

How to Filter Dates Between Two Specific Dates in Power BI

Authored by
mohammed looti

January 11, 2026

RECOMMENDED CITATION

mohammed looti (2026). *How to Filter Dates Between Two Specific Dates in Power BI*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=125470>

Introduction: Mastering Date Range Filtering in Power BI

The ability to filter data based on specific time boundaries is fundamental to effective data analysis. In short, the answer is a resounding yes: dates can be filtered accurately between two specific dates within Power BI. This capability allows users to narrow down vast datasets to focused intervals, thereby facilitating the identification of temporal trends, seasonal fluctuations, and performance comparisons over precise periods. Whether you are using the visual filter pane for quick adjustments or leveraging the advanced capabilities of DAX (Data Analysis Expressions) for complex derived tables, Power BI provides robust tools to handle time-based segmentation with precision. Understanding how to implement these techniques is crucial for anyone performing serious business intelligence tasks.

Date filtering typically involves defining a clear starting date and a clear ending date. When these parameters are applied, the underlying dataset is instantaneously pruned, showing only those records whose date field falls inclusively within the selected range. For instance, an analyst might want to review sales figures exclusively between the beginning of the last fiscal quarter and the end of the current month. This process moves beyond simple single-date selection and requires a function or interface that handles the 'between' condition effectively. While visual filters offer an accessible entry point, the use of DAX formulas, specifically those designed for time intelligence, provides unmatched control and flexibility when creating permanent or calculated data structures based on filtered time segments.

Dual Approaches to Date Filtering: Visual Tools vs. DAX

In Power BI, users typically encounter two primary methodologies for date range filtering. The first method involves using the built-in visual filter pane or slicers within a report. When a date column is used on a visual, the filter pane provides options such as "Relative Date," "Start Date/End Date," or "Advanced Filtering," allowing users to select parameters interactively without writing code. This approach is highly effective for ad-hoc exploration and dynamic user interaction within published reports. However, these visual filters are temporary and only affect the visuals they are applied to; they do not persist as a new, foundational table structure within the data model itself.

The second, more powerful method involves utilizing DAX. DAX allows the creation of new calculated tables or measures where the filtering logic is hard-coded into the model definition. This is particularly useful when the filtered dataset needs to serve as the basis for multiple subsequent calculations, relationships, or visualizations, ensuring consistent results across the entire report. By defining a new filtered table using functions like CALCULATETABLE combined with time intelligence functions like DATESBETWEEN, the analyst establishes a robust, fixed snapshot of data corresponding exactly to the desired time frame. This structural permanence is a key advantage of the DAX approach over transient visual filtering.

Choosing between these methods depends entirely on the objective. If the goal is rapid exploration or providing end-users with interactive control over a report's timeframe, visual filters suffice. If the requirement is to establish a permanent, derived dataset essential for complex metrics or baseline comparisons--such as creating a static "Pre-COVID" sales table--then the DAX method is the professional standard. The following sections focus on the superior control offered by using DAX functions for this precise task.

Implementing DAX for Precise Date Range Selection

To create a filtered table specifically for a defined date range, we leverage a combination of powerful DAX functions. The primary goal is to evaluate the existing table context and apply a filter condition based on the date column. The function that allows us to return an entire table, filtered by a specific expression, is **CALCULATETABLE**. Within this function, we introduce the **DATESBETWEEN** function, which is specifically designed to handle inclusive date range specification efficiently. This combination ensures that the output is a clean, new table containing only the rows meeting the temporal criteria.

The basic structure of this DAX command is highly readable and declarative, making the intent clear even for those new to the syntax. It instructs Power BI to calculate a new table based on an existing one, imposing a time filter defined by the column reference and the explicit start and end dates. This method bypasses the need for complex logical operators (like combining `Date >= StartDate` AND `Date <= EndDate`), simplifying the code and reducing potential errors associated with date formatting or boundary conditions.

The syntax below illustrates the precise structure required in the formula bar when creating a new calculated table. Note the use of the **DATE** function within the parameters of **DATESBETWEEN** to ensure that the start and end boundaries are correctly recognized as datetime objects, preventing ambiguity in the calculation engine.

You can use the following syntax in DAX to filter a table for rows where a date column is between two specific dates:

```
filtered_data =  
CALCULATETABLE (  
  'my_data',  
  DATESBETWEEN ('my_data', DATE(2022, 5, 1), DATE(2023, 8, 20))  
)
```

This particular example creates a new table named **filtered_data** that contains only the rows from the table named **my_data** where the date in the **Date** column is between **5/1/2022** and **8/20/2023**.

The result is a persistent table object within the Power BI data model.

Detailed Breakdown of the DAX Components

Understanding the function of each component within the provided DAX formula is essential for effective customization and troubleshooting. The outer function, **CALCULATETABLE**, is crucial because it takes the original table, in this case, 'my_data', and applies a filter context defined by the second argument. Unlike **CALCULATE**, which returns a scalar value (a measure), **CALCULATETABLE** returns a full table object, which is exactly what is needed when creating a filtered subset of rows. If this function were omitted, the filtering logic would not yield the desired table result.

The core filtering mechanism is handled by the **DATESBETWEEN** function. This is a powerful time intelligence function that requires three key arguments: the date column reference, the start date, and the end date. It returns a table containing a single column of dates that fall within or are equal to the specified start and end boundaries. When nested within **CALCULATETABLE**, this returned table acts as a filter on the date column of the primary table, effectively limiting the rows to those that match the dates produced by **DATESBETWEEN**.

The dates themselves are constructed using the standard **DATE** function: `DATE(Year, Month, Day)`. Using the **DATE** function explicitly guarantees that the expression passes a proper date scalar value to **DATESBETWEEN**, ensuring the boundaries are interpreted correctly by the Power BI engine, regardless of regional settings or underlying data type complexities. In the example `DATE(2022, 5, 1)` and `DATE(2023, 8, 20)`, we are explicitly setting the start and end points of our analysis, respectively. To adapt this formula for any other period, an analyst simply needs to modify these three numerical parameters (Year, Month, Day) for both the start and end points.

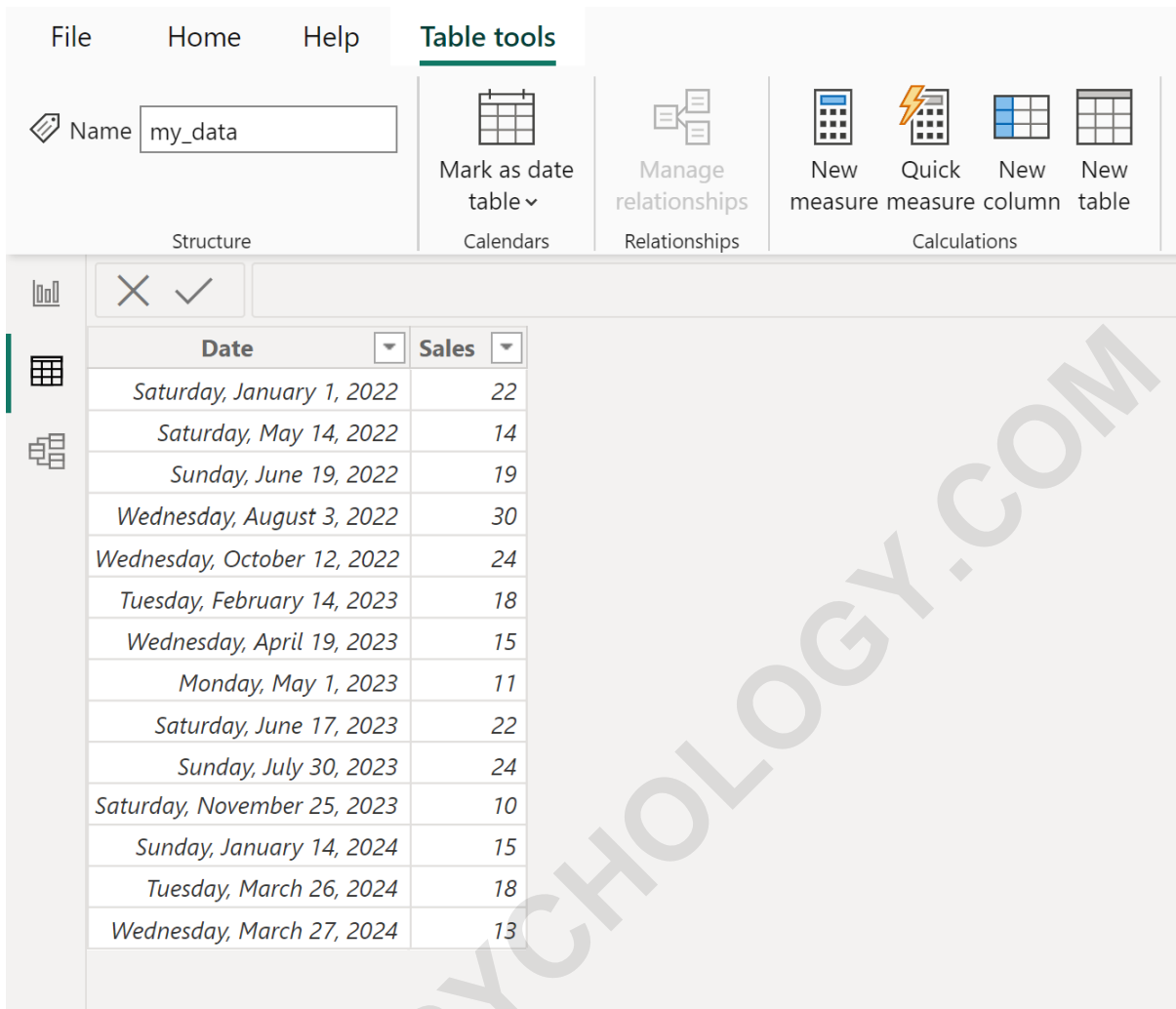
Practical Example: Step-by-Step Filtering Implementation

The following example demonstrates the process of applying this advanced DAX filtering technique within Power BI Desktop. Suppose we have a standard sales table named **my_data** containing transactional records, and our objective is to isolate all sales data spanning from May 1, 2022, through August 20, 2023, for targeted analysis. This exercise involves navigating the data modeling environment to define a new calculated object based on the filtering criteria.

The following example shows how to use this syntax in practice.

Example: How to Filter Between Two Dates in Power BI Using DAX

Suppose we have the following table in Power BI named **my_data** that contains information about sales made on various dates at some company:

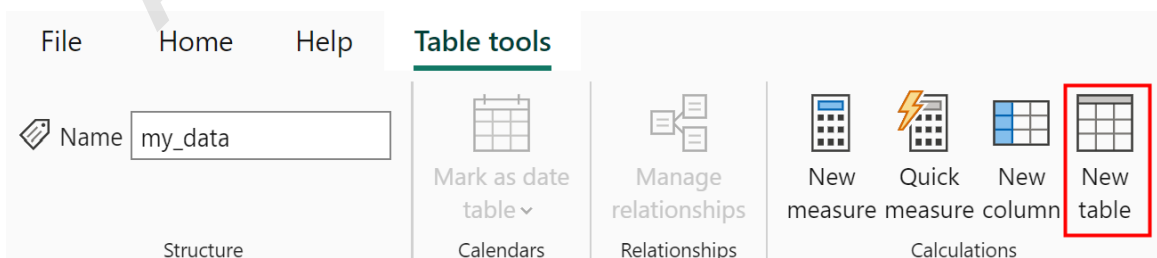


The screenshot shows the Power BI Desktop interface with the 'Table tools' ribbon selected. The ribbon includes options for 'Mark as date table', 'Manage relationships', and 'Calculations' (New measure, Quick measure, New column, New table). Below the ribbon, a table is displayed with the following data:

Date	Sales
Saturday, January 1, 2022	22
Saturday, May 14, 2022	14
Sunday, June 19, 2022	19
Wednesday, August 3, 2022	30
Wednesday, October 12, 2022	24
Tuesday, February 14, 2023	18
Wednesday, April 19, 2023	15
Monday, May 1, 2023	11
Saturday, June 17, 2023	22
Sunday, July 30, 2023	24
Saturday, November 25, 2023	10
Sunday, January 14, 2024	15
Tuesday, March 26, 2024	18
Wednesday, March 27, 2024	13

Suppose we would like to filter the table to only show the rows where the date in the **Date** column is between **5/1/2022** and **8/20/2023**.

To do so, click the **Table tools** tab in the modeling view of Power BI Desktop and then click the **New table** icon:

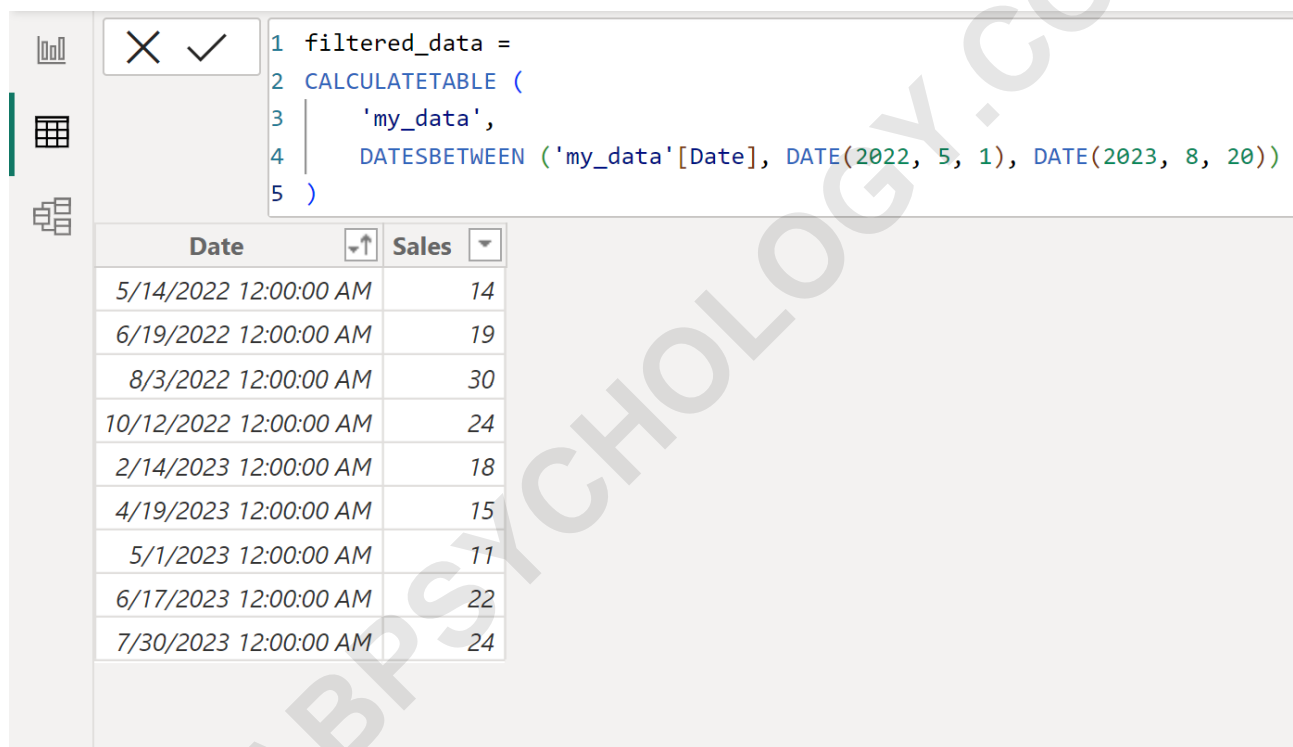


The screenshot shows the 'Table tools' ribbon in Power BI Desktop. The 'New table' icon, located in the 'Calculations' group, is highlighted with a red box.

Upon clicking **New table**, a formula bar will appear. Then type the following formula into the formula bar to define the calculated table:

```
filtered_data =  
CALCULATETABLE (  
    'my_data',  
    DATESBETWEEN ('my_data', DATE(2022, 5, 1), DATE(2023, 8, 20))  
)
```

This will successfully create a new table named **filtered_data** that contains only the rows from the table named **my_data** where the date in the **Date** column is between **5/1/2022** and **8/20/2023**. This new table is now ready to be used in reports and measures for focused data analysis:



The screenshot shows the Power BI interface. On the left, there are icons for a chart, a table, and a grid. The main area displays a DAX formula in a text editor with a 'Close' (X) and 'Check' (✓) button. The formula is: `1 filtered_data =
2 CALCULATETABLE (
3 'my_data',
4 DATESBETWEEN ('my_data'[Date], DATE(2022, 5, 1), DATE(2023, 8, 20))
5)`. Below the formula, a table is displayed with two columns: 'Date' and 'Sales'. The 'Date' column contains dates from 5/14/2022 to 7/30/2023, and the 'Sales' column contains corresponding sales values.

Date	Sales
5/14/2022 12:00:00 AM	14
6/19/2022 12:00:00 AM	19
8/3/2022 12:00:00 AM	30
10/12/2022 12:00:00 AM	24
2/14/2023 12:00:00 AM	18
4/19/2023 12:00:00 AM	15
5/1/2023 12:00:00 AM	11
6/17/2023 12:00:00 AM	22
7/30/2023 12:00:00 AM	24

Customizing the Date Range and Function Flexibility

The true power of using the **DATESBETWEEN** function lies in its flexibility and ease of modification. To filter using any two different dates, an analyst simply needs to change the start and end date parameters within the function call. For example, if the requirement shifted to analyzing data from the entire year of 2023, the formula would be modified to: `DATESBETWEEN ('my_data', DATE(2023, 1, 1), DATE(2023, 12, 31))`. This adaptability makes **DAX** an extremely efficient tool for repetitive or scenario-based time-series filtering tasks.

Furthermore, while **CALCULATETABLE** is used here to create a persistent new table, the filtering logic provided by **DATESBETWEEN** can also be utilized within measures using the **CALCULATE**

function. When used in a measure, **DATESBETWEEN** shifts the filter context of the measure calculation to only include the specified date range, allowing for the calculation of metrics like "Total Sales within Q3 2023" without needing to create a physical new table. This versatility demonstrates why mastering DAX time intelligence functions is a cornerstone of advanced Power BI development.

It is important to acknowledge that **DATESBETWEEN** is inclusive, meaning it includes the starting date and the ending date in the filtered output. If an exclusive range (e.g., records after the start date but before the end date) is required, analysts would need to rely on conditional logic within the **FILTER** function, combined with comparison operators (> and <). However, for most standard business intelligence tasks requiring a bounded period, the inclusive nature of **DATESBETWEEN** is perfectly suited and highly optimized for performance.

Alternative DAX Time Intelligence Functions

While **DATESBETWEEN** is excellent for fixed, custom date ranges, DAX provides several other time intelligence functions that achieve similar filtering goals but based on different criteria. For instance, the **DATESINPERIOD** function allows filtering by a specific number of intervals (e.g., 6 months or 90 days) starting from a particular date, which is ideal for rolling analyses. Similarly, functions like **DATESYTD** (Year-To-Date) or **DATESMTD** (Month-To-Date) automatically calculate time ranges relative to the current context, simplifying the process of standard financial reporting.

Analysts should select the appropriate function based on whether the time period is static and arbitrary (use **DATESBETWEEN**), relative to a current calculation context (use MTD/QTD/YTD), or defined by a length of time starting from a reference point (use **DATESINPERIOD**). All these functions work seamlessly within the filter argument of CALCULATETABLE or **CALCULATE**, providing a comprehensive toolkit for temporal data manipulation in Power BI.

For detailed official documentation and a complete list of parameters, analysts are encouraged to consult the Microsoft documentation. Understanding these nuances ensures that the correct function is applied to the specific time-based query, maximizing both accuracy and calculation efficiency within the data model.

Conclusion and Further Resources

The ability to accurately filter data between two distinct dates is a core requirement for insightful data analysis in Power BI. While visual filters offer interactivity, the DAX approach using **CALCULATETABLE** and DATESBETWEEN provides a robust, persistent solution for creating filtered subsets of data directly within the model. This calculated table method is particularly valuable for establishing fixed baselines, comparative periods, and simplifying subsequent calculations that rely on a specific historical window.

By mastering the syntax demonstrated in this guide, particularly the integration of **CALCULATE** and the parameter structure of **DATESBETWEEN**, Power BI professionals gain a critical skill necessary for advanced data modeling. Always ensure that the date column referenced in the formula is correctly marked as a date/time data type in the model for these time intelligence functions to operate correctly.

Note: You can find the complete documentation for the [DATESBETWEEN](#) function in [DAX](#) by visiting the official Microsoft site.

This detailed approach to date filtering is key to transforming raw data into actionable business intelligence. Whether segmenting sales cycles, tracking project timelines, or monitoring system performance during critical windows, defining precise temporal boundaries allows for highly focused and accurate reporting, solidifying [Power BI](#)'s role as a leading analytical tool.