**Introduction**

Completed100 XP

* 5 minutes

Creating a great data model is one of the most important tasks that a data analyst can perform in Microsoft Power BI. By doing this job well, you help make it easier for people to understand your data, which will make building valuable Power BI reports easier for them and for you.

The pages in this module are instructional only, no data files are provided. You have a chance to work with real data in the labs.

[Introduction - Training | Microsoft Learn](https://learn.microsoft.com/en-us/training/modules/design-model-power-bi/1-introduction)

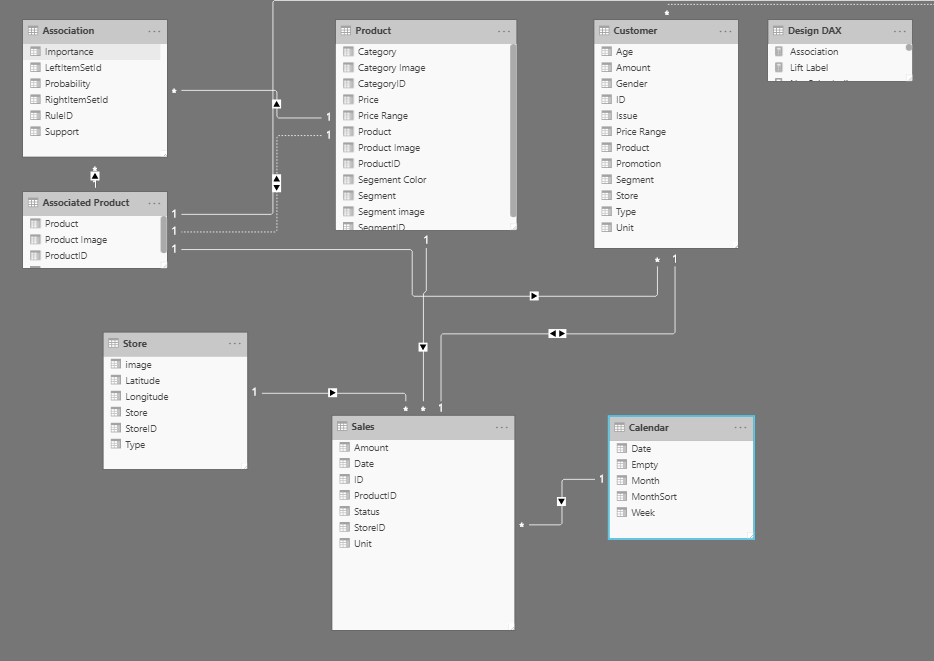
A good data model offers the following benefits:

* Data exploration is faster.
* Aggregations are simpler to build.
* Reports are more accurate.
* Writing reports takes less time.
* Reports are easier to maintain in the future.

Providing set rules for what makes a good data model is difficult because all data is different, and the usage of that data varies. Generally, a smaller data model is better because it performs faster and will be simpler to use. However, defining what a smaller data model entails is equally as problematic because it's a heuristic and subjective concept.

Typically, a smaller data model is composed of fewer tables and fewer columns in each table that the user can see. If you import all necessary tables from a sales database, but the total table count is 30 tables, the user will not find that intuitive. Collapsing those tables into five tables make the data model more intuitive to the user, whereas if the user opens a table and finds 100 columns, they might find it overwhelming. Removing unneeded columns to provide a more manageable number increases the likelihood that the user reads all column names. To summarize, you should aim for simplicity when designing your data models.

The following image is an example data model. The boxes contain tables of data, where each line item within the box is a column. The lines that connect the boxes represent relationships between the tables. These relationships can be complex, even in such a simplistic model. The data model can become easily disorganized, and the total table count in the model can gradually increase. Keeping your data model simple, comprehensive, and accurate requires constant effort.

[](https://learn.microsoft.com/en-us/training/modules/design-model-power-bi/media/01-example-data-model-01-ss.png#lightbox)

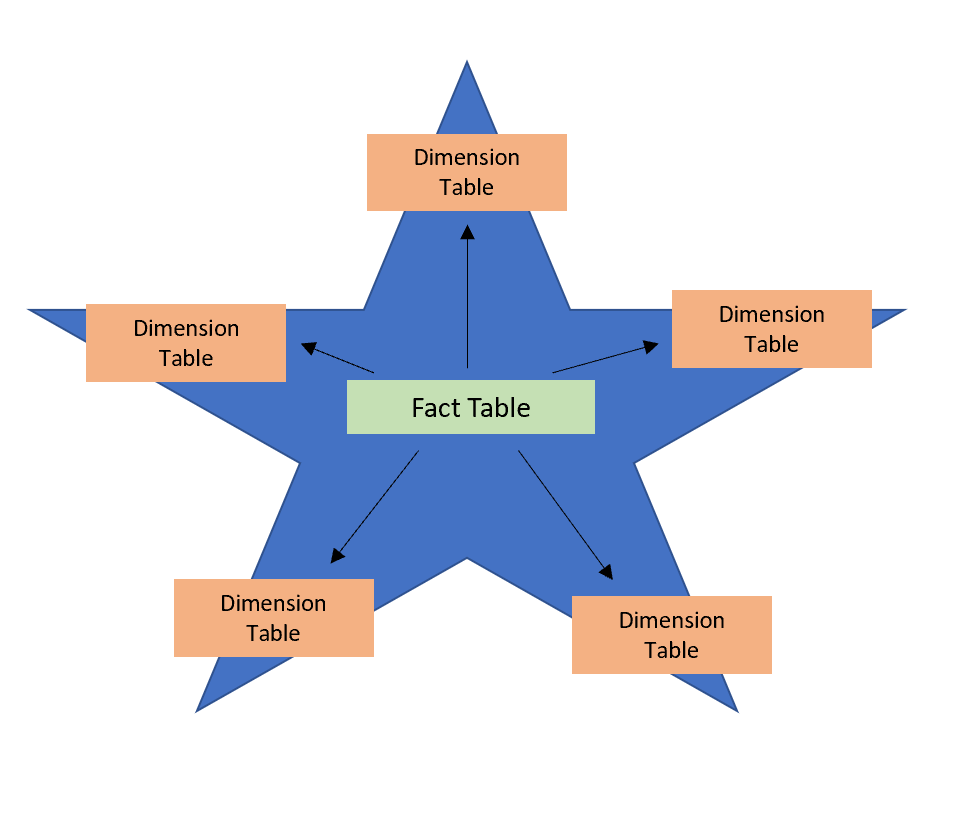
Relationships are defined between tables through primary and foreign keys. Primary keys are column(s) that identify each unique, non-null data row. For instance, if you have a Customers table, you could have an index that identifies each unique customer. The first row has an ID of 1, the second row an ID of 2, and so on. Each row is assigned a unique value, which can be referred to by this simple value: the primary key. This process becomes important when you are referencing rows in a different table, which is what foreign keys do. Relationships between tables are formed when you have primary and foreign keys in common between different tables.

Power BI allows relationships to be built from tables with different data sources, a powerful function that enables you to pull one table from Microsoft Excel and another from a relational database. You would then create the relationship between those two tables and treat them as a unified dataset.

Now that you have learned about the relationships that make up the data schema, you are able to explore a specific type of schema design, the star schema, which is optimized for high performance and usability.

**Star schemas**

You can design a star schema to simplify your data. It's not the only way to simplify your data, but it is a popular method; therefore, every Power BI data analyst should understand it. In a star schema, each table within your dataset is defined as a dimension or a fact table, as shown in the following visual.

[](https://learn.microsoft.com/en-us/training/modules/design-model-power-bi/media/01-star-schema-example-01-ss.png#lightbox)

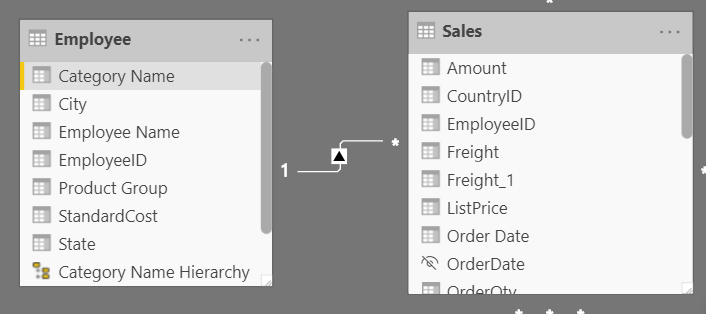
**Fact tables** contain observational or event data values: sales orders, product counts, prices, transactional dates and times, and quantities. Fact tables can contain several repeated values. For example, one product can appear multiple times in multiple rows, for different customers on different dates. These values can be aggregated to create visuals. For instance, a visual of the total sales orders is an aggregation of all sales orders in the fact table. With fact tables, it is common to see columns that are filled with numbers and dates. The numbers can be units of measurement, such as sale amount, or they can be keys, such as a customer ID. The dates represent time that is being recorded, like order date or shipped date.

**Dimension tables** contain the details about the data in fact tables: products, locations, employees, and order types. These tables are connected to the fact table through key columns. Dimension tables are used to filter and group the data in fact tables. The fact tables, on the other hand, contain the measurable data, such as sales and revenue, and each row represents a unique combination of values from the dimension tables. For the total sales orders visual, you could group the data so that you see total sales orders by product, in which product is data in the dimension table.

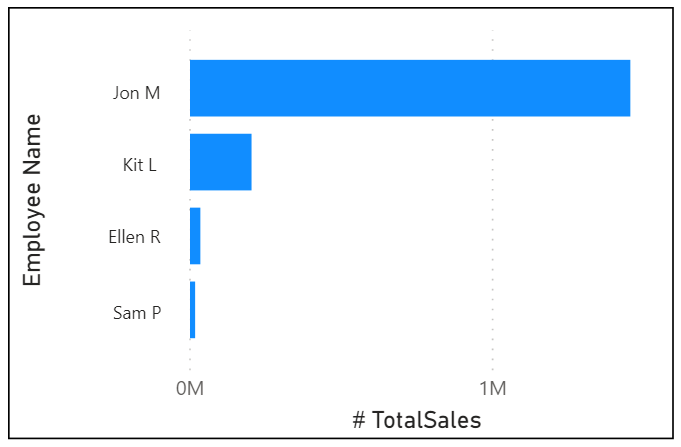
Fact tables are much larger than dimension tables because numerous events occur in fact tables, such as individual sales. Dimension tables are typically smaller because you are limited to the number of items that you can filter and group on. For instance, a year contains only so many months, and the United States are composed of only a certain number of states.

Considering this information about fact tables and dimension tables, you might wonder how you can build this visual in Power BI.

The pertinent data resides in two tables, Employee and Sales, as shown in the following data model. Because the Sales table contains the sales order values, which can be aggregated, it is considered a fact table. The Employee table contains the specific employee name, which filters the sales orders, so it would be a dimension table. The common column between the two tables, which is the primary key in the Employee table, is **EmployeeID**, so you can establish a relationship between the two tables based on this column.

[](https://learn.microsoft.com/en-us/training/modules/design-model-power-bi/media/01-data-model-relationships-ss.png#lightbox)

When creating this relationship, you can build the visual according to the requirements, as shown in the following figure. If you did not establish this relationship, while keeping in mind the commonality between the two tables, you would have had more difficulty building your visual.

[](https://learn.microsoft.com/en-us/training/modules/design-model-power-bi/media/01-data-schema-example-04-ss.png#lightbox)

Star schemas and the underlying data model are the foundation of organized reports; the more time you spend creating these connections and design, the easier it will be to create and maintain reports.

**Next unit: Work with tables**

**Work with tables**

Completed100 XP

* 3 minutes

When users see fewer tables, they will enjoy using your data model considerably more. For example, suppose you've imported dozens of tables from many data sources and now the visual appears disorderly. In this case, you need to ensure that, before you begin working on building reports, your data model and table structure are simplified.

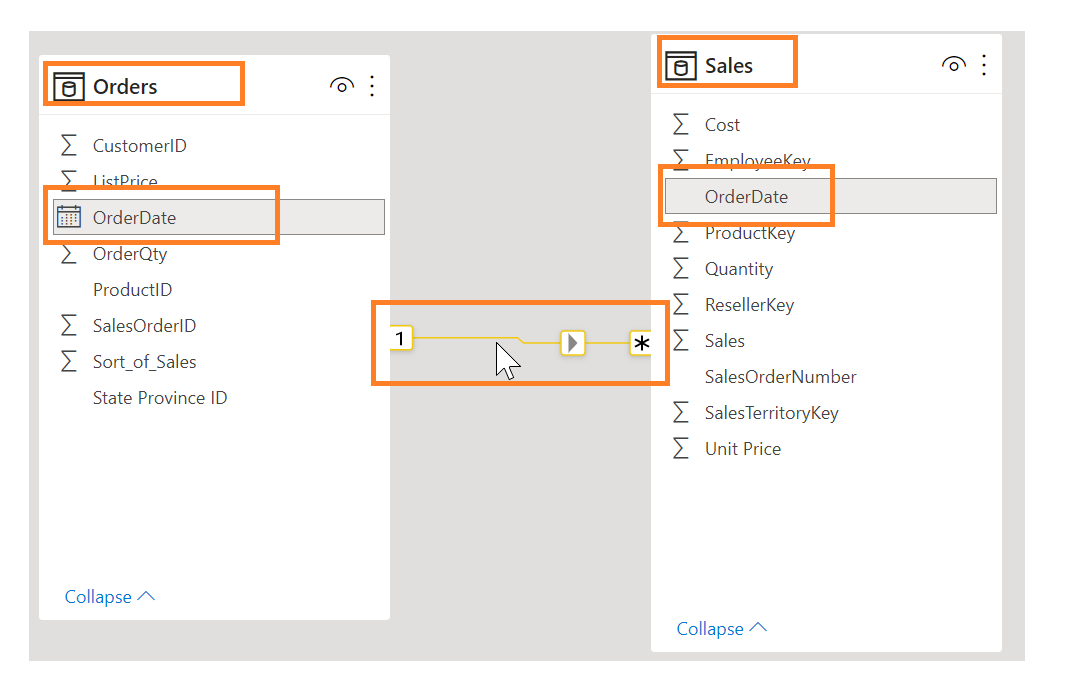
A simple table structure will:

* Be simple to navigate because of column and table properties that are specific and user-friendly.
* Have merged or appended tables to simplify the tables within your data structure.
* Have good-quality relationships between tables that make sense.

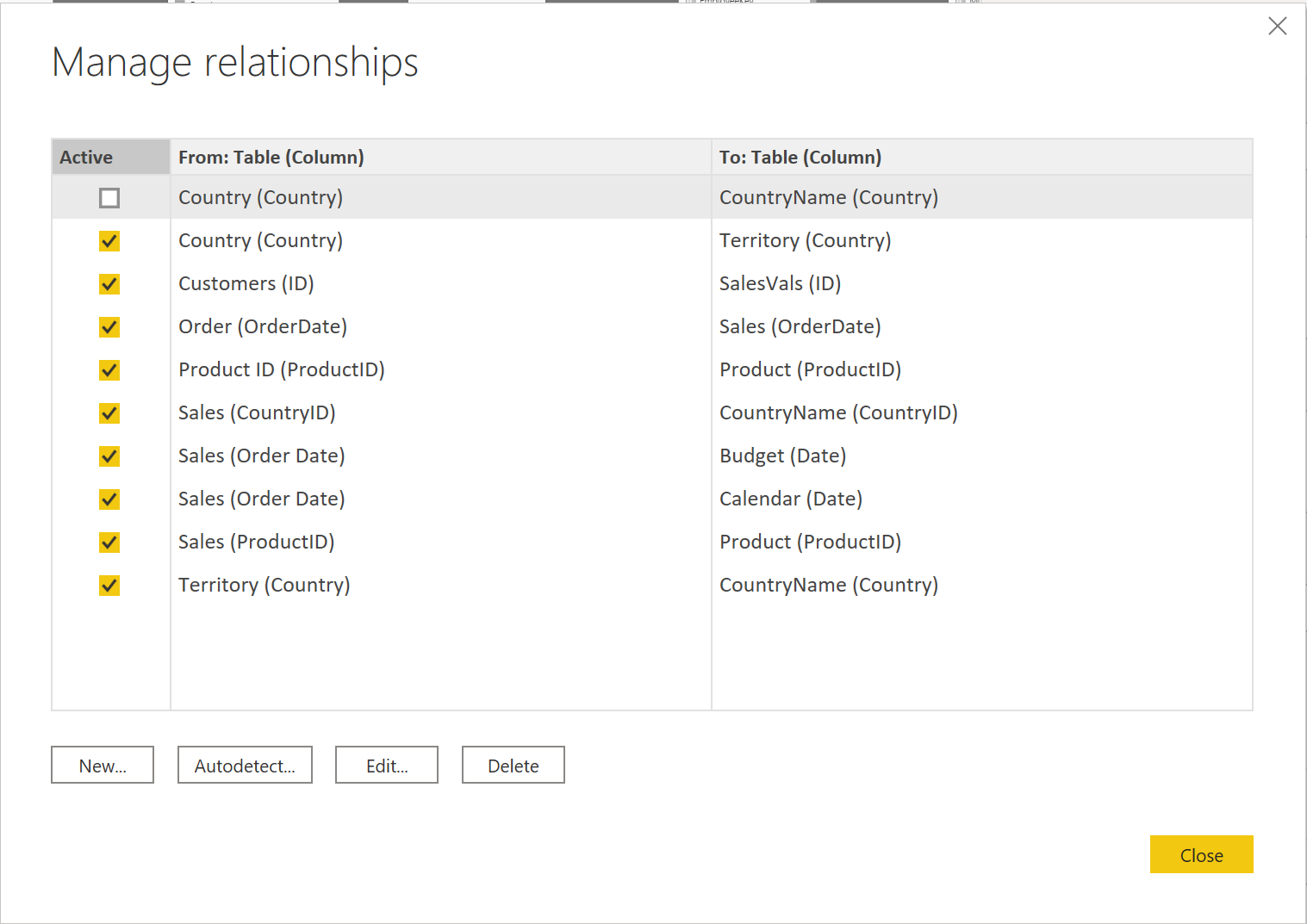
The following sections further explain how you might work with your tables to ensure a simple and readable table structure.

**Configure data model and build relationships between tables**

Assuming that you've already retrieved your data and cleaned it in Power Query, you can then go to the **Model** tab, where the data model is located. The following image shows how the relationship between the **Order** and **Sales** tables can be seen through the **OrderDate** column.

[](https://learn.microsoft.com/en-us/training/modules/design-model-power-bi/media/02-data-model-example-01-ssm.png#lightbox)

To manage these relationships, go to **Manage Relationships** on the ribbon, where the following window will appear.

[](https://learn.microsoft.com/en-us/training/modules/design-model-power-bi/media/02-manage-relationships-window-02-ss.png#lightbox)

In this view, you can create, edit, and delete relationships between tables and also autodetect relationships that already exist. When you load your data into Power BI, the **Autodetect** feature will help you establish relationships between columns that are named similarly. Relationships can be inactive or active. Only one active relationship can exist between tables, which is discussed in a future module.

While the **Manage Relationships** feature allows you to configure relationships between tables, you can also configure table and column properties to ensure organization in your table structure.

**Configure table and column properties**

The **Model** view in Power BI desktop provides many options within the column properties that you can view or update. A simple method to get to this menu to update the tables and fields is by Ctrl+clicking or Shift+clicking items on this page.

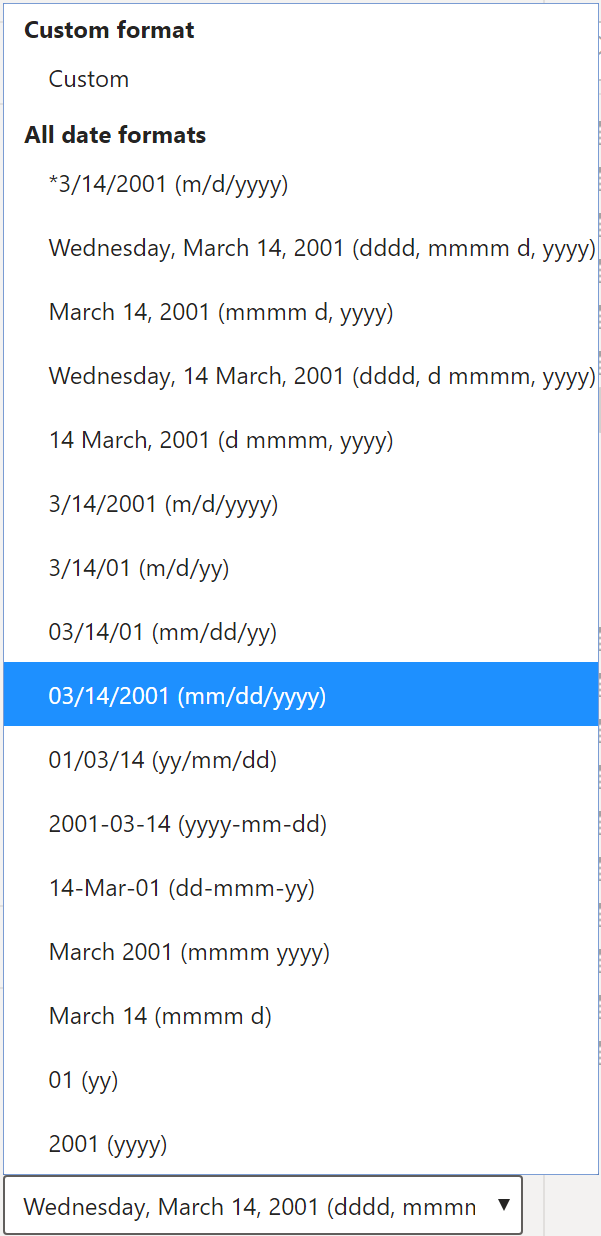
Under the **General** tab, you can:

* Edit the name and description of the column.
* Add synonyms that can be used to identify the column when you are using the Q&A feature.
* Add a column into a folder to further organize the table structure.
* Hide or show the column.

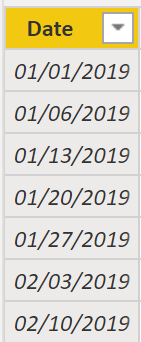
Under the **Formatting** tab, you can:

* Change the data type.
* Format the date.

For instance, suppose that the dates in your column are formatted, as seen in the previous screenshot, in the form of "Wednesday, March 14, 2001". If you want to change the format so that the date was in the "mm/dd/yyyy" format, you would select the drop-down menu under **All date time formats** and then choose the appropriate date format, as shown in the following figure.

[](https://learn.microsoft.com/en-us/training/modules/design-model-power-bi/media/02-change-date-format-05-ss.png#lightbox)

After selecting the appropriate date format, return to the **Date** column, where you should see that the format has indeed changed, as shown in the following figure.

[](https://learn.microsoft.com/en-us/training/modules/design-model-power-bi/media/02-result-changing-format-06-ss.png#lightbox)

Under the **Advanced** tab, you can:

* Sort by a specific column.
* Assign a specific category to the data.
* Summarize the data.
* Determine if the column or table contains null values.

Additionally, Power BI has a new functionality to update these properties on many tables and fields by Ctrl+clicking or Shift+clicking items.

These examples are only some of the many types of transformations that you can make to simplify the table structure. This step is important to take before you begin making your visuals so that you don't have to go back and forth when making formatting changes. This process of formatting and configuring tables can also be done in Power Query.

**Next unit: Create a date table**

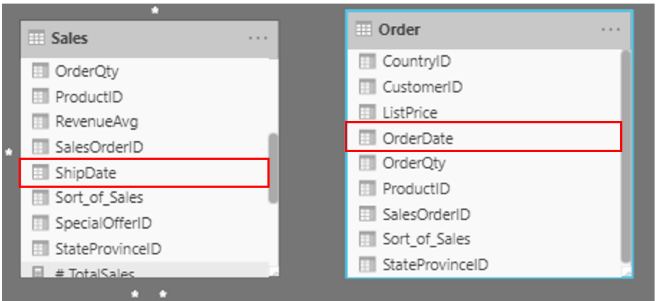
**Create a date table**

Completed100 XP

* 9 minutes

During report creation in Power BI, a common business requirement is to make calculations based on date and time. Organizations want to know how their business is doing over months, quarters, fiscal years, and so on. For this reason, it is crucial that these time-oriented values are formatted correctly. Power BI autodetects for date columns and tables; however, situations can occur where you will need to take extra steps to get the dates in the format that your organization requires.

For example, suppose that you are developing reports for the Sales team at your organization. The database contains tables for sales, orders, products, and more. You notice that many of these tables, including Sales and Orders, contain their own date columns, as shown by the **ShipDate** and **OrderDate** columns in the Sales and Orders tables. You are tasked with developing a table of the total sales and orders by year and month. How can you build a visual with multiple tables, each referencing their own date columns?

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

To solve this problem, you can create a common date table that can be used by multiple tables. The following section explains how you can accomplish this task in Power BI.

**Create a common date table**

Ways that you can build a common date table are:

* Source data
* DAX
* Power Query

**Source data**

Occasionally, source databases and data warehouses already have their own date tables. If the administrator who designed the database did a thorough job, these tables can be used to perform the following tasks:

* Identify company holidays
* Separate calendar and fiscal year
* Identify weekends versus weekdays

Source data tables are mature and ready for immediate use. If you have a table as such, bring it into your data model and don't use any other methods that are outlined in this section. We recommend that you use a source date table because it is likely shared with other tools that you might be using in addition to Power BI.

If you do not have a source data table, you can use other ways to build a common date table.

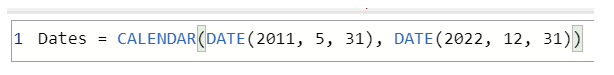
**DAX**

You can use the Data Analysis Expression (DAX) functions CALENDARAUTO() or CALENDAR() to build your common date table. The CALENDAR() function returns a contiguous range of dates based on a start and end date that are entered as arguments in the function. Alternatively, the CALENDARAUTO() function returns a contiguous, complete range of dates that are automatically determined from your dataset. The starting date is chosen as the earliest date that exists in your dataset, and the ending date is the latest date that exists in your dataset plus data that has been populated to the fiscal month that you can choose to include as an argument in the CALENDARAUTO() function. For the purposes of this example, the CALENDAR() function is used because you only want to see the data from May 31, 2011 (the first day that Sales began its tracking of this data) and forward for the next 10 years.

In Power BI Desktop, go to the **Table** tab on the ribbon. Select **New Table**, and then enter in the following DAX formula:

DAXCopy

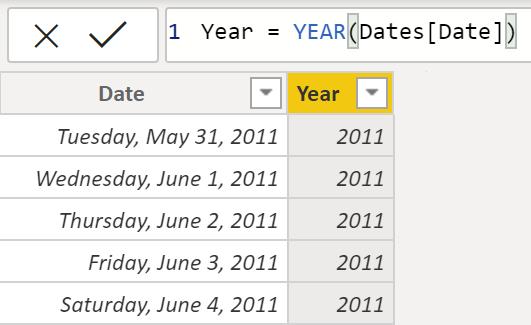
Dates = CALENDAR(DATE(2011, 5, 31), DATE(2022, 12, 31))

[](https://learn.microsoft.com/en-us/training/modules/design-model-power-bi/media/03-dax-function-calendar-auto-01-ss.png#lightbox)

Now, you have a column of dates that you can use. However, this column is slightly sparse. You also want to see columns for just the year, the month number, the week of the year, and the day of the week. You can accomplish this task by selecting **New Column** on the ribbon and entering the following DAX equation, which will retrieve the year from your Date table.

DAXCopy

Year = YEAR(Dates[Date])

[](https://learn.microsoft.com/en-us/training/modules/design-model-power-bi/media/03-adding-columns-dax.png#lightbox)

You can perform the same process to retrieve the month number, week number, and day of the week:

DAXCopy

MonthNum = MONTH(Dates[Date])

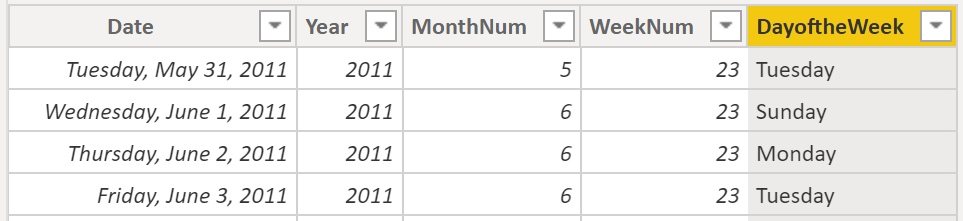
DAXCopy

WeekNum = WEEKNUM(Dates[Date])

DAXCopy

DayoftheWeek = FORMAT(Dates[Date], "DDDD")

When you have finished, your table will contain the columns that are shown in the following figure.

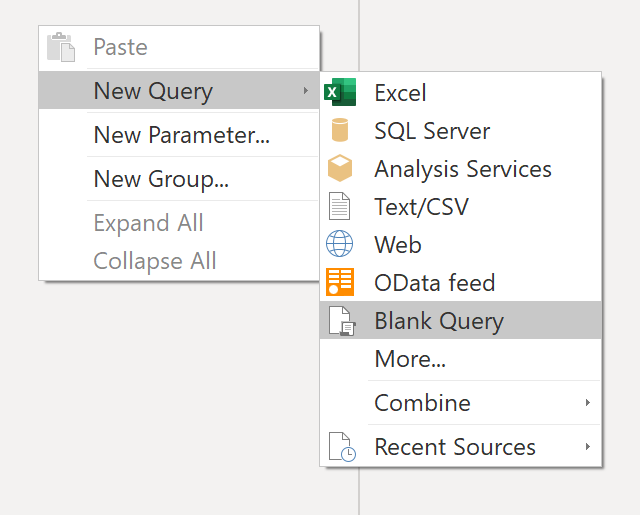
[](https://learn.microsoft.com/en-us/training/modules/design-model-power-bi/media/03-final-columns-dax-table-2-ss.png#lightbox)

You have now created a common date table by using DAX. This process only adds your new table to the data model; you will still need to establish relationships between your date table and the Sales and Order tables, and then mark your table as the official date table of your data model. However, before you complete those tasks, make sure that you consider another way of building a common date table: by using Power Query.

**Power Query**

You can use M-language, the development language that is used to build queries in Power Query, to define a common date table.

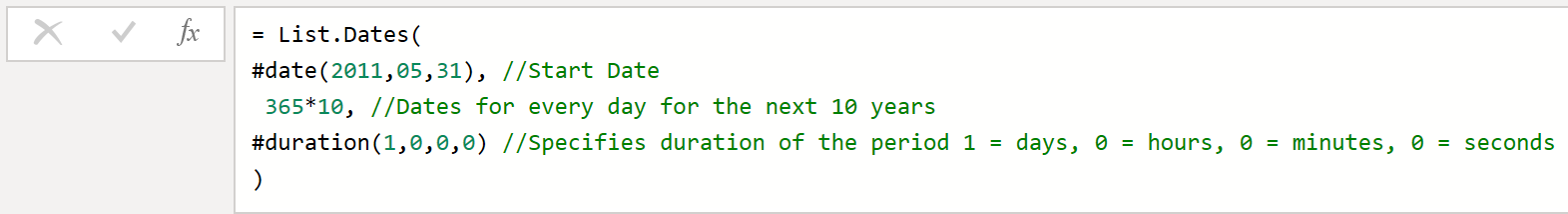
Select **Transform Data** in Power BI Desktop, which will direct you to Power Query. In the blank space of the left **Queries** pane, right-click to open the following drop-down menu, where you will select **New Query > Blank Query**.

[](https://learn.microsoft.com/en-us/training/modules/design-model-power-bi/media/03-new-query-common-date-table-03-ss.png#lightbox)

In the resulting **New Query** view, enter the following M-formula to build a calendar table:

MCopy

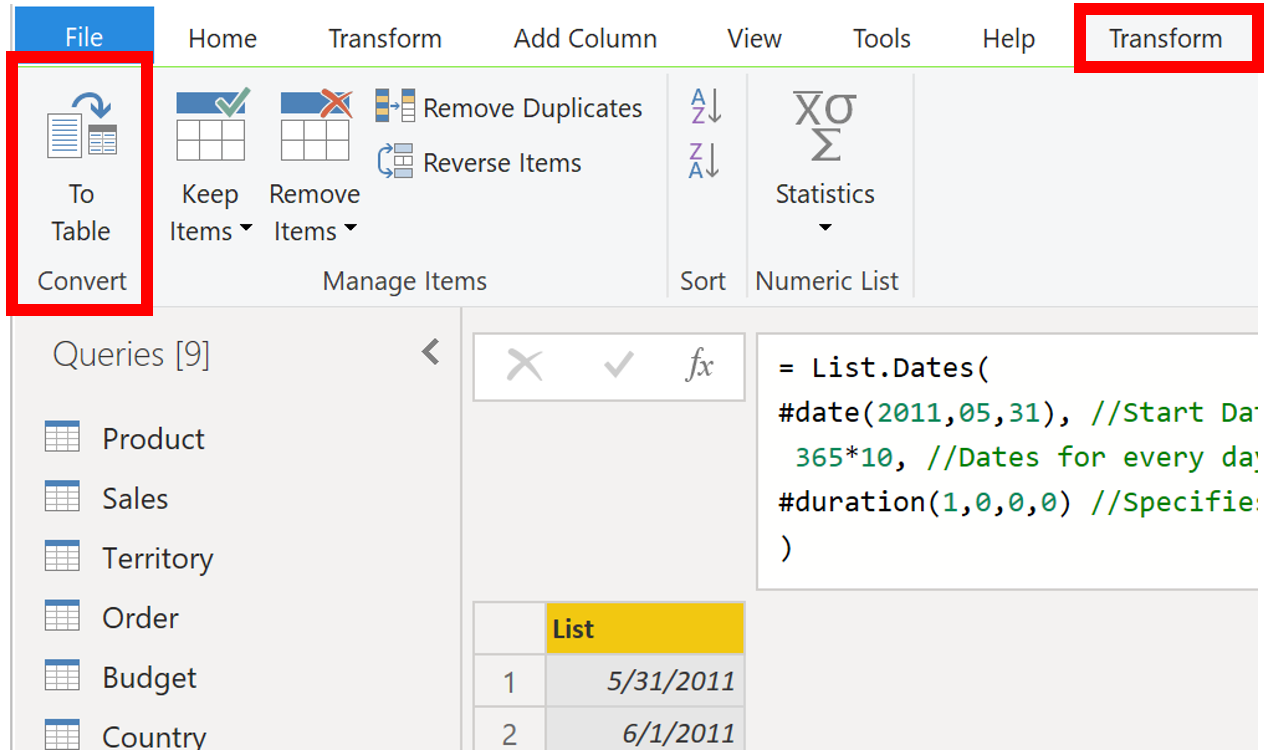
= List.Dates(#date(2011,05,31), 365\*10, #duration(1,0,0,0))

[](https://learn.microsoft.com/en-us/training/modules/design-model-power-bi/media/03-m-query-common-data-table-04-ss.png#lightbox)

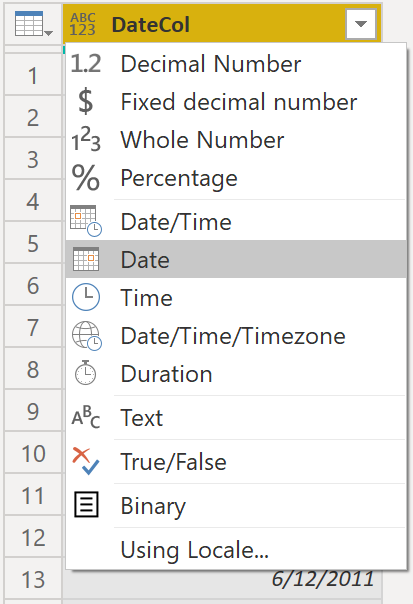
For your sales data, you want the start date to reflect the earliest date that you have in your data: May 31, 2011. Additionally, you want to see dates for the next 10 years, including dates in the future. This approach ensures that, as new sales data flows in, you won't have to re-create this table. You can also change duration. In this case, you want a data point for every day, but you can also increment by hours, minutes, and seconds. The following figure shows the result.

[](https://learn.microsoft.com/en-us/training/modules/design-model-power-bi/media/03-list-power-query-11-ss.png#lightbox)

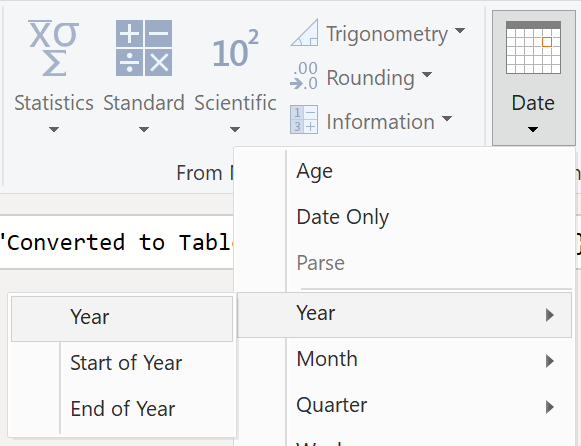
After you have realized success in the process, you notice that you have a list of dates instead of a table of dates. To correct this error, go to the **Transform** tab on the ribbon and select **Convert > To Table**. As the name suggests, this feature will convert your list into a table. You can also rename the column to **DateCol**.

[](https://learn.microsoft.com/en-us/training/modules/design-model-power-bi/media/03-converting-list-table-05-ssm.png#lightbox)

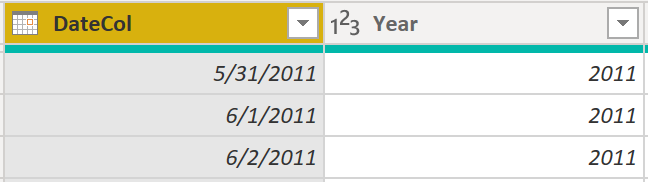
Next, you want to add columns to your new table to see dates in terms of year, month, week, and day so that you can build a hierarchy in your visual. Your first task is to change the column type by selecting the icon next to the name of the column and, in the resulting drop-down menu, selecting the **Date** type.

[](https://learn.microsoft.com/en-us/training/modules/design-model-power-bi/media/03-change-type-date-3-ss.png#lightbox)

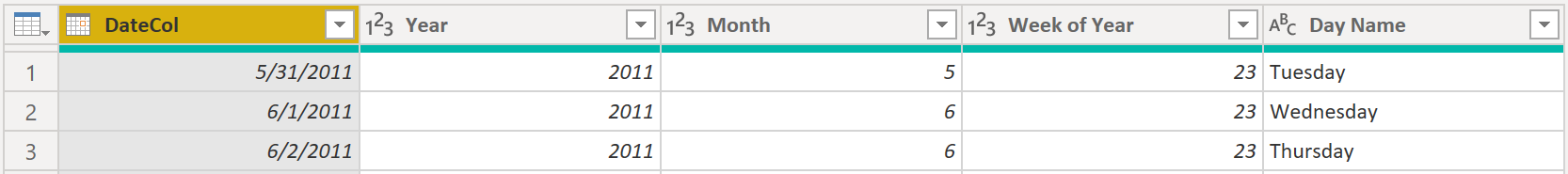
After you have finished selecting the **Date** type, you can add columns for year, months, weeks, and days. Go to **Add Column**, select the drop-down menu under **Date**, and then select **Year**, as shown in the following figure.

[](https://learn.microsoft.com/en-us/training/modules/design-model-power-bi/media/03-adding-columns-power-query-5-ss.png#lightbox)

Notice that Power BI has added a column of all years that are pulled from **DateCol**.

[](https://learn.microsoft.com/en-us/training/modules/design-model-power-bi/media/03-adding-columns-table-power-query-6-ss.png#lightbox)

Complete the same process for months, weeks, and days. After you have finished this process, the table will contain the columns that are shown in the following figure.

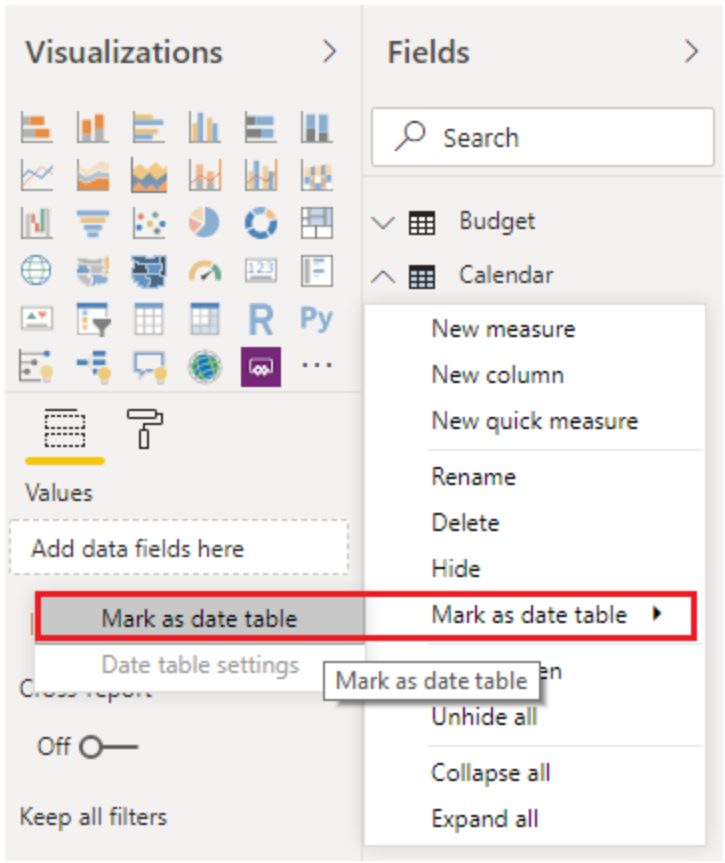
[](https://learn.microsoft.com/en-us/training/modules/design-model-power-bi/media/03-final-columns-using-power-query-7-ss.png#lightbox)

You have now successfully used Power Query to build a common date table.

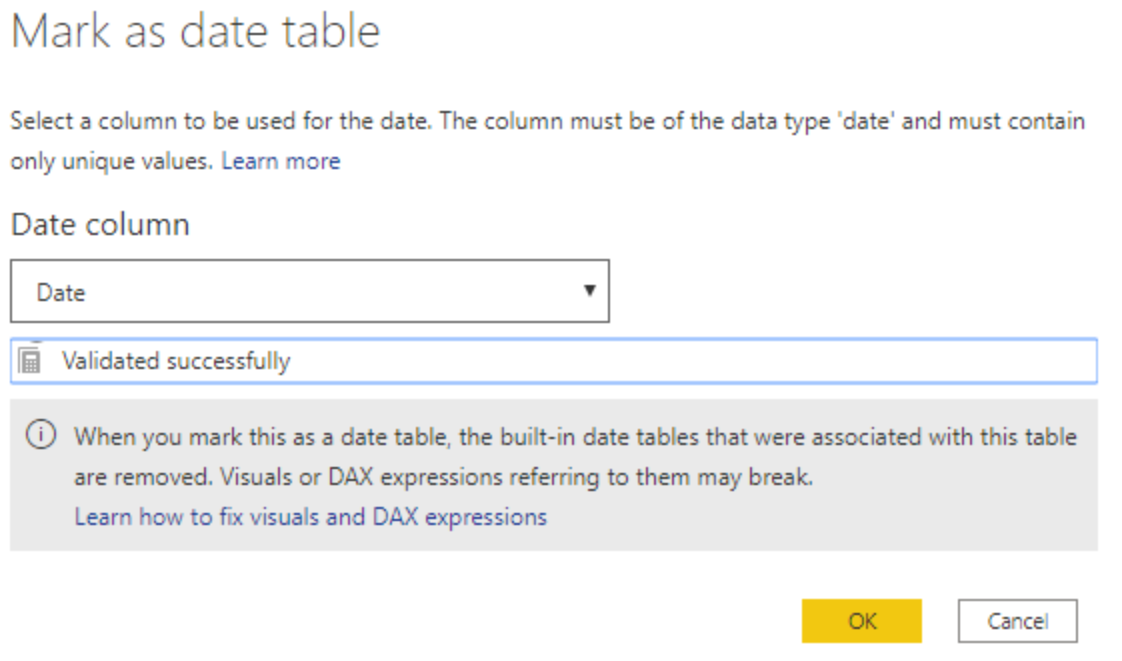
The previous steps show how to get the table into the data model. Now, you need to mark your table as the official date table so that Power BI can recognize it for all future values and ensure that formatting is correct.

**Mark as the official date table**

Your first task in marking your table as the official date table is to find the new table on the **Fields** pane. Right-click the name of the table and then select **Mark as date table**, as shown in the following figure.

[](https://learn.microsoft.com/en-us/training/modules/design-model-power-bi/media/03-mark-date-table-06-ss.png#lightbox)

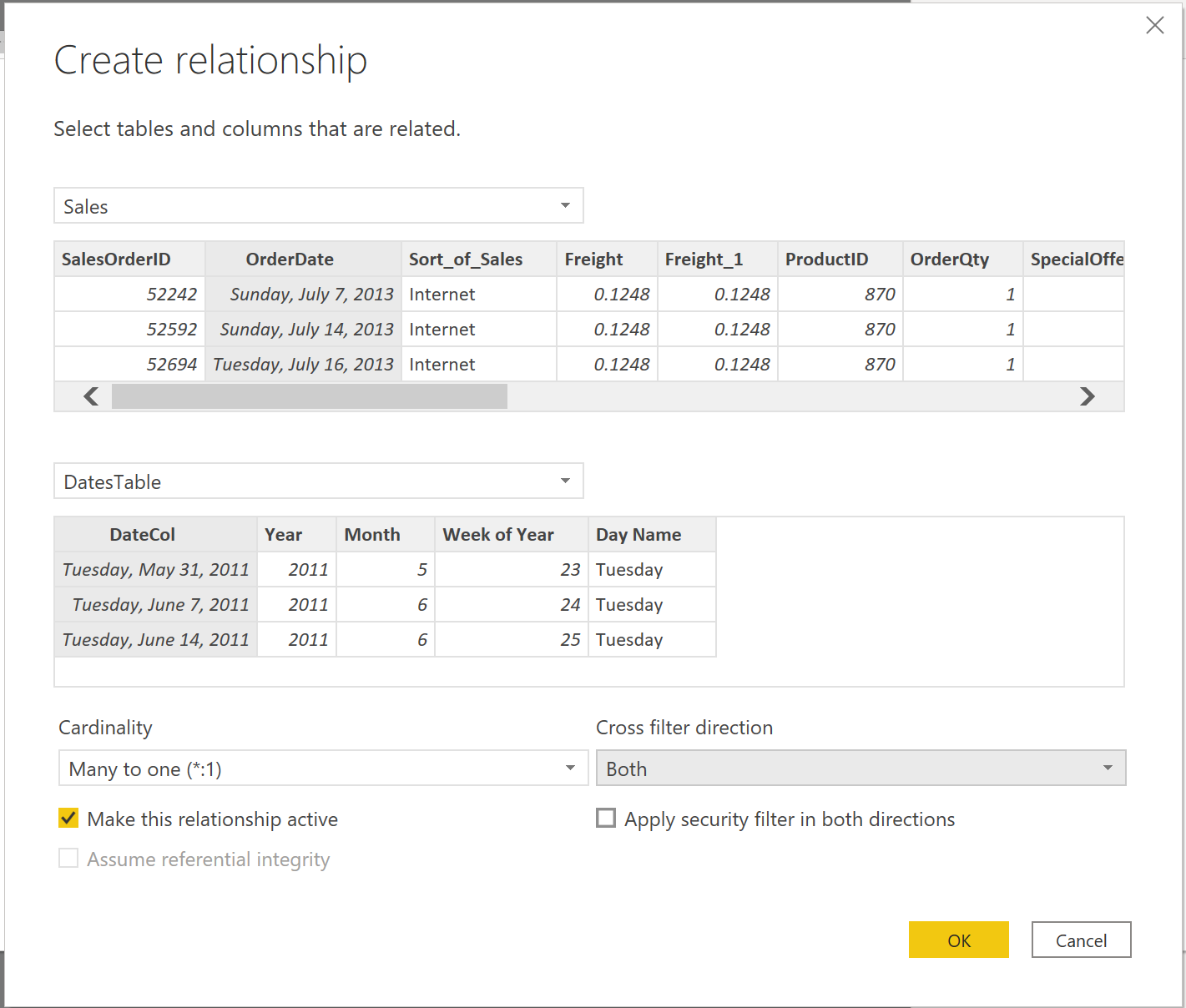
By marking your table as a date table, Power BI performs validations to ensure that the data contains zero null values, is unique, and contains continuous date values over a period. You can also choose specific columns in your table to mark as the date, which can be useful when you have many columns within your table. Right-click the table, select **Mark as date table**, and then select **Date table settings.** The following window will appear, where you can choose which column should be marked as **Date**.

[](https://learn.microsoft.com/en-us/training/modules/design-model-power-bi/media/03-mark-date-table-07-ss.png#lightbox)

Selecting **Mark as date table** will remove autogenerated hierarchies from the **Date** field in the table that you marked as a date table. For other date fields, the auto hierarchy will still be present until you establish a relationship between that field and the date table or until you turn off the **Auto Date/Time** feature. You can manually add a hierarchy to your common date table by right-clicking the year, month, week, or day columns in the **Fields** pane and then selecting **New hierarchy.** This process is further discussed later in this module.

**Build your visual**

To build your visual between the Sales and Orders tables, you will need to establish a relationship between this new common date table and the Sales and Orders tables. As a result, you will be able to build visuals by using the new date table. To complete this task, go to **Model** tab **>** **Manage Relationships**, where you can create relationships between the common date table and the Orders and Sales tables by using the **OrderDate** column. The following screenshot shows an example of one such relationship.

[](https://learn.microsoft.com/en-us/training/modules/design-model-power-bi/media/03-create-relationship-8-ss.png#lightbox)

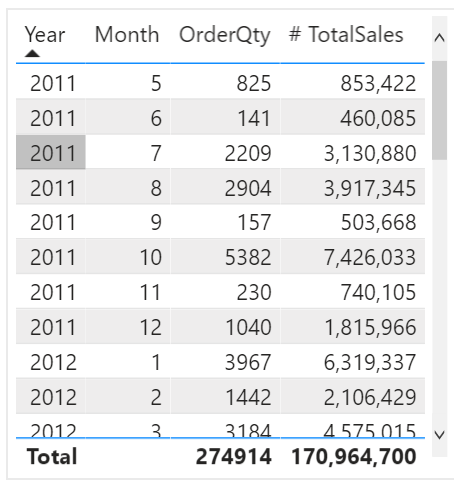
After you have built the relationships, you can build your **Total Sales and Order Quantity by Time** visual with your common date table that you developed by using the DAX or Power Query method.

To determine the total sales, you need to add all sales because the **Amount** column in the Sales table only looks at the revenue for each sale, not the total sales revenue. You can complete this task by using the following measure calculation, which will be explained in later discussions. The calculation that you will use when building this measure is as follows:

DAXCopy

#Total Sales = SUM(Sales[‘Amount’])

After you have finished, you can create a table by returning to the **Visualizations** tab and selecting the **Table** visual. You want to see the total orders and sales by year and month, so you only want to include the Year and Month columns from your date table, the **OrderQty** column, and the **#TotalSales** measure. When you learn about hierarchies, you can also build a hierarchy that will allow you drill down from years to months. For this example, you can view them side-by-side. You have now successfully created a visual with a common date table.

[](https://learn.microsoft.com/en-us/training/modules/design-model-power-bi/media/03-common-date-dax-5-ss.png#lightbox)

**Next unit: Work with dimensions**