

How to Extract the Hour from a Date & Time in Power BI

Authored by
stats writer

January 29, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Extract the Hour from a Date & Time in Power BI*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=128440>

Analyzing temporal data effectively is foundational to robust Power BI reporting. One common requirement in data analysis is the ability to isolate specific components from a datetime field, such as the hour. This extraction is critical for grouping records, identifying peak activity times, and building accurate time-series visualizations. Fortunately, the DAX language provides a straightforward solution: the **HOUR** function.

The **HOUR** function is specifically designed to accept a datetime input and return the hour component as a whole number ranging from 0 (midnight) to 23 (11 PM). Utilizing this function allows analysts to transform granular timestamp data into manageable time intervals, thereby enabling powerful insights into hourly trends. Whether you are analyzing website traffic, sales transactions, or operational logs, extracting the hour is a necessary preliminary step for deep temporal segmentation. This article provides an expert guide on implementing this crucial function within your Power BI data model.

Power BI: Mastering the Extraction of Hour from Datetime

To perform this operation efficiently within the Power BI environment, we rely on **Data Analysis Expressions (DAX)**. DAX is the formula language used throughout Power BI, Analysis Services, and Power Pivot in Excel. When the goal is to create a new, calculated column based on existing time fields, the **HOUR** function is the most appropriate tool.

Understanding the DAX HOUR Function Syntax

The syntax for using the **HOUR** function is extremely clean and intuitive. It requires only one argument: a column containing datetime values. The function processes each row independently, extracting and returning the numerical hour value for that specific timestamp. This resulting integer is then populated into a new column, ready for use in filtering, grouping, or complex measure calculations.

You can use the following syntax in DAX to extract the hour from a datetime column in Power BI, typically applied via the 'New Column' feature:

```
hour_of_day = HOUR('my_data')
```

This specific formula creates a new column, which we have titled **hour_of_day**, that extracts only the hour component from the source column named **Datetime** within the table designated as **my_data**. Defining the table name before the column name ensures accuracy and prevents ambiguity, especially when dealing with complex data models containing numerous tables.

The Importance of Time Segmentation in Data Analysis

Why is isolating the hour so critical? Detailed time segmentation enables analysts to uncover hidden patterns that are often obscured when dealing with full timestamps. For instance, in retail analysis, knowing the hour of sale allows you to identify peak shopping periods, which can inform staffing schedules or targeted marketing campaigns. In server monitoring, analyzing traffic by hour helps pinpoint exactly when system load spikes occur.

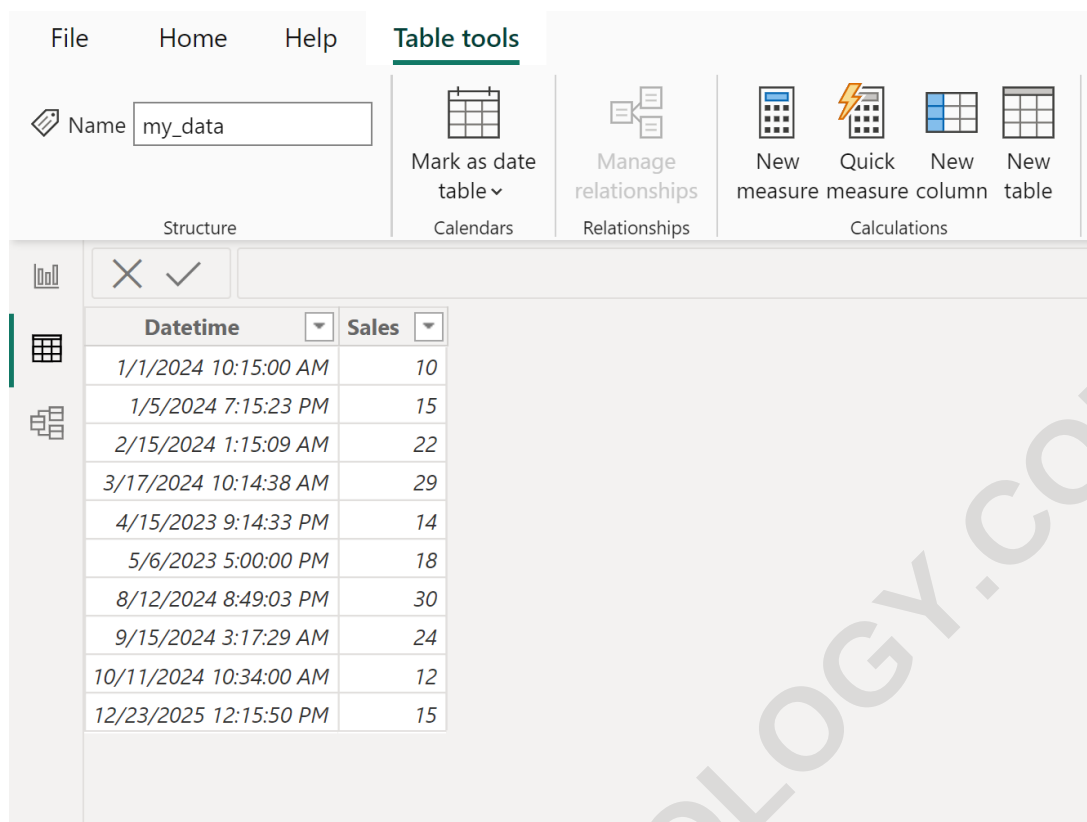
By transforming the continuous datetime field into a discrete hourly category, we facilitate better data aggregation. This categorization is essential for effective visualization techniques, such as creating bar charts or line graphs that display metrics (like total sales or error counts) across the 24 hours of the day. This level of granularity vastly improves the depth of your data analysis and reporting.

Example: Step-by-Step Guide to Extracting Hour in Power BI

To demonstrate the practical application of the **HOUR** function, we will walk through a common scenario. Suppose we have imported a dataset into Power BI, named **my_data**, which tracks total sales made by a company. Crucially, this table includes a detailed **Datetime** column containing the exact timestamp of each transaction. Our goal is to derive a new column that captures only the transaction hour.

Practical Example Setup: Initial Dataset Review

Consider the structure of our example table, **my_data**, which clearly illustrates the format of the mixed date and time values. Notice that the **Sales Amount** is tied directly to a specific timestamp in the **Datetime** column:



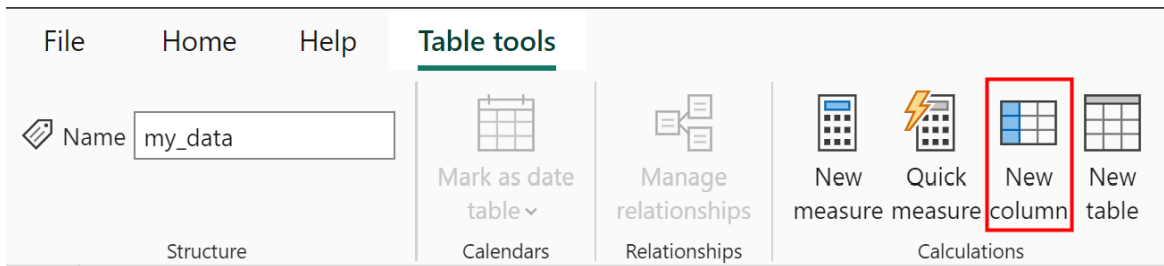
Datetime	Sales
1/1/2024 10:15:00 AM	10
1/5/2024 7:15:23 PM	15
2/15/2024 1:15:09 AM	22
3/17/2024 10:14:38 AM	29
4/15/2023 9:14:33 PM	14
5/6/2023 5:00:00 PM	18
8/12/2024 8:49:03 PM	30
9/15/2024 3:17:29 AM	24
10/11/2024 10:34:00 AM	12
12/23/2025 12:15:50 PM	15

As you can see, the **Datetime** column contains data ranging from morning times (AM) to evening times (PM). For analytical purposes, we need these times converted into the 24-hour clock format (0 to 23) and stored separately. This conversion ensures consistency across all visualizations and measures derived from this column.

Implementing the Calculation: Creating a New Column

The process for adding a calculated column using DAX is initiated through the Power BI user interface. This is typically done within the Data View or Report View, ensuring that the selected table (**my_data** in this case) is active.

To begin, click on the **Table tools** tab in the Power BI ribbon interface. Within this tab, locate and click the icon called **New column**. This action opens the formula bar, allowing you to input your DAX expression:



Once the formula bar is visible, enter the following precise formula. We rename the resulting column simply as **Hour** for clarity:

Hour = HOUR('my_data')

Reviewing the Output and Hourly Conversion

Upon executing the DAX formula, Power BI automatically processes every row in the **Datetime** column and generates the new calculated field. This new column, named **Hour**, contains only the integer representing the hour, standardized to the 24-hour clock format (military time).

This successfully extracts the hour portion of the datetime and creates a new column with these standardized numerical hour values. Observe how the new column integrates seamlessly with the existing data:

The screenshot shows the Power BI interface with the DAX formula bar at the top. The formula entered is `1 hour = HOUR('my_data'[Datetime])`. Below the formula bar, a table is displayed with three columns: **Datetime**, **Sales**, and **hour**. The **hour** column contains the extracted hour values for each row.

Datetime	Sales	hour
1/1/2024 10:15:00 AM	10	10
1/5/2024 7:15:23 PM	15	19
2/15/2024 1:15:09 AM	22	1
3/17/2024 10:14:38 AM	29	10
4/15/2023 9:14:33 PM	14	21
5/6/2023 5:00:00 PM	18	17
8/12/2024 8:49:03 PM	30	20
9/15/2024 3:17:29 AM	24	3
10/11/2024 10:34:00 AM	12	10
12/23/2025 12:15:50 PM	15	12

Demonstration of Time Conversion Logic

It is important to understand the conversion logic applied by the **HOUR** function, particularly how it handles AM and PM designations and converts them into the 0-23 range. The function strictly adheres to the 24-hour clock standard for consistent numerical representation:

The formula extracts **10** from 1/1/2024 10:15:00 AM.

The formula extracts **19** from 1/5/2024 7:15:23 PM.

The formula extracts **1** from 2/15/2024 1:15:09 AM.

If the timestamp were 12:30:00 PM (Noon), the result would be **12**.

If the timestamp were 12:30:00 AM (Midnight), the result would be **0**.

Further Applications of Hourly Data

Once the hour is isolated, its utility extends far beyond simple display. The **Hour** column becomes a powerful dimension for creating measures and defining time-based analysis. For example, you could create a measure to calculate the average sales during specific hours, or use it as a slicer to isolate activity during business hours versus after hours.

This hourly segmentation is often the first step in creating advanced time intelligence features within Power BI. By combining the **Hour** column with date components (extracted using the DAY, MONTH, or YEAR functions), you can build a comprehensive calendar table suitable for high-level reporting and accurate comparisons across different time periods. Utilizing the **HOUR** function is a fundamental skill for any professional engaged in serious temporal data analysis and reporting tasks.

For those seeking to expand their Power BI expertise, the following tutorials explain how to perform other common tasks related to date and time manipulation, aggregation, and visualization: