**Introduction**

Completed100 XP

* 2 minutes

Consider the scenario where you have imported data into Power BI from several different sources and, when you examine the data, it is not prepared for analysis. What could make the data unprepared for analysis?

When examining the data, you discover several issues, including:

* A column called **Employment status** only contains numerals.
* Several columns contain errors.
* Some columns contain null values.
* The customer ID in some columns appears as if it was duplicated repeatedly.
* A single address column has combined street address, city, state, and zip code.

You start working with the data, but every time you create visuals on reports, you get bad data, incorrect results, and simple reports about sales totals are wrong.

Dirty data can be overwhelming and, though you might feel frustrated, you decide to get to work and figure out how to make this data model as pristine as possible.

Fortunately, Power BI and Power Query offer you a powerful environment to clean and prepare the data. Clean data has the following advantages:

* Measures and columns produce more accurate results when they perform aggregations and calculations.
* Tables are organized, where users can find the data in an intuitive manner.
* Duplicates are removed, making data navigation simpler. It will also produce columns that can be used in slicers and filters.
* A complicated column can be split into two, simpler columns. Multiple columns can be combined into one column for readability.
* Codes and integers can be replaced with human readable values.

In this module, you will learn how to:

* Resolve inconsistencies, unexpected or null values, and data quality issues.
* Apply user-friendly value replacements.
* Profile data so you can learn more about a specific column before using it.
* Evaluate and transform column data types.
* Apply data shape transformations to table structures.
* Combine queries.
* Apply user-friendly naming conventions to columns and queries.
* Edit M code in the Advanced Editor.

**Next unit: Shape the initial data**

# Shape the initial data

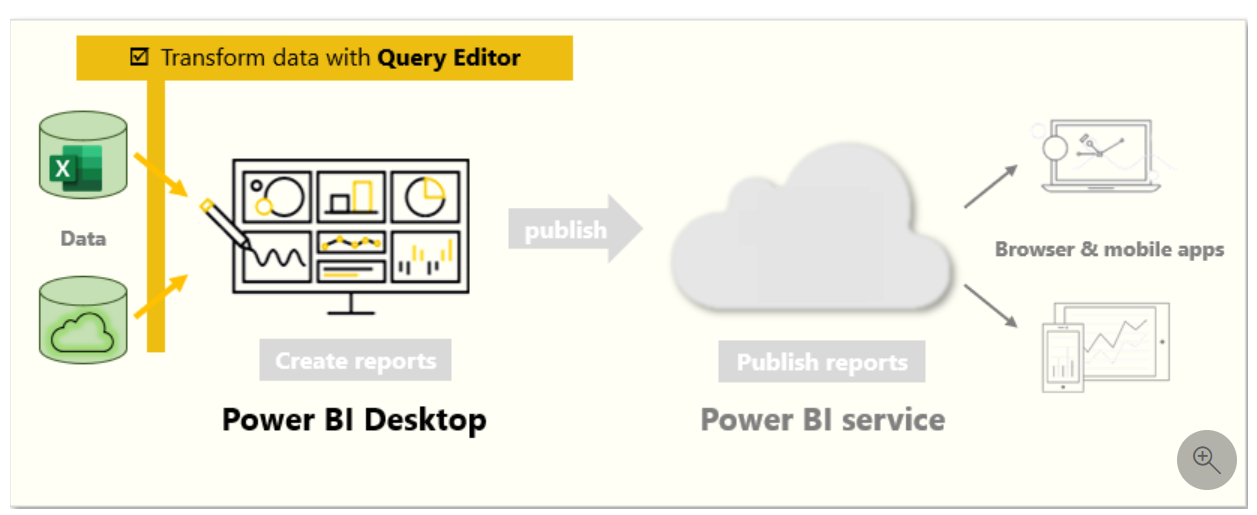
Completed100 XP

* 18 minutes

Power Query Editor in Power BI Desktop allows you to shape (transform) your imported data. You can accomplish actions such as renaming columns or tables, changing text to numbers, removing rows, setting the first row as headers, and much more. It is important to shape your data to ensure that it meets your needs and is suitable for use in reports.

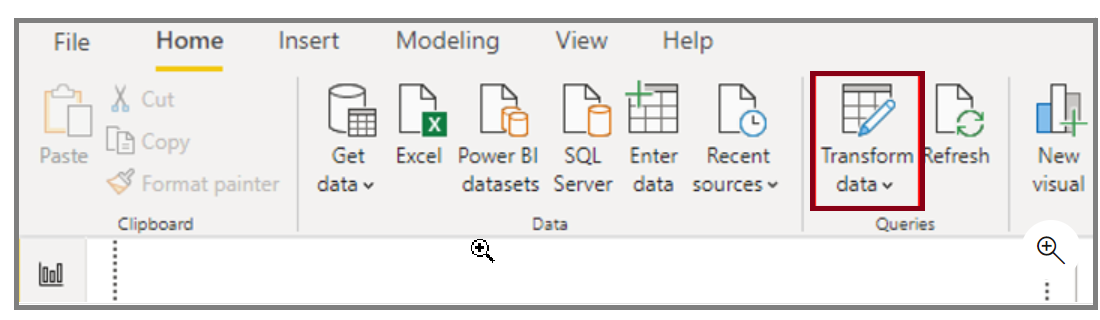
You have loaded raw sales data from two sources into a Power BI model. Some of the data came from a .csv file that was created manually in Microsoft Excel by the Sales team. The other data was loaded through a connection to your organization's Enterprise Resource Planning (ERP) system. Now, when you look at the data in Power BI Desktop, you notice that it's in disarray; some data that you don't need and some data that you do need are in the wrong format.

You need to use Power Query Editor to clean up and shape this data before you can start building reports.

[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/02-shape-data-query-editor-overview-ss.png#lightbox)

## Get started with Power Query Editor

To start shaping your data, open Power Query Editor by selecting the **Transform data** option on the **Home** tab of Power BI Desktop.

[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/02-open-query-editor-transform-button-ssm.png#lightbox)

In Power Query Editor, the data in your selected query displays in the middle of the screen and, on the left side, the **Queries** pane lists the available queries (tables).

When you work in Power Query Editor, all steps that you take to shape your data are recorded. Then, each time the query connects to the data source, it automatically applies your steps, so your data is always shaped the way that you specified. Power Query Editor only makes changes to a particular view of your data, so you can feel confident about changes that are being made to your original data source. You can see a list of your steps on the right side of the screen, in the **Query Settings** pane, along with the query's properties.

The Power Query Editor ribbon contains many buttons you can use to select, view, and shape your data.

To learn more about the available features and functions, see [The query ribbon](https://learn.microsoft.com/en-us/power-query/power-query-quickstart-using-power-bi#the-query-ribbon).

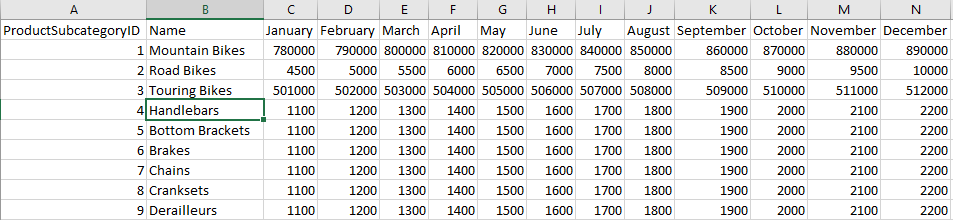
**Note**

In Power Query Editor, the right-click context menus and **Transform** tab in the ribbon provide many of the same options.

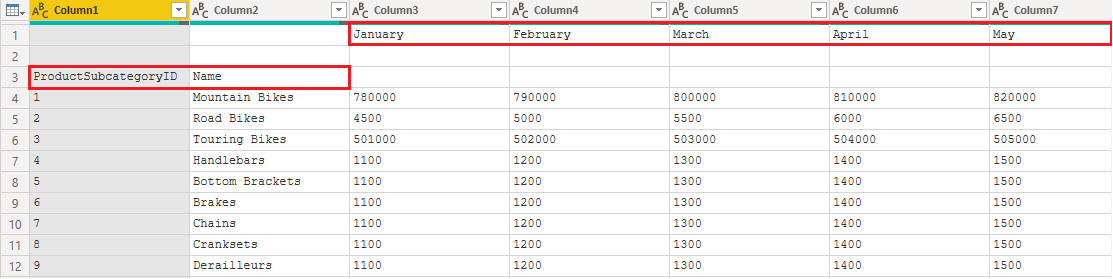
### Identify column headers and names

The first step in shaping your initial data is to identify the column headers and names within the data and then evaluate where they are located to ensure that they are in the right place.

In the following screenshot, the source data in the csv file for SalesTarget (sample not provided) had a target categorized by products and a subcategory split by months, both of which are organized into columns.

[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/02-identify-columns-ss.png#lightbox)

However, you notice that the data did not import as expected.

[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/02-identify-headers-names-ssm.png#lightbox)

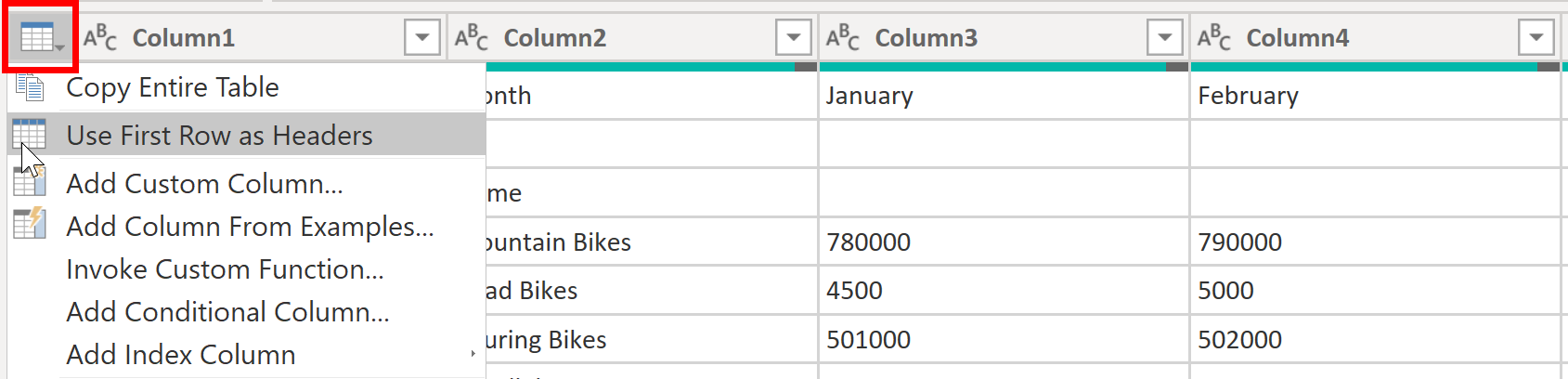
Consequently, the data is difficult to read. A problem has occurred with the data in its current state because column headers are in different rows (marked in red), and several columns have undescriptive names, such as **Column1**, **Column2**, and so on.

When you have identified where the column headers and names are located, you can make changes to reorganize the data.

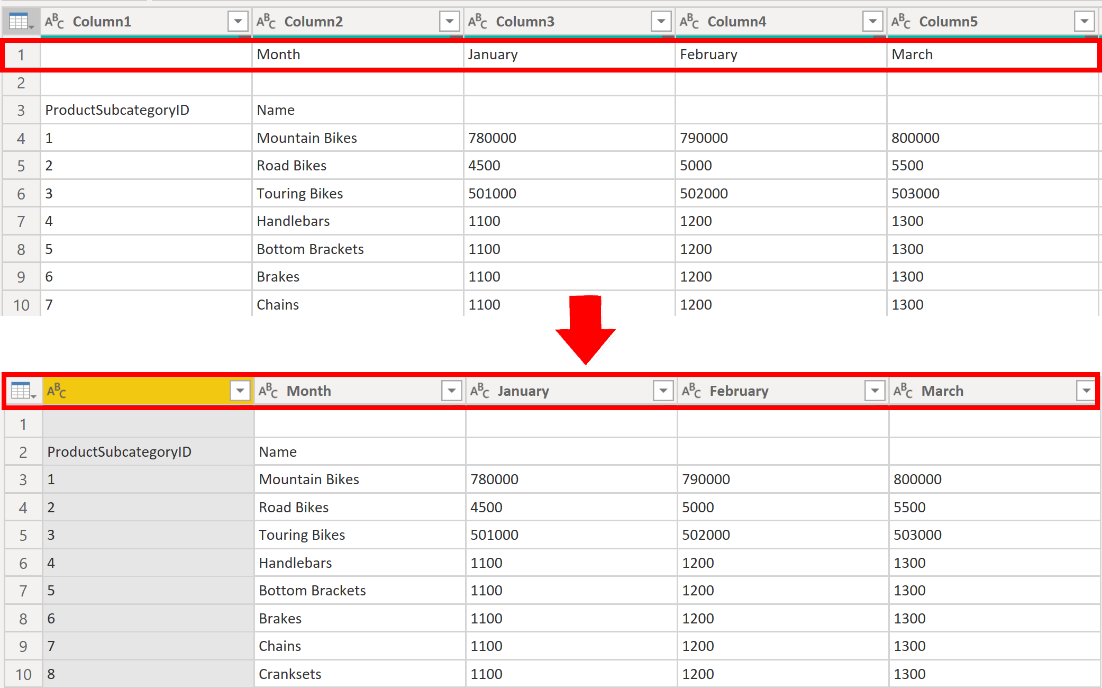
### Promote headers

When a table is created in Power BI Desktop, Power Query Editor assumes that all data belongs in table rows. However, a data source might have a first row that contains column names, which is what happened in the previous SalesTarget example. To correct this inaccuracy, you need to promote the first table row into column headers.

You can promote headers in two ways: by selecting the **Use First Row as Headers** option on the **Home** tab or by selecting the drop-down button next to **Column1** and then selecting **Use First Row as Headers**.

[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/02-use-first-row-headers-ssm.png#lightbox)

The following image illustrates how the **Use First Row as Headers** feature impacts the data:

[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/02-use-first-row-headers-result-ssm.png#lightbox)

## Rename columns

The next step in shaping your data is to examine the column headers. You might discover that one or more columns have the wrong headers, a header has a spelling error, or the header naming convention is not consistent or user-friendly.

Refer to the previous screenshot, which shows the impact of the **Use First Row as Headers** feature. Notice that the column that contains the subcategory **Name** data now has **Month** as its column header. This column header is incorrect, so it needs to be renamed.

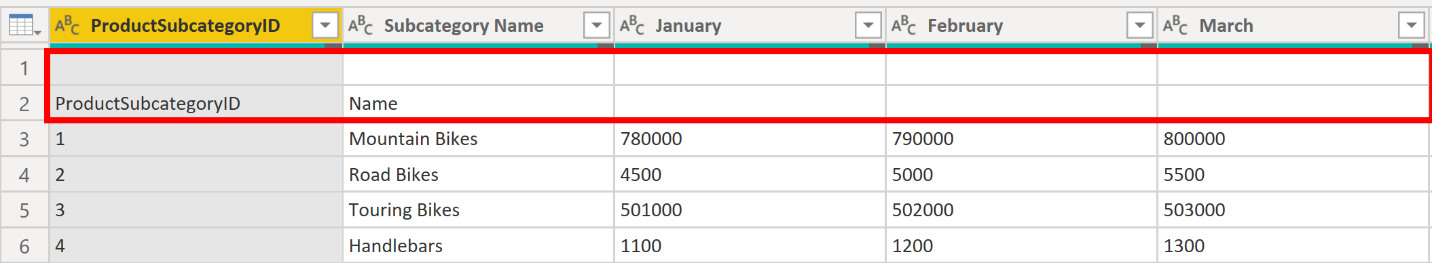
You can rename column headers in two ways. One approach is to right-click the header, select **Rename**, edit the name, and then press **Enter**. Alternatively, you can double-click the column header and overwrite the name with the correct name.

You can also work around this issue by removing (skipping) the first two rows and then renaming the columns to the correct name.

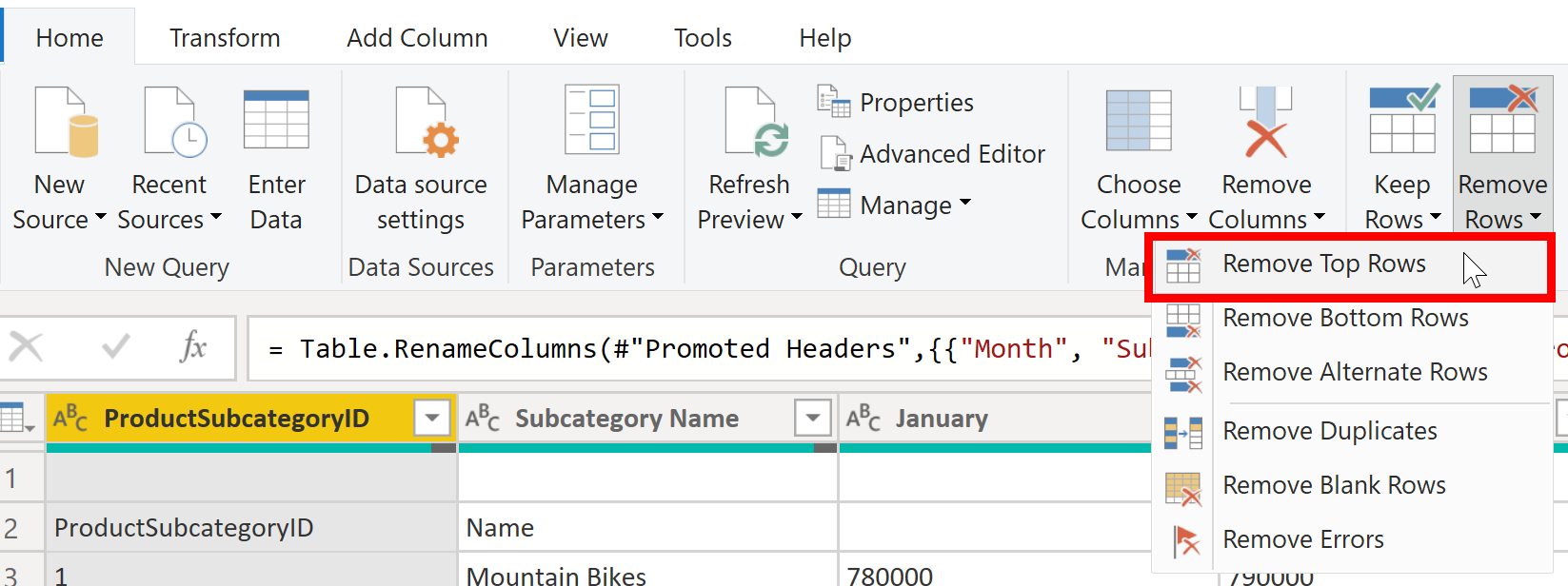
## Remove top rows

When shaping your data, you might need to remove some of the top rows, for example, if they are blank or if they contain data that you do not need in your reports.

Continuing with the SalesTarget example, notice that the first row is blank (it has no data) and the second row has data that is no longer required.

[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/02-remove-top-rows-ssm.png#lightbox)

To remove these excess rows, select **Remove Rows** > **Remove Top Rows** on the **Home** tab.

[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/02-remove-top-rows-feature-ssm.png#lightbox)

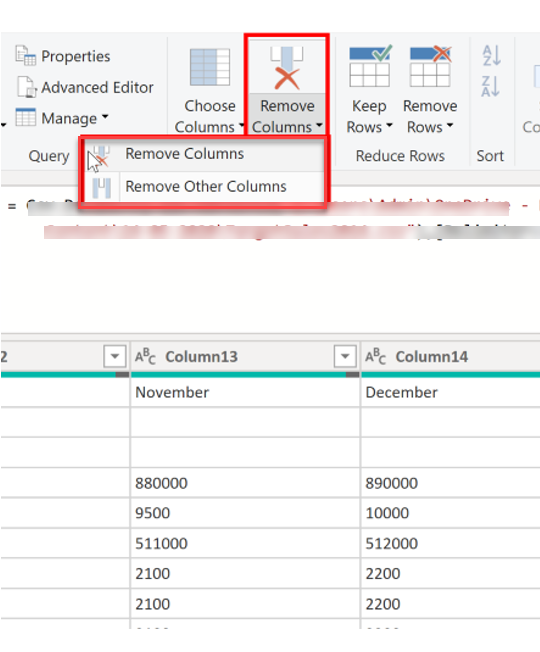
## Remove columns

A key step in the data shaping process is to remove unnecessary columns. It is much better to remove columns as early as possible. One way to remove columns would be to limit the column when you get data from data source. For instance, if you are extracting data from a relational database by using SQL, you would want to limit the column that you extract by using a column list in the SELECT statement.

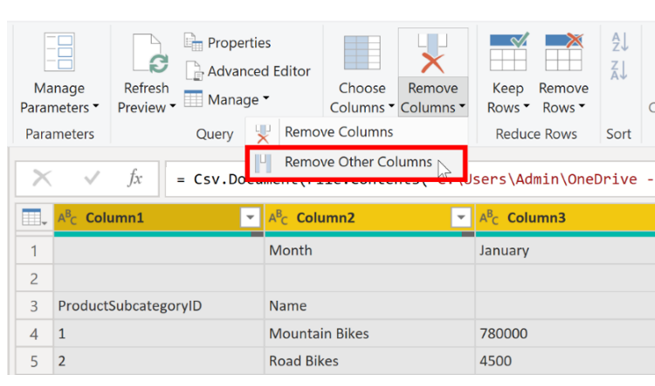
Removing columns at an early stage in the process rather than later is best, especially when you have established relationships between your tables. Removing unnecessary columns will help you to focus on the data that you need and help improve the overall performance of your Power BI Desktop datasets and reports.

Examine each column and ask yourself if you really need the data that it contains. If you don't plan on using that data in a report, the column adds no value to your data model. Therefore, the column should be removed. You can always add the column later, if your requirements change over time.

You can remove columns in two ways. The first method is to select the columns that you want to remove and then, on the **Home** tab, select **Remove Columns**.

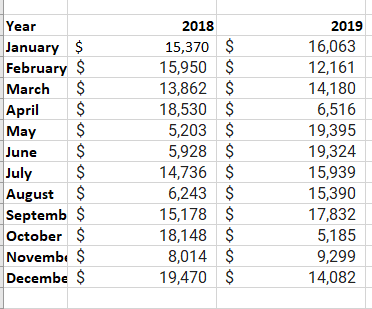
[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/02-remove-columns-ssm.png#lightbox)

Alternatively, you can select the columns that you want to keep and then, on the **Home** tab, select **Remove Columns** > **Remove Other Columns**.

[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/02-remove-other-columsn-ssm.png#lightbox)

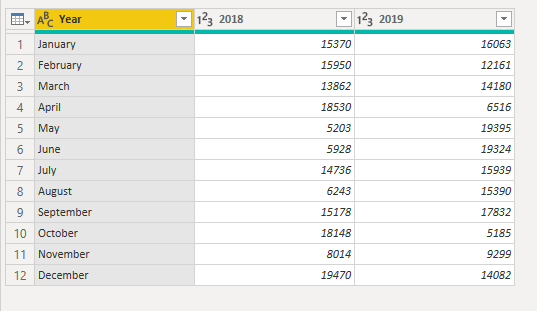
## Unpivot columns

Unpivoting is a useful feature of Power BI. You can use this feature with data from any data source, but you would most often use it when importing data from Excel. The following example shows a sample Excel document with sales data.

[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/02-excel-data-multiple-columns-ss.png#lightbox)

Though the data might initially make sense, it would be difficult to create a total of all sales combined from 2018 and 2019. Your goal would then be to use this data in Power BI with three columns: **Month**, **Year**, and **SalesAmount**.

When you import the data into Power Query, it will look like the following image.

[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/02-original-data-ss.png#lightbox)

Next, rename the first column to **Month**. This column was mislabeled because that header in Excel was labeling the 2018 and 2019 columns. Highlight the 2018 and 2019 columns, select the **Transform** tab in Power Query, and then select **Unpivot**.

[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/02-unpivot-ss.png#lightbox)

You can rename the **Attribute** column to **Year** and the **Value** column to **SalesAmount**.

Unpivoting streamlines the process of creating DAX measures on the data later. By completing this process, you have now created a simpler way of slicing the data with the **Year** and **Month** columns.

## Pivot columns

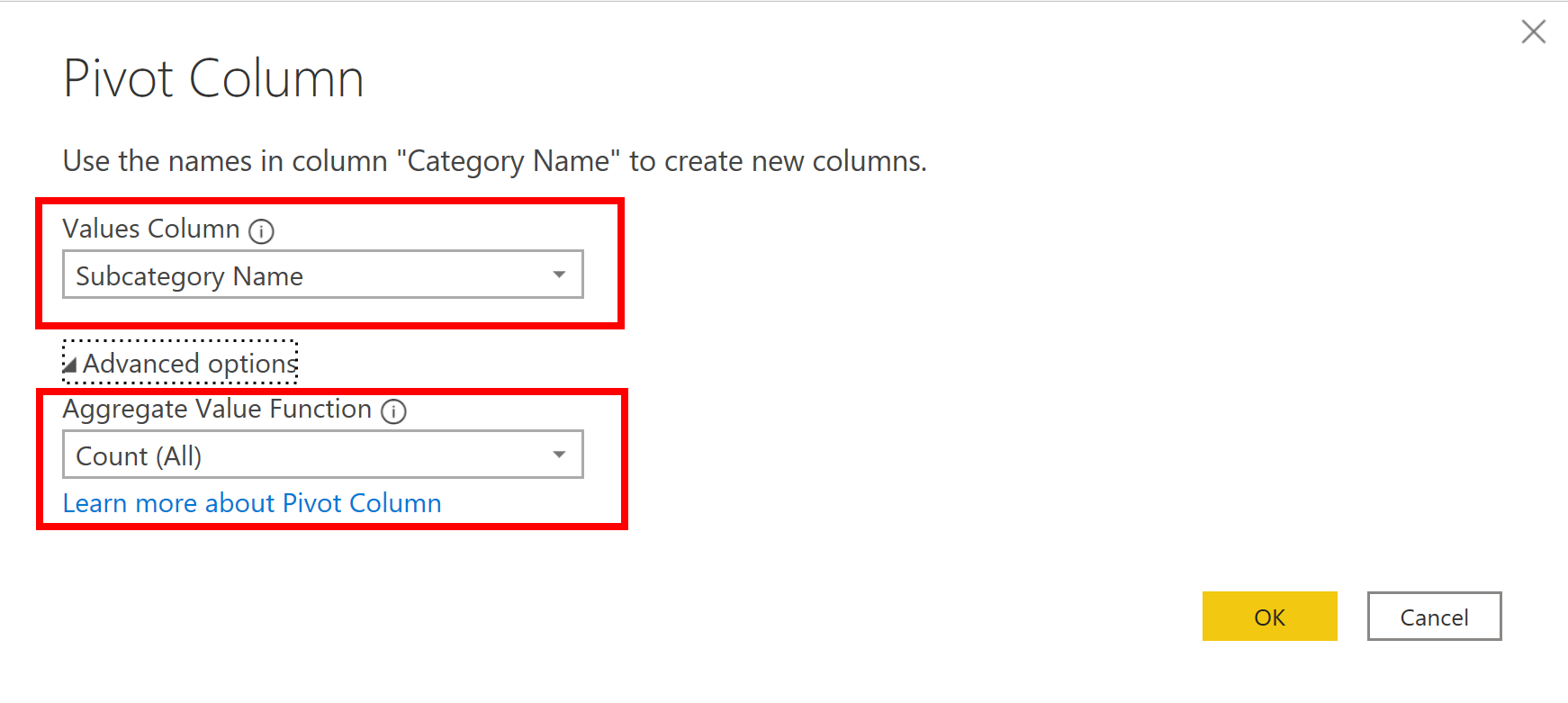
If the data that you are shaping is flat (in other words, it has lot of detail but is not organized or grouped in any way), the lack of structure can complicate your ability to identify patterns in the data.

You can use the **Pivot Column** feature to convert your flat data into a table that contains an aggregate value for each unique value in a column. For example, you might want to use this feature to summarize data by using different math functions such as **Count**, **Minimum**, **Maximum**, **Median**, **Average**, or **Sum**.

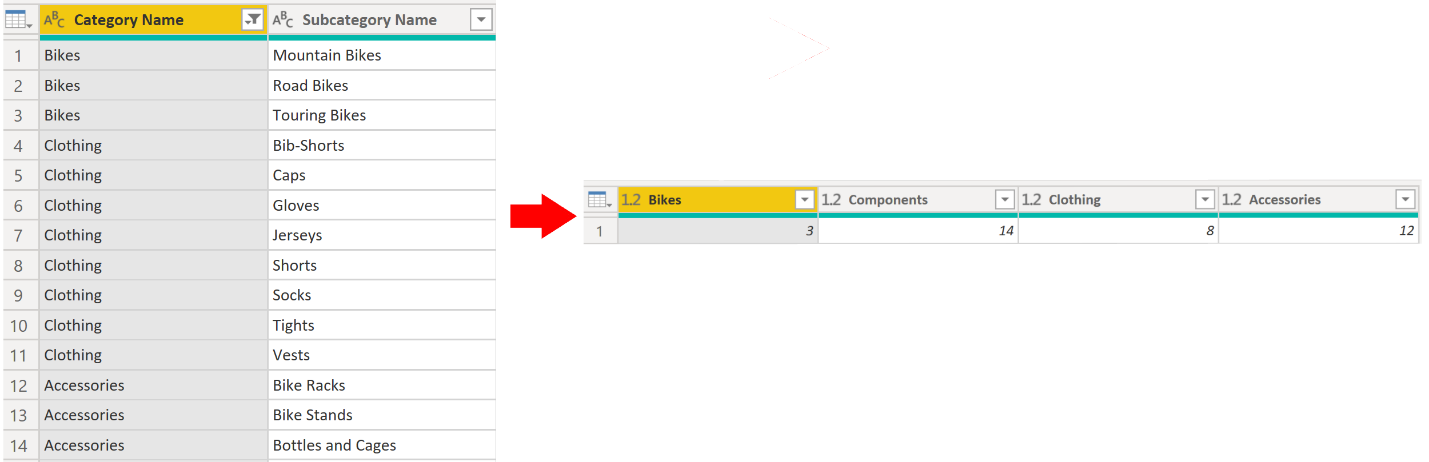
In the SalesTarget example, you can pivot the columns to get the quantity of product subcategories in each product category.

On the **Transform** tab, select **Transform > Pivot Columns**.

On the **Pivot Column** window that displays, select a column from the **Values Column** list, such as **Subcategory name**. Expand the advanced options and select an option from the **Aggregate Value Function** list, such as **Count (All)**, and then select **OK**.

[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/02-aggregate-value-function-ssm.png#lightbox)

The following image illustrates how the **Pivot Column** feature changes the way that the data is organized.

[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/02-pivot-column-feature-display-data-ssm.png#lightbox)

Power Query Editor records all steps that you take to shape your data, and the list of steps are shown in the **Query Settings** pane. If you have made all the required changes, select **Close & Apply** to close Power Query Editor and apply your changes to your data model. However, before you select **Close & Apply**, you can take further steps to clean up and transform your data in Power Query Editor. These additional steps are covered later in this module.

## Next unit: Simplify the data structure

**Simplify the data structure**

Completed100 XP

* 8 minutes

When you import data from multiple sources into Power BI Desktop, the data retains its predefined table and column names. You might want to change some of these names so that they are in a consistent format, easier to work with, and more meaningful to a user. You can use Power Query Editor in Power BI Desktop to make these name changes and simplify your data structure.

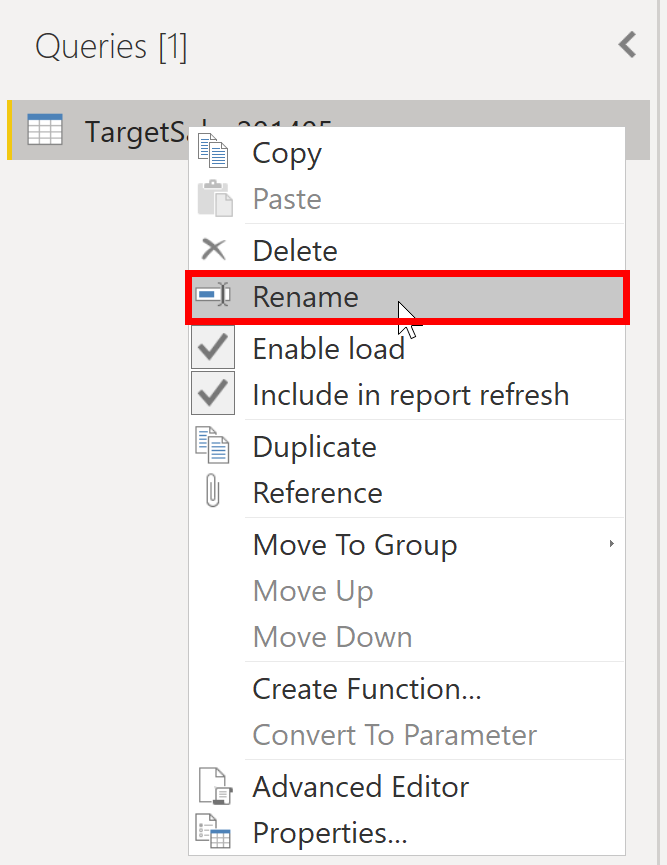
To continue with the previous scenario where you shaped the initial data in your model, you need to take further action to simplify the structure of the sales data and get it ready for developing reports for the Sales team. You have already renamed the columns, but now you need to examine the names of the queries (tables) to determine if any improvements can be made. You also need to review the contents of the columns and replace any values that require correction.

**Rename a query**

It's good practice to change uncommon or unhelpful query names to names that are more obvious or that the user is more familiar with. For instance, if you import a product fact table into Power BI Desktop and the query name displays as *FactProductTable*, you might want to change it to a more user-friendly name, such as *Products*. Similarly, if you import a view, the view might have a name that contains a prefix of *v*, such as *vProduct*. People might find this name unclear and confusing, so you might want to remove the prefix.

In this example, you have examined the name of the TargetSales query and realize that this name is unhelpful because you'll have a query with this name for every year. To avoid confusion, you want to add the year to the query name.

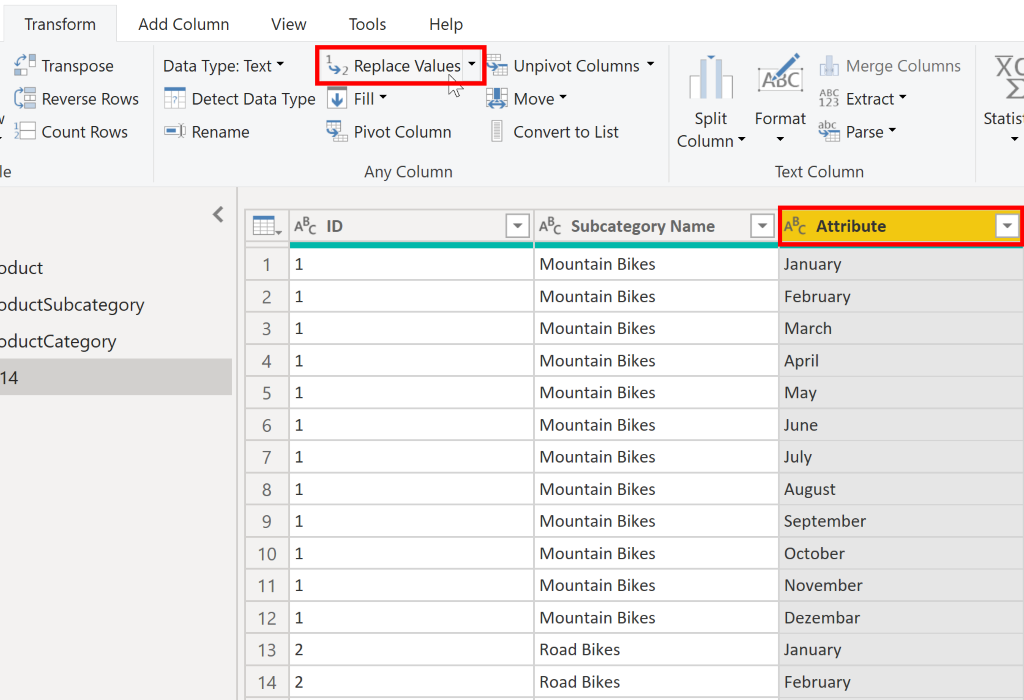
In Power Query Editor, in the **Queries** pane to the left of your data, select the query that you want to rename. Right-click the query and select **Rename**. Edit the current name or type a new name, and then press **Enter**.

[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/03-rename-query-ssm.png#lightbox)

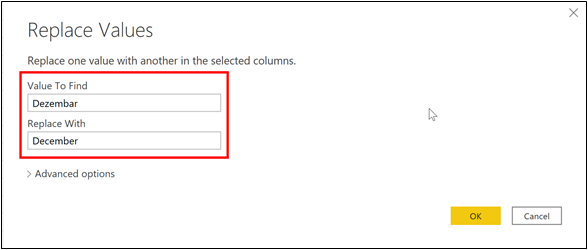
**Replace values**

You can use the **Replace Values** feature in Power Query Editor to replace any value with another value in a selected column.

In this example, you notice that, in the **Attribute** column, the month December is misspelled. You need to correct this spelling mistake. Select the column that contains the value that you want to replace (**Attribute** in this case), and then select **Replace Values** on the **Transform** tab.

[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/03-replace-value-ssm.png#lightbox)

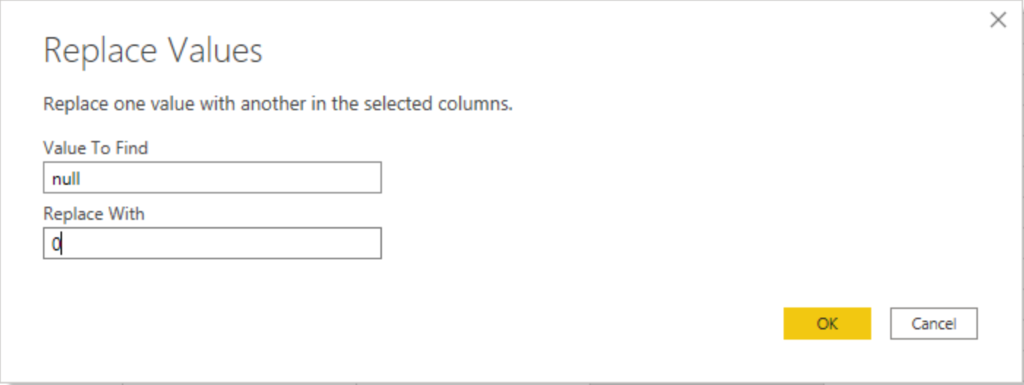
In the **Value to Find** box, enter the name of the value that you want to replace, and then in the **Replace With** box, enter the correct value name and then select **OK**. In Power Query, you can't select one cell and change one value, like you might have done in Excel.

[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/03-value-find-ssm.png#lightbox)

You can review the list of steps that you took to restructure and correct your data in the **Query Settings** pane. When you have completed all steps that you want to take, you can select **Close & Apply** to close Power Query Editor and apply your changes to your data model. However, you can take further action to clean and transform your data.

**Replace null values**

Occasionally, you might find that your data sources contain null values. For example, a freight amount on a sales order might have a null value if it's synonymous with zero. If the value stays null, the averages will not calculate correctly. One solution would be to change the nulls to zero, which will produce the more accurate freight average. In this instance, using the same steps that you followed previously will help you replace the null values with zero.

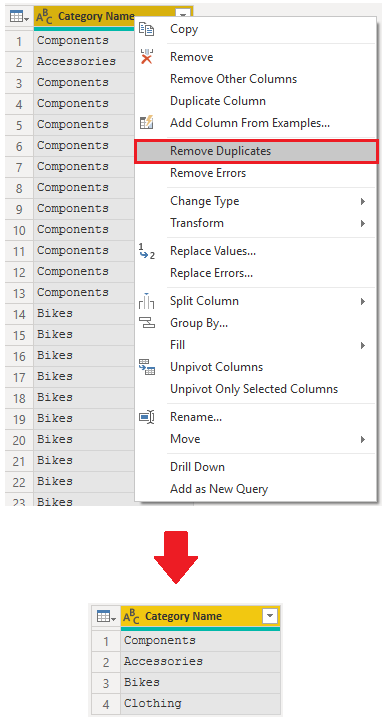
[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/03-replace-null-zero-ss.png#lightbox)

**Remove duplicates**

You can also remove duplicates from columns to only keep unique names in a selected column by using the **Remove Duplicates** feature in Power Query.

In this example, notice that the **Category Name** column contains duplicates for each category. As a result, you want to create a table with unique categories and use it in your data model. You can achieve this action by selecting a column, right-clicking on the header of the column, and then selecting the **Remove Duplicates** option.

You might consider copying the table before removing the duplicates. The **Copy** option is at the top of the context menu, as shown in the following screenshot. Copying the table before removing duplicates will give you a comparison of the tables and will let you use both tables, if needed.

[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/03-remove-duplicates-ssm.png#lightbox)

**Best practices for naming tables, columns, and values**

Naming conventions for tables, columns, and values have no fixed rules; however, we recommend that you use the language and abbreviations that are commonly used within your organization and that everyone agrees on and considers them as common terminology.

A best practice is to give your tables, columns, and measures descriptive business terms and replace underscores ("\_") with spaces. Be consistent with abbreviations, prefaces, and words like "number" and "ID." Excessively short abbreviations can cause confusion if they are not commonly used within the organization.

Also, by removing prefixes or suffixes that you might use in table names and instead naming them in a simple format, you will help avoid confusion.

When replacing values, try to imagine how those values will appear on the report. Values that are too long might be difficult to read and fit on a visual. Values that are too short might be difficult to interpret. Avoiding acronyms in values is also a good idea, provided that the text will fit on the visual.

**Next unit: Evaluate and change column data types**

# Evaluate and change column data types

Completed100 XP

* 10 minutes

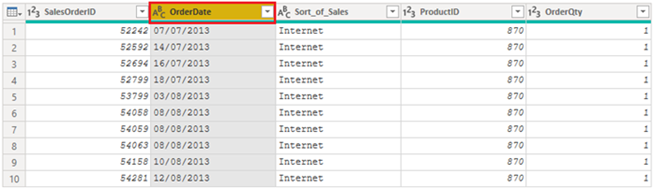
When you import a table from any data source, Power BI Desktop automatically starts scanning the first 1,000 rows (default setting) and tries to detect the type of data in the columns. Some situations might occur where Power BI Desktop does not detect the correct data type. Where incorrect data types occur, you will experience performance issues.

You have a higher chance of getting data type errors when you are dealing with flat files, such as comma-separated values (.CSV) files and Excel workbooks (.XLSX), because data was entered manually into the worksheets and mistakes were made. Conversely, in databases, the data types are predefined when tables or views are created.

A best practice is to evaluate the column data types in Power Query Editor before you load the data into a Power BI data model. If you determine that a data type is incorrect, you can change it. You might also want to apply a format to the values in a column and change the summarization default for a column.

To continue with the scenario where you are cleaning and transforming sales data in preparation for reporting, you now need to evaluate the columns to ensure that they have the correct data type. You need to correct any errors that you identify.

You evaluate the **OrderDate** column. As expected, it contains numeric data, but Power BI Desktop has incorrectly set the column data type to Text. To report on this column, you need to change the data type from Text to Date.

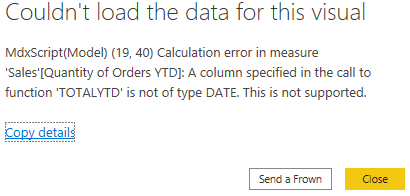
[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/04-column-data-type-setas-text-ssm.png#lightbox)

## Implications of incorrect data types

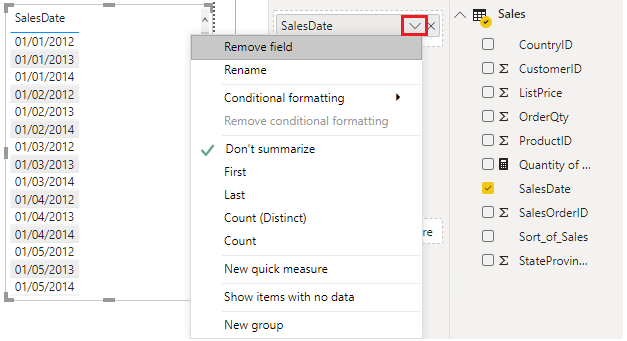
The following information provides insight into problems that can arise when Power BI does not detect the correct data type.

Incorrect data types will prevent you from creating certain calculations, deriving hierarchies, or creating proper relationships with other tables. For example, if you try to calculate the Quantity of Orders YTD, you will get the following error stating that the **OrderDate** column data type is not Date, which is required in time-based calculations.

Quantity of Orders YTD = TOTALYTD(SUM('Sales'[OrderQty]), 'Sales'[OrderDate])

[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/04-error-calculated-measure-ss.png#lightbox)

Another issue with having an incorrect data type applied on a date field is the inability to create a date hierarchy, which would allow you to analyze your data on a yearly, monthly, or weekly basis. The following screenshot shows that the SalesDate field is not recognized as type Date and will only be presented as a list of dates in the Table visual. However, it is a best practice to use a date table and turn off the auto date/time to get rid of the auto generated hierarchy. For more information about this process, see [Auto generated data type](https://learn.microsoft.com/en-us/power-bi/guidance/auto-date-time/) documentation.

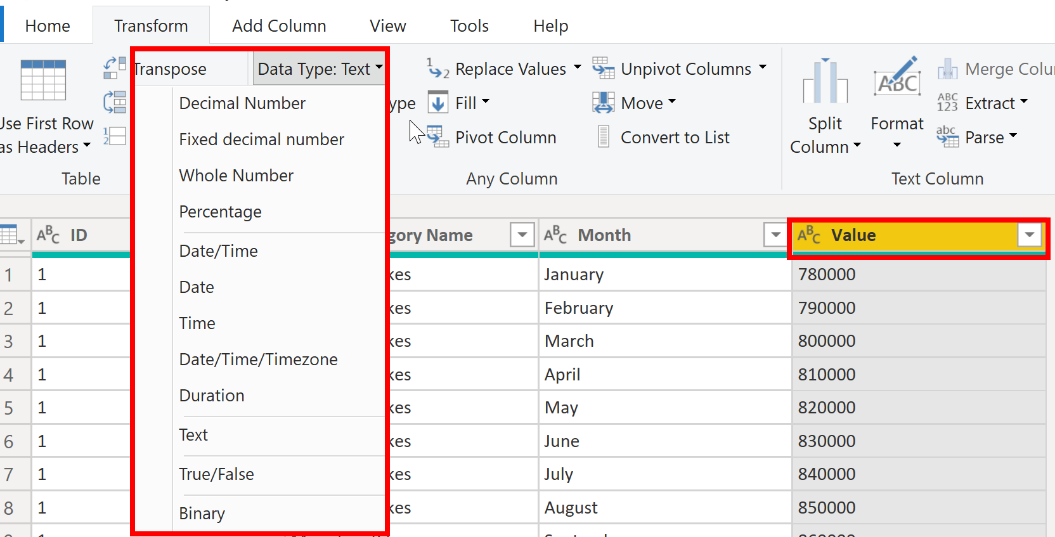
[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/04-additions-sales-date-options-ssm.png#lightbox)

## Change the column data type

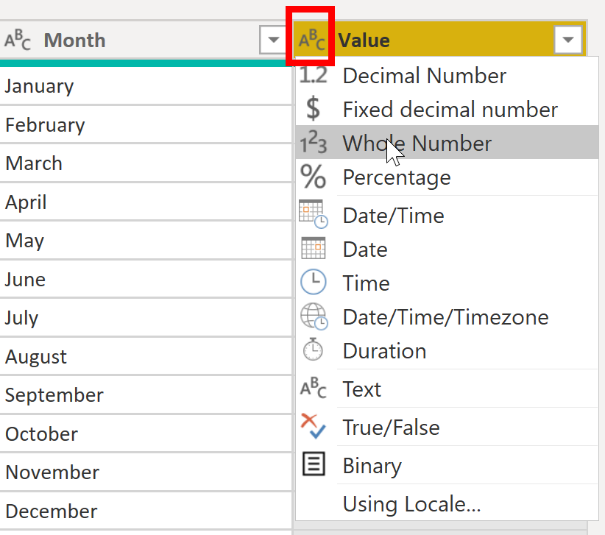
You can change the data type of a column in two places: in Power Query Editor and in the Power BI Desktop Report view by using the column tools. It is best to change the data type in the Power Query Editor before you load the data.

### Change the column data type in Power Query Editor

In Power Query Editor, you can change the column data type in two ways. One way is to select the column that has the issue, select **Data Type** in the **Transform** tab, and then select the correct data type from the list.

[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/04-select-data-type-ssm.png#lightbox)

Another method is to select the data type icon next to the column header and then select the correct data type from the list.

[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/04-select-data-type-from-list-ssm.png#lightbox)

As with any other changes that you make in Power Query Editor, the change that you make to the column data type is saved as a programmed step. This step is called **Changed Type** and it will be iterated every time the data is refreshed.

After you have completed all steps to clean and transform your data, select **Close & Apply** to close Power Query Editor and apply your changes to your data model. At this stage, your data should be in great shape for analysis and reporting.

For more information, see [Data types in Power BI Desktop](https://learn.microsoft.com/en-us/power-bi/connect-data/desktop-data-types/).

## Next unit: Combine multiple tables into a single table

**Combine multiple tables into a single table**

Completed100 XP

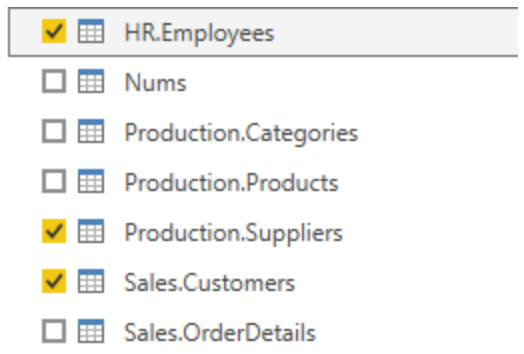
* 8 minutes

The ability to combine queries is powerful because it allows you to append or merge different tables or queries together. You can combine tables into a single table in the following circumstances:

* Too many tables exist, making it difficult to navigate an overly-complicated data model.
* Several tables have a similar role.
* A table has only a column or two that can fit into a different table.
* You want to use several columns from different tables in a custom column.

You can combine the tables in two different ways: merging and appending.

Assume that you are developing Power BI reports for the Sales and HR teams. They have asked you to create a contact information report that contains the contact information and location of every employee, supplier, and customer. The data is in the HR.Employees, Production.Suppliers, and the Sales.Customers tables, as shown in the following image.

[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/05-choosing-tables-combine-ss.png#lightbox)

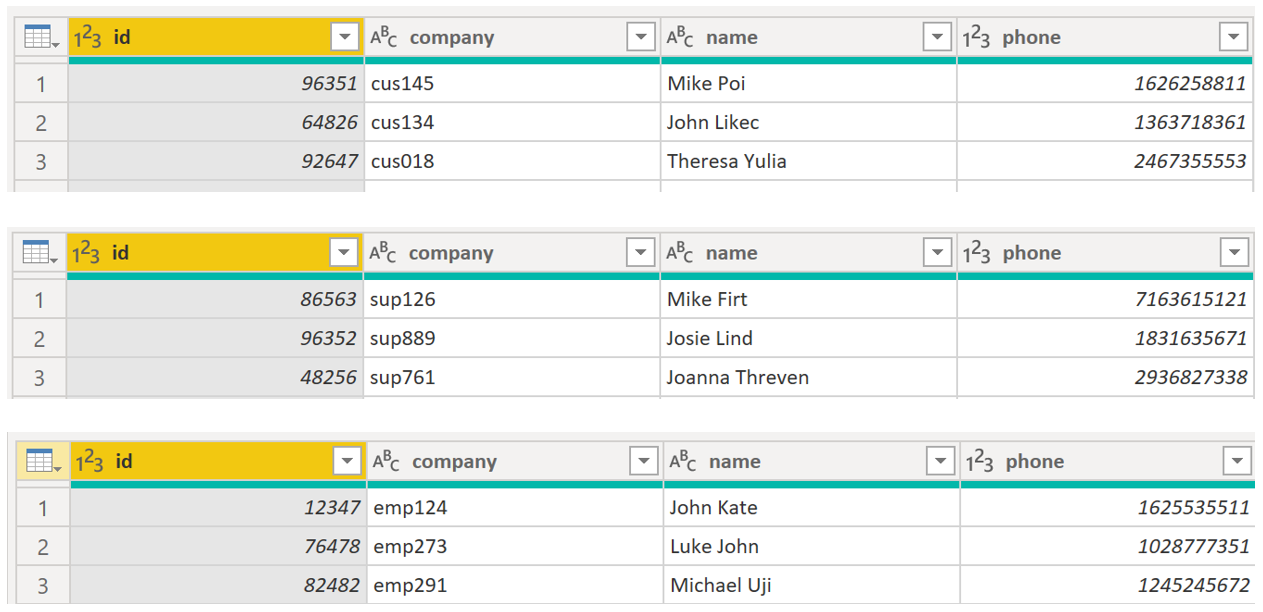
However, this data comes from multiple tables, so the dilemma is determining how you can merge the data in these multiple tables and create one source-of-truth table to create a report from. The inherent functionality of Power BI allows you to combine and merge queries into a single table.

**Append queries**

When you append queries, you will be adding rows of data to another table or query. For example, you could have two tables, one with 300 rows and another with 100 rows, and when you append queries, you will end up with 400 rows. When you merge queries, you will be adding columns from one table (or query) into another. To merge two tables, you must have a column that is the key between the two tables.

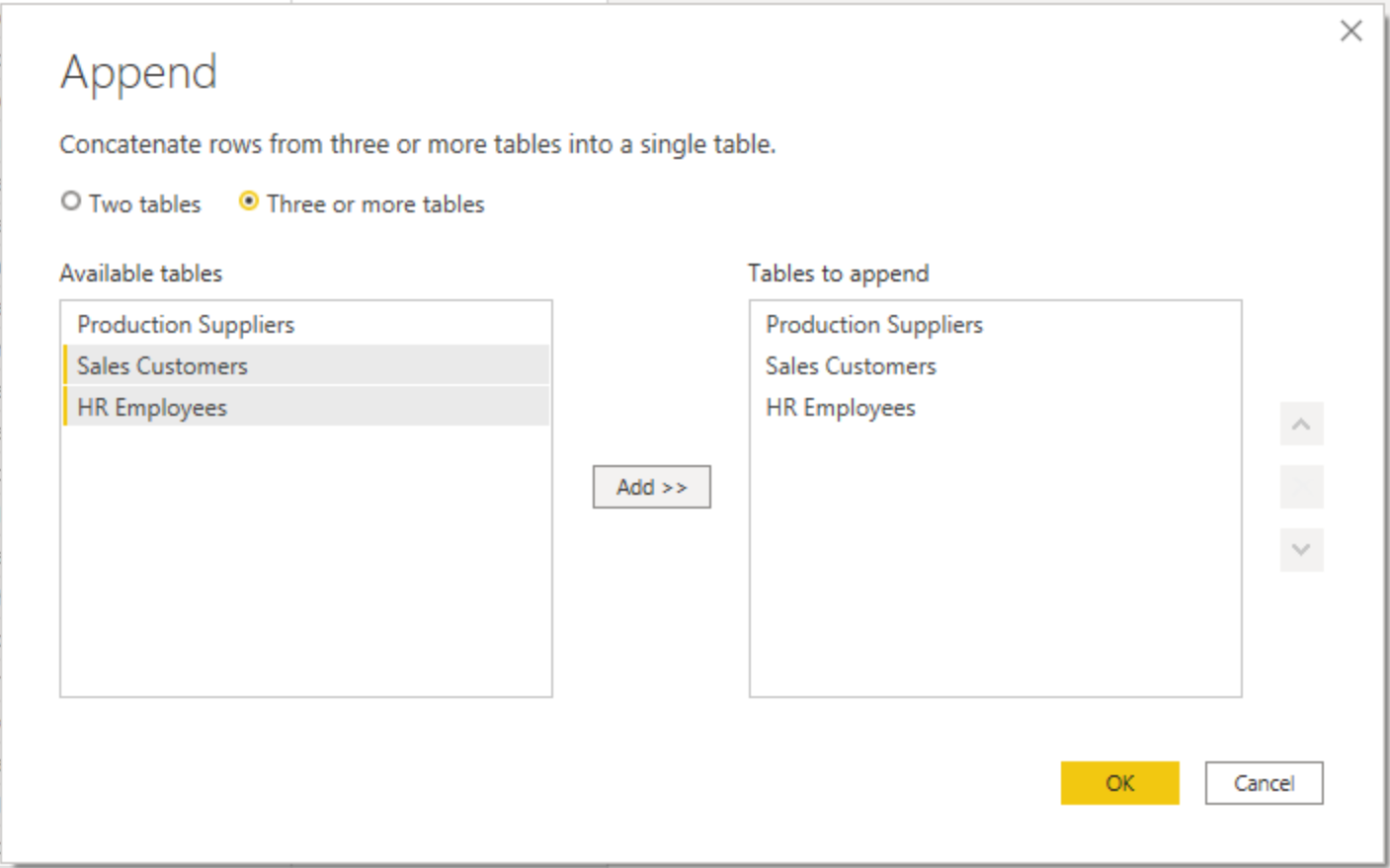
For the previously mentioned scenario, you will append the HR.Employees table with the Production.Suppliers and Sales.Customers tables so that you have one master list of contact information. Because you want to create one table that has all contact information for employees, suppliers, and customers, when you combine the queries, the pertinent columns that you require in your combined table must be named the same in your original data tables to see one consolidated view.

Before you begin combining queries, you can remove extraneous columns that you don't need for this task from your tables. To complete this task, format each table to have only four columns with your pertinent information, and rename them so they all have the same column headers: ID, company, name, and phone. The following images are snippets of the reformatted Sales.Customers, Production.Suppliers, and HR.Employees tables.

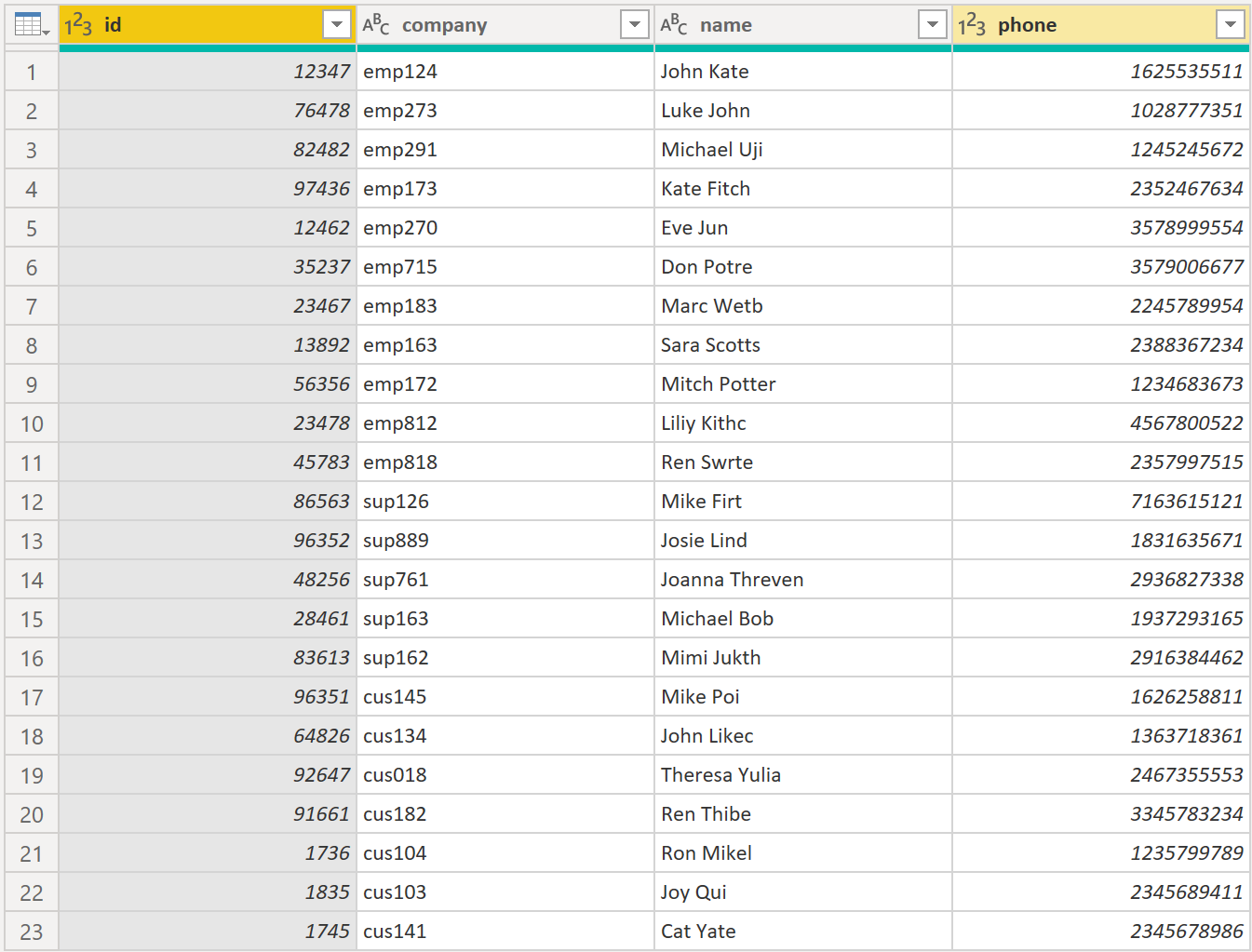
[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/05-reformatting-appending-ss.png#lightbox)

After you have finished reformatting, you can combine the queries. On the **Home** tab on the Power Query Editor ribbon, select the drop-down list for **Append Queries**. You can select **Append Queries as New**, which means that the output of appending will result in a new query or table, or you can select **Append Queries**, which will add the rows from an existing table into another.

Your next task is to create a new master table, so you need to select **Append Queries as New**. This selection will bring you to a window where you can add the tables that you want to append from **Available Tables** to **Tables to Append**, as shown in the following image.

[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/05-append-new-window-ss.png#lightbox)

After you have added the tables that you want to append, select **OK**. You will be routed to a new query that contains all rows from all three of your tables, as shown in the following image.

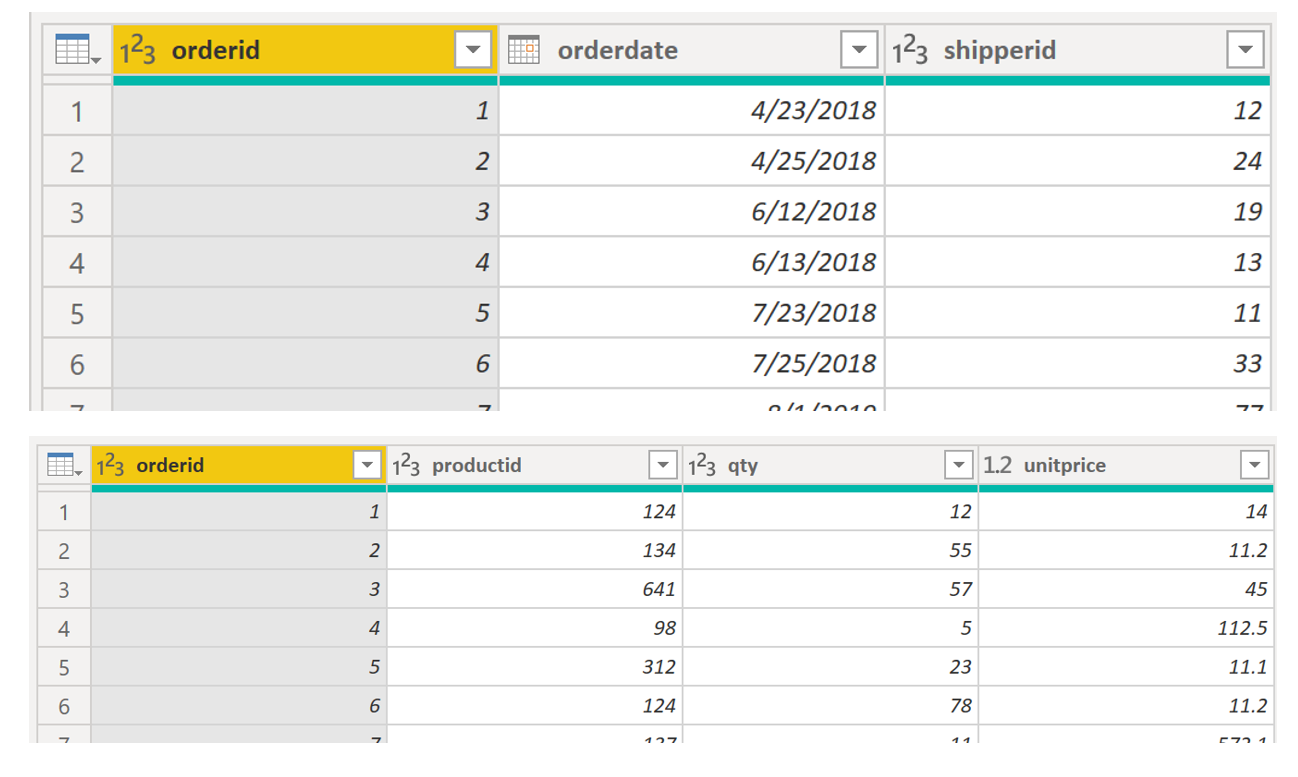
[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/05-appending-new-final-ss.png#lightbox)

You have now succeeded in creating a master table that contains the information for the employees, suppliers, and customers. You can exit Power Query Editor and build any report elements surrounding this master table.

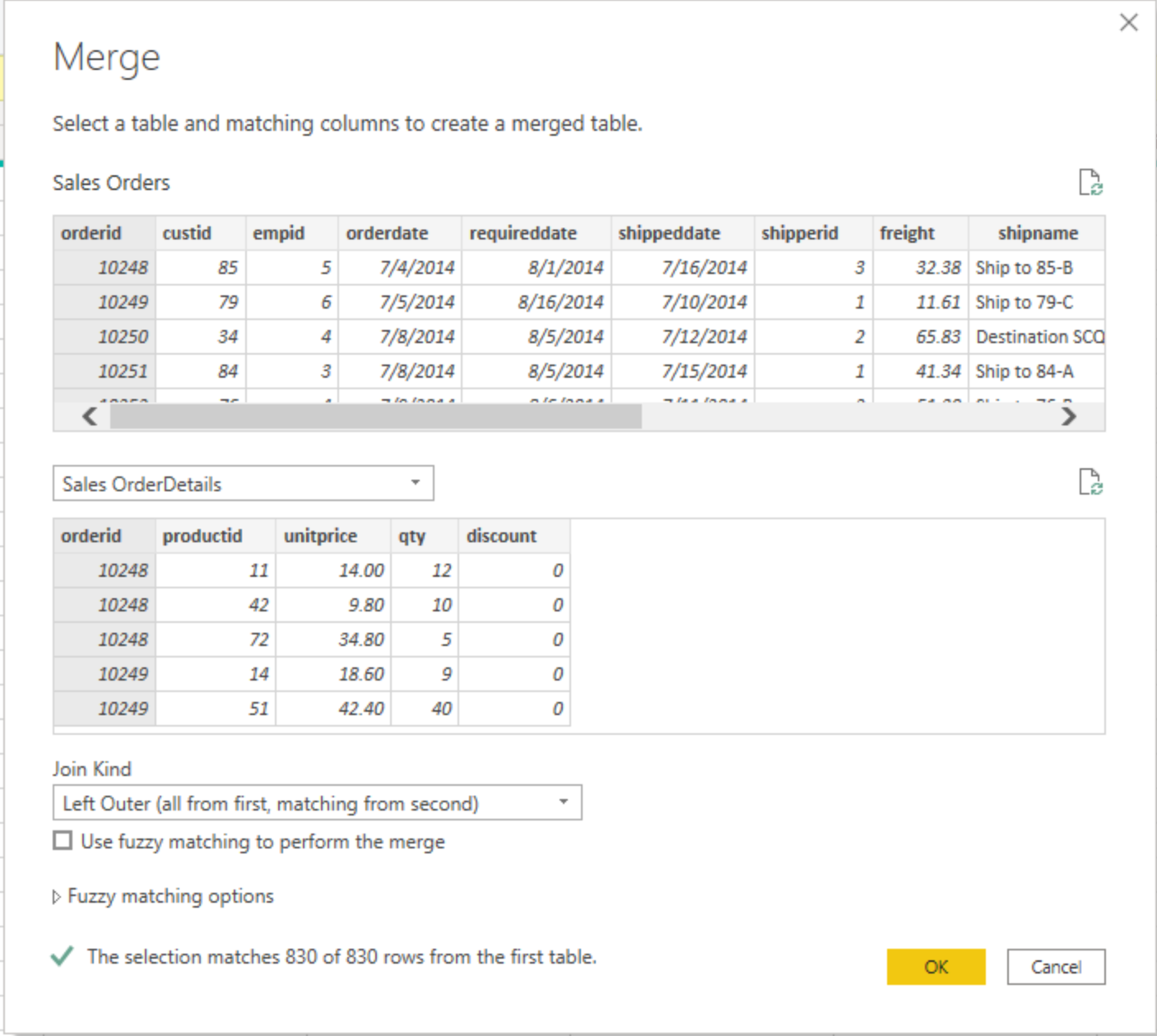
However, if you wanted to merge tables instead of appending the data from one table to another, the process would be different.

**Merge queries**

When you merge queries, you are combining the data from multiple tables into one based on a column that is common between the tables. This process is similar to the JOIN clause in SQL. Consider a scenario where the Sales team now wants you to consolidate orders and their corresponding details (which are currently in two tables) into a single table. You can accomplish this task by merging the two tables, Orders and OrderDetails, as shown in the following image. The column that is shared between these two tables is **OrderID**.

[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/05-merging-tables-example-ss.png#lightbox)

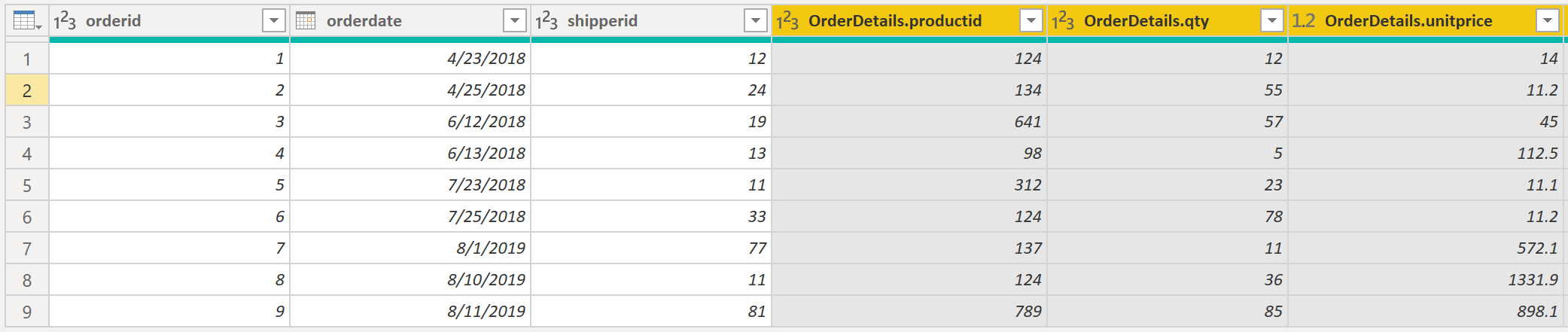
Go to **Home** on the Power Query Editor ribbon and select the **Merge Queries** drop-down menu, where you can select **Merge Queries as New**. This selection will open a new window, where you can choose the tables that you want to merge from the drop-down list, and then select the column that is matching between the tables, which in this case is **orderid**.

[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/05-merge-queries-new-ss.png#lightbox)

You can also choose how to join the two tables together, a process that is also similar to JOIN statements in SQL. These join options include:

* **Left Outer** - Displays all rows from the first table and only the matching rows from the second.
* **Full Outer** - Displays all rows from both tables.
* **Inner** - Displays the matched rows between the two tables.

For this scenario, you will choose to use a **Left Outer** join. Select **OK**, which will route you to a new window where you can view your merged query.

[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/05-merge-queries-final-view-ss.png#lightbox)

Now, you can merge two queries or tables in different ways so that you can view your data in the most appropriate way for your business requirements.

For more information on this topic, see the [Shape and Combine Data in Power BI](https://learn.microsoft.com/en-us/power-bi/connect-data/desktop-shape-and-combine-data/) documentation.

**Next unit: Profile data in Power BI**

**Profile data in Power BI**

Completed100 XP

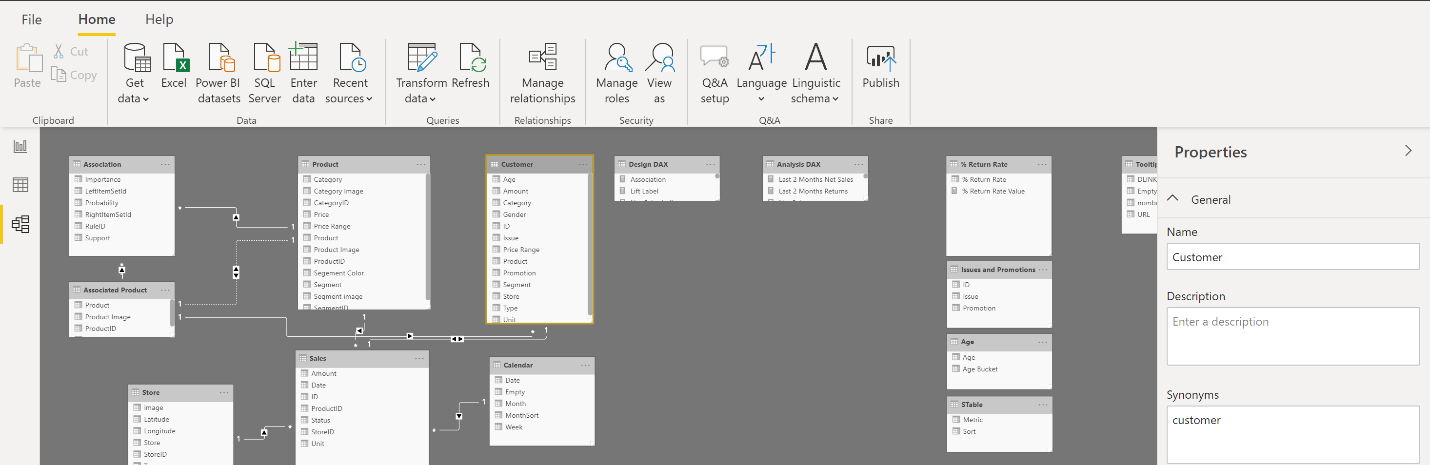
* 20 minutes

Profiling data is about studying the nuances of the data: determining anomalies, examining and developing the underlying data structures, and querying data statistics such as row counts, value distributions, minimum and maximum values, averages, and so on. This concept is important because it allows you to shape and organize the data so that interacting with the data and identifying the distribution of the data is uncomplicated, therefore helping to make your task of working with the data on the front end to develop report elements near effortless.

Assume that you are developing reports for the Sales team at your organization. You are uncertain how the data is structured and contained within the tables, so you want to profile the data behind the scenes before you begin developing the visuals. Power BI has inherent functionality that makes these tasks user-friendly and straightforward.

**Examine data structures**

Before you begin examining the data in Power Query Editor, you should first learn about the underlying data structures that data is organized in. You can view the current data model under the **Model** tab on Power BI Desktop.

[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/06-data-model-ribbon-ss.png#lightbox)

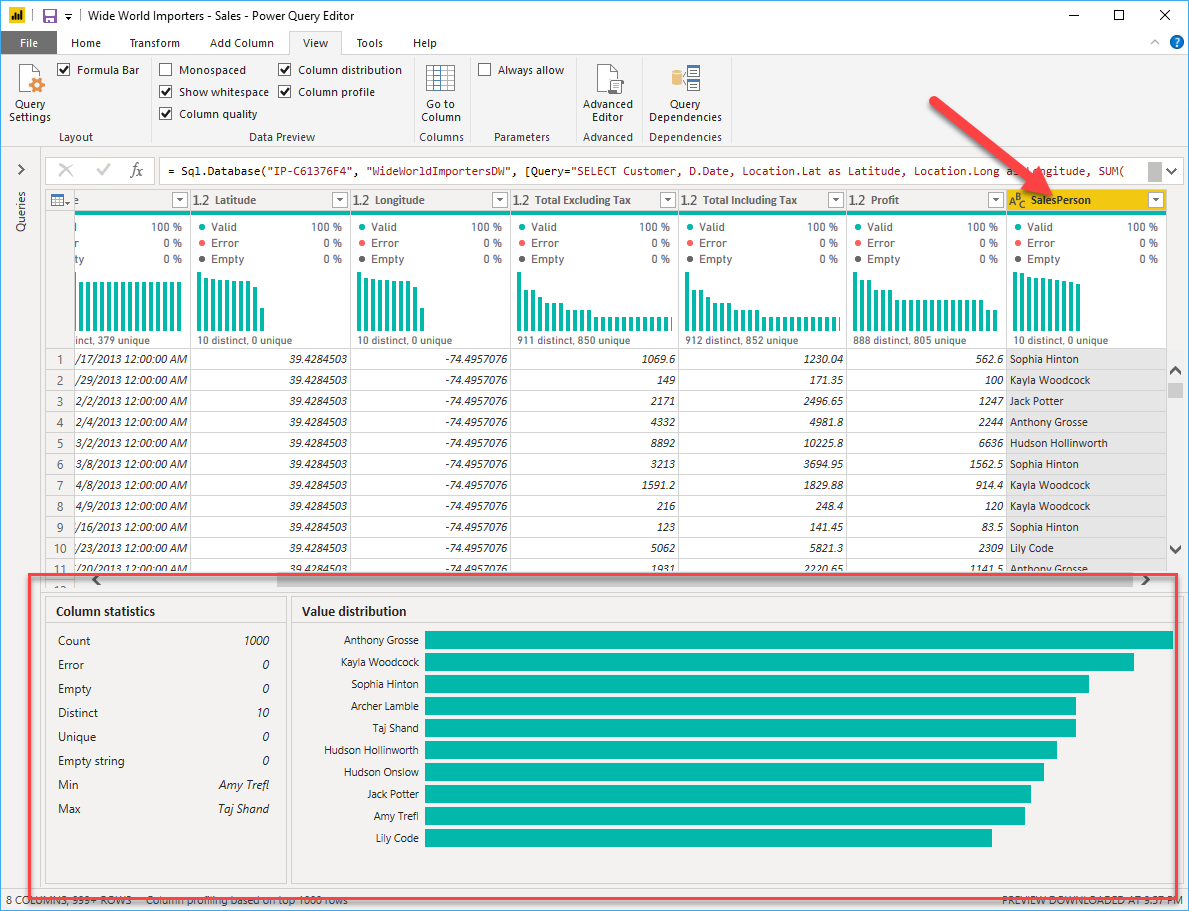
On the **Model** tab, you can edit specific column and table properties by selecting a table or columns, and you can transform the data by using the **Transform Data** button, which takes you to Power Query Editor. Additionally, you can manage, create, edit, and delete relationships between different tables by using **Manage Relationships**, which is located on the ribbon.

**Find data anomalies and data statistics**

After you have created a connection to a data source and have selected **Transform Data**, you are brought to Power Query Editor, where you can determine if anomalies exist within your data. Data anomalies are outliers within your data. Determining what those anomalies are can help you identify what the normal distribution of your data looks like and whether specific data points exist that you need to investigate further. Power Query Editor determines data anomalies by using the **Column Distribution** feature.

Select **View** on the ribbon, and under **Data Preview**, you can choose from a few options. To understand data anomalies and statistics, select the **Column Distribution**, **Column Quality**, and **Column Profile** options. The following figure shows the statistics that appear.

**Column quality** and **Column distribution** are shown in the graphs above the columns of data. **Column quality** shows you the percentages of data that is valid, in error, and empty. In an ideal situation, you want 100 percent of the data to be valid.

[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/06-column-statistics-ssm.png#lightbox)

**Note**

By default, Power Query examines the first 1000 rows of your data set. To change this, select the profiling status in the status bar and select **Column profiling based on entire data set**. ]

**Column distribution** shows you the distribution of the data within the column and the counts of distinct and unique values, both of which can tell you details about the data counts. Distinct values are all the different values in a column, including duplicates and null values, while unique values do not include duplicates or nulls. Therefore, **distinct** in this table tells you the total count of how many values are present, while **unique** tells you how many of those values only appear once.

**Column profile** gives you a more in-depth look into the statistics within the columns for the first 1,000 rows of data. This column provides several different values, including the count of rows, which is important when verifying whether the importing of your data was successful. For example, if your original database had 100 rows, you could use this row count to verify that 100 rows were, in fact, imported correctly. Additionally, this row count will show how many rows that Power BI has deemed as being outliers, empty rows and strings, and the min and max, which will tell you the smallest and largest value in a column, respectively. This distinction is particularly important in the case of numeric data because it will immediately notify you if you have a maximum value that is beyond what your business identifies as a "maximum." This value calls to your attention these values, which means that you can then focus your efforts when delving deeper into the data. In the case where data was in the text column, as seen in the previous image, the minimum value is the first value and the maximum value is the last value when in alphabetical order.

Additionally, the **Value distribution** graph tells you the counts for each distinct value in that specific column. When looking at the graph in the previous image, notice that the value distribution indicates that "Anthony Grosse" appears the greatest number of times within the **SalesPerson** column and that "Lily Code" appears the least amount of times. This information is particularly important because it identifies outliers. If a value appears far more than other values in a column, the **Value distribution** feature allows you to pinpoint a place to begin your investigation into why this is so.

On a numeric column, **Column Statistics** will also include how many zeroes and null values exist, along with the average value in the column, the standard deviation of the values in the column, and how many even and odd values are in the column. These statistics give you an idea of the distribution of data within the column, and are important because they summarize the data in the column and serve as a starting point to determine what the outliers are.

For example, while looking through invoice data, you notice that the **Value distribution** graph shows that a few salespeople in the **SalesPerson** column appear the same amount of times within the data. Additionally, you notice the same situation has occurred in the **Profit** column and in a few other tables as well. During your investigation, you discover that the data you were using was bad data and needed to be refreshed, so you immediately complete the refresh. Without viewing this graph, you might not have seen this error so quickly and, for this reason, value distribution is essential.

After you have completed your edits in Power Query Editor and are ready to begin building visuals, return to **Home** on the Power Query Editor ribbon. Select **Close & Apply**, which will return you to Power BI Desktop and any column edits/transformations will also be applied.

You have now determined the elements that make up profiling data in Power BI, which include loading data in Power BI, interrogating column properties to gain clarity about and make further edits to the type and format of data in columns, finding data anomalies, and viewing data statistics in Power Query Editor. With this knowledge, you can include in your toolkit the ability to study your data in an efficient and effective manner.

**Next unit: Use Advanced Editor to modify M code**

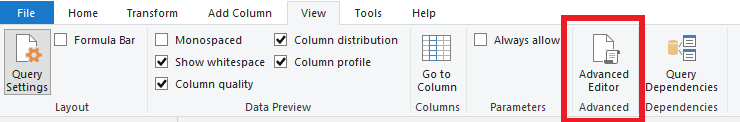
**Use Advanced Editor to modify M code**

100 XP

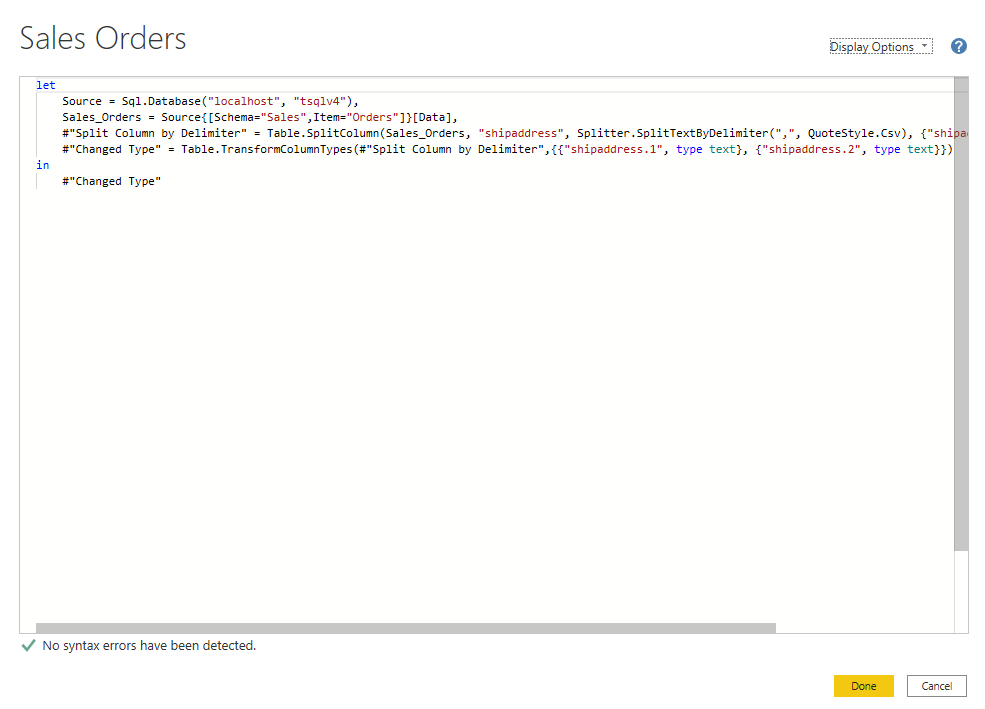
* 4 minutes

Each time you shape data in Power Query, you create a step in the Power Query process. Those steps can be reordered, deleted, and modified where it makes sense. Each cleaning step that you made was likely created by using the graphical interface, but Power Query uses the M language behind the scenes. The combined steps are available to read by using the Power Query Advanced Editor. The M language is always available to be read and modified directly.  It is not required that you use M code to take advantage of Power Query. You will rarely need to write M code, but it can still prove useful. Because each step in Power Query is written in M code, even if the UI created it for you, you can use those steps to learn M code and customize it to suit your needs.

After creating steps to clean data, select the **View** ribbon of Power Query and then select **Advanced Editor**.

[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/08-view-ribbon-advanced-button-ssm.png#lightbox)

The following screen should appear.

[](https://learn.microsoft.com/en-us/training/modules/clean-data-power-bi/media/08-m-code.png#lightbox)

Each Power Query step will roughly align with one or two lines of M code. You don't have to be an expert in M code to be able to read it. You can even experiment with changing it. For instance, if you need to change the name of a database, you could do it right in the code and then select **Done**.

You might notice that M code is written top-down. Later steps in the process can refer to previous steps by the variable name to the left of the equal sign. Be careful about reordering these steps because it could ruin the statement dependencies. Write to a query formula step by using the **in** statement. Generally, the last query step is used as the **in final data set** result.

**Next unit: Exercise - Load data in Power BI Desktop**