Power BI Tutorial: Calculating Sums with Filters Using DAX

Authored by

Mohammed looti

November 12, 2025

Mastering Conditional Aggregation in Power BI using DAX

The ability to perform conditional aggregation is fundamental for advanced data analysis. In Power BI, calculating a sum based on specific criteria--often referred to as a filtered sum--requires leveraging the powerful capabilities of DAX (Data Analysis Expressions). Unlike standard Excel formulas, DAX introduces concepts like filter context and row context, which are managed primarily by the versatile CALCULATE function.

This guide details two primary methods for calculating a sum while applying filters to your data model. Whether you need to filter based on a single condition or combine multiple complex criteria, these DAX patterns provide the precise control necessary for generating accurate and insightful reports. Understanding how to correctly apply filters within the aggregation context is essential for building robust and dynamic reports in Power BI.

We will explore both the single-filter and multiple-filter approaches, providing practical syntax examples and step-by-step instructions for implementation within the Power BI desktop environment. These techniques are crucial for data professionals aiming to move beyond simple totals and create sophisticated, context-aware business intelligence solutions.

Method 1: Calculating Sums with a Single Filter Condition

The most straightforward approach to conditional summation involves using the CALCULATE function to modify the filter context of the underlying data before the aggregation occurs. The CALCULATE function is arguably the most important function in DAX, as it allows you to evaluate an expression (like SUM) under a modified set of filters.

When defining a new Measure, the structure for applying a single filter is concise and highly readable. You provide the calculation expression (e.g., summing a column) as the first argument, followed by the specific filter expression you wish to enforce. This filter expression temporarily overrides or adds to any existing filters applied by the visual context.

The following syntax demonstrates how to calculate the sum of values in the **Points** column, restricted only to rows where the **Team** column equals "Mavs."

Sum Points =
CALCULATE (SUM ('my_data'), 'my_data' = "Mavs")

This formula generates a new measure named **Sum Points**. Crucially, it calculates the sum of values exclusively within the **Points** column for those rows where the corresponding value in the **Team** column is exactly "Mavs." The CALCULATE function ensures that the aggregation function, SUM, operates only on the subset of data defined by the filter argument.

Method 2: Applying Conditional Sums with Multiple Filters

Frequently, business logic demands aggregation based on multiple criteria that must all be met simultaneously. The CALCULATE function natively supports this requirement by allowing you to introduce multiple filter arguments, separated by commas. When multiple filters are passed to CALCULATE, they are inherently treated as a logical AND condition.

This approach allows for highly specific data segmentation. For instance, you might want to sum sales only for a specific product line AND only within a certain geographical region AND only if the order quantity exceeds a threshold. Each condition is listed sequentially after the main aggregation expression.

In this example, we calculate the sum of **Points**, but only for players belonging to the "Mavs" team AND whose **Assists** count is greater than 4.

```
Sum Points =
CALCULATE (
SUM ( 'my_data' ),
'my_data' = "Mavs",
'my_data' > 4
)
```

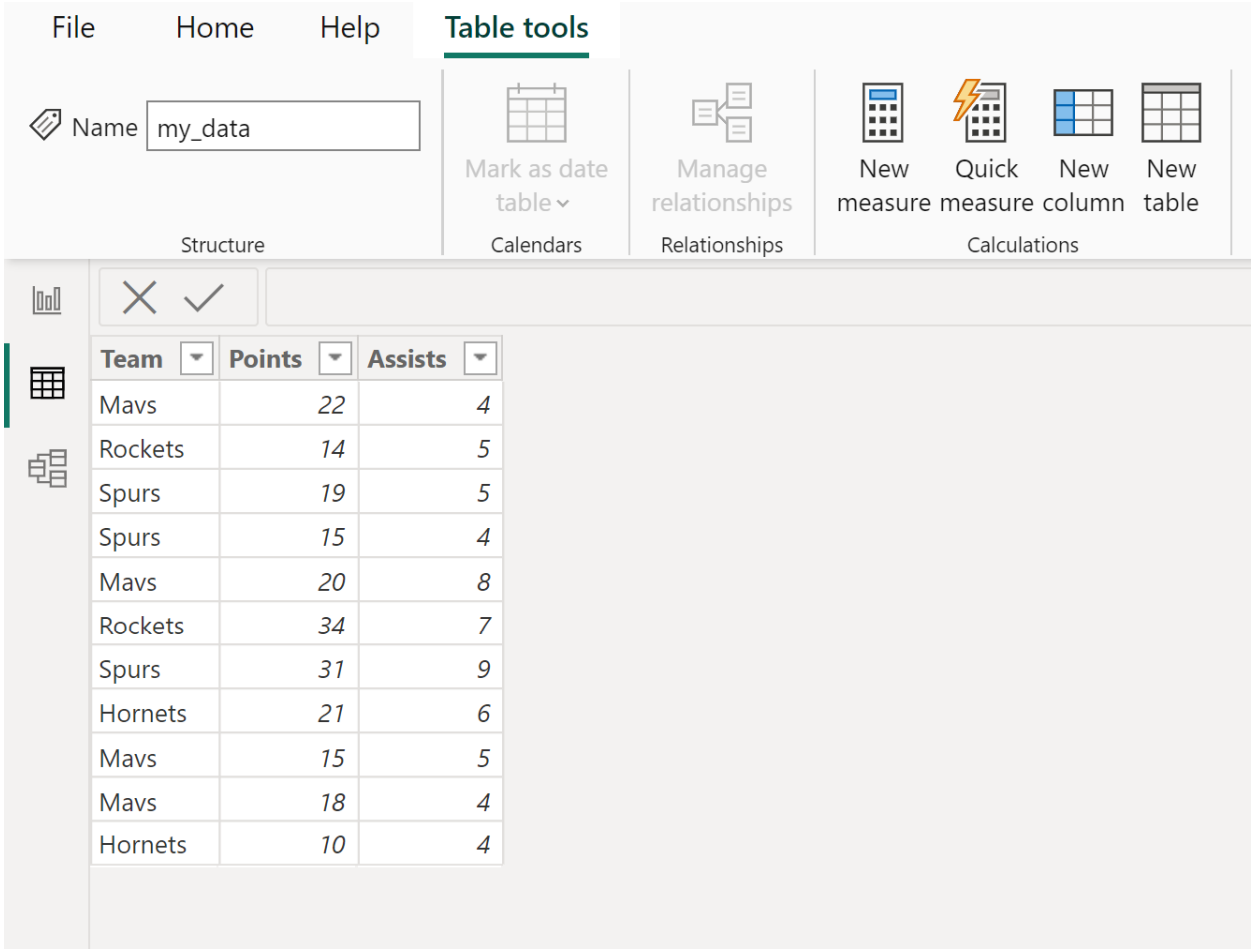
The resultant Measure, **Sum Points**, computes the sum of the **Points** column only for rows where the value in the **Team** column is equal to "Mavs" and where the value in the **Assists** column satisfies the condition of being greater than 4. This simultaneous application of filters ensures that only records meeting both criteria contribute to the final aggregated total.

This method highlights the efficiency of DAX, as it eliminates the need for complex nested IF statements often required in other calculation environments, relying instead on the straightforward application of filter arguments within the CALCULATE function.

Practical Demonstration: Understanding the Data Context

To illustrate these concepts, we will utilize a sample dataset named **my_data**. This table contains performance metrics for various basketball players, including their team affiliation, points scored, and assists made. Before diving into the measure creation, it is helpful to visualize the data structure upon which our conditional calculations will operate.

The structure of **my_data**, shown below, reveals the columns we will reference in our DAX formulas: **Points** (the column to be summed), **Team** (our primary categorical filter), and **Assists** (our numerical filter).



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

Our goal is to accurately segregate and sum the points based on the specified team and assist criteria, demonstrating how CALCULATE dynamically changes the evaluation context of the sum.

Case Study 1: Calculating Sum with One Filter Applied

Consider a scenario where the requirement is to aggregate the total points scored exclusively by players on the "Mavs" team. This task requires defining a new Measure that implements the single-filter logic discussed previously.

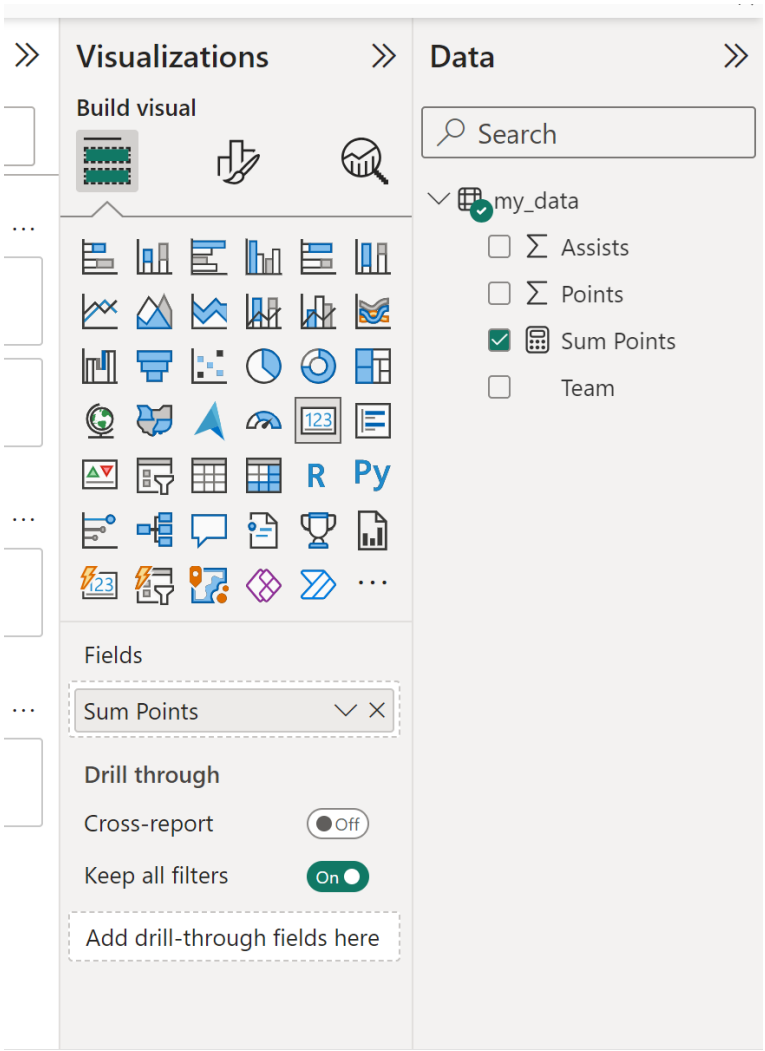
To begin, navigate to the **Table tools** tab in the Power BI desktop interface. Within this ribbon, locate and click the **New measure** icon. This action initiates the creation of a persistent calculation that can be used across various visuals in your report.

In the formula bar that appears, input the DAX formula designed for single-filter aggregation, referencing our defined table and columns.

Sum Points =
CALCULATE (SUM ('my_data'), 'my_data' = "Mavs")

Once confirmed, a new measure named **Sum Points** is successfully created. This measure holds the conditional total of points, restricted only to the players associated with the Mavs team within the **my_data** table. This calculation is now ready to be visualized.

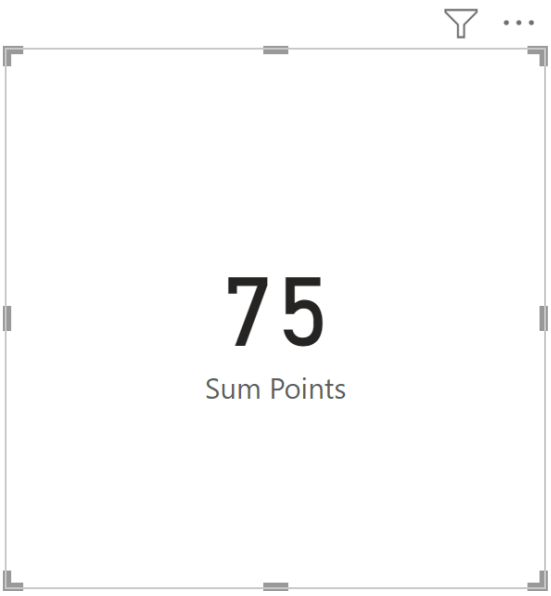
To display this result, drag the newly created **Sum Points** measure onto a visualization, such as a Card visual. The visual will immediately reflect the calculated total, adhering strictly to the applied filter context.



The resultant Card visual confirms the aggregation, showing the total sum of values in the **Points**

column exclusively for Mavs players.

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



Based on the data and the calculated output, we can definitively state that the players on the Mavs team scored a total of **75** points. This demonstrates the precision and effectiveness of using CALCULATE for simple conditional aggregation.

Case Study 2: Calculating Sum with Multiple Filters Enforced

For a more complex analytical request, let us determine the sum of points scored only by players on the Mavs team who also recorded more than four assists. This requires implementing the multiple-filter logic, ensuring both conditions are met simultaneously.

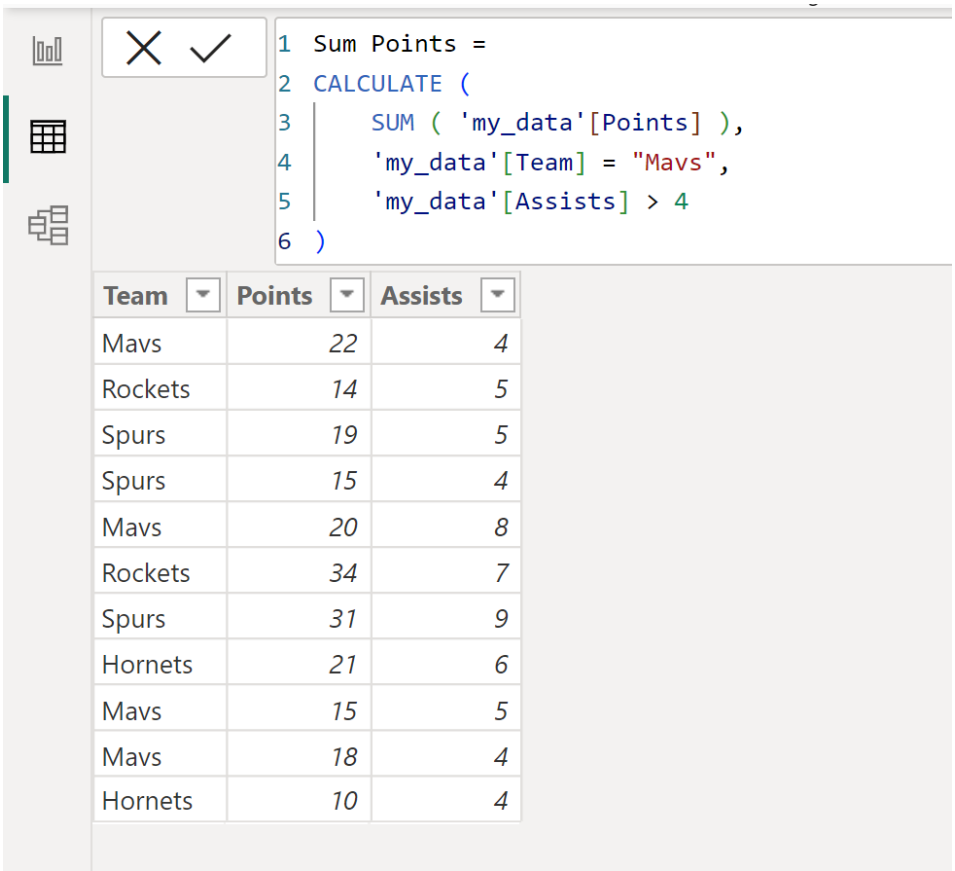
As before, start by navigating to the **Table tools** tab and selecting the **New measure** icon to initiate the measure creation process in Power BI.

Next, input the multi-conditional DAX formula into the measure bar. Note the use of the comma delimiter to separate the two distinct filter arguments within the CALCULATE function.

```
Sum Points =
CALCULATE (
SUM ( 'my_data' ),
```

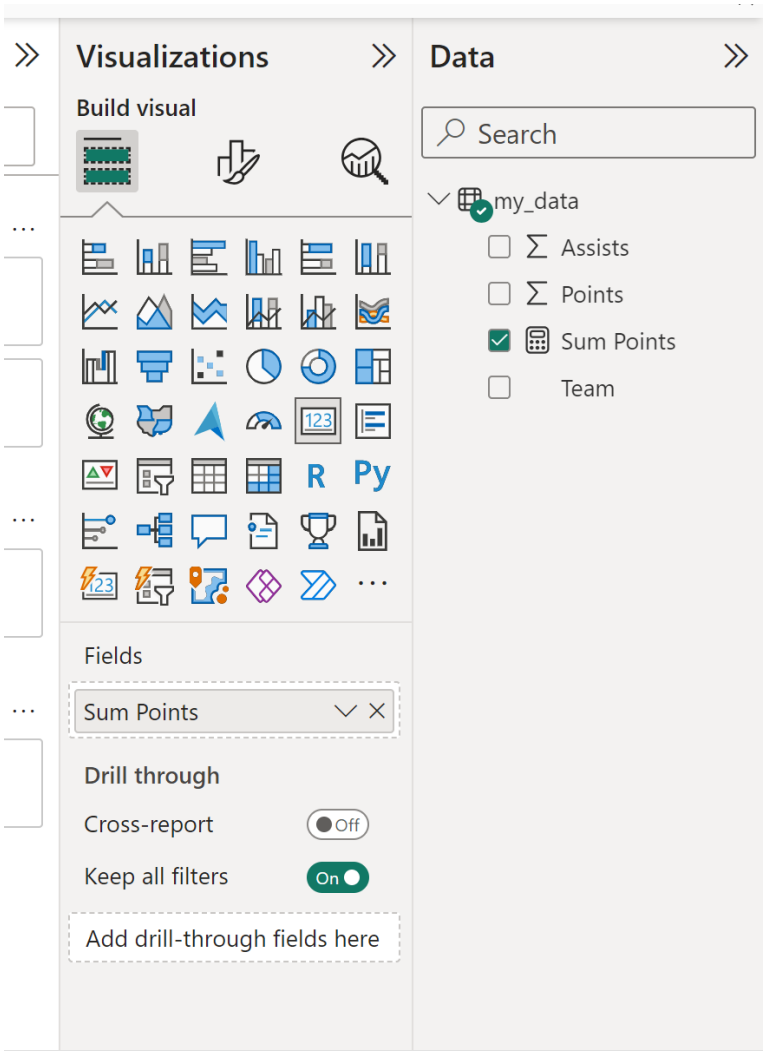
'my_data' = "Mavs",
'my_data' > 4
)

This process creates the **Sum Points** measure, which now encapsulates the sum of points scored exclusively by Mavs players whose assists exceeded the threshold of 4. The resulting measure is now highly segmented and specific to our analytical need.



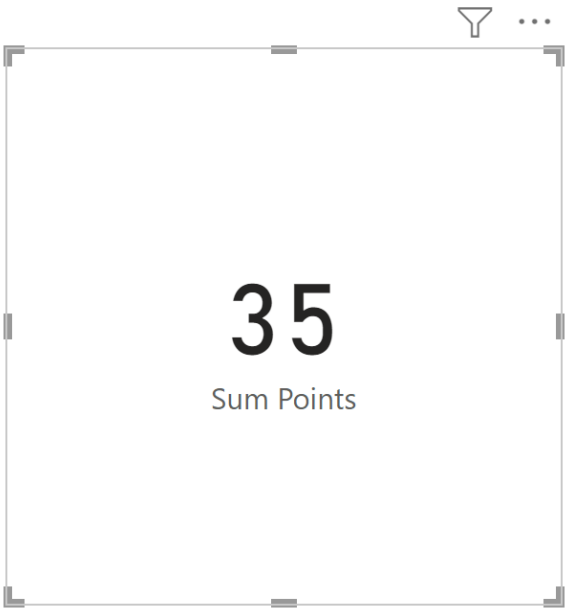
Team	Points	Assists
Mavs	22	4
Rockets	14	5
Spurs	19	5
Spurs	15	4
Mavs	20	8
Rockets	34	7
Spurs	31	9
Hornets	21	6
Mavs	15	5
Mavs	18	4
Hornets	10	4

To visualize this specific total, switch to the **Report View** in Power BI. Click the **Card** icon located under the **Visualizations** pane, and then drag the newly created **Sum Points** Measure into the **Fields** well of the visual.



The final Card visual displays the filtered sum, representing the total points achieved by the highly specific subset of players who satisfied both the team and the assists criteria.

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



The output confirms that the players on the Mavs team who had more than 4 assists scored a total of **35** points. This successful implementation showcases the power and flexibility of the CALCULATE function in managing complex filter requirements within a single Measure definition.

Advanced DAX Concepts and Further Resources

The techniques demonstrated here form the cornerstone of advanced DAX development. While we focused on SUM, the CALCULATE pattern is universally applicable to other aggregation functions like AVERAGE, COUNT, and MAX. Mastery of the filter context manipulation is critical for writing efficient and accurate business logic in Power BI.

For those seeking to expand their knowledge beyond simple conditional sums, explore related DAX functions such as FILTER, ALLEXCEPT, and ALL. These functions provide even greater control over how external filters interact with the internal context defined within a measure.

The following resources offer tutorials and documentation to help you perform other common and advanced tasks within the Power BI ecosystem:

How to calculate averages conditionally.
Techniques for using time intelligence functions in DAX.
Detailed guides on creating calculated columns versus measures.