Learning DAX: Extracting the Day of the Week from Dates in Power BI

Authored by

Mohammed looti

November 12, 2025

Harnessing Date Intelligence: Extracting Day of Week in Power BI

Analyzing temporal data is fundamental to modern business intelligence, providing deep insights into performance cycles, seasonal trends, and operational anomalies. Often, the specific day of the week--whether it is a peak sales Monday or a slow administrative Friday--holds critical keys to understanding these patterns. To effectively extract and leverage this crucial time dimension within Microsoft Power BI, practitioners rely heavily on DAX (Data Analysis Expressions). This comprehensive guide details the precise formulas and methodologies required to transform standard date fields into usable day-of-week formats, ensuring your data model is optimized for time-based analysis.

The inherent flexibility of DAX is essential, allowing data modelers to tailor the output based on specific analytical requirements. For instance, creating a highly readable visualization of weekly operational metrics typically demands the full descriptive day name (e.g., "Wednesday"), whereas implementing advanced conditional logic or ensuring correct chronological sorting requires a simple numeric index. Understanding these distinct output requirements is the first step toward effective time intelligence.

We will explore the three primary DAX formulas used for this common data transformation task: generating the full name, the standard abbreviation, and the numerical representation. Mastering these functions is a foundational skill in data modeling, enabling the creation of powerful time-based filters, calculated measures, and aggregated calculations that accurately reflect the rhythm and flow of the business week.

Core DAX Functions for Day of Week Extraction

The ability to manipulate dates in DAX primarily relies on two powerful, purpose-built functions: the FORMAT function and the WEEKDAY function. A thorough comprehension of their syntax, capabilities, and nuances is essential for advanced data modeling, particularly when dealing with international or regional date standards. The FORMAT function is exceptionally versatile, designed to allow developers to define custom output strings based on standard Date and time formatting specifiers, thereby controlling the presentation of the data attribute.

When utilizing the FORMAT function, we must pass two key arguments: the column containing the underlying date value, and a meticulously crafted format string. This format string acts as the instruction set, defining precisely how the date component should be presented. For extracting day names, we employ sequences of the 'd' specifier. The quantity of 'd' characters--specifically three ('ddd') or four ('dddd')--determines whether the output is an abbreviation or the full descriptive name. This approach is generally preferred when the desired output is a clear, descriptive text label for end-user reports.

In contrast, the WEEKDAY function is engineered specifically for numeric output. It requires the date column as its primary argument and an optional return type argument. This optional argument is critical as it dictates the starting day of the week for the numerical index (e.g., Return Type 1 sets Sunday = 1; Return Type 2 sets Monday = 1). Since we often focus on the default behavior (Sunday=1) for basic implementations, the return type is frequently omitted. This function is absolutely vital not for presentation, but for underlying calculation logic and, most importantly, for ensuring proper chronological sorting of textual day names within charts and tables.

To summarize the core techniques, the following standard formulas are used in DAX to derive the day of the week from any date field within your Power BI data model:

Formula 1: Get Day of Week as Full Name ("Sunday") - Ideal for descriptive labeling.

day_of_week = FORMAT('my_data', "dddd")

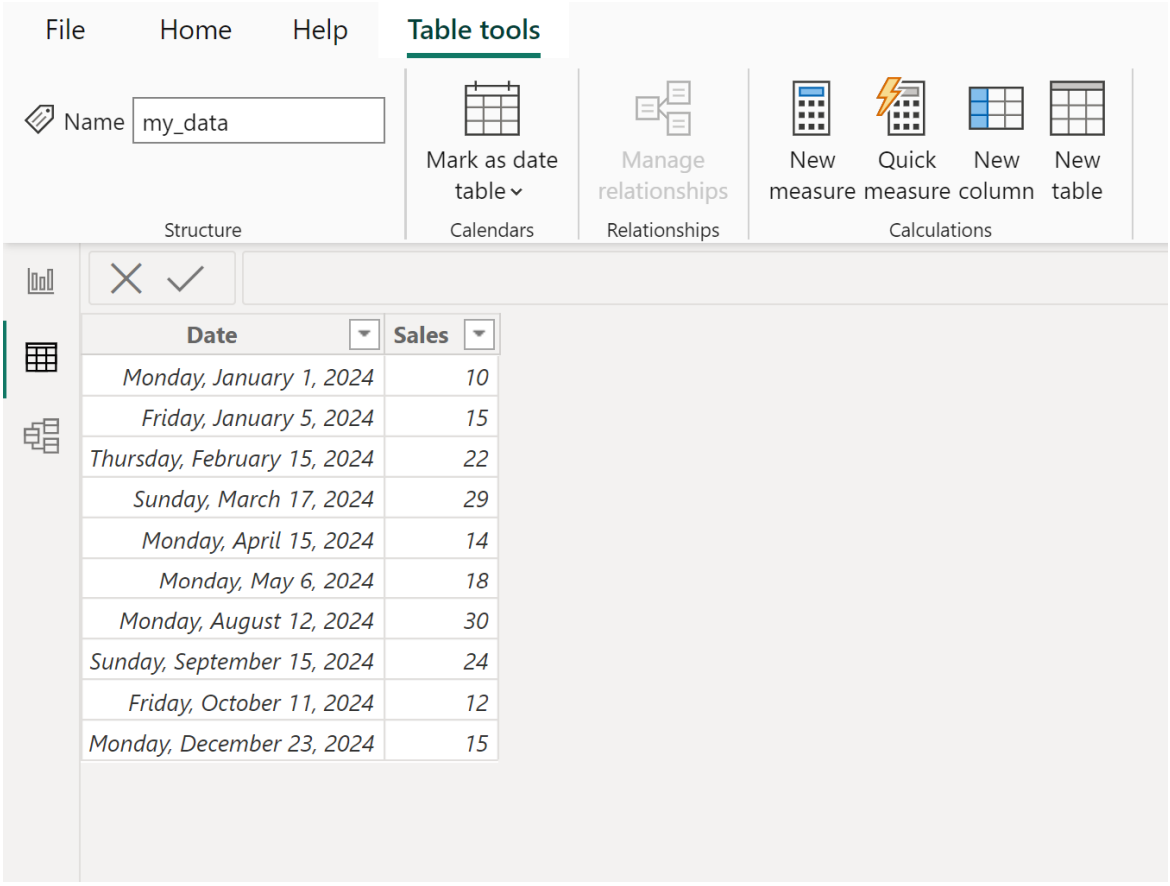
Formula 2: Get Day of Week as Abbreviated Name ("Sun") - Ideal for space-constrained visualizations.

day_of_week = FORMAT('my_data', "ddd")

Formula 3: Get Day of Week as Number (Sunday = 1 by default) - Essential for sorting and calculation logic.

day_of_week = WEEKDAY('my_data')

The following examples illustrate how to implement each method in practice, using the sample table below within the Power BI data environment:

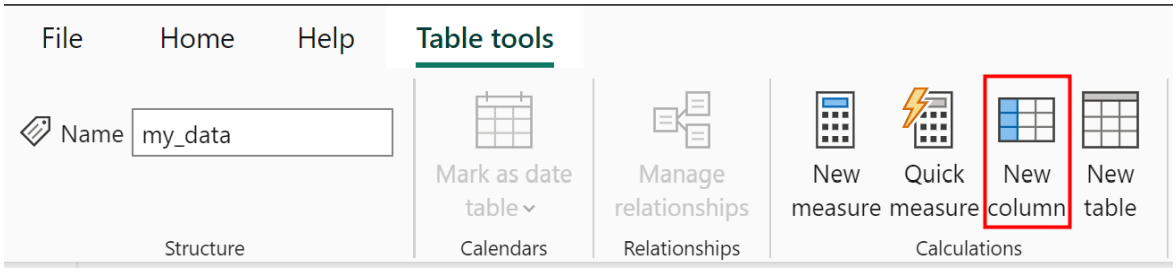


Date	Sales
Monday, January 1, 2024	10
Friday, January 5, 2024	15
Thursday, February 15, 2024	22
Sunday, March 17, 2024	29
Monday, April 15, 2024	14
Monday, May 6, 2024	18
Monday, August 12, 2024	30
Sunday, September 15, 2024	24
Friday, October 11, 2024	12
Monday, December 23, 2024	15

Example 1: Generating the Full Day Name

Our initial and perhaps most common objective is to establish a new calculated column that displays the unabbreviated, descriptive name of the weekday. This format provides the highest degree of clarity and is therefore the preferred choice for inclusion in primary reports, summary tables, and visualizations where maximizing readability and minimizing ambiguity are paramount design considerations. This process begins directly in the Data View or Report View within the Power BI Desktop environment, assuming the user possesses the necessary permissions to modify the underlying data model structure.

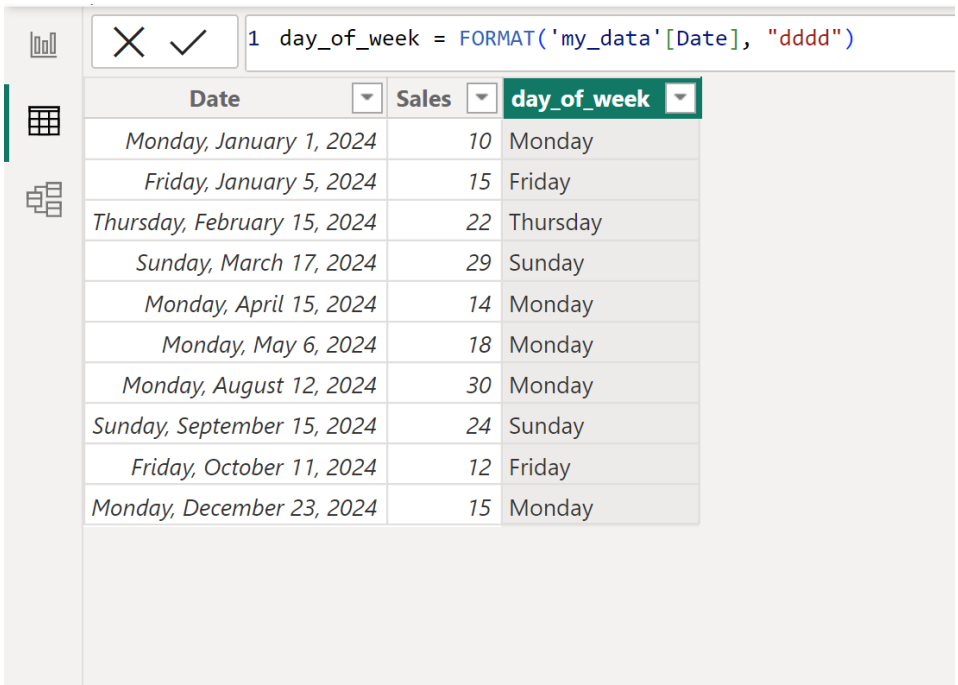
To initiate the creation of this new column, navigate to the **Table tools** tab located in the top ribbon of the interface. Locate and then click the **New column** icon. This action immediately populates the formula bar, enabling the input of your custom DAX expression. It is critically important at this stage to confirm that you have correctly selected the target table within your data model before proceeding, a necessary step, especially in complex datasets containing numerous related tables.



Once the formula bar is visible and ready for input, enter the following DAX expression. Pay close attention to the use of the 'dddd' string, which explicitly directs the FORMAT function to return the full, localized name of the day. This function respects the current regional and locale settings configured for the Power BI report, ensuring international compliance:

day_of_week = FORMAT('my_data', "dddd")

Execution of this expression will result in a new calculated column named **day_of_week**. This column will display the full textual day name corresponding to each date entry in the source **Date** column, as clearly illustrated in the image below. This full text output is highly valuable for creating user-friendly filters and slicing data based on specific days of operational activity, such as isolating weekend sales data.

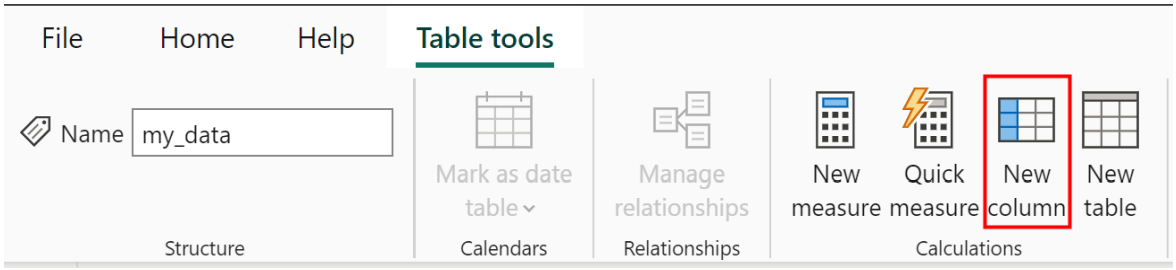


Date	Sales	day_of_week
Monday, January 1, 2024	10	Monday
Friday, January 5, 2024	15	Friday
Thursday, February 15, 2024	22	Thursday
Sunday, March 17, 2024	29	Sunday
Monday, April 15, 2024	14	Monday
Monday, May 6, 2024	18	Monday
Monday, August 12, 2024	30	Monday
Sunday, September 15, 2024	24	Sunday
Friday, October 11, 2024	12	Friday
Monday, December 23, 2024	15	Monday

Example 2: Retrieving the Abbreviated Day Name

While the full names of weekdays offer unmatched superior clarity and detail, abbreviated names prove indispensable in scenarios where visual space is a premium. Abbreviated day names are often the preferred format for inclusion in dense visualizations, smaller dashboards, or when integrating axis labels into detailed charts where optimizing space is critical. Utilizing the standard three-letter abbreviation conserves visual real estate without sacrificing the immediate recognition of the weekday, striking an excellent balance between detail and efficiency.

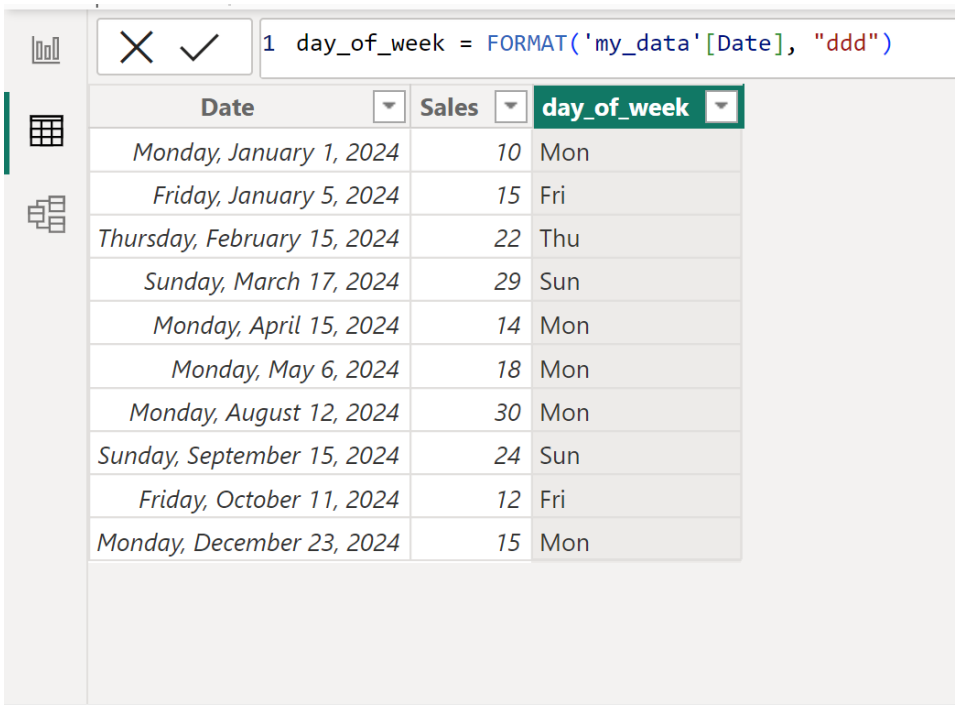
The procedural steps for generating the abbreviated name column are almost entirely identical to the previous example, focusing on the creation of a new column within the existing data table structure. The analyst must first ensure they utilize the **New column** tool within the **Table tools** ribbon to open the necessary DAX formula input environment:



The fundamental difference, defining the abbreviated output, lies solely in the format string provided as the second argument to the FORMAT function. Instead of specifying four 'd' characters ('dddd') for the full name, we specify only three 'd' characters ('ddd'). This instructs the function to return the standard localized abbreviation for the day:

day_of_week = FORMAT('my_data', "ddd")

Successful execution of this formula creates a new column named **day_of_week**. This column displays the day of the week as an abbreviated name for the corresponding date in the source **Date** column. This abbreviated output successfully maintains high readability while offering efficient spatial utilization for visual presentation elements.

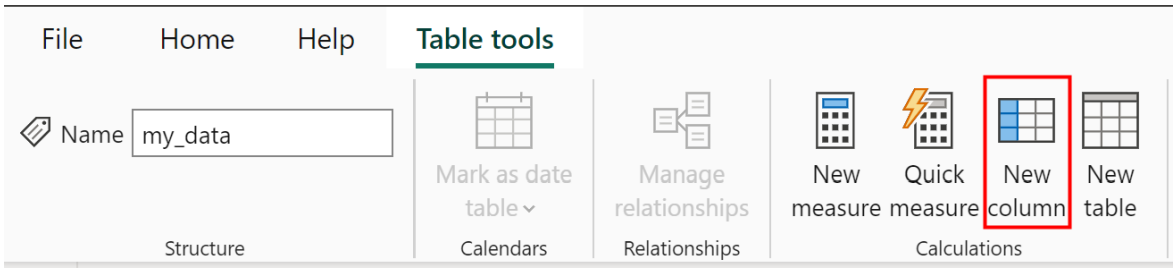


Date	Sales	day_of_week
Monday, January 1, 2024	10	Mon
Friday, January 5, 2024	15	Fri
Thursday, February 15, 2024	22	Thu
Sunday, March 17, 2024	29	Sun
Monday, April 15, 2024	14	Mon
Monday, May 6, 2024	18	Mon
Monday, August 12, 2024	30	Mon
Sunday, September 15, 2024	24	Sun
Friday, October 11, 2024	12	Fri
Monday, December 23, 2024	15	Mon

Example 3: Leveraging Numeric Indices for Chronological Sorting

The final and arguably most critical method involves extracting the day of the week as a numerical index, typically ranging from 1 to 7. This technique is absolutely essential for ensuring proper chronological sorting within reports and visualizations. If day names are sorted alphabetically (the default for text fields), 'Friday' would erroneously appear before 'Monday'. Using numeric indices overcomes this inherent flaw, guaranteeing that days are arranged in the correct, natural sequence. Furthermore, numerical indices significantly simplify the implementation of complex conditional logic and calculated columns based on specific weekday ranges.

To generate the day of the week as a numerical index ranging from 1 to 7, the process begins, as before, by navigating to the **Table tools** tab, and selecting the **New column** icon:



For this specific task, we bypass the FORMAT function entirely and utilize the dedicated WEEKDAY function. The syntax required for the WEEKDAY function is significantly less complex

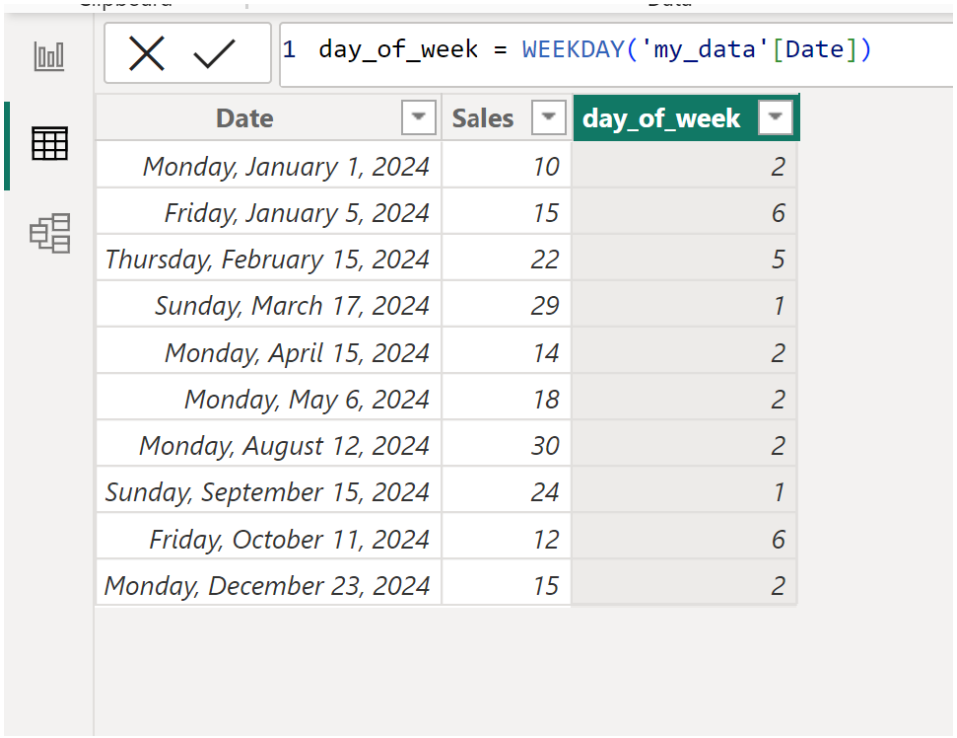
than that of the FORMAT function when aiming for a numerical output.

Subsequently, enter the following streamlined formula into the formula bar:

day_of_week = WEEKDAY('my_data')

It is vital to understand the function's default behavior: by omitting the optional second argument (the return type), the WEEKDAY function defaults to return type 1. In this standard configuration, Sunday is assigned the index 1, and the sequence concludes with Saturday assigned 7. If organizational standards or regulatory requirements dictate that Monday must be the start of the week (Monday = 1), you must explicitly append the return type '2' to the function call (e.g., `WEEKDAY('my_data', 2)`). Choosing the correct return type ensures calculated measures align with business calendars.

This formula successfully creates a new column named **day_of_week** that displays the day of the week as its numerical index for the corresponding date in the **Date** column:



Date	Sales	day_of_week
Monday, January 1, 2024	10	2
Friday, January 5, 2024	15	6
Thursday, February 15, 2024	22	5
Sunday, March 17, 2024	29	1
Monday, April 15, 2024	14	2
Monday, May 6, 2024	18	2
Monday, August 12, 2024	30	2
Sunday, September 15, 2024	24	1
Friday, October 11, 2024	12	6
Monday, December 23, 2024	15	2

Advanced Data Modeling: Sorting Text Columns Chronologically

While the three formulas discussed cover the fundamental requirements for day-of-week extraction, effective data modeling in Power BI demands an understanding of how to link these text and numeric outputs. A frequent challenge arises when a user creates a column for display using

the descriptive 'dddd' format--this column must still be sorted chronologically within visualizations. If left unsorted, it will default to alphabetical order, rendering charts and slicers useless for time-series analysis.

To resolve this, the numeric day index generated by the WEEKDAY function must be utilized as a separate, hidden sorting column. The process involves creating both the descriptive text column (using FORMAT) and the numeric index column (using WEEKDAY). Then, within the Data View, the analyst selects the descriptive text column and uses the "Sort by Column" feature in the Modeling tab, pointing it to the numerical index column. This technique ensures that the visually appealing text labels are always arranged in the correct chronological order, regardless of alphabetical placement.

Mastering date and time manipulation in Power BI is key to building robust, insightful, and usable reports. Regardless of whether the immediate need is for descriptive text labels using the FORMAT function or for reliable numeric indices necessary for sorting and conditional logic, DAX provides the precise tools for controlling temporal attributes. Always remember the critical importance of locale settings when generating text-based output to ensure maximum international compatibility and accurate presentation across different user locations.

Note: The complete and authoritative documentation for the **FORMAT** function, along with all associated formatting specifications in DAX, can be found in the official Microsoft documentation.

Additional Resources

The following tutorials explain how to perform other common date and time tasks in Power BI, further enhancing your overall data analysis capabilities and time intelligence modeling: