Rounding Time in Power BI with DAX: A Step-by-Step Guide to the Nearest Quarter Hour

Authored by

Mohammed looti

November 12, 2025

Why Standardize Time Data in Power BI?

When conducting extensive time-based analysis within Power BI, achieving absolute consistency across all data points is paramount for generating reliable insights. Raw timestamp data is inherently highly granular, often including seconds and milliseconds, which makes accurate aggregation and meaningful comparisons across different time periods exceptionally difficult. By proactively rounding timestamps to standardized, fixed intervals--such as the nearest quarter hour--analysts can establish cleaner, more manageable categories for reporting.

This vital step facilitates more accurate assessments of operational flow, identification of key shift patterns, or precise tracking of sales spikes. Such standardization represents a cornerstone of effective Data modeling and ensures that subsequent visualizations are built upon a foundation of structured data, leading to superior data quality and analytical precision.

To efficiently manage this essential data preparation requirement, the DAX (Data Analysis Expressions) language offers a robust suite of mathematical and time intelligence functions. The subsequent sections will meticulously detail the specific syntax and implementation steps required within Power BI to round any given datetime value precisely to the nearest 15-minute interval, thereby transforming unstructured time data into a readily analyzable format.

The Core DAX Syntax for Quarter-Hour Rounding

Achieving precise quarter-hour rounding relies on intelligently combining the **MROUND** and **TIME** functions available within the DAX language. This powerful analytical technique ensures that irrespective of the underlying seconds or minutes in the original timestamp, the resultant time value flawlessly aligns with one of the four standard quarter-hour marks: :00, :15, :30, or :45. This systematic approach is critical for maintaining data integrity across large datasets.

The specific formula detailed below is the most efficient expression for this operation. It is typically implemented when defining a new calculated column directly within your data model in Power BI Desktop:

Rounded Time = MROUND('my_data', TIME(0, 15, 0)) + TIME(0, 0, 0)

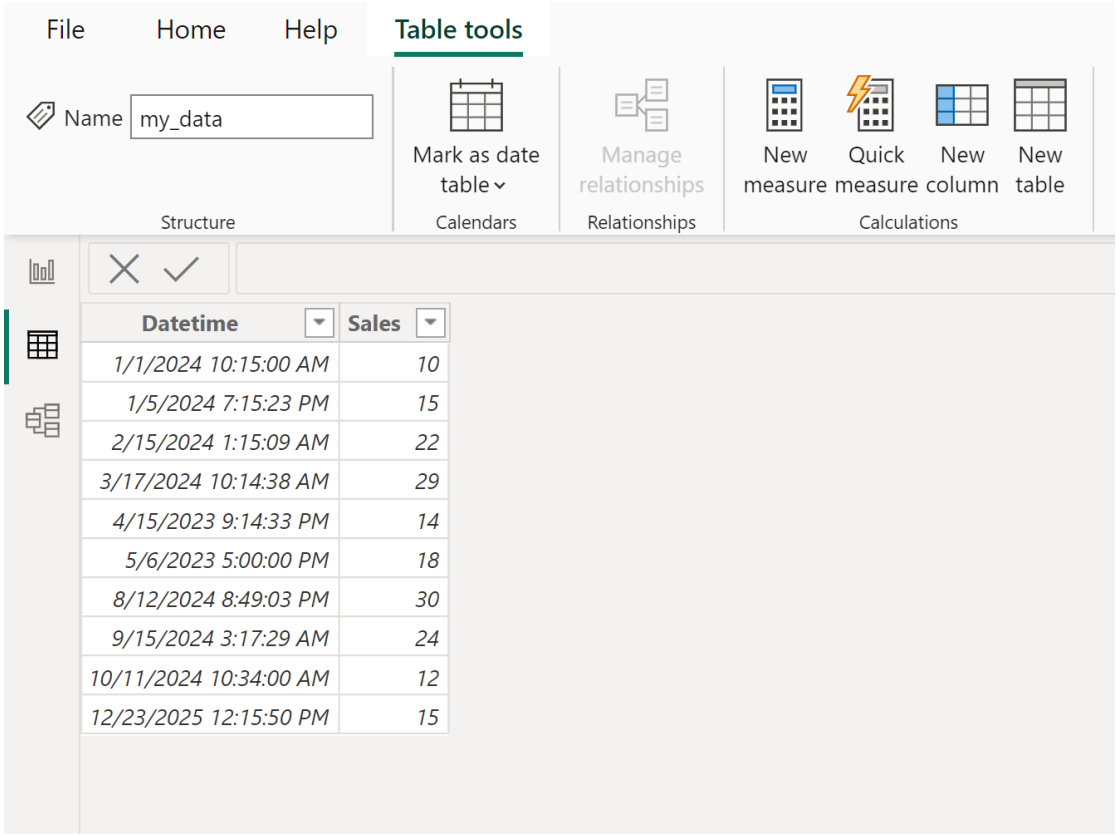
This expression generates a new column named **Rounded Time**. It processes the numerical values from the **Datetime** column found within the specified table (here, **my_data**), rounding them efficiently to the desired 15-minute multiple. For instance, if the original highly granular timestamp was **8:49:03 PM**, this formula would correctly apply the rounding logic and yield the standardized time of **8:45:00 PM**, aligning it to the closest preceding quarter-hour mark. If the time were 8:53:00 PM, conversely, it would round up to 9:00:00 PM. The following comprehensive guide walks

through the practical application of this syntax step-by-step.

Step-by-Step Example: Implementing Quarter-Hour Rounding

To solidify this concept, let us work through a practical business scenario. Imagine we are analyzing operational data captured within a Power BI data table named **my_data**. This table holds detailed records, perhaps relating to service requests or manufacturing cycles, each associated with a highly specific, granular timestamp located in the **Datetime** column. Our primary goal is to standardize these timestamps, allowing us to accurately analyze service volume or cycle throughput aggregated into precise 15-minute blocks.

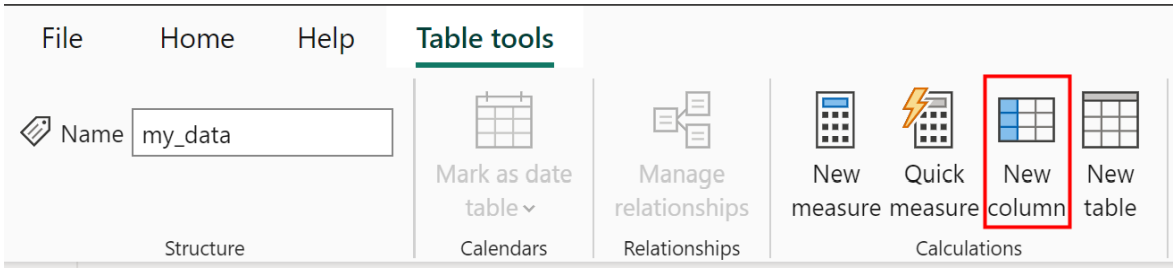
The initial state of the **my_data** table demonstrates the challenge inherent in the raw data--the high granularity of the timestamps makes direct comparison and reporting cumbersome:



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

Our objective is to apply the calculated rounding logic to every entry. To begin this transformation, we must navigate the Power BI interface to initiate the creation of a new calculated column, which is where the standardized time values will reside. Ensure you are operating within either the Data view or the Model view of Power BI Desktop. Locate and click the **Table tools** tab positioned prominently in the ribbon menu. Within this tab, select the icon labeled **New column**. This action immediately activates the formula bar, providing the necessary canvas to input our custom DAX

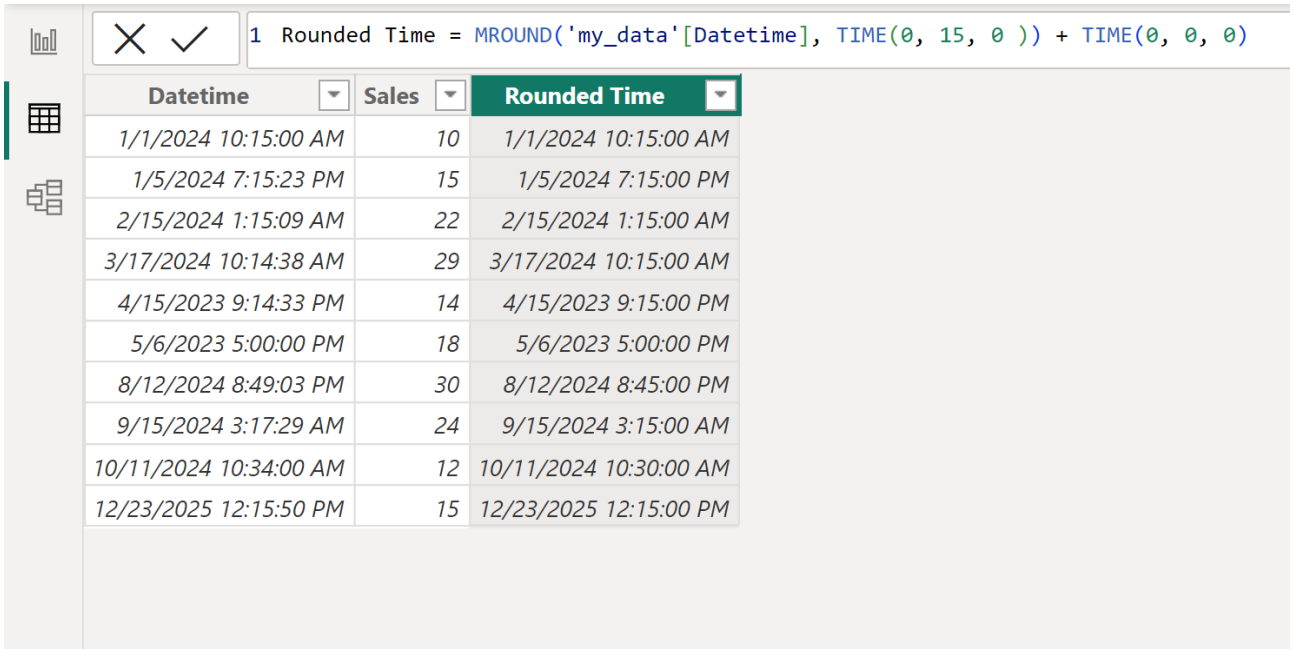
expression.



Once the formula bar is active and ready for input, copy and paste the full comprehensive formula below. It is critically important to verify that the table and column names used in the expression--in this case, `'my_data'`--perfectly match the actual names within your underlying data model:

Rounded Time = MROUND('my_data', TIME(0, 15, 0)) + TIME(0, 0, 0)

Upon execution, this formula instantly generates the new **Rounded Time** column. This column contains the fully standardized time values, accurately derived from the original **Datetime** column and rounded to the nearest quarter hour. This transformation significantly streamlines subsequent aggregations, time-series analysis, and reporting tasks, making the data model far more accessible for analytical purposes.



Datetime	Sales	Rounded Time
1/1/2024 10:15:00 AM	10	1/1/2024 10:15:00 AM
1/5/2024 7:15:23 PM	15	1/5/2024 7:15:00 PM
2/15/2024 1:15:09 AM	22	2/15/2024 1:15:00 AM
3/17/2024 10:14:38 AM	29	3/17/2024 10:15:00 AM
4/15/2023 9:14:33 PM	14	4/15/2023 9:15:00 PM
5/6/2023 5:00:00 PM	18	5/6/2023 5:00:00 PM
8/12/2024 8:49:03 PM	30	8/12/2024 8:45:00 PM
9/15/2024 3:17:29 AM	24	9/15/2024 3:15:00 AM
10/11/2024 10:34:00 AM	12	10/11/2024 10:30:00 AM
12/23/2025 12:15:50 PM	15	12/23/2025 12:15:00 PM

Deep Dive: Deconstructing the DAX Formula Components

A thorough understanding of the underlying mechanics is essential for adapting this solution to more complex data manipulation requirements. Let us revisit the complete expression used to achieve the precise quarter-hour rounding, which relies on treating time values mathematically:

Rounded Time = MROUND('my_data', TIME(0, 15, 0)) + TIME(0, 0, 0)

The efficacy of this calculation hinges primarily on the MROUND function. In the context of DAX, this function is designed to return a numerical value rounded to the specified multiple. Crucially, within the DAX engine, all datetime values are stored internally as simple numerical values, where the integer portion denotes the date and the decimal portion accurately represents the time component. This numerical representation is what allows mathematical functions like MROUND to manipulate temporal data.

The formula's first parameter, `'my_data'`, feeds the raw numerical timestamp value that requires rounding into the MROUND calculation. The second parameter is perhaps the most critical component, as it defines the precise multiple to which the rounding must occur. This multiple is dynamically calculated using the TIME function: `TIME(0, 15, 0)`. This specific expression generates a numerical equivalent that corresponds exactly to a 15-minute interval (0 hours, 15 minutes, 0 seconds). Consequently, the MROUND function successfully rounds the original time value to the nearest multiple of 15 minutes, achieving our desired standardization.

The final, seemingly redundant component, `+ TIME(0, 0, 0)`, serves a vital formatting purpose. While the MROUND calculation accurately yields the correct numerical value for the rounded time, adding `TIME(0, 0, 0)`--which mathematically adds zero time--forces DAX to correctly interpret the final output. This ensures the resulting column is displayed as a proper time format, typically "HH:MM:SS PM/AM," rather than as a general numerical or raw decimal value, guaranteeing clarity and usability for end-users and reports.

Analyzing the Results and Adapting Intervals

The resulting data table provides compelling evidence of the formula's accuracy and consistency in aligning disparate timestamps with the strict quarter-hour convention. This standardization process is non-negotiable when constructing reliable visualizations, dashboards, or Key Performance Indicators (KPIs) that necessitate analysis based on fixed, periodic time buckets.

We can confirm the effectiveness of the rounding mechanism by examining several calculated examples from the dataset:

A timestamp already aligned, such as 10:15:00 AM, correctly remains rounded to **10:15:00 AM**.

A time slightly exceeding the quarter-hour threshold, such as 7:15:23 PM, is rounded down to the nearest multiple, resulting in **7:15:00 PM**.
Similarly, 1:15:09 AM is rounded down to **1:15:00 AM**.
Crucially, if a time like 1:16:00 AM were encountered, the inherent 'nearest' logic of the MROUND function would round it up to the subsequent quarter-hour, yielding **1:30:00 AM**.

This consistent and predictable application of rounding ensures that all subsequent analytical tasks--such as calculating average transaction times or pinpointing peak operational performance--are based exclusively on clean, comparable, and reliable time intervals. Furthermore, this technique is highly adaptable: if your analytical requirements shift to different intervals (e.g., 5 minutes, 10 minutes, or 30 minutes), you only need to adjust the minute parameter within the TIME function (e.g., use `TIME(0, 5, 0)` for 5-minute rounding or `TIME(0, 30, 0)` for half-hour rounding).

For developers or data architects seeking to integrate this logic into broader data models, the complete technical documentation for the MROUND function and the TIME function in DAX is the definitive reference, accessible via official Microsoft resources.

Expanding Your Expertise in DAX Time Manipulation

While mastering time rounding is a significant step, it represents only one facet of effective data analysis and modeling in Power BI. To continue enhancing your capabilities in handling complex temporal data, consider exploring these related resources:

A focused tutorial detailing the calculation of precise duration between two distinct dates using DAX formulas.
A comprehensive guide on creating and implementing dynamic fiscal calendars customized to specific reporting needs.
Advanced techniques for effectively managing time zones and addressing daylight savings time adjustments within large-scale data models.

These tutorials will further equip you with the essential skills required to clean, model, and analyze highly complex date and time fields accurately in your professional reports and dashboards.