

# How to Round Time to the Nearest Quarter Hour in Power BI

Authored by  
**mohammed looti**

January 10, 2026

## RECOMMENDED CITATION

mohammed looti (2026). *How to Round Time to the Nearest Quarter Hour in Power BI*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=125320>

Power BI is recognized globally as a leading business intelligence tool, empowering users to transform complex datasets into actionable insights. When performing data analysis, especially involving time-series or transactional records, it is frequently necessary to standardize time values. Raw timestamps, which often include seconds or milliseconds, can introduce unnecessary granularity and inconsistency into reports. To mitigate this issue and ensure that reporting is consistent across all data points, Power BI offers robust methods for rounding time, specifically allowing users to round these values to the nearest quarter hour. This process is crucial for scenarios like shift scheduling, production throughput reporting, and aggregation of instantaneous events into standardized 15-minute buckets, thereby significantly improving the quality and interpretation of the analytical results. This approach ensures that all timestamps are normalized to common intervals, making comparisons and calculations far more accurate and consistent for executive summaries and detailed reports.

## Understanding the Need for Time Rounding in Power BI

Modern datasets frequently contain timestamps recorded with high precision, often down to the millisecond. While this precision is valuable for auditing purposes, it poses significant challenges when attempting to aggregate data meaningfully for high-level business reporting. For instance, if one event occurred at 8:44:59 AM and another at 8:45:01 AM, standard grouping functions might place them in different buckets, obscuring trends that should logically be analyzed together within the same fifteen-minute interval. Rounding the time value to the nearest quarter hour, therefore, is a fundamental technique used to harmonize data prior to aggregation.

The concept of rounding ensures that all events within a narrow temporal window are assigned to a single, standardized checkpoint (e.g., :00, :15, :30, or :45 minutes past the hour). This standardization is not merely an aesthetic choice; it dramatically simplifies the process of creating visuals, calculating average processing times, or analyzing utilization rates based on fixed time slots. Without this critical step, analysts risk fragmented data views and misleading conclusions drawn from overly precise, yet inconsistently aligned, timestamps.

This particular requirement is often satisfied using Data Analysis Expressions (DAX), the powerful formula language native to Power BI, Analysis Services, and Power Pivot in Excel. DAX provides specialized functions designed to handle complex date and time calculations, allowing users to define precise rules for manipulating these data types directly within the data model. By leveraging DAX, we can implement a robust and repeatable solution for quarter-hour rounding that applies dynamically to any dataset.

## The Core DAX Formula for Quarter-Hour Rounding

To implement time rounding in Power BI, we utilize a combination of DAX functions. The primary

function employed for this task is the MROUND function, which mathematically rounds a given number to the nearest specified multiple. Since dates and times are stored internally in DAX as numerical values--where time is represented as a decimal fraction of a full day--we can use MROUND function to align the fractional time component to a multiple of 15 minutes.

The desired rounding increment of a quarter hour (15 minutes) must be expressed as a time value. We define this increment using the TIME function, specifying 0 hours, 15 minutes, and 0 seconds. This calculates the numerical equivalent of 15 minutes as a fraction of a 24-hour day, which serves as the precise multiple for the MROUND function. The resulting calculation is highly precise and ensures adherence to the 15-minute rule.

The complete formula below not only performs the rounding using the MROUND function but also includes a crucial addition of `TIME(0, 0, 0)`. While adding zero time seems redundant mathematically, this step is essential for ensuring that the final output is correctly formatted as a time or datetime type in Power BI, rather than displaying as a simple numeric decimal value, which is how DAX handles internal time calculations.

You can use the following syntax in DAX to round time to the nearest quarter hour:

**Rounded Time = MROUND('my\_data', TIME(0, 15, 0)) + TIME(0, 0, 0)**

This particular example creates a new column named **Rounded Time** that rounds the time component within the **Datetime** column of the table named **my\_data** to the nearest quarter hour. If the original timestamp was, for instance, **8:49:03 PM**, this formula would correctly round it down to **8:45:00 PM**, aligning it to the closest valid quarter-hour marker. This standardization is foundational for consistent reporting.

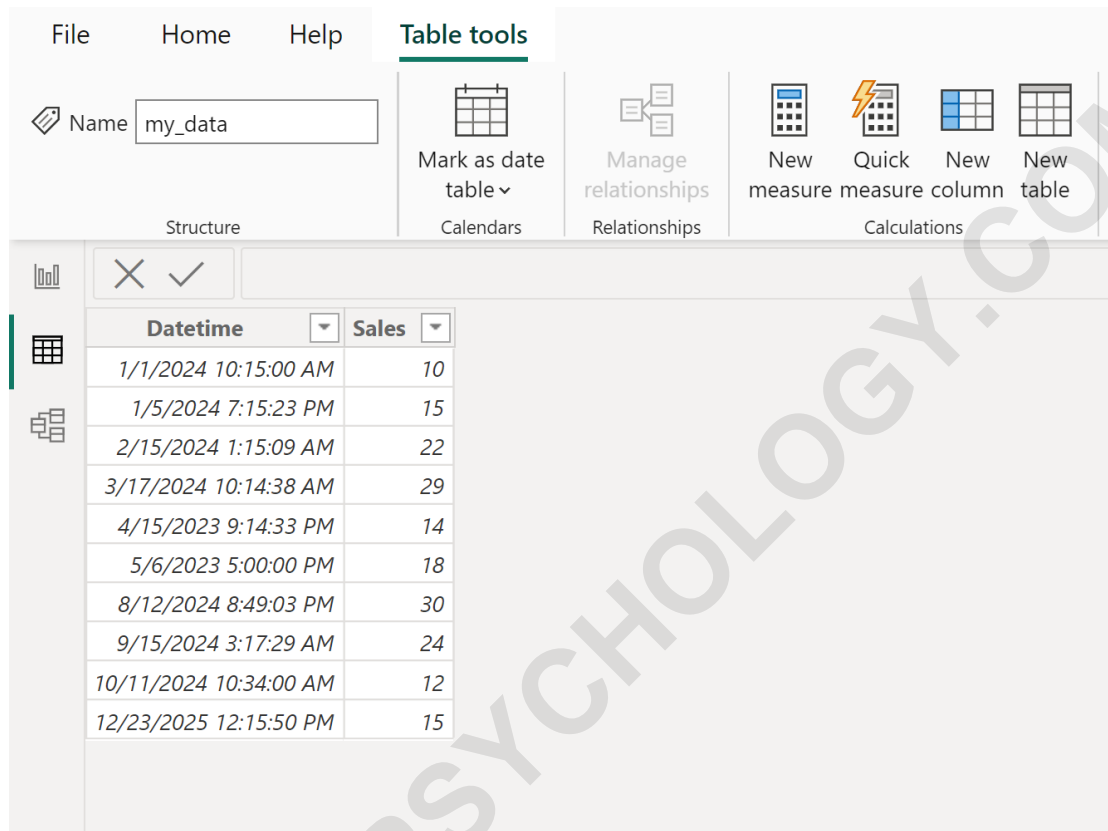
The following step-by-step procedure illustrates how to apply this syntax in a live Power BI data model, ensuring that all time entries conform to the required 15-minute interval.

### **Example: Setting Up the Data Scenario in Power BI**

To demonstrate this powerful rounding technique, let us consider a common scenario in business data analysis. Suppose we are tracking the total sales made by a company, where each transaction is logged with a precise timestamp. The source data, named **my\_data** in our Power BI model, contains a key column called **Datetime** which captures the exact moment the sale occurred. These precise times, however, make it difficult to quickly analyze sales volume within structured 15-minute operational slots.

Our objective is to create a new, derived column where every entry in the **Datetime** column is

mathematically adjusted to the nearest quarter hour. This derived column, which we will name **Rounded Time**, will allow us to accurately group sales activity and visualize trends based on standardized intervals, leading to cleaner and more effective business intelligence reports. Before applying the DAX formula, examine the structure of the initial dataset, which shows varied and unrounded timestamps:



The screenshot shows the Power BI Desktop interface. The 'Table tools' ribbon is active, with options for 'Mark as date table', 'Manage relationships', and 'Calculations' (New measure, Quick measure, New column, New table). Below the ribbon, a table is displayed with two columns: 'Datetime' and 'Sales'. The 'Datetime' column contains various timestamps, and the 'Sales' column contains numerical values.

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 illustrated in the table above, the **Datetime** column contains various minutes and seconds. We now proceed with the implementation steps to transform these high-granularity timestamps into standardized quarter-hour intervals.

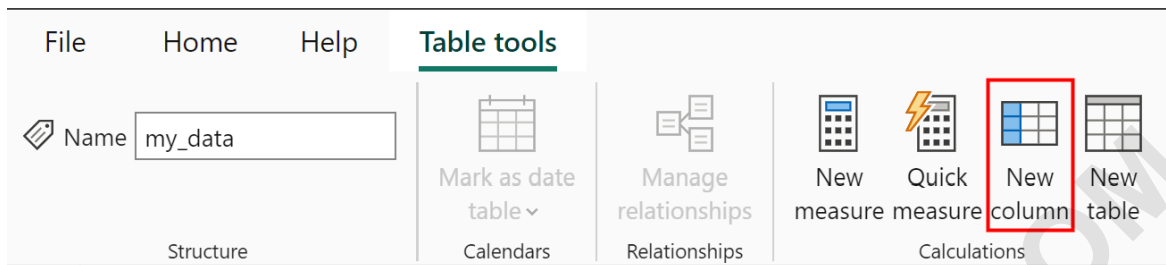
## Step-by-Step Implementation in Power BI Desktop

The process of adding a calculated column in Power BI is straightforward and requires navigating to the modeling tools within the Desktop environment. This calculation must be performed using a calculated column, not a measure, because we are deriving a new static time value for every row in the table, rather than calculating an aggregated value for a visual context.

To begin, ensure you are viewing the data model or the table view where the **my\_data** table is accessible. Locate the necessary options within the user interface. The primary method involves accessing the ribbon menu at the top of the Power BI Desktop application, specifically under the

data modeling or table tools sections.

To create the column that holds our rounded time, click the **Table tools** tab in the ribbon, then click the icon labeled **New column**:



Once you click **New column**, the DAX formula bar will appear, prompting you to define the calculation for this new field. This is where we will enter the complex expression involving the MROUND function and the specific quarter-hour increment.

## Applying the MROUND function and TIME function Formula

With the DAX formula bar active, we now define the calculation. It is essential to correctly reference the table and column names used in the data model. In this scenario, we are targeting the column within the 'my\_data' table, and setting the rounding multiple to be exactly 15 minutes.

The calculation is entered as follows, carefully combining the MROUND function for the mathematical rounding and the TIME function to specify the interval and ensure proper data type formatting. This precise instruction tells DAX to look at the numerical representation of the date and time, find the closest multiple of 15 minutes, and return that result as a formatted time value.

Type the following robust formula into the formula bar and press Enter:

**Rounded Time = MROUND('my\_data', TIME(0, 15, 0)) + TIME(0, 0, 0)**

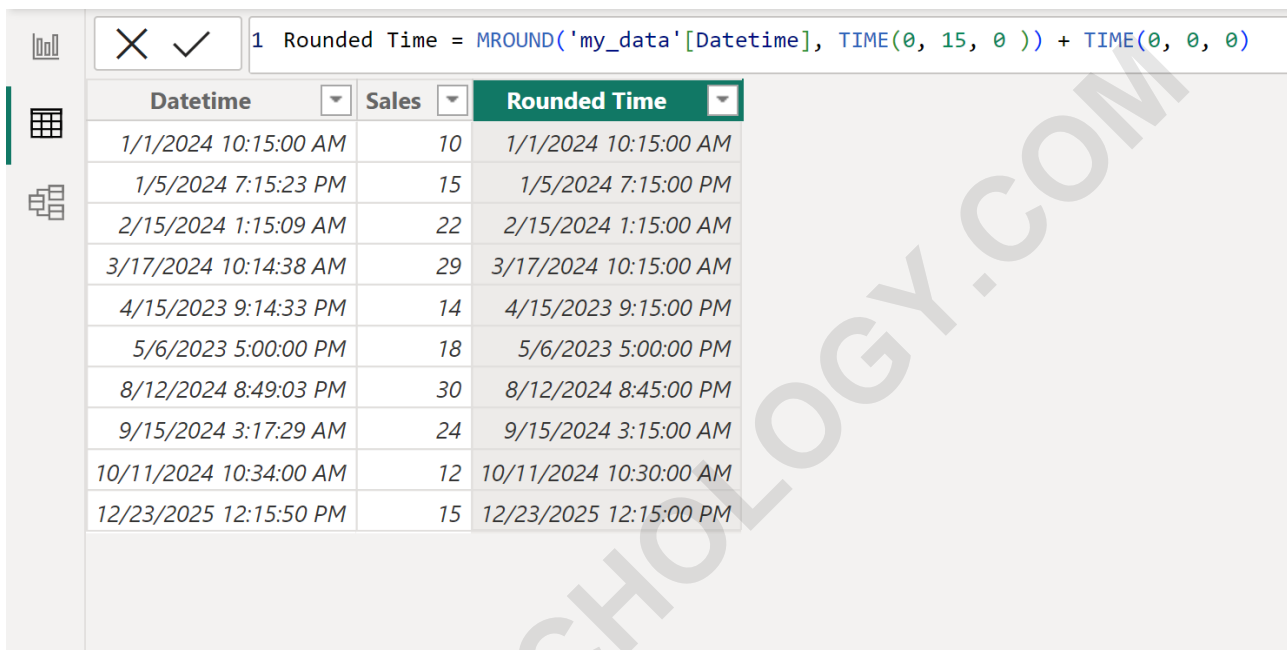
Upon execution, Power BI processes this DAX instruction across every row in the **my\_data** table. A new column, **Rounded Time**, is generated instantly, containing the standardized and rounded time value for each corresponding transaction. This immediate transformation makes the data ready for reporting and visualization.

## Verifying the Results and Output

After successfully applying the DAX formula, the data table is updated. This will create a new column named **Rounded Time** that contains the time value from the **Datetime** column rounded to

the nearest quarter hour, ensuring data consistency across the entire dataset.

A review of the resulting table confirms that the timestamps have been successfully normalized. Timestamps that fell slightly before or after a 15-minute mark (e.g., 10:14:59 AM or 7:16:05 PM) are now consolidated into the standardized quarter-hour buckets (10:15:00 AM or 7:15:00 PM). This structured approach is fundamental for any serious time-series data analysis.



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

The transformation is visually apparent and functionally effective. For instance, notice how precise timestamps like 10:15:00 AM, which was already aligned, remain unchanged, while slightly offset times are cleanly adjusted to the closest quarter-hour marker. This result confirms the precision and validity of our DAX implementation, providing a clean foundation for generating aggregate reports.

## Case Study Examples of Rounded Values

To fully grasp how the MROUND function handles different scenarios, considering specific examples is beneficial. The function rounds up or down based on standard mathematical rules: if the time falls exactly halfway between two quarters, it typically rounds away from zero. For practical time rounding, this means that a time value slightly closer to the previous quarter hour will round down, while a value slightly closer to the next quarter hour will round up.

For example, if the current time is 10:07:00 AM (the halfway point between 10:00 and 10:15 is 10:07:30), the formula would round this down to 10:00:00 AM. Conversely, if the time was 10:08:00 AM, it would round up to 10:15:00 AM. This precise midpoint handling ensures accurate allocation

of events to the correct 15-minute slot for reporting purposes. The following list showcases how various timestamps are accurately rounded:

The time 10:15:00 AM has been rounded to **10:15:00 AM** (No change, already aligned).

The time 7:15:23 PM has been rounded to **7:15:00 PM** (Rounded down, closest quarter hour).

The time 1:15:09 AM has been rounded to **1:15:00 AM** (Rounded down, closest quarter hour).

The time 4:22:45 PM has been rounded to **4:15:00 PM** (Rounded down, closer to 4:15 than 4:30).

The time 9:39:11 AM has been rounded to **9:45:00 AM** (Rounded up, closer to 9:45 than 9:30).

This systematic rounding mechanism provides the necessary foundation for advanced business intelligence applications where time-slot consistency is mandatory.

## Deep Dive: Mechanics of the MROUND function and Time Format

Understanding the fundamental mechanics of how DAX handles dates and times is crucial for implementing this calculation successfully. In Power BI, a date is stored as an integer, representing the number of days since December 30, 1899. Time, conversely, is stored as a decimal fraction of a day. For example, 12:00 PM is represented as 0.5, since it is exactly half of a 24-hour day.

When we use `TIME(0, 15, 0)`, DAX calculates the fractional numeric value equivalent to 15 minutes. Since there are 1,440 minutes in a day (24 hours \* 60 minutes), 15 minutes equates to 15/1440, or approximately 0.010416666... This fractional value becomes the rounding increment for the MROUND function.

The MROUND function takes the original datetime value (e.g., 44583.3567 for a specific date and time) and rounds the fractional part to the nearest multiple of 0.010416666.... The addition of `TIME(0, 0, 0)` is a common DAX technique. Although mathematically equivalent to adding zero, this action forces Power BI to interpret and format the resulting numerical value explicitly as a datetime data type, ensuring that the user sees **8:45:00 PM** instead of a confusing decimal number like 44583.3645.

Recall the formula that we used to round each time to the nearest quarter hour:

**Rounded Time = MROUND('my\_data', TIME(0, 15, 0)) + TIME(0, 0, 0)**

This formula first uses the **MROUND** function to round the time in the **Datetime** column to the nearest 15 minutes, which is the nearest quarter hour increment.

Then, we add **TIME(0, 0, 0)** which adds zero time to the rounded value. While mathematically neutral, this step is critically important as it ensures that the final result is correctly shown as a formatted time or datetime value instead of the underlying numeric data type used internally by

DAX.

**Note:** You can find the complete documentation for the MROUND function in DAX on the official Microsoft documentation pages for further detailed study.

## **Conclusion: Enhancing Reporting Consistency**

Rounding time to the nearest quarter hour in Power BI is more than just a data cleaning step; it is a critical component of preparing time-series data for effective data analysis and visualization. By utilizing the MROUND and TIME functions within a DAX calculated column, analysts can transform highly granular timestamps into standardized, uniform intervals. This consistency enables cleaner aggregation, simplified reporting, and ultimately, more reliable insights for business decision-making.

Mastering this technique ensures that your reports avoid ambiguity caused by fractional minutes and allows key stakeholders to analyze operational patterns based on clearly defined, predictable 15-minute intervals. This methodology can be adapted for any desired rounding increment, making it a versatile tool in the Power BI analyst's toolkit.