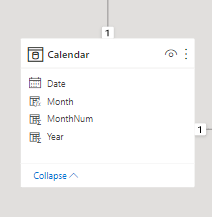
**How to Create Date Tables in Power BI**

**What are Power BI Date Tables?**

Date tables in Power BI are tables that only contain date-related data. It is a standard dimension table that can be used to reference dates in your model and analyze data based on these dates. They are also useful for time intelligence calculations and when creating reports that require precise date information.

**Why are Date Tables Useful in Data Analysis?**

Date and time-based analyses are usually required in Power BI reports. This is where making a date table comes in handy. Date tables allow you to slice and dice your data by date attributes such as weekday, month, quarter, and year. They also allow you to use DAX time intelligence functions that would not normally work without a date table. It is necessary to have all the columns formatted correctly when utilizing date tables for proper analysis.



**Requirements for Creating a Data Table in Power BI**

Some of the requirements for a date table are as follows:

* The date table should have a date column with datatype date/time.
* A date column should not have blanks.
* A date column must have unique values.
* There should be no dates missing from a date column.
* The date table must span whole years, i.e. it must encompass all date values stored in the date table, which could be calendar years (January-December) or fiscal years, for example.
* The date column must be marked as a Date Table (this will be discussed in detail later in this tutorial).

**Names for Power BI Date Tables**

Date tables are also known by various other names, such as calendar table, date dimension table, and calendar dimension table. All these names refer to the same thing: a table with one record per day and a column that displays the date's attribute.

**Generating Power BI Date Tables**

There are four major ways in which date tables can be generated in Power BI:

* Source Data
* Auto Date/Time
* DAX
* Power Query

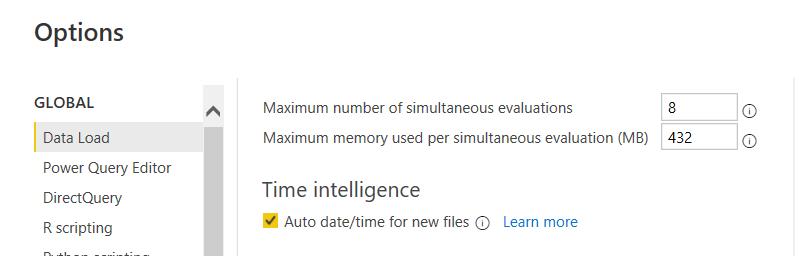
**Source Data**

When you import your data, it may already have a date table that was created in the data source. In this case, there is no need to create another date table. This date table is ready to use, so you can simply bring it into the data model and a relationship with other tables in your data model will be created.

If the date table does not come with the source data, there are several methods for creating it.

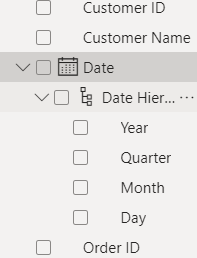
**Auto Date/Time**

When filtering your data over date periods, the auto date/time approach makes use of simple time intelligence based on date columns already loaded into your model. It does not, however, provide a date table that may be used to slice and dice other tables. To use this method, you must first enable Auto date/time in Power BI. Navigate to the **File** ribbon **> Options and Settings > Options > Data Load > Current File > Time Intelligence > Enable Auto date/time.**

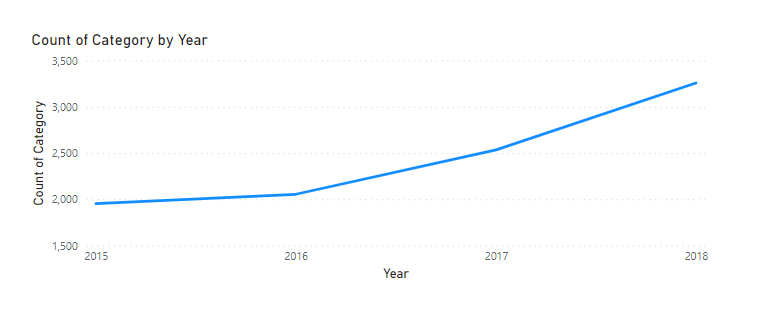


After the option is enabled, Power BI Desktop will create a hidden auto date/time table based on the dates in the date column. It then creates a relationship between the hidden auto date/time date column and the date column in the model.

When there is an auto date/time table, it will not appear as a field in the **Fields** pane. Instead, it can be found as an extendable drop-down with the name of the date column, as shown below. When you expand the date column, you will see a hierarchy called the Date Hierarchy, which includes Year, Quarter, Month, and Day.



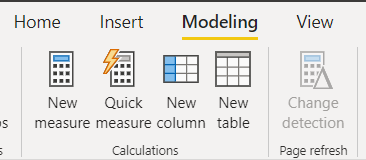
This can then be used to create a visual.



**DAX**

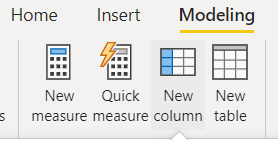
Another method for creating a date dimension table in Power BI is to use Data Analysis Expression (DAX) methods. CalendarAuto and Calendar are frequently used to create these tables. The difference between these two functions is that the Calendar function returns a range of dates based on the start and end dates specified as parameters within the function, whereas the CalendarAuto function returns a range of dates that are automatically detected from the dataset. The start date is the earliest date in your dataset, and the end date is the most recent date in your dataset.

To use the Calendar function, navigate to the Table tab on the ribbon in Power BI Desktop. Select **New Table**, then input the DAX formula as shown below.



Power BI DAX Formula 2

The above DAX function creates a date table with a date column. Other columns, such as Year, Month, Weekday, and Week of the Year, can be added to the table. To do so, select the **New Column** button on the ribbon and input the DAX equation for each column you want to add. In the following examples, we will write the DAX equation to get the year, month, and month numbers from the date table.

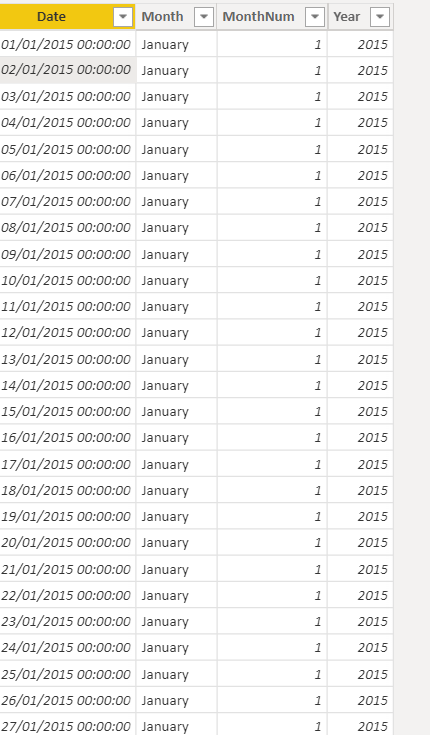


Power BI DAX Date Formula

Power BI DAX Date Formula 2

Power BI DAX Date Formula  3

The results of the DAX equations written for all of these new columns are shown below:



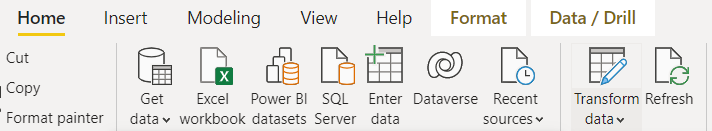
Other expressions can be used to get as many additional date-related columns as you want.

We have just used DAX to create a date table. This method, however, simply adds your new table to the data model; you must still create relationships between your date tables and then mark your table as the data model's official date table.

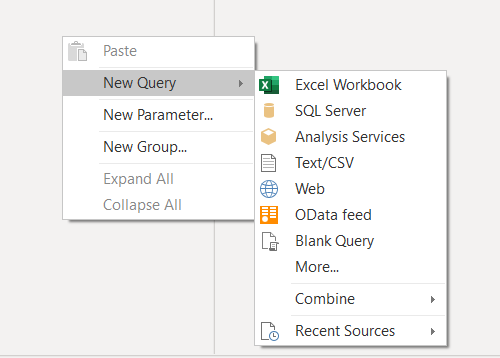
**Power Query**

To create a date table in Power BI, use the mash-up language, often known as M-Query.

To do so, click the **Transform Data** button on the ribbon and then navigate to Power Query.



Right-click in the empty space of the left Queries pane to access the following drop-down menu where you will select **New Query**and **Blank Query**.



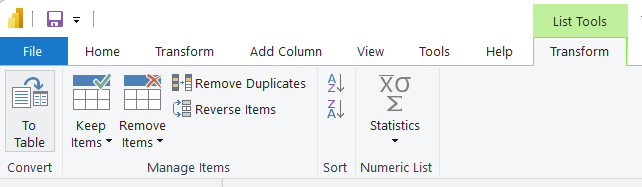
In the blank query tab, enter the M-query to create the date table as seen below:

Power BI M-query

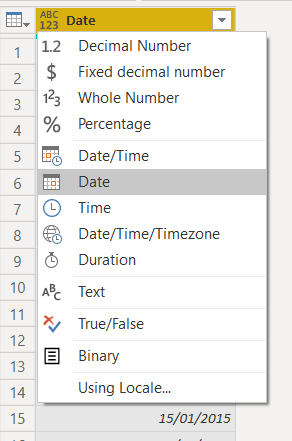
The **#date** argument indicates the earliest day in your data's start year, month, and day, and **365\*7**represents the date for the next 7 years. The days, hours, minutes, and seconds are represented by **#duration**, and **#duration(1,0,0,0)** indicates 1 day, 0 hours, 0 minutes, and 0 seconds in the query above. The advantage of this approach for creating date tables over others is that it will automatically update when new data comes in, omitting the need to recreate the table.



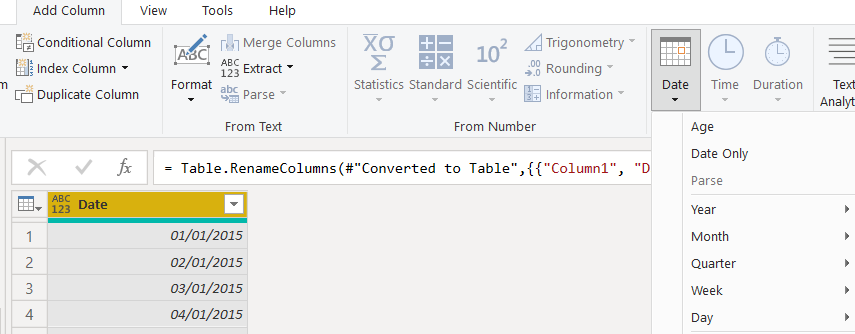
To change the result of the M-equation from a list of dates to a table of dates, navigate to the **Transform** tab on the ribbon and select **Convert** and then **To Table**.



After this has been done, you can include other date-related columns just as we did with the DAX equation approach for creating date tables. To do this, you have to first change the data type of the date column to **Date** by selecting the icon on the left side of the column name.



After changing the data type, you may add new columns to the table by navigating to the **Add Column** ribbon, selecting the dropdown just beneath **Date**, and then selecting **Year** or any other column you want to add.

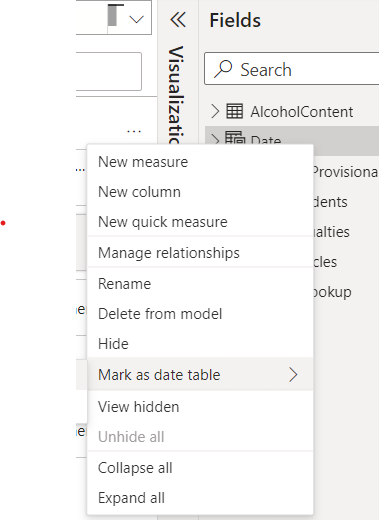


As seen above, the date drop-down allows you to enter the Year, Month, Quarter, Week, Day, and Age.

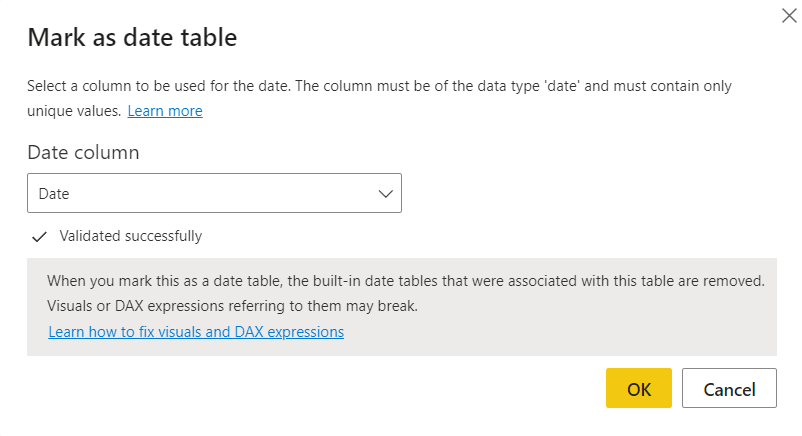
We have now successfully used Power Query to create a date table. You may now mark your newly created date table as such after pulling it into the data model.

**Marking a Table as the Date Table**

After creating the date table using one of the approaches described above, the following step is to mark the table as a Date Table. To do so, right-click the table's name in the **Fields** pane and select **Mark as date table**.



Power BI verifies the data in the table by marking it as a date table, ensuring that the date column is of data type Date and contains unique values.

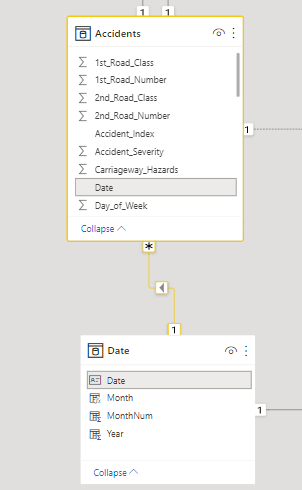


When a table is marked as a date table, the autogenerated hierarchies for the date field in the date table are deleted; but the hierarchies for other date fields in other tables remain until a relationship is established between that field and the generated date table.

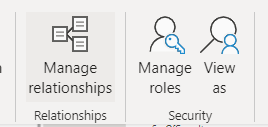
**Creating Relationships between Date Tables and Other Tables**

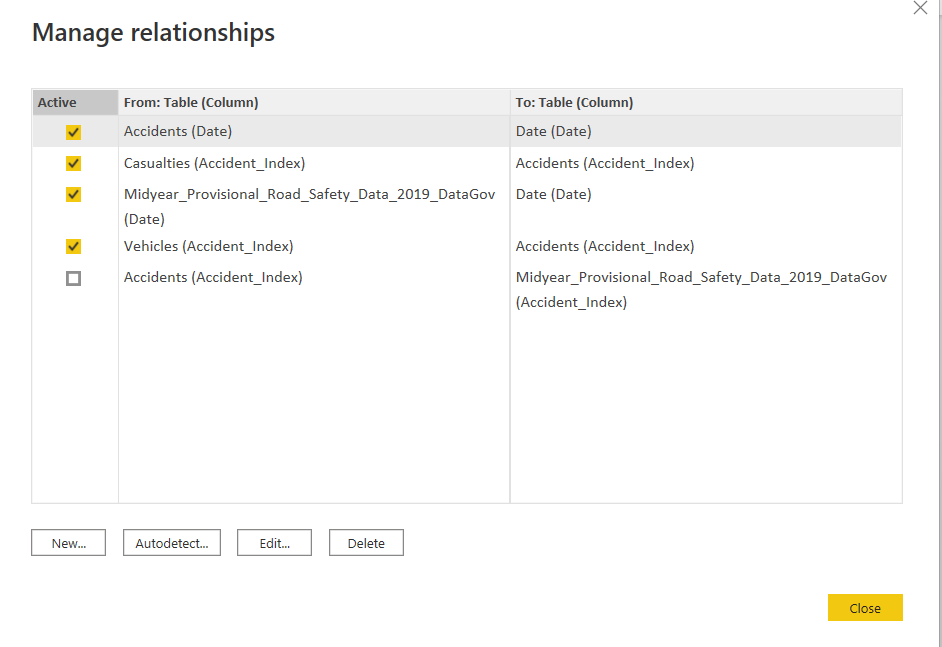
Power BI relationships provide a clear understanding of how tables are linked to one another. It demonstrates how two or more tables are typically linked by a column, and then joins the columns from the separate tables. These relationships are formed either automatically by Power BI when the data is loaded, or manually. When you enter your data into Power BI, the Autodetect feature will help you in establishing relationships between columns with similar names.

To manually create relationships between the date table and other tables, navigate to the model tab, where the data model is placed, and then drag the relevant column from one table and drop it into the corresponding column from the other table. In the following example, the date field from the Accidents table was dragged and dropped onto the date field from the date table.



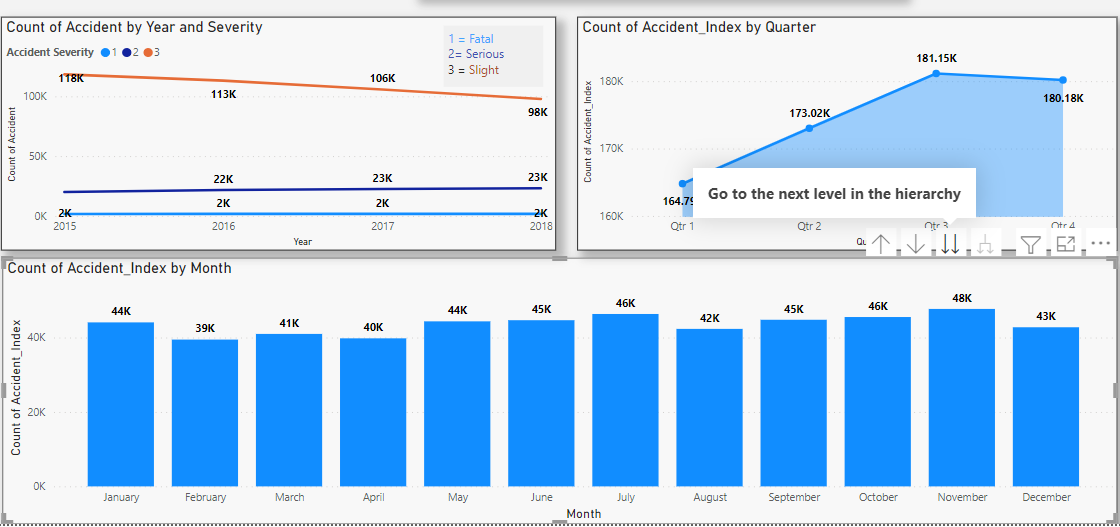
Another option for creating the relationship is to go to the **Manage Relationship** ribbon in the model tab's relationship’s view. In this view, you can create, update, and delete relationships between tables, as well as autodetect existing relationships.





Creating relationships with the date table propagates filters to several tables, allowing you to display accurate information in your report between the tables connected.

A few illustrations of how filters are propagated from date tables to other tables are given below. Each of these instances can be drilled down to the next level in the hierarchy (i.e. you can drill down from year to month to quarter to day, etc):



**Pros and Cons of DAX vs Power Query**

There are not many differences between the DAX and Power Query approaches to creating a date table in Power BI. The following points are the main distinctions that can be made:

* The DAX approach is slightly simpler to use because you do not need to open Power Query Editor.
* Power Query allows for reusability since the date table can be built in the Power BI dataflow, which allows you to create, store, and manage tables in the Power BI service.

**Conclusion**

When creating a data table with Power BI, you have more than one option, as demonstrated in this tutorial. However, considerations such as usability, simplicity, reusability, and your requirements may influence your choice.