

# How to Extract the Date from a Datetime Value in Power BI

Authored by  
**stats writer**

January 29, 2026

## RECOMMENDED CITATION

stats writer (2026). *How to Extract the Date from a Datetime Value in Power BI*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=128434>

Extracting the date component from a comprehensive datetime value is a foundational step in data preparation within Power BI. This essential procedure involves isolating the specific calendar date, disregarding the time stamp, to facilitate targeted reporting and analytical processes. Power BI offers several powerful built-in functions, notably within the DAX language, such as the FORMAT function, to achieve this goal efficiently. By successfully extracting the date, users gain the ability to perform accurate, date-based analyses--including period-over-period comparisons, trend identification, and aggregated sales summaries--making this technique indispensable for robust business intelligence and reporting.

The ability to manipulate time data types is crucial because transactional databases often store timestamps with millisecond precision, which is rarely needed for high-level summaries. Converting this complex data type into a simpler date format ensures uniformity and significantly improves the ease of creating relationships and measures in the DAX environment. This detailed guide outlines the precise method using the FORMAT function to ensure clean and valid date extraction.

## Mastering Date Extraction from Datetime in Power BI

### Understanding Datetime Values and Analytical Needs

**In data warehousing and analytical modeling, data is frequently captured as a datetime data type, which encapsulates both calendar date information and time of day (hours, minutes, seconds). While this level of granularity is excellent for auditing and detailed logging, it often hinders high-level data analysis. For instance, if you are calculating daily sales totals, the presence of the time component means that two transactions occurring on the same calendar day but at different times are treated as unique values, complicating aggregation. Therefore, the essential first step is to normalize the data by stripping away the time**

element.

This normalization process is critical for establishing consistent reporting axes. Without pure date values, attempts to connect tables using date keys or build standard time intelligence functions (like Year-to-Date or Moving Averages) will fail or produce incorrect results. By extracting just the date, we create a clean attribute suitable for linking to a separate Date Dimension table, which is a best practice in dimensional modeling within **Power BI**.

Introducing the DAX Approach: The FORMAT Function

The most reliable and versatile method for extracting the date component in **Power BI** is through a calculated column utilizing the powerful Data Analysis Expressions (**DAX**) language. Specifically, the `FORMAT` function is leveraged here, as it allows users to convert any data type into a string representation based on a specified format code. When applied to a datetime column, we can specify a format that only retains the date elements, effectively discarding the time.

You can employ the following specific syntax in a new

calculated column in Power BI to perform this date extraction:

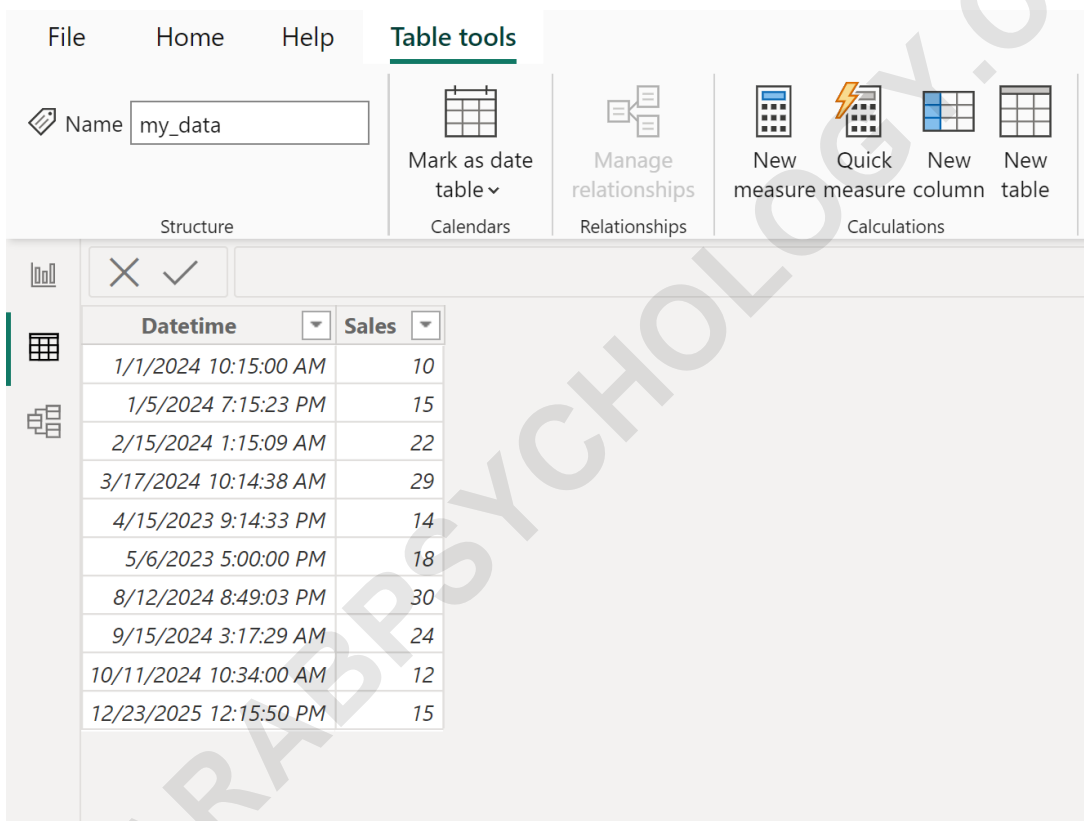
```
date = FORMAT('my_data', "M/D/YYYY")
```

This particular example creates a newly calculated column named `date`. This column systematically extracts the date portion from the source column named `date`, which resides within the data table named `'my_data'`. It is important to remember that while the result looks like a date, the `FORMAT` function returns a text string, which can sometimes require further conversion if complex date calculations are needed afterward, though for basic display and filtering, it works perfectly.

#### Step-by-Step Example Setup in Power BI

To demonstrate this process practically, consider a standard scenario where a business analyst is presented with raw transactional data. Suppose we have a table loaded into our Power BI data model, typically named `my_data`, containing comprehensive records of total sales along with the precise datetime stamp of each transaction:

This table structure is common in operational systems, where the `Datetime` column serves as the primary time reference. Our goal is to derive a separate, clean date column to analyze daily sales performance, isolating us from the noise introduced by the hour, minute, and second values.



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

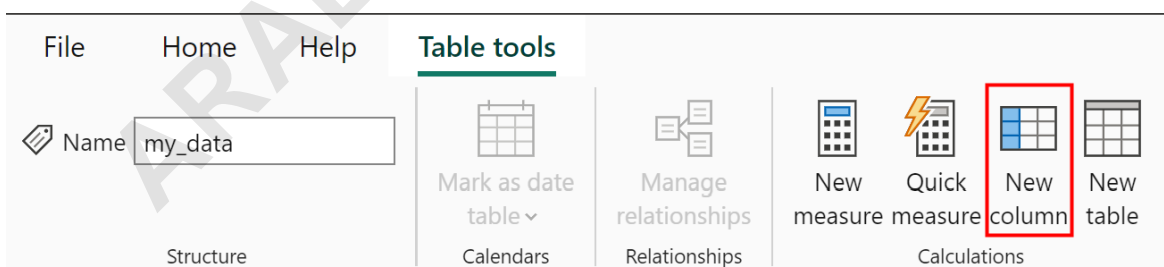
Datetime	Sales
1/1/2024 10:15:00 AM	10
1/5/2024 7:15:23 PM	15
2/15/2024 1:15:09 AM	22
3/17/2024 10:14:38 AM	29
4/15/2023 9:14:33 PM	14
5/6/2023 5:00:00 PM	18
8/12/2024 8:49:03 PM	30
9/15/2024 3:17:29 AM	24
10/11/2024 10:34:00 AM	12
12/23/2025 12:15:50 PM	15

The subsequent steps detail how to modify this table to include the extracted date field, ensuring that the necessary foundation for accurate data analysis is correctly established within the model environment.

## Creating the New Column and Applying DAX

The process of adding a calculated column begins in the Data view or Report view within the **Power BI Desktop** application. To start the creation of our derived date column, you must navigate to the appropriate menu interface. First, ensure you have the target table selected, and then proceed to initiate the column creation process.

To execute this, click on the Table tools tab located in the ribbon menu at the top of the interface. Once this tab is active, locate and click the specific icon designated for creating a New column. This action opens the **DAX** formula bar, allowing you to input the necessary expression:



Once the formula bar is visible and ready for input, meticulously type or paste the previously defined **DAX** expression. This formula instructs **Power BI** to calculate

the new column's values based on the formatting logic provided:

```
date = FORMAT('my_data', "M/D/YYYY")
```

Upon committing this formula (by pressing Enter or clicking the checkmark), **Power BI** instantly evaluates the expression row by row, generating a new column labeled `date`. This new column will contain only the date component derived from the original `Datetime` column, resulting in a cleaner, text-based date field that is optimized for aggregated reporting and visualizations.

#### Analyzing the FORMAT Function Syntax

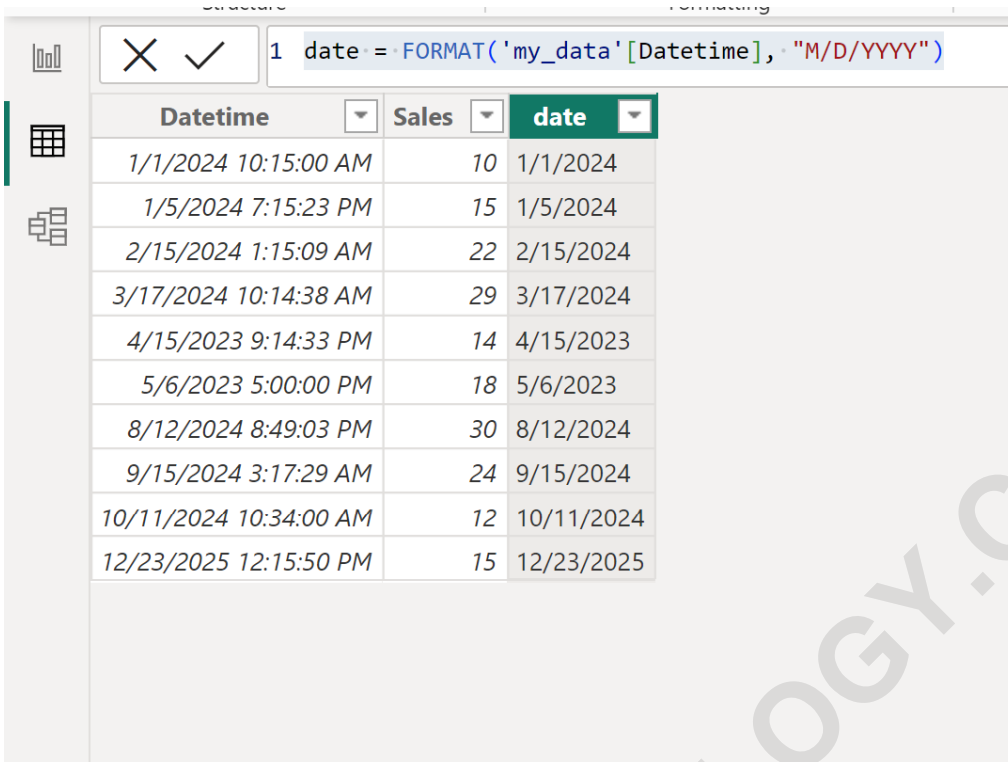
A deeper examination of the `FORMAT` function syntax reveals why it is so effective for this specific task. The **FORMAT function** generally requires two arguments: the value to be formatted, and the string specifying the desired format. In our example, `'my_data'` serves as the input value, which is a datetime value.

The crucial element is the format string: `"M/D/YYYY"`. This format code explicitly instructs the function to return only the month, day, and four-digit year, effectively

truncating the time element (hours, minutes, seconds) that is inherent in the source column. Other useful format strings might include "DD-MM-YYYY" or "YYYYMMDD", depending on regional standards or required sorting formats. Choosing the right format string is paramount for ensuring the resulting date string is interpretable and usable by end-users and other analytical processes.

#### Interpreting the Results of the Date Extraction

Executing the **DAX** formula successfully results in the creation of the `date` column appended to the `my_data` table. This visual confirmation is essential, demonstrating that the time element has been successfully removed across all rows, providing a unified date field for analysis:



The screenshot shows the Power BI interface with a DAX formula bar and a data table. The formula bar contains the following formula:

```
1 date = FORMAT('my_data'[Datetime], "M/D/YYYY")
```

The table below shows the result of this formula, with columns for Datetime, Sales, and date.

Datetime	Sales	date
1/1/2024 10:15:00 AM	10	1/1/2024
1/5/2024 7:15:23 PM	15	1/5/2024
2/15/2024 1:15:09 AM	22	2/15/2024
3/17/2024 10:14:38 AM	29	3/17/2024
4/15/2023 9:14:33 PM	14	4/15/2023
5/6/2023 5:00:00 PM	18	5/6/2023
8/12/2024 8:49:03 PM	30	8/12/2024
9/15/2024 3:17:29 AM	24	9/15/2024
10/11/2024 10:34:00 AM	12	10/11/2024
12/23/2025 12:15:50 PM	15	12/23/2025

The output clearly illustrates the transformation applied to each transactional record. For every complex datetime value in the source column, the newly created column contains only the corresponding date. This clean result set allows for accurate aggregation and reporting, fulfilling the primary requirement for simplified data analysis.

For instance, observe the detailed transformation results:

The formula successfully extracts `1/1/2024` from the original timestamp `1/1/2024 10:15:00 AM`.

Similarly, it isolates `1/5/2024` from the entry `1/5/2024 7:15:23 PM`. Furthermore, the formula correctly pulls `2/15/2024` from `2/15/2024 1:15:09 AM`.

This systematic removal of the time component ensures that all daily activities are grouped under the correct calendar date, regardless of when they occurred during that 24-hour period. The resulting column is now ideal for filtering, visualization, and joining operations with other data tables.

#### Common Pitfalls and Best Practices

While the `FORMAT` function provides a straightforward solution, users must be aware that the output is a text string, not a true date data type. Text dates can cause issues if you later attempt complex time intelligence calculations (which require a date type) or if sorting is not handled correctly (as text sorts alphabetically, e.g., '10/1' comes before '2/1'). If complex calculations are required, the best practice is to use the `DATE` or `INT` function in **DAX** alongside `TRUNC` or `DATEVALUE`, or use Power Query (M Language) transformations (such as `Date.From`) for superior data type handling at the source level.

However, for analysts primarily focused on reporting and filtering based on the extracted date, the `FORMAT` method remains a quick and highly readable solution. Always verify the data type of the new column after creation and ensure that the format string used (e.g., "M/D/YYYY") aligns with the regional settings or reporting requirements for your organizational standards.

These tutorials offer guidance on performing related tasks and advanced data preparation techniques in **Power BI**: