

How to Calculate the Maximum Value in Power BI with an Example

Authored by
mohammed looti

January 11, 2026

RECOMMENDED CITATION

mohammed looti (2026). *How to Calculate the Maximum Value in Power BI with an Example*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=125620>

The calculation of the maximum value is a fundamental operation in Data Analysis, providing immediate insight into the upper bounds of a dataset. Within Power BI, this crucial task is performed using the powerful MAX function. The MAX function is specifically designed to calculate the absolute highest value present in a designated column or set of numerical data. Understanding how to correctly implement this function is essential for effective data summarization and reporting, as it allows users to quickly identify peak performance metrics, highest sales records, or maximum recorded scores.

The process generally involves identifying the specific data context--be it an entire column within a fact table or a filtered subset of data--and then applying the MAX function through a Measure definition. For instance, imagine a scenario where a company tracks its monthly revenue across several regions. By utilizing the MAX function on the revenue column, analysts can immediately pinpoint the single month that generated the highest income, thereby establishing a benchmark. Furthermore, the functionality extends beyond simple column scanning; MAX can be dynamically integrated with other functions, such as filters, to ensure that the calculation only considers specific subsets--like finding the maximum revenue only for the "East" region--providing a granular level of detail crucial for strategic decision-making.

This introductory guide will explore the precise definition, syntax, and step-by-step implementation of the MAX function within the Power BI environment, leveraging the capabilities of Data Analysis Expressions (DAX). We will transition from theoretical concepts to a practical, detailed example, demonstrating how to create a new measure and display the resulting maximum value clearly within a report visualization. Mastery of this function is a prerequisite for any serious Power BI user aiming to transform raw data into actionable business intelligence.

Defining the MAX Function within Power BI

In the context of Power BI, the MAX function is a scalar function belonging to the DAX library. Its primary purpose is to evaluate a column and return the largest numerical value contained within it. When applied to textual data, it returns the value closest to the end of the alphabet, and when applied to date/time data, it returns the latest date or time. This versatility makes MAX indispensable for various data types, though it is most commonly employed for quantitative analysis, particularly in defining key performance indicators (KPIs) and summarizing large datasets efficiently. It is important to note that the MAX function operates directly on the physical column data within the specified table.

The calculated result of MAX is typically stored within a Measure in Power BI. A measure is a dynamic calculation that changes based on the filtering context applied by the report user. Unlike calculated columns, which compute values row-by-row and consume memory, measures calculate aggregated results on the fly. Therefore, defining the maximum value as a measure ensures

optimal performance and adaptability across different visuals and slicer selections in the report, maintaining data integrity and responsiveness even with massive data models. This structural choice is fundamental to effective Power BI development.

To implement this aggregation, users must first identify the target data field. Whether the data is sourced from a SQL database, Excel sheets, or cloud services, as long as it is loaded into the Power BI Desktop data model, it is accessible for the `MAX` function. The syntax is straightforward, requiring only the column name as an argument. The power of this function truly shines when combined with filtering mechanisms, allowing analysts to move beyond global maximums to calculate context-specific maximums, such as the highest sales recorded by a specific employee or during a particular financial quarter.

Core Principles of the Data Analysis Expressions (DAX) MAX Function

DAX, or Data Analysis Expressions, is the specialized formula language used throughout Power BI (as well as Analysis Services and Power Pivot in Excel) to create new information from data already in the model. The `MAX` function is one of the core aggregation functions available in this language. It is categorized as a statistical function, specifically designed to handle numerical data summarization, although its application extends to text and date types as previously mentioned. The function evaluates the entire column provided in its argument and returns the single largest element encountered.

A crucial principle when working with `MAX` in DAX is its handling of non-numeric and blank values. When the function encounters non-numeric data that cannot be converted to a number for comparison, it typically ignores those values if the column is primarily numeric. However, if the column contains entirely mixed data types, the result can be unpredictable or error-prone. Blank values, or nulls, are generally treated as zero in mathematical operations in DAX, though the specific behavior depends on the context and whether the value is truly blank or zero. For robust calculations, data cleansing and type conversion should precede advanced DAX formulas.

The standard syntax for calling the MAX Function in DAX, when defining a Measure, requires specifying the table and the column, using the format `'Table Name'`. This explicit reference ensures that the DAX engine knows precisely which data scope to examine. For instance, if you have a sales table called `'Financial Records'` and a column detailing transaction amounts called `Amount`, the calculation would be `MAX('Financial Records', Amount)`. This clear structure promotes readability and maintainability of complex data models.

Syntax and Structure for Calculating Maximum Values

The fundamental syntax for calculating the maximum value within a column in Power BI using DAX is simple and powerful. When creating a new Measure, you assign a descriptive name followed by

the assignment operator (=), and then the function call. The core of the aggregation relies entirely on the function signature and its argument, which must be a column reference.

You can use the following syntax in DAX to calculate the max value in a column:

Max Points = MAX('my_data')

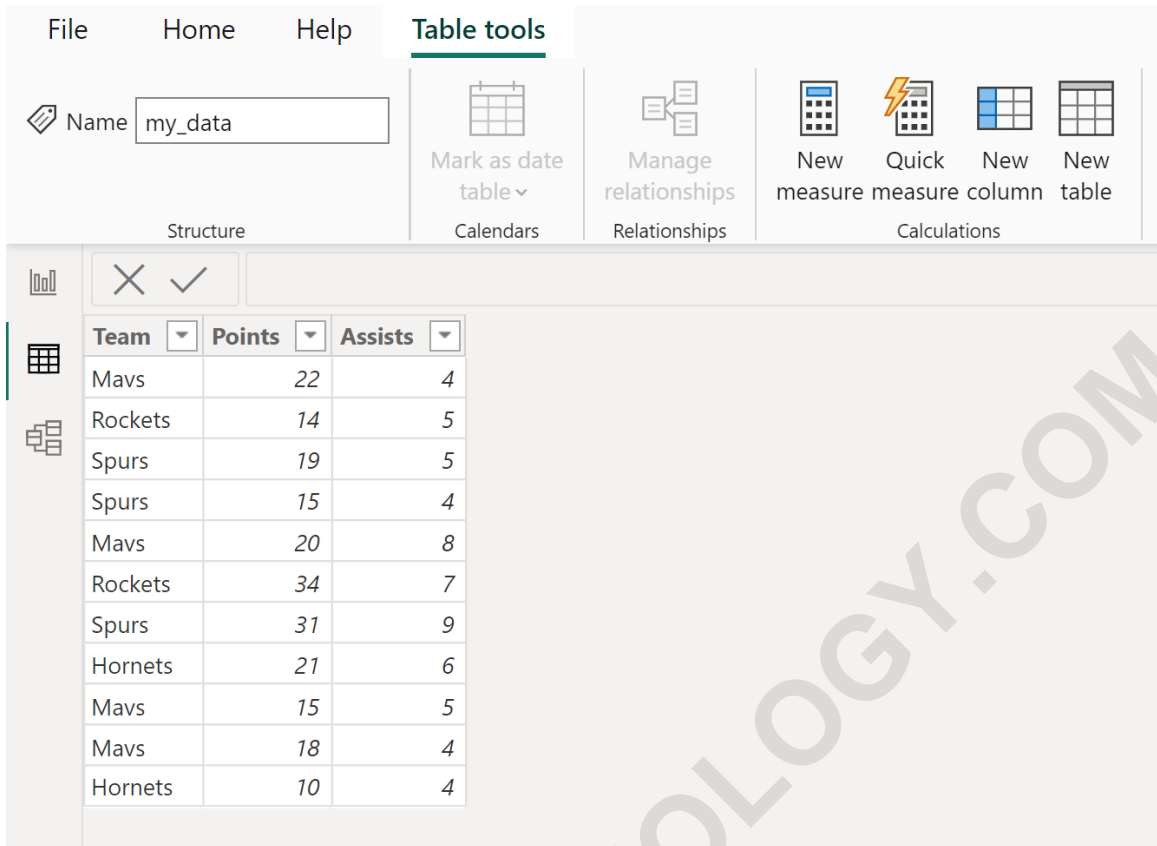
This particular example creates a new measure named **Max Points**. This measure performs the crucial task of calculating the maximum value observed across all rows in the **Points** column, which belongs to the table designated as **my_data**. Defining the measure this way ensures that whenever **Max Points** is used in a visualization (like a card or a table), it will dynamically return the largest available score based on any filters active on the report canvas. This efficiency highlights why measures are the preferred method for aggregation results in Power BI reporting.

Understanding the component parts of this syntax is essential. `Max Points` is the chosen, human-readable name for the resulting measure. The `=` sign signifies the formula definition. `MAX` is the DAX Function itself. Finally, `'my_data'` is the fully qualified name of the column, where the single quotes enclose the table name (necessary if the table name contains spaces or special characters, though used here for clarity) and the brackets enclose the column name. This structure ensures that the DAX engine can resolve the reference unambiguously within the data model schema.

Setting Up the Practical Scenario in Power BI

To illustrate the practical application of the `MAX` function, we will utilize a sample dataset representing sports statistics. Suppose we have imported a simple table into the Power BI data model, which is named **my_data**. This table contains performance metrics for basketball players, specifically tracking the points scored per game by various team members. This setup mirrors many real-world scenarios where analysts need to quickly determine the highest performance metric achieved within a category.

Suppose we have the following table in Power BI named **my_data** that contains information about points scored by basketball players on various teams:



Our objective is clearly defined: we need to calculate the single maximum value recorded in the **Points** column. This calculation will serve as a valuable KPI, perhaps indicating the highest scoring game recorded across the entire dataset. Before proceeding with the formula, it is always wise to visually inspect the data to confirm the data type is numeric and that there are no obvious anomalies that might skew the maximum calculation, though the `MAX` function is generally resilient to minor data quality issues compared to average functions.

The benefit of using a Measure for this task is that the resulting highest score will be immediately available for use across all visual elements in the report. If we were to use a calculated column instead, the calculation would be redundant, as calculating the maximum point score for every single row is inefficient and unnecessary. Measures ensure that computational resources are only utilized when the data is requested by a visual element, making the report highly performant. The next step involves navigating the Power BI Desktop interface to initiate the creation of this crucial aggregation.

Detailed Procedure: Creating the MAX Measure

The creation of a new Measure in Power BI Desktop is initiated via the ribbon interface, specifically designed for data modeling tasks. This process ensures that the calculation is correctly registered in the data model and associated with the appropriate table, typically the table containing the

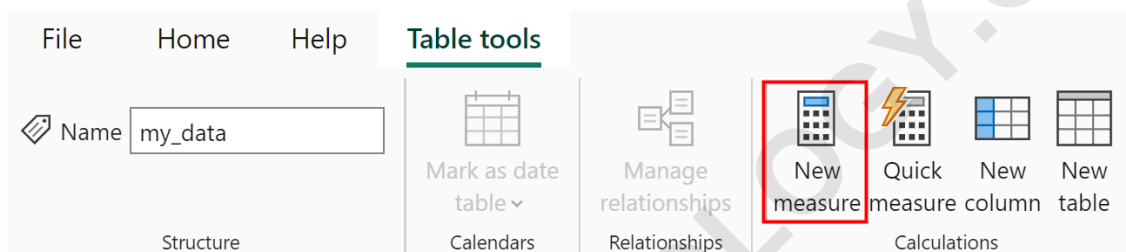
column referenced in the aggregation.

To calculate the maximum value in the **Points** column, the following procedural steps must be executed sequentially:

Navigate to the **Table tools** tab: Along the top ribbon of the Power BI Desktop interface, click the **Table tools** tab. This tab provides essential functions for managing tables and creating new data model objects.

Click the **New measure** icon: Within the **Table tools** section, locate and click the **New measure** icon. This action opens the formula bar, allowing the user to input the DAX formula.

The visualization below depicts the location of the **New measure** button within the interface:



Once the formula bar is active, the precise DAX syntax must be input to define the aggregation. The formula is typed directly into the bar, ensuring correct spelling and referencing the exact table and column names as they appear in the data model. Remember that DAX is case-insensitive for function names but sensitive for table and column names depending on the underlying data source, though standard practice is to match the model's casing exactly for clarity.

Then type in the following formula into the formula bar:

Max Points = MAX('my_data')

Upon committing this formula (by pressing Enter or clicking the checkmark), Power BI creates a new measure named **Max Points**. This measure instantly calculates the maximum score from the **Points** column of the specified table. The new measure will appear in the Fields pane, typically identifiable by a calculator icon next to its name, indicating its nature as a dynamic calculation. At this stage, the calculation engine has successfully determined the highest value based on the current context, which, without any filters applied, represents the global maximum score.

This will create a new measure named **Max Points** that contains the maximum value found in the **Points** column of the table:

The screenshot shows the DAX editor interface. At the top, a formula bar contains the measure definition: `1 Max Points = MAX('my_data'[Points])`. Below the formula bar is a table with three columns: Team, Points, and Assists. The table contains 12 rows of data representing different players and their statistics.

| Team | Points | Assists |
|---------|--------|---------|
| Mavs | 22 | 4 |
| Rockets | 14 | 5 |
| Spurs | 19 | 5 |
| Spurs | 15 | 4 |
| Mavs | 20 | 8 |
| Rockets | 34 | 7 |
| Spurs | 31 | 9 |
| Hornets | 21 | 6 |
| Mavs | 15 | 5 |
| Mavs | 18 | 4 |
| Hornets | 10 | 4 |

Visualizing the Maximum Value using the Card Element

After successfully defining the **Max Points** measure, the next step is to integrate this crucial statistic into the user-facing report. Since the maximum value is a single, aggregated numerical result, the most appropriate visualization to display this KPI clearly and prominently is the **Card** visual in Power BI.

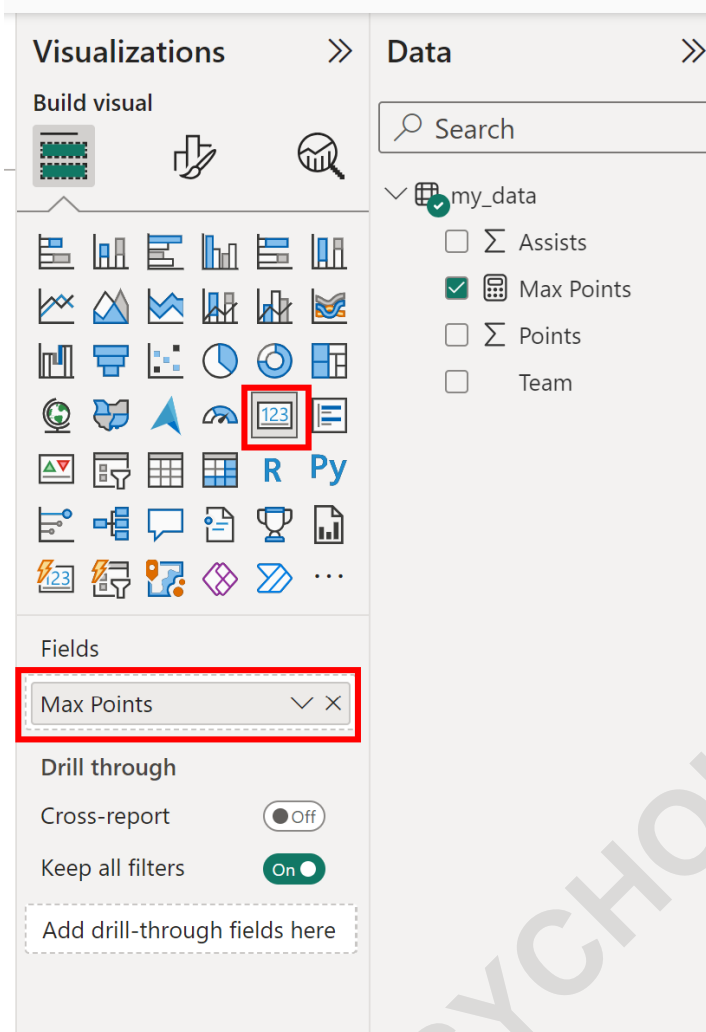
To display this calculated maximum value, the user must navigate to the **Report View** within Power BI Desktop. The visualization process involves selecting the appropriate visual element and assigning the newly created Measure to its data fields:

Accessing the Report View: Ensure you are in the **Report View** (the main design canvas) of Power BI.

Selecting the Card Visual: In the **Visualizations** pane, click the **Card** icon. This adds an empty Card visual to the canvas.

Assigning the Measure: Locate the **Max Points** measure in the Fields pane and drag it onto the Card visual, or into the **Fields** well under the Visualizations tab.

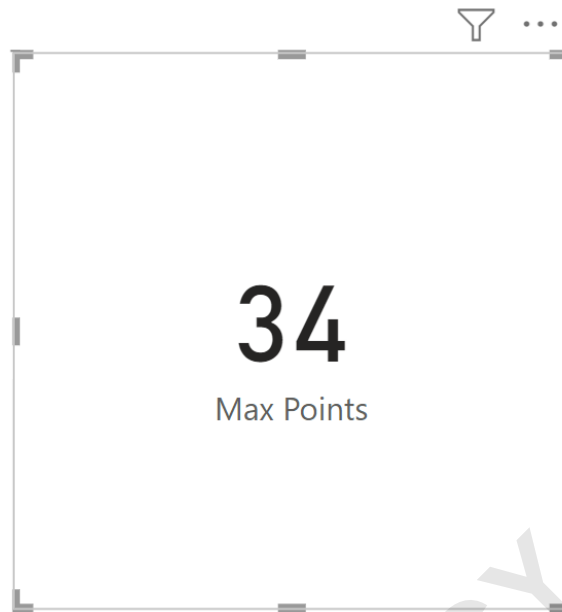
The following illustration shows the process of dragging the **Max Points** measure into the Card visual's fields area, demonstrating the link between the measure definition and its visual representation:



Once the measure is assigned, the Card visual automatically calculates and displays the aggregated result defined by the `MAX` function. This immediate feedback confirms the successful implementation of the DAX formula. The Card visual is invaluable for presenting key metrics, offering highly customizable formatting options to ensure the number stands out and is easily understood by report consumers. It typically formats large numbers automatically (e.g., displaying 1,000,000 as 1M), although for smaller integers like scores, the raw value is usually displayed.

This will produce the following card that clearly displays the maximum value calculated from the **Points** column of the table:

| Team | Points | Assists |
|---------|--------|---------|
| Hornets | 10 | 4 |
| Hornets | 21 | 6 |
| Mavs | 15 | 5 |
| Mavs | 18 | 4 |
| Mavs | 20 | 8 |
| Mavs | 22 | 4 |
| Rockets | 14 | 5 |
| Rockets | 34 | 7 |
| Spurs | 15 | 4 |
| Spurs | 19 | 5 |
| Spurs | 31 | 9 |



From the visual output, we can clearly ascertain that the maximum value within the **Points** column across the entire dataset is **34**. This simple, elegant display provides instantaneous insight into the highest single performance recorded in the source data, fulfilling the analytical objective established at the beginning of the scenario.

Differentiating MAX from MAXX: Iteration Context

While the standard `MAX` function is perfect for finding the highest value directly contained within a single column, `DAX` also provides an advanced, iterative variant: `MAXX`. Understanding the distinction between these two functions is crucial for complex data modeling, especially when the maximum value needs to be calculated based on an expression evaluated row-by-row, rather than just pulling a value from an existing column.

The `MAX` function requires a column reference as its sole argument and is strictly an aggregation function that operates on physical data. In contrast, `MAXX` is an iterator function. Iterator functions evaluate an expression over every row of a specified table before aggregating the results. The syntax for `MAXX` is `MAXX (`