How to Display Dates as Month and Year in Power BI

Authored by

stats writer

January 29, 2026

ARABPSYCHOLOGY.COM

RECOMMENDED CITATION

stats writer (2026). *How to Display Dates as Month and Year in Power BI*. PSYCHOLOGICAL SCALES. Retrieved from https://scales.arabpsychology.com/?p=128531

Effectively manipulating temporal data is a fundamental requirement for impactful data analysis in Microsoft Power BI. When visualizing trends or grouping metrics, converting a raw date field into a clean month and year format is essential. The most efficient method for achieving this conversion involves utilizing the powerful DAX (Data Analysis Expressions) language, specifically the FORMAT function.

By applying the **FORMAT** function and specifying a custom format code, such as "MMM-YYYY", you can instantly transform complex date stamps into a highly readable structure like "Jan-2020". This standardized display is crucial for improving the readability of charts and tables, enabling users to perform time-series analysis based on calendar periods rather than specific days. This guide will demonstrate various techniques using **DAX** to control the output format precisely, ensuring your reports are both accurate and aesthetically pleasing for advanced reporting in Power BI.

Core Methodology: Using the FORMAT Function in DAX

The foundation of date manipulation in Power BI lies in the **DAX** language. To achieve a custom month and year format, we create a new calculated column utilizing the **FORMAT** function. This function takes two mandatory arguments: the value (your date column) and the format string (the specific pattern you wish to apply). Mastery of these format strings allows for granular control over the final output, tailoring the data presentation exactly to your reporting needs.

Below, we outline four primary formula variations, each designed to meet a different reporting standard. These examples assume your date column is named and resides within a table named `'my_data'`. Understanding these syntax differences is key to successful data modeling and DAX implementation.

Formula 1: Numeric Month and Full Four-Digit Year (e.g., 01/2022)

This format is standard for many financial and operational reports, providing a clear, numerical representation. The `MM` code ensures the month is always displayed with a leading zero (01 through 12), guaranteeing consistency in sorting, while `YYYY` ensures the full century is represented.

month_year = FORMAT('my_data', "MM/YYYY")

Formula 2: Numeric Month and Two-Digit Year (e.g., 01/22)

For reports where space is limited or context implies the century, using the two-digit year format (`YY`) can simplify the visual presentation. This is commonly seen in summary tables or dashboards where users are already familiar with the reporting period. Note that `MM/YY` is concise but should be

ARABPSYCHOLOGY.COM

used cautiously to avoid ambiguity regarding the century in long-term data analysis.

month_year = FORMAT('my_data', "MM/YY")

Formula 3: Abbreviated Month Name and Full Year (e.g., Jan. 2022)

Using abbreviated month names (`MMM`) offers a balance between clarity and conciseness. This is often preferred for presentation layers, as text labels are inherently easier for humans to read than numerical month codes. We use `YYYY` here to ensure the full year is explicitly stated, minimizing potential misinterpretations when applying the FORMAT function.

month_year = FORMAT('my_data', "MMM. YYYY")

Formula 4: Full Month Name and Full Year (e.g., January 2022)

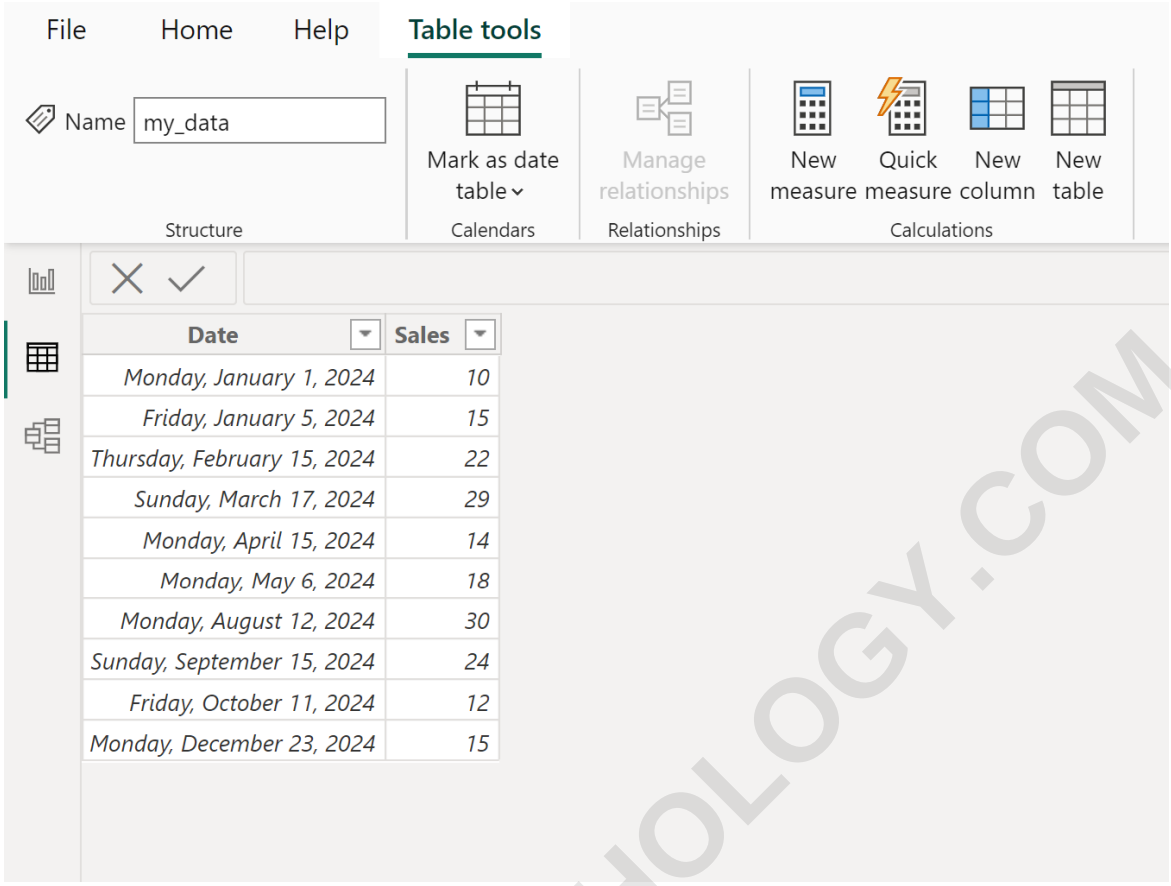
For formal documentation or detailed reports requiring maximum clarity, the full month name (`MMMM`) provides the most descriptive output. Although this format consumes more horizontal space in tables and visualizations, it eliminates all possibility of date interpretation errors for the end user and provides a highly accessible date/time format.

month_year = FORMAT('my_data', "MMMM YYYY")

To illustrate the practical application of these **DAX** formulas, we will use a sample dataset containing a standard column. The following image displays the initial state of the data table within the Power BI environment. Our goal is to derive new, time-based columns from this source data using the techniques outlined above.

It is important to remember that when using the **FORMAT** function, you are creating a text field, not a true date field. While this is perfect for display and sorting purposes (provided you include the year), it may require specific formatting adjustments if you intend to perform complex chronological calculations later.

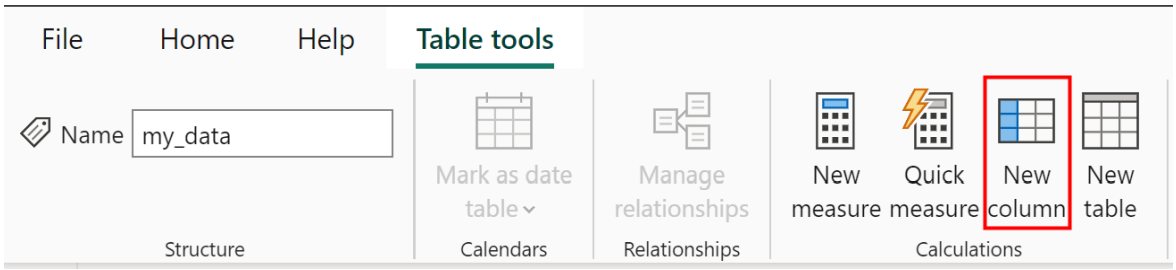
ARABPSYCHOLOGY.COM



Example 1: Numeric Month and Full Year Conversion (MM/YYYY)

This example demonstrates the creation of a new column that formats the date using the standard numerical month and four-digit year pattern. This pattern is exceptionally useful when the primary goal is strict chronological sorting, as the leading zeros in the month (`MM`) ensure that '01/2022' sorts correctly before '12/2022', which is critical for accurate time series representation in visualizations.

To begin the transformation process, navigate to the **Table tools** tab within the Power BI interface. Locate and click the **New column** icon. This action opens the DAX formula bar, preparing the environment for the calculation.

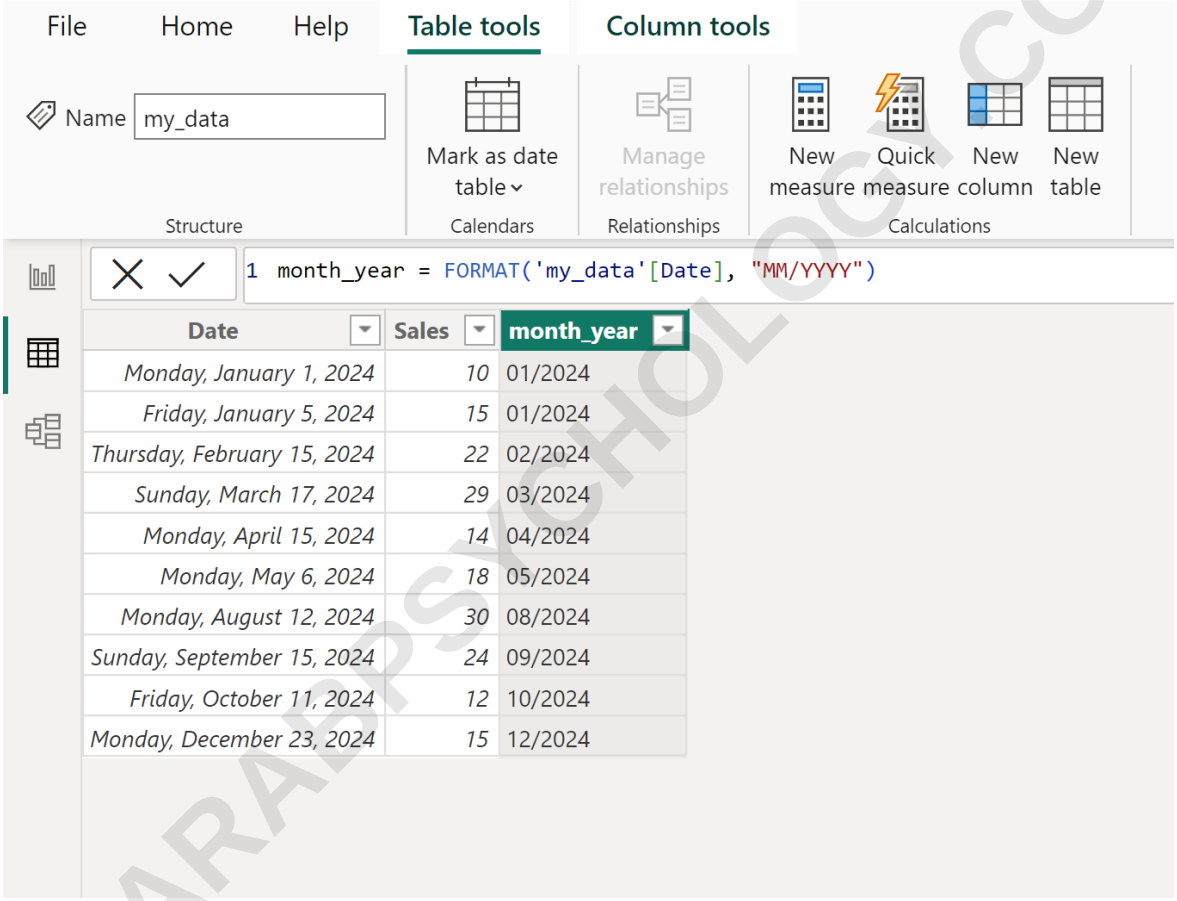


ARABPSYCHOLOGY.COM

Enter the following formula into the **DAX** formula bar. Ensure that the table and column names (`'my_data'`) accurately reflect your data model:

month_year = FORMAT('my_data', "MM/YYYY")

Upon committing the formula, Power BI generates a new column labeled `month_year`. This column displays the original date values transformed into the **MM/YYYY** format, as illustrated in the resulting table view below. This method provides an immediate, clear categorization of data by month-year grouping, enabling straightforward aggregation.



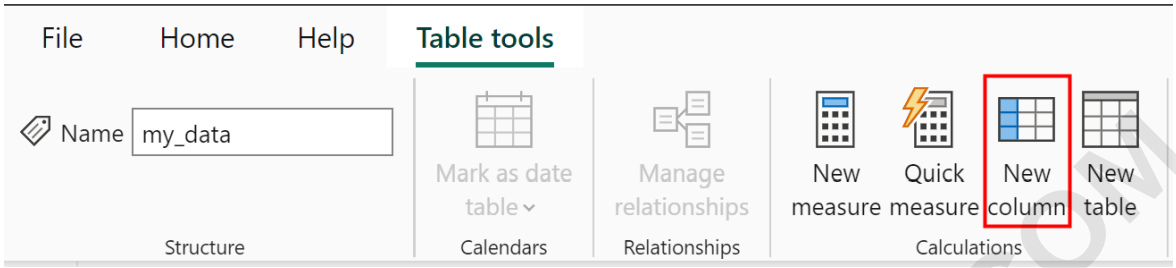
Date	Sales	month_year
Monday, January 1, 2024	10	01/2024
Friday, January 5, 2024	15	01/2024
Thursday, February 15, 2024	22	02/2024
Sunday, March 17, 2024	29	03/2024
Monday, April 15, 2024	14	04/2024
Monday, May 6, 2024	18	05/2024
Monday, August 12, 2024	30	08/2024
Sunday, September 15, 2024	24	09/2024
Friday, October 11, 2024	12	10/2024
Monday, December 23, 2024	15	12/2024

ARABPSYCHOLOGY.COM

Example 2: Concise Format with Two-Digit Year (MM/YY)

When generating reports intended for internal audiences or those focusing strictly on recent data, the concise format using only the last two digits of the year (`YY`) can significantly reduce visual clutter. While sacrificing some detail, this approach maintains the chronological structure needed for effective sorting and grouping, assuming the century context is already understood by the report consumers.

As with the previous example, start by selecting the **Table tools** tab and initiating the creation of a **New column**. The process of adding a calculated field remains consistent across all DAX conversions, emphasizing the importance of utilizing the calculated column functionality for derived temporal attributes.

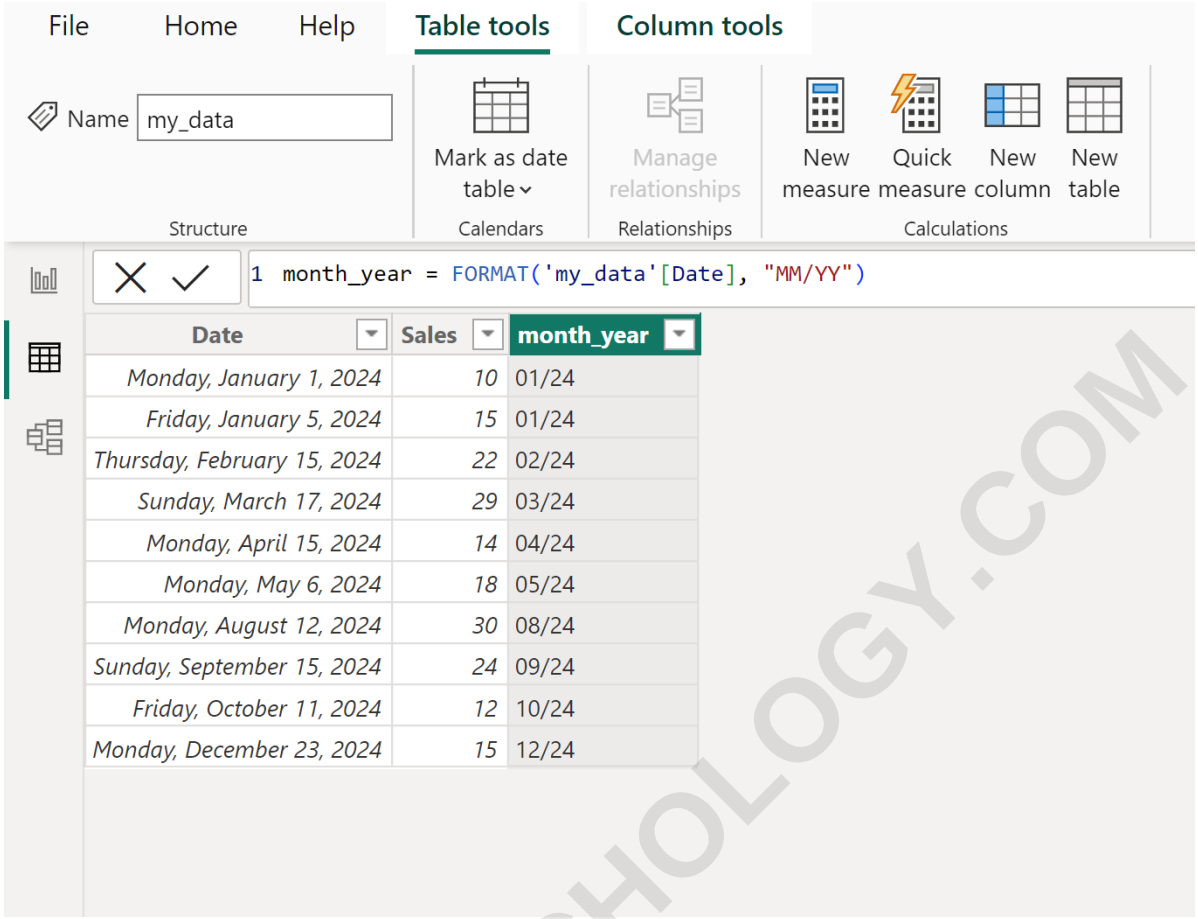


Input the following modified formula into the calculation bar. The key differentiator here is the use of `YY` instead of `YYYY` in the format string argument of the FORMAT function:

month_year = FORMAT('my_data', "MM/YY")

The resultant `month_year` column will display the dates in the tighter **MM/YY** structure. This is especially useful for visualizations like sparklines or small multiples where space optimization is critical to the overall dashboard design without compromising the sequential order of time periods.

ARABPSYCHOLOGY.COM

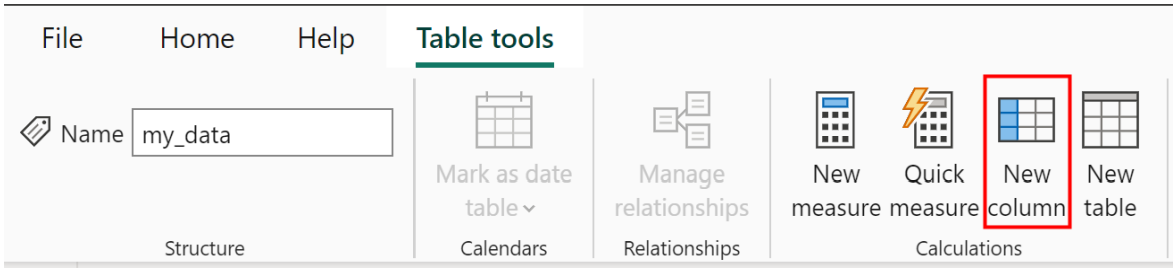


Example 3: Abbreviated Month Name (MMM. YYYY)

Using text representations for months often enhances user experience and report intuitiveness. The abbreviated format, denoted by `MMM`, translates the numeric month into a three-letter code (e.g., Jan, Feb, Mar). This format is highly popular in visual reports because it is immediately recognizable and uses less space than the full month name, making it ideal for axis labels and category summaries.

To implement this textual conversion, repeat the standard procedure: navigate to **Table tools** and click **New column** to open the calculation editor. This systematic approach ensures that all date derivations are managed consistently within the data model, improving maintainability.

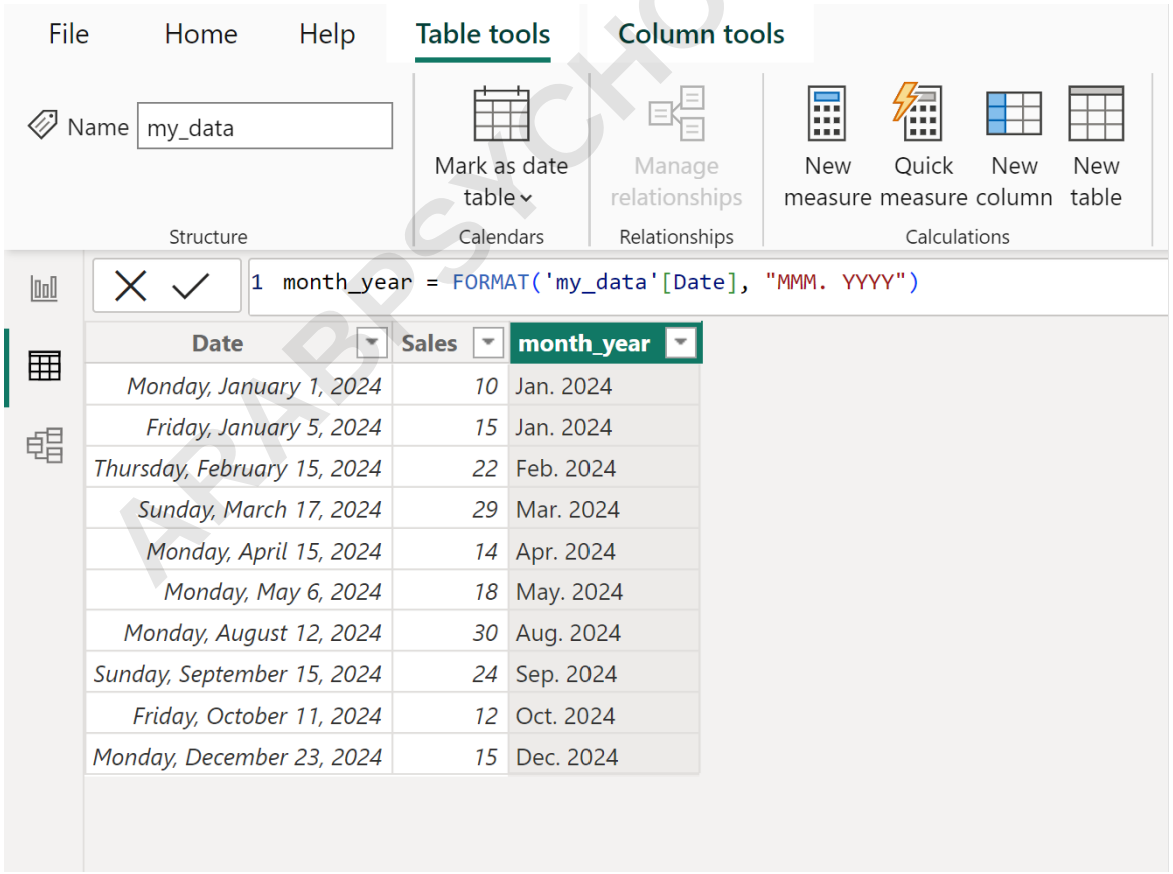
ARABPSYCHOLOGY.COM



Apply the format string `"MMM. YYYY"` within the **DAX** formula. Note the inclusion of the period after the month abbreviation, which is optional but often used for stylistic separation and improved visual parsing:

month_year = FORMAT('my_data', "MMM. YYYY")

The newly calculated column, `month_year`, now displays the dates using the abbreviated month format coupled with the four-digit year. This provides robust readability without sacrificing the necessary detail for multi-year comparisons in complex data analysis reports.



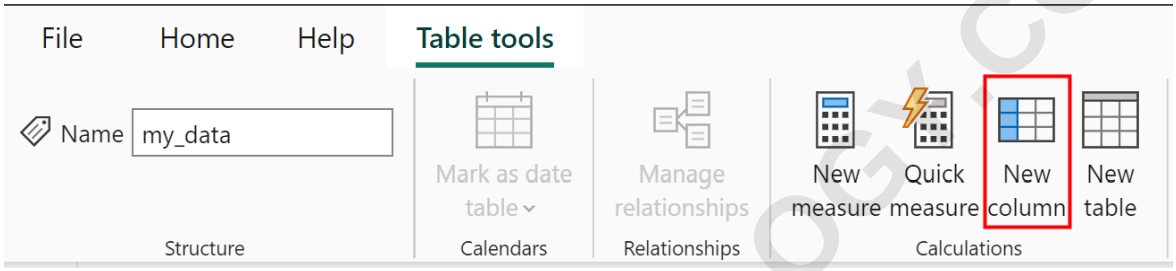
Date	Sales	month_year
Monday, January 1, 2024	10	Jan. 2024
Friday, January 5, 2024	15	Jan. 2024
Thursday, February 15, 2024	22	Feb. 2024
Sunday, March 17, 2024	29	Mar. 2024
Monday, April 15, 2024	14	Apr. 2024
Monday, May 6, 2024	18	May. 2024
Monday, August 12, 2024	30	Aug. 2024
Sunday, September 15, 2024	24	Sep. 2024
Friday, October 11, 2024	12	Oct. 2024
Monday, December 23, 2024	15	Dec. 2024

ARABPSYCHOLOGY.COM

Example 4: Full Month Name and Full Year (MMMM YYYY)

The fullest textual format, employing four 'M's (`MMMM`), outputs the complete, spelled-out month name (e.g., January, February). While this is the least space-efficient option, it offers the highest level of formality and clarity, making it ideal for executive reports or documents where the visual presentation must be highly descriptive and unambiguous.

As before, initiate the process by selecting the **New column** option under the **Table tools** menu. It is crucial to manage these calculated columns efficiently to maintain model performance and ensure smooth deployment of your DAX solutions.

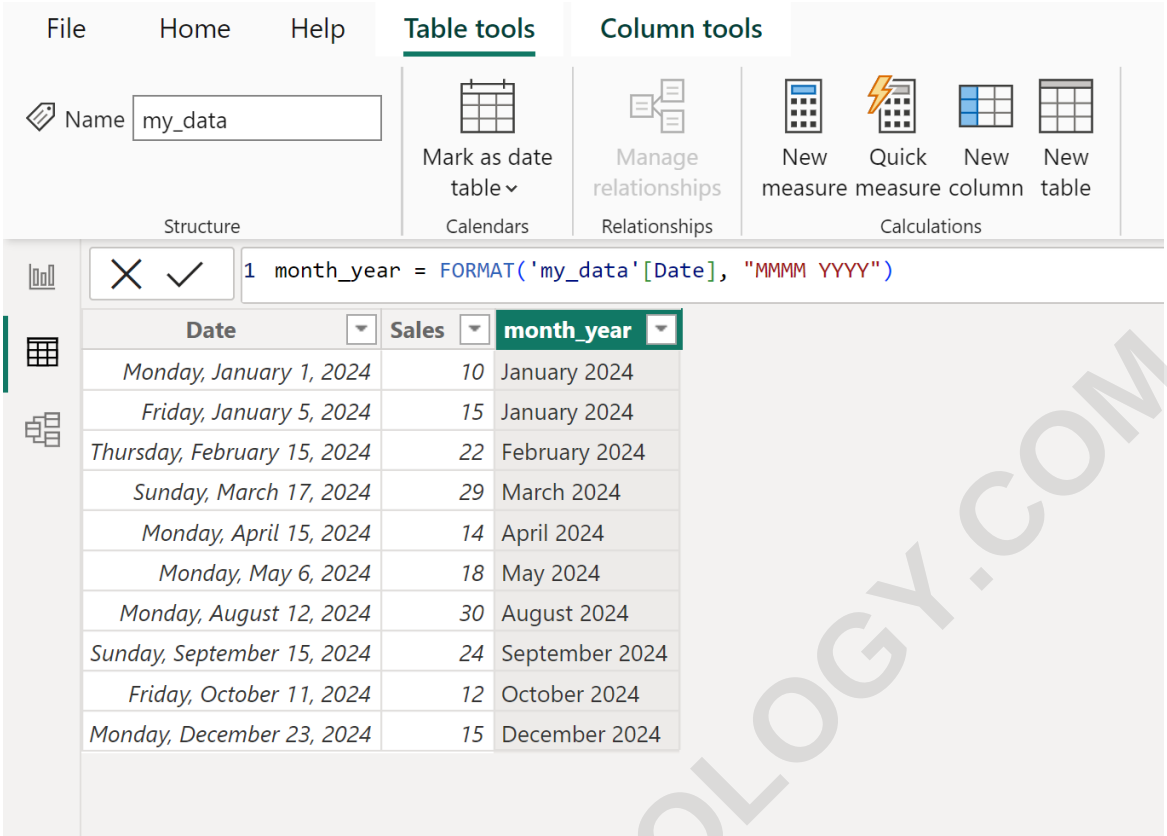


Input the formula using the `MMMM YYYY` format string. This instructs the FORMAT function to return the full month name and the complete four-digit year:

month_year = FORMAT('my_data', "MMMM YYYY")

The result is a highly descriptive `month_year` column, providing the most accessible date/time format for non-technical users. This format minimizes cognitive load and improves accessibility across different regions and languages, although localization settings in Power BI may influence the language of the month name output.

ARABPSYCHOLOGY.COM



ARABPSYCHOLOGY.COM

Summary of Format Codes and Best Practices

Choosing the correct format string for your **FORMAT** function is paramount to generating useful time dimensions. While `MM/YYYY` is excellent for sorting, `MMM YYYY` or `MMMM YYYY` provides superior visual clarity for reports. Always consider the audience and the required level of detail before finalizing your format string. Remember that the output of the **FORMAT** function is a text string, which can sometimes complicate advanced chronological filtering or date arithmetic, necessitating alternative data modeling techniques.

For complex data models, it is often best practice to create a dedicated Date table and manage the Month/Year attributes there. However, for quick, ad-hoc analysis or when needing a simple derived column, the **FORMAT** function remains the fastest and most accessible tool in the DAX toolkit for immediate display transformation.

Further Power BI and DAX Resources

For users looking to expand their knowledge of temporal data manipulation and advanced modeling techniques, the following resources are highly recommended. These tutorials provide comprehensive guidance on performing other common data transformation tasks essential for becoming proficient in data modeling:

Official Microsoft documentation provides the complete reference for the FORMAT function and its various customizable parameters, which are essential for precise format control.
Explore tutorials on creating calculated tables and columns, managing relationships, and time intelligence functions in DAX to unlock the full potential of your data model.
Learn how to optimize your data analysis performance by refining calculated measures versus calculated columns.

ARABPSYCHOLOGY.COM