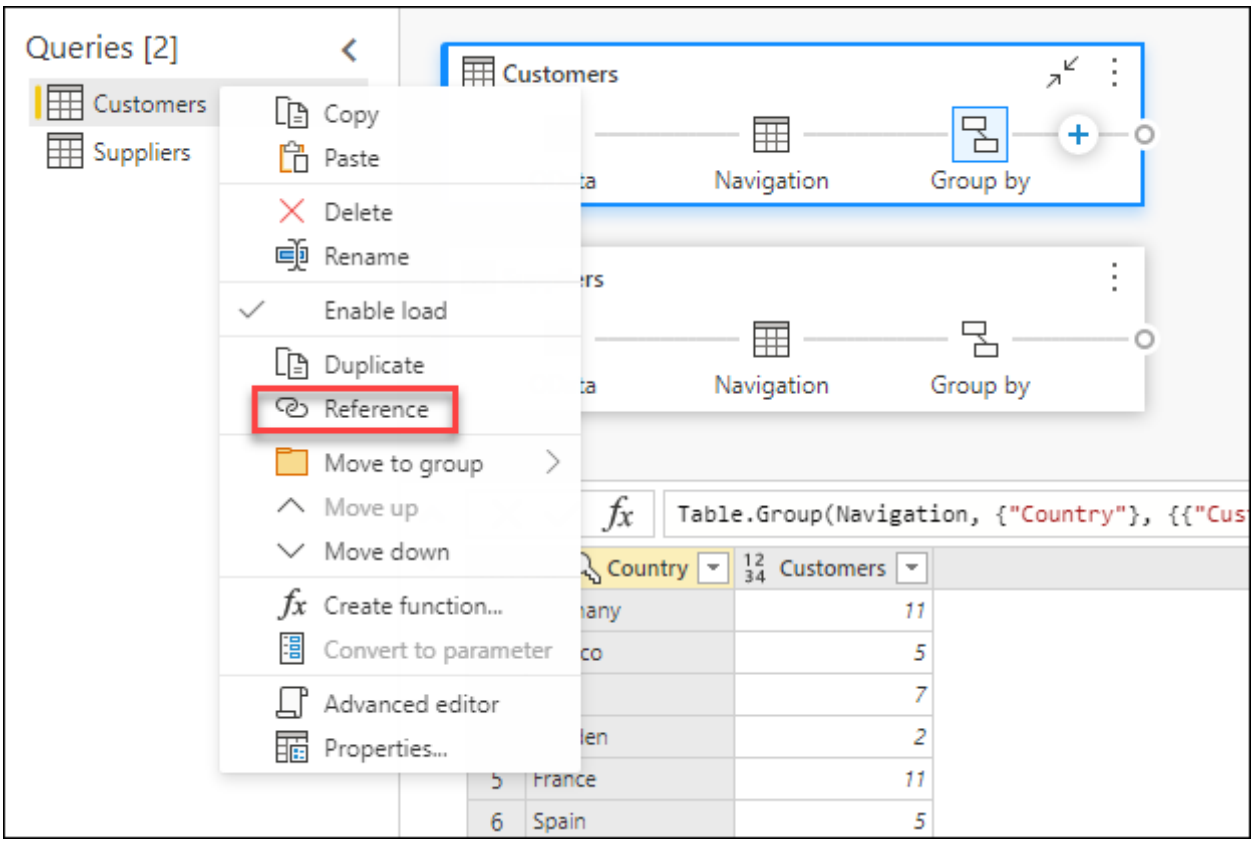


Note

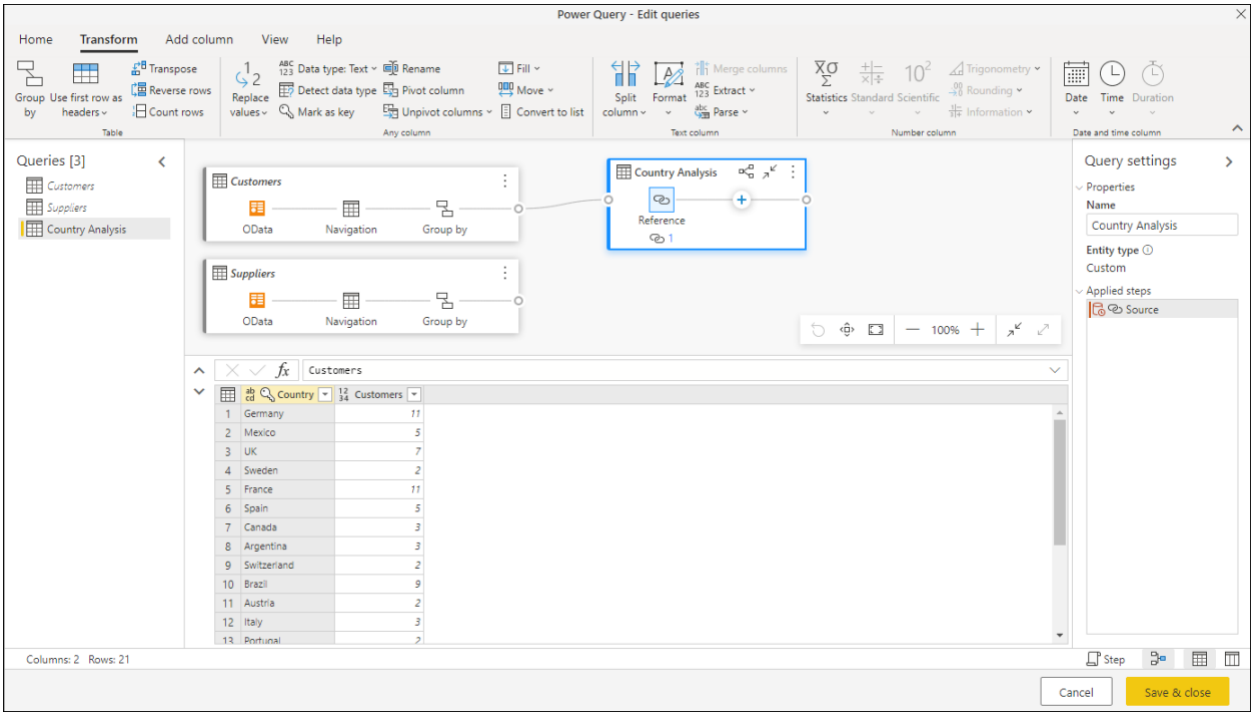
To learn more about the **Group by** transform, go to **Grouping or summarizing rows**.

Referencing queries

Now that you have a query for customers and a query for suppliers, your next goal is to combine these queries into one. There are many ways to accomplish this, including using the **Merge** option in the **Customers** table, duplicating a query, or referencing a query. For this example, you'll create a reference by right-clicking the **Customers** table and selecting **Reference**, which effectively creates a new query that references the **Customers** query.



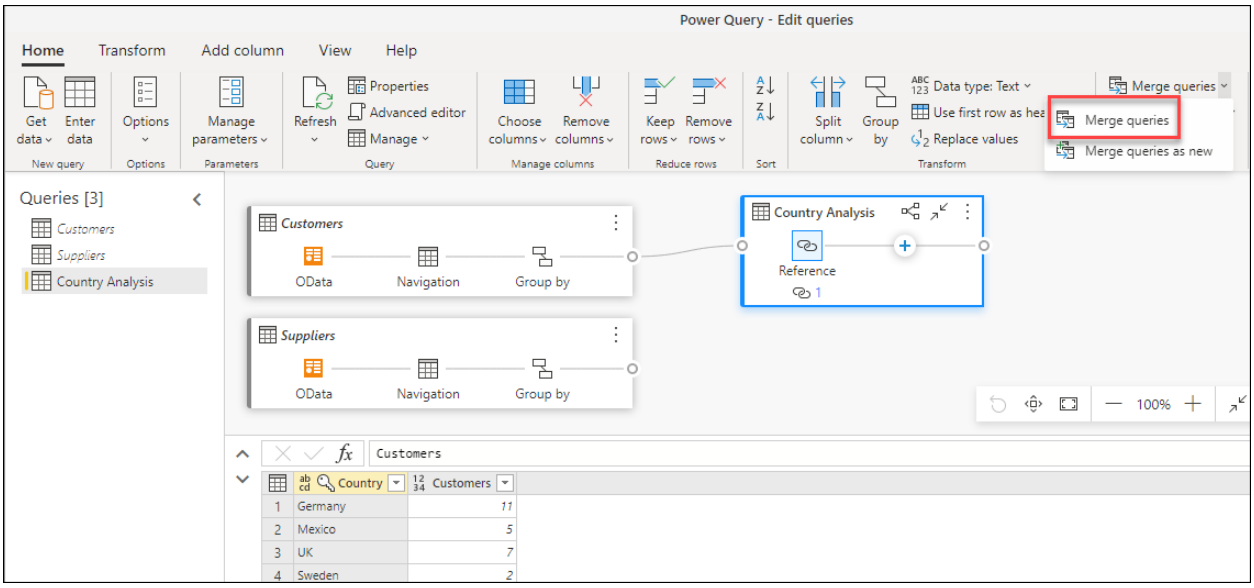
After creating this new query, change the name of the query to **Country Analysis** and disable the load of the **Customers** table by unmarking the **Enable load** option from the **Suppliers** query.



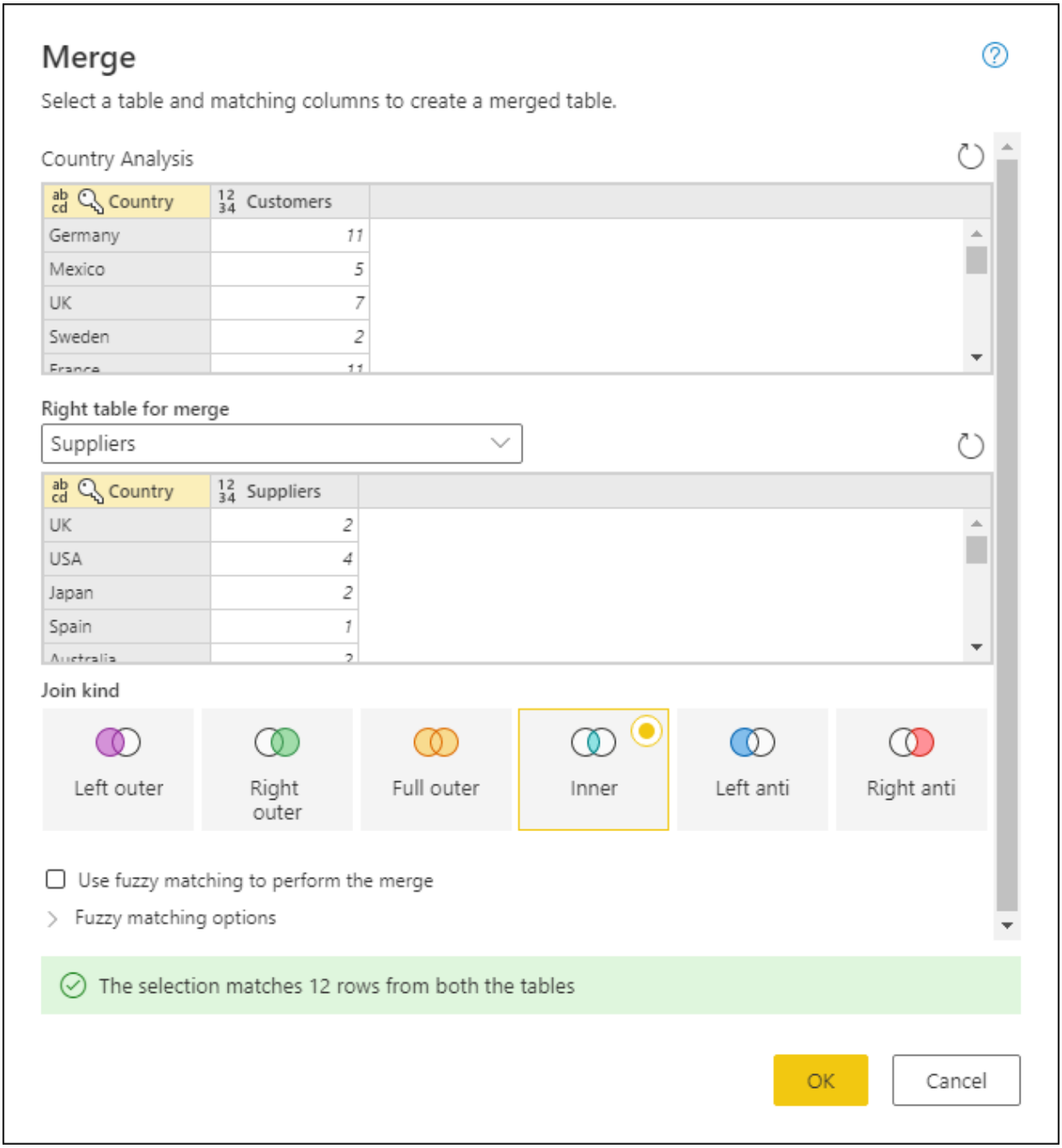
Merging queries

A **merge queries** operation joins two existing tables together based on matching values from one or multiple columns. In this example, the goal is to join both the **Customers** and **Suppliers** tables into one table only for the countries that have both **Customers** and **Suppliers**.

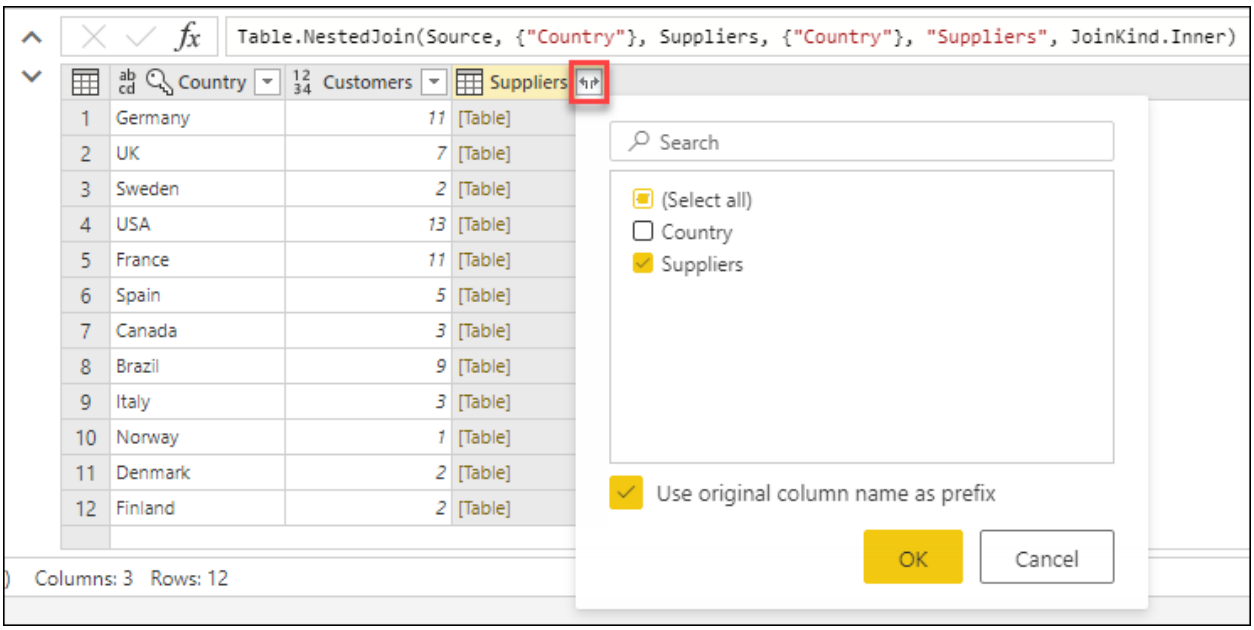
Inside the **Country Analysis** query, select the **Merge queries** option from the **Home** tab in the ribbon.



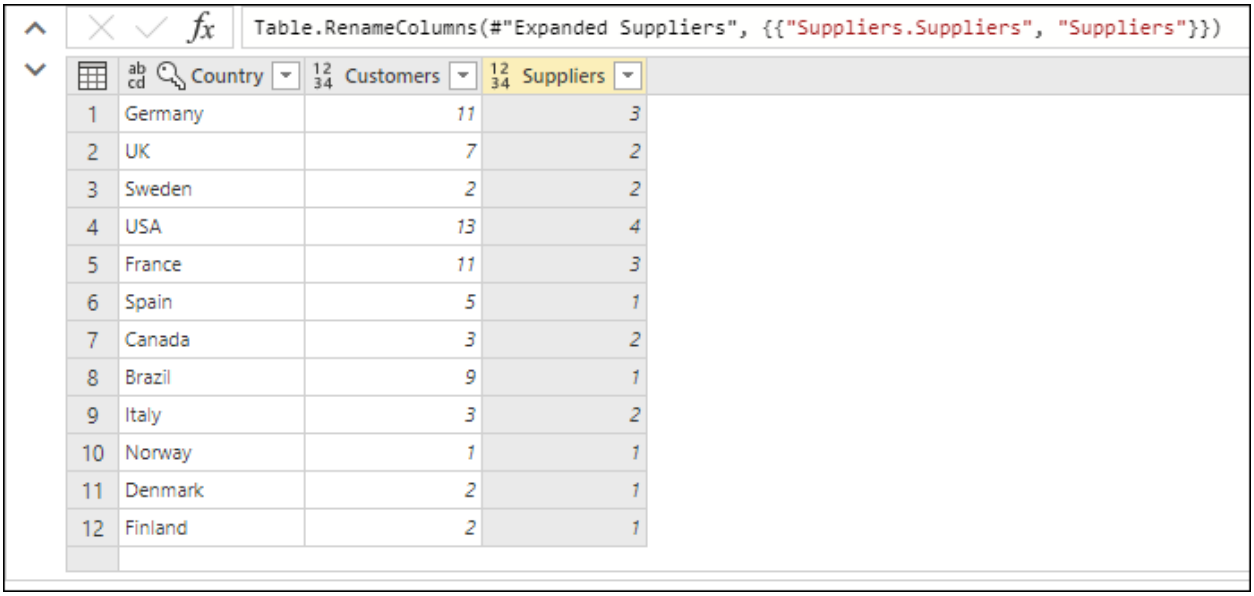
A new dialog for the **Merge** operation appears. You can then select the query to merge with your current query. Select the **Suppliers** query and select the **Country** field from both queries. Finally, select the **Inner** join kind, as you only want the countries where you have **Customers** and **Suppliers** for this analysis.



After selecting the **OK** button, a new column is added to your **Country Analysis** query that contains the data from the **Suppliers** query. Select the icon next to the **Suppliers** field, which displays a menu where you can select which fields you want to expand. Select only the **Suppliers** field, and then select the **OK** button.



The result of this **expand** operation is a table with only 12 rows. Rename the **Suppliers.Suppliers** field to just **Suppliers** by double-clicking the field name and entering the new name.



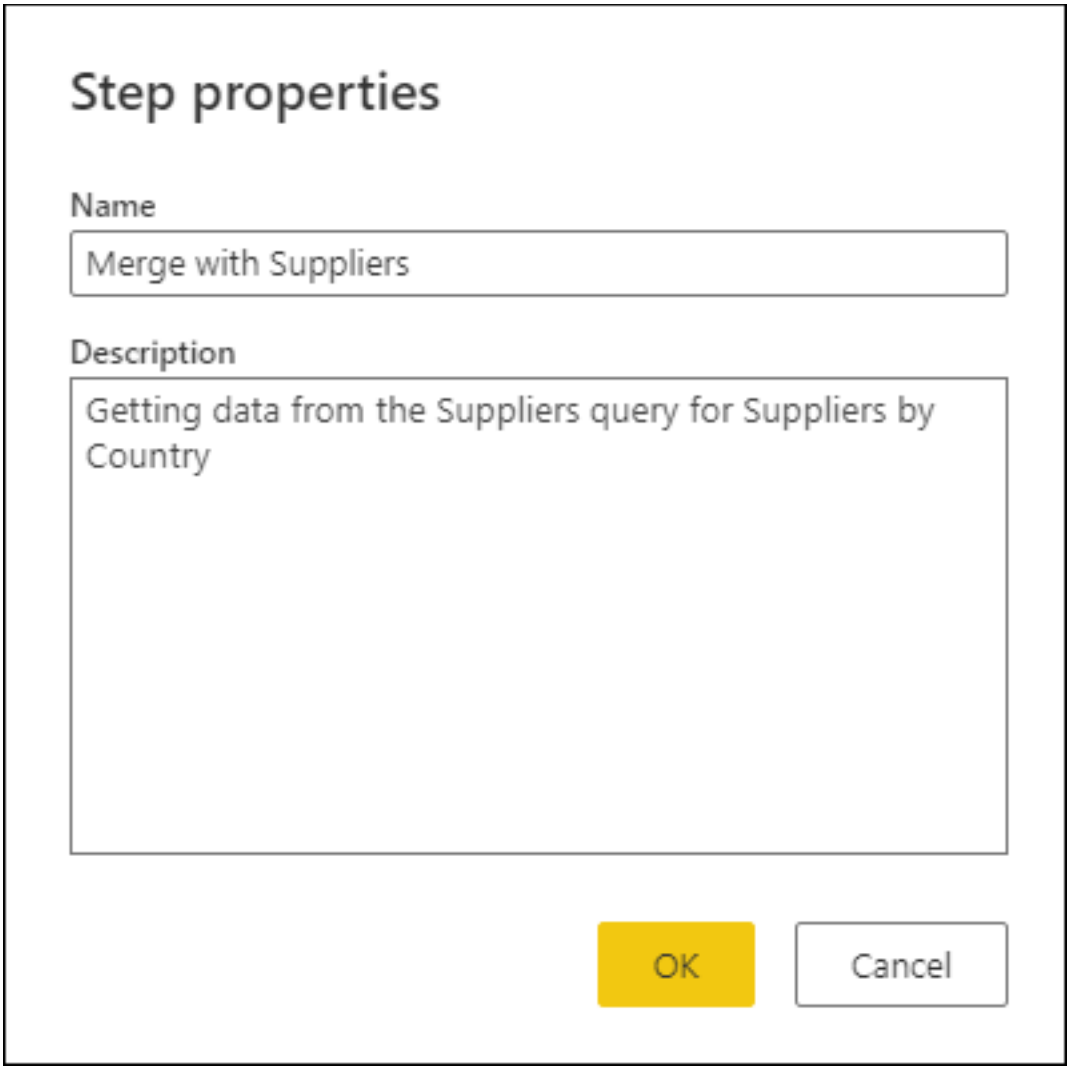
Note

To learn more about the **Merge queries** feature, go to **Merge queries overview**.

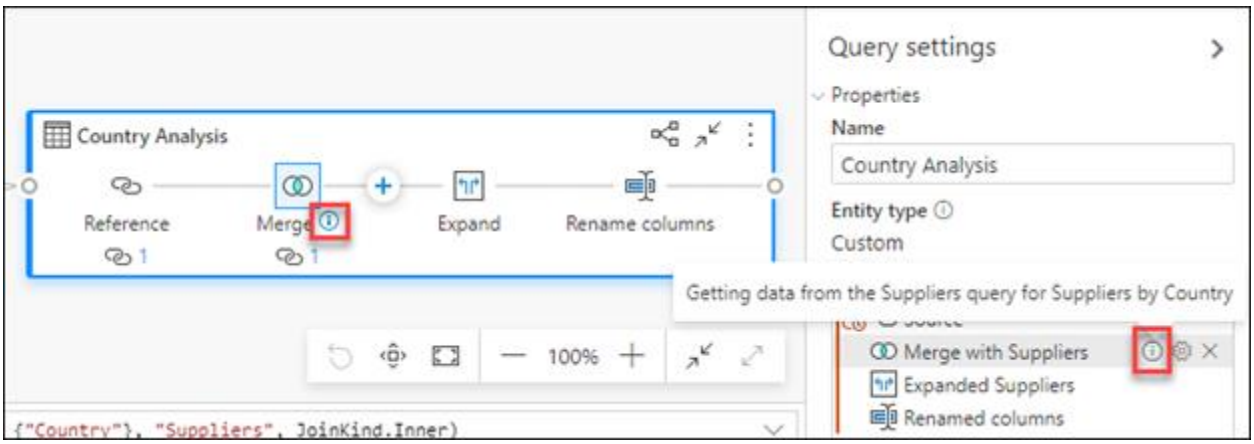
Applied steps

Every transformation that is applied to your query is saved as a step in the **Applied steps** section of the query settings pane. If you ever need to check how your query is transformed from step to step, you can select a step and preview how your query resolves at that specific point.

You can also right-click a query and select the **Properties** option to change the name of the query or add a description for the query. For example, right-click the **Merge queries** step from the **Country Analysis** query and change the name of the query to be **Merge with Suppliers** and the description to be **Getting data from the Suppliers query for Suppliers by Country**.



This change adds a new icon next to your step that you can hover over to read its description.



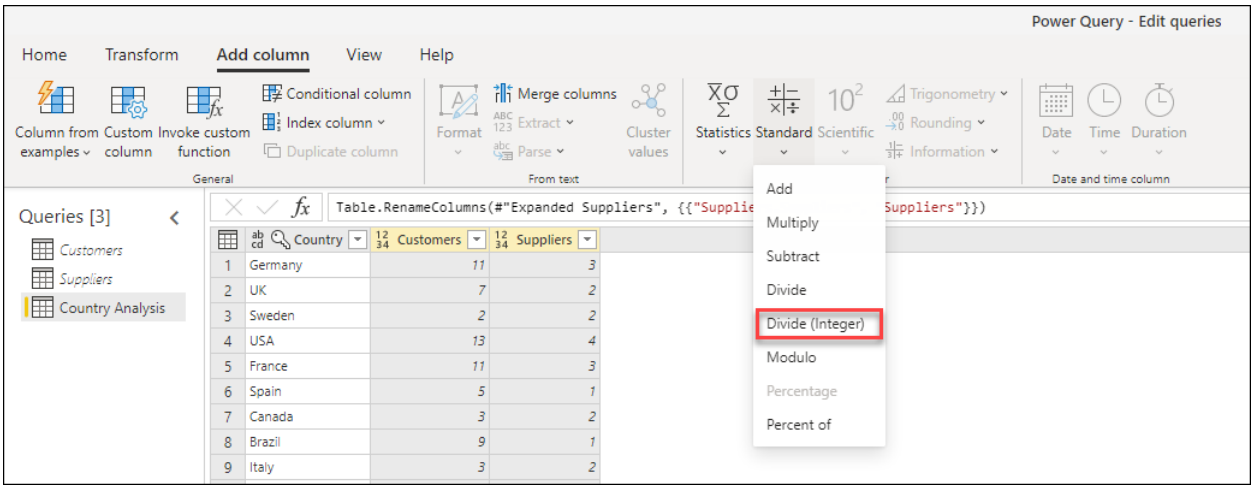
Note

To learn more about **Applied steps**, go to **Using the Applied Steps list**.

Before moving on to the next section, disable the **Diagram view** to only use the **Data preview**.

Adding a new column

With the data for customers and suppliers in a single table, you can now calculate the ratio of customers-to-suppliers for each country. Select the last step of the **Country Analysis** query, and then select both the **Customers** and **Suppliers** columns. In the **Add column tab** in the ribbon and inside the **From number** group, select **Standard**, and then **Divide (Integer)** from the dropdown.

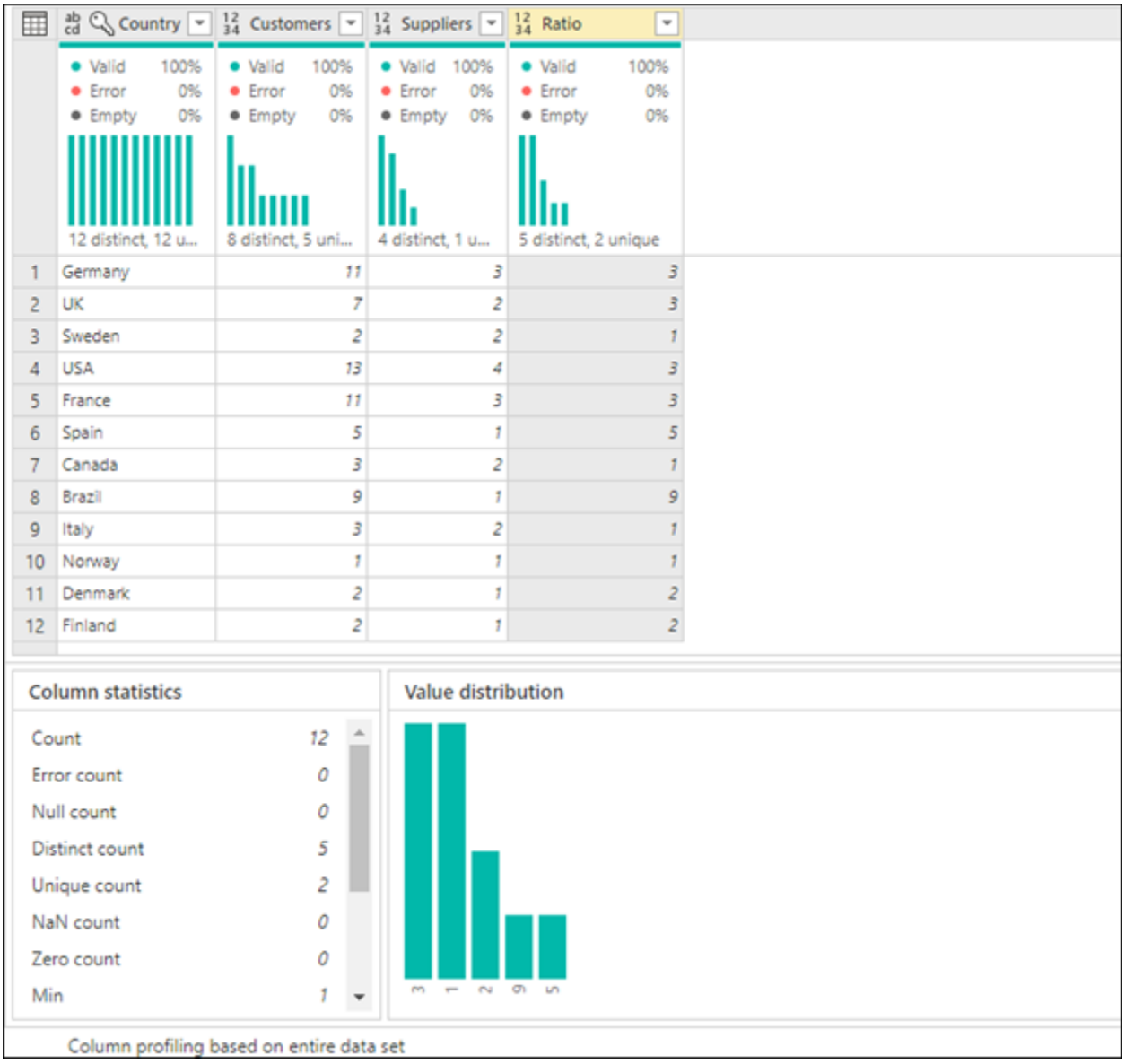


This change creates a new column called **Integer-division** that you can rename to **Ratio**. This change is the final step of your query, and provides the customer-to-supplier ratio for the countries where the data has customers and suppliers.

Data profiling

Another Power Query feature that can help you better understand your data is **Data Profiling**. By enabling the data profiling features, you'll get feedback about the data inside your query fields, such as value distribution, column quality, and more.

We recommended that you use this feature throughout the development of your queries, but you can always enable and disable the feature at your convenience. The following image shows all the data profiling tools enabled for your **Country Analysis** query.

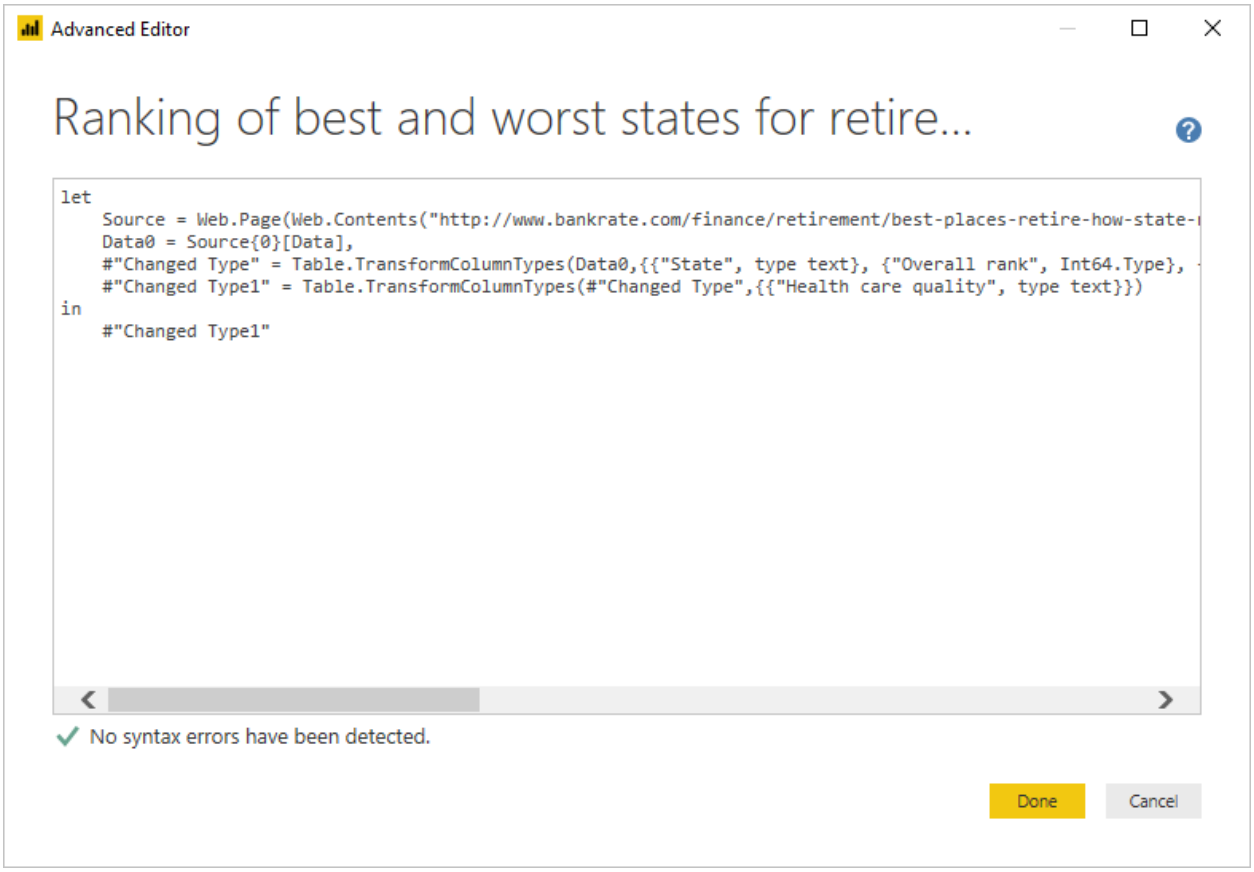


Note

To learn more about **Data profiling**, go to **Using the data profiling tools**.

The advanced editor

If you want to observe the code that the Power Query editor is creating with each step, or want to create your own shaping code, you can use the advanced editor. To open the advanced editor, select the **View** tab on the ribbon, and then select **Advanced Editor**. A window appears, showing the existing query code.



You can directly edit the code in the **Advanced Editor** window. The editor indicates if your code is free of syntax errors. To close the window, select the **Done** or **Cancel** button.

Accessing Power Query help

There are various levels of help that can be accessed in the Microsoft apps that use Power Query. This section describes these help levels.

Power Query help in Excel

There are a couple of ways to access Power Query help information that specifically applies to Power Query in Excel. One way to access the online Excel Power Query documentation is to select **File** > **Help** > **Help** in the Power Query editor.

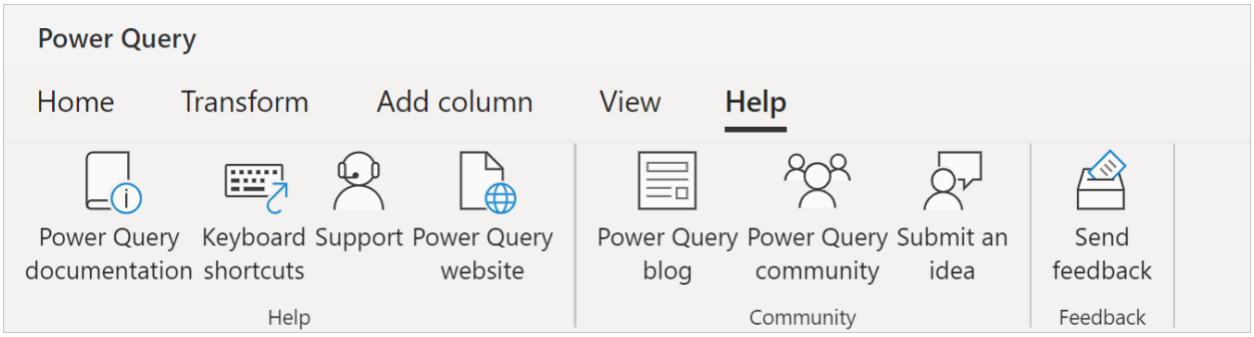
To access the inline Power Query help information in Excel, select the **Help** tab on the Excel ribbon, and then enter **Power Query** in the search text box.

Power Query help in Power BI Desktop

There's a **Help** tab in the Power Query editor running on Power BI Desktop, but at this time all of the options go to locations that supply Power BI information. You can also reach this same Power BI **Help** information in the Power Query editor in Power BI Desktop by selecting **File** > **Help**, and then selecting one of the help options.

Power Query help in Power BI service, Power Apps, Customer Insights, and Azure Data Factory

The Power Query **Help** tab in Power BI service, Power Apps, Dynamics 365 Customer Insights, and Azure Data Factory contains links to important sources of information about Power Query.



The options in the help tab include links to:

- Power Query documentation: Links to the Power Query documentation
- Keyboard shortcuts: Links to the keyboard shortcuts article in the Power Query documentation.
- **Support**: Links to the support website of the Microsoft product that Power Query is currently running in.
- Power Query websight: Links to the Power Query website.
- Power Query blog: Links to the Power Query blog
- Power Query community: Links to the Power Query community.
- **Submit an idea**: Links to the ideas website of the Microsoft product that Power Query is currently running in.
- **Send feedback**: Opens a window in Power Query that asks you to rate your experience with Power Query, and to provide any additional information you would like to supply.

Note

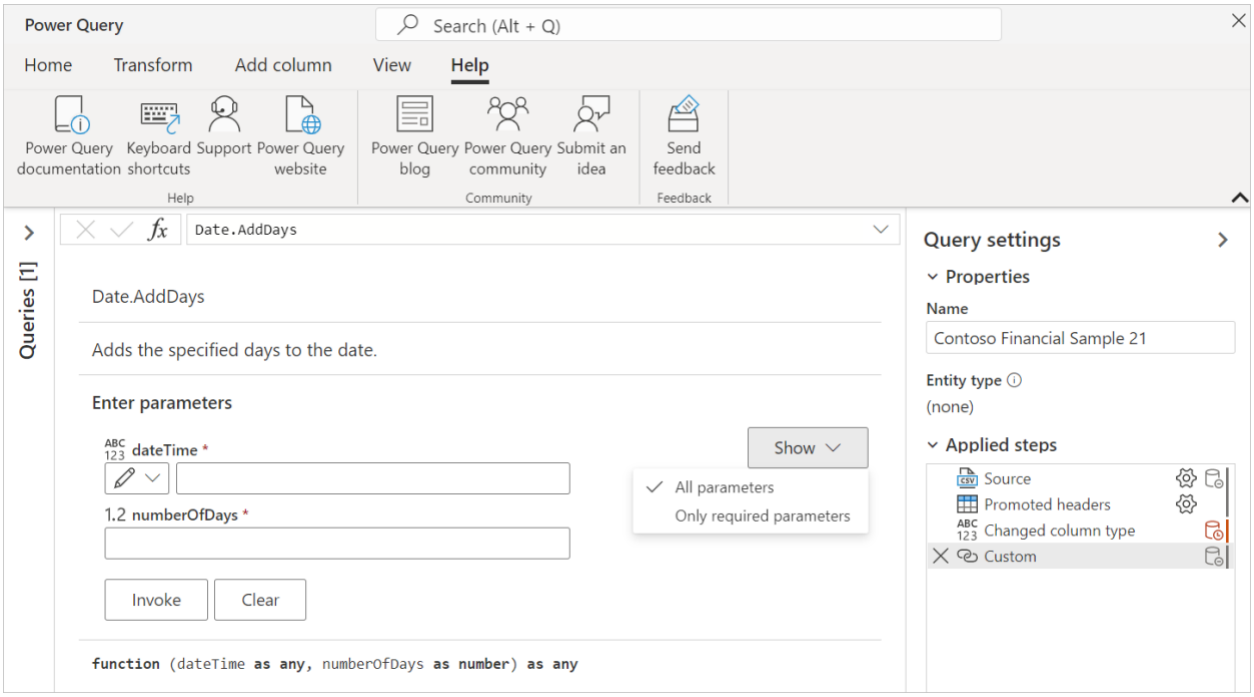
Currently, Azure Analysis Services doesn't contain any inline Power Query help links. However, you can get help for Power Query M functions. More information is contained in the next section.

Help for Power Query M functions

In addition to the Power Query help functions listed in the previous sections, you can also get help for individual M functions from inside the Power Query editor. The following steps describe how to get to this inline help.

1. With the Power Query editor open, select the insert step (fx) button.
2. In the formula bar, enter the name of a function you want to check.
    1. If you are using Power Query Desktop, enter an equal sign, a space, and the name of a function.
    2. If you are using Power Query Online, enter the name of a function.
3. Select the properties of the function.
    1. If you are using Power Query Desktop, in the **Query Settings** pane, under **Properties**, select **All properties**.
    2. If you are using Power Query Online, in the **Query Settings** pane, select **Properties**.

These steps will open the inline help information for your selected function, and let you enter individual properties used by the function.



Summary

In this article, you created a series of queries with Power Query that provides a customer-to-supplier ratio analysis at the country level for the Northwind corporation.

You learned the components of the Power Query user interface, how to create new queries inside the query editor, reference queries, merge queries, understand the applied steps section, add new columns, and how to use the data profiling tools to better understand your data.

Power Query is a powerful tool used to connect to many different data sources and transform the data into the shape you want. The scenarios outlined in this article are examples to show you how you can use Power Query to transform raw data into important actionable business insights.