

How to Calculate a Sum with Filters in Power BI

Authored by
mohammed looti

January 12, 2026

RECOMMENDED CITATION

mohammed looti (2026). *How to Calculate a Sum with Filters in Power BI*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=125645>

Analyzing subsets of data is fundamental to effective business intelligence. While standard visualizations in **Power BI** allow users to apply interactive filters, generating specific, static sums based on predefined criteria requires leveraging the power of **DAX** (Data Analysis Expressions). This deep dive explores how to create calculated **Measures** that permanently define both the aggregation (the sum) and the specific conditions (the filter) simultaneously, ensuring reliable and reusable metrics across your reports. To achieve this level of precision, we rely heavily on the powerful **CALCULATE** function within the **Power BI** environment.

Understanding Calculated Measures vs. Visual Filters

When working in a **Power BI** report, you typically have two main ways to apply filtering and aggregation. The first is through visual or page-level filters, which dynamically adjust the data displayed based on user selection or slicers. The second, and far more robust method for creating fixed key performance indicators (KPIs), is by defining explicit **Measures** using **DAX**. These calculated measures embed the filter logic directly into the metric definition, ensuring the sum always adheres to the criteria, regardless of the visual context in which it is used. This is essential for creating reliable reporting metrics that remain consistent across different views and pages.

The requirement to calculate a sum under a specific condition (e.g., total sales for a single region, or total points scored by a specific team) necessitates overriding the inherent **Filter Context** of the visual. While the simple **SUM** function aggregates all visible rows, we need a function capable of evaluating an expression in a modified context. This critical function is **CALCULATE**, which acts as the cornerstone of complex aggregation logic in **DAX**.

The Role of CALCULATE in Defining Context

The **CALCULATE** function is arguably the most important function in **DAX**. It evaluates an expression in a context modified by new filters. When we want to calculate a conditional sum, we wrap the **SUM** aggregation within **CALCULATE**, followed by the specific conditions we wish to impose. The expression provided to **CALCULATE** is the calculation we want to perform (e.g., **SUM('Table')**), and subsequent arguments are the filters that define the new evaluation environment.

Understanding how **Filter Context** works is crucial here. When you place a measure on a report, the measure inherits the context defined by the rows, columns, and slicers applied to that visual. **CALCULATE** allows us to introduce a new, internal filter that modifies or overrides this external context. For instance, if a visual is showing all teams, but you want a measure specifically for the "Mavs," **CALCULATE** imposes "Mavs" as an additional, permanent filter on that aggregation.

The fundamental structure for applying conditional aggregation in **Power BI** using **DAX** is provided below, demonstrating how the core aggregation function, **SUM**, is evaluated under specific

conditions defined by **CALCULATE**.

Method 1: Calculating Sum with a Single Filter

When you need to aggregate a column based on a single condition--for example, calculating the total points scored only by members of a specific team--the simplest form of the **CALCULATE** function is used. This method clearly isolates the aggregation (the sum of points) from the filtering condition (the team name).

The following syntax creates a new **Measure** named **Sum Points**. This measure uses the **SUM** function to aggregate the values in the column, but only where the corresponding value in the column is exactly "Mavs."

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

This particular formula creates a new **Measure** named **Sum Points** that contains the sum of values in the **Points** column only for the rows in the table where the value in the **Team** column is equal to "Mavs." The filter argument **'my_data' = "Mavs"** is applied to the data model before the **SUM** function is executed.

Method 2: Calculating Sum with Multiple Filters

For more complex analytical requirements, you often need to satisfy multiple conditions simultaneously. For instance, you might need the total points scored only by players on the "Mavs" team who also achieved more than four assists. **CALCULATE** easily accommodates this by allowing multiple filter arguments, separated by commas.

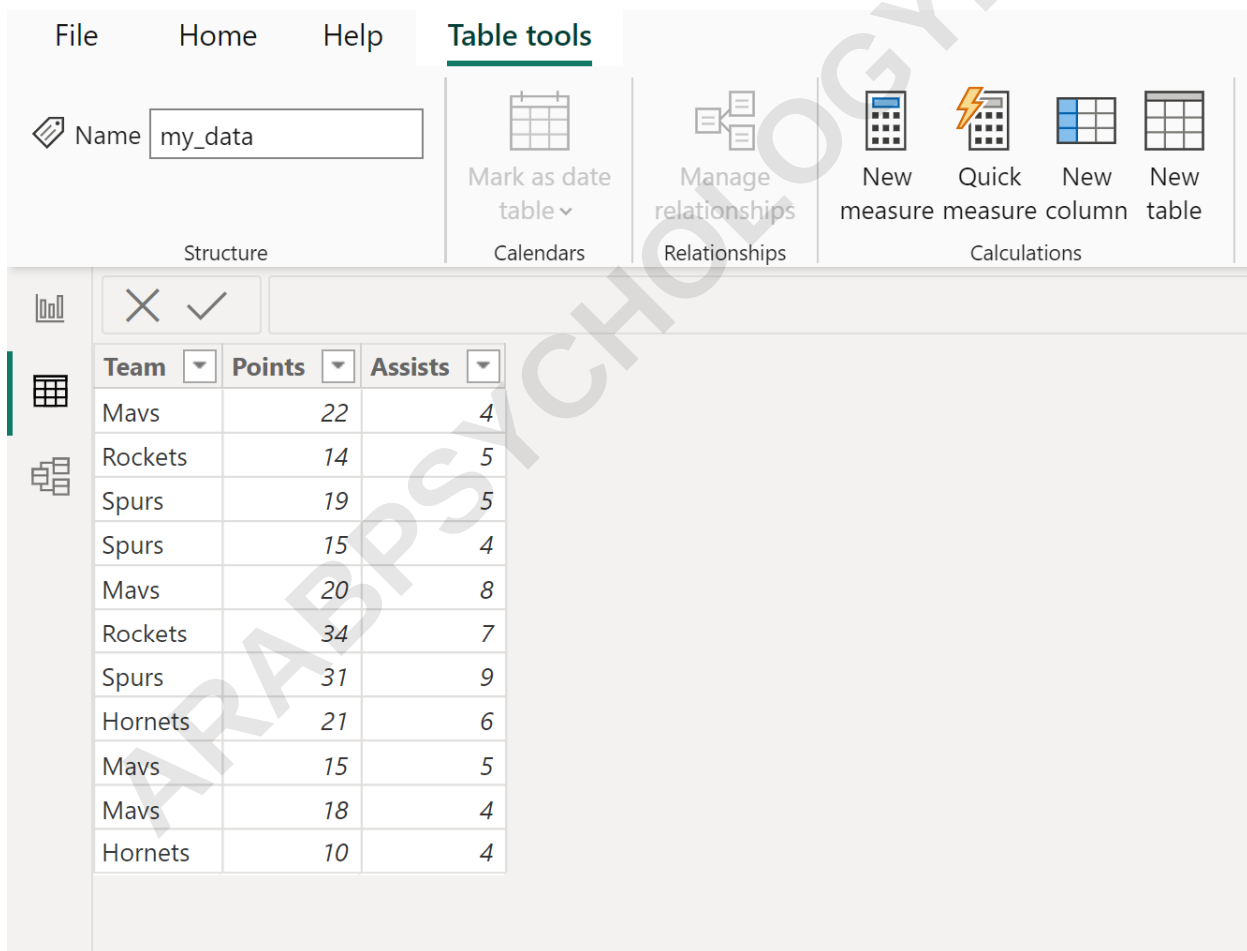
When multiple filters are supplied to **CALCULATE**, they are implicitly treated as logical **AND** conditions. This means that for a row to be included in the final aggregation, it must satisfy all specified filter conditions. This is a highly efficient way to define narrow, precise subsets of data for aggregation within **DAX**.

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

This particular formula creates a new **Measure** named **Sum Points** that contains the sum of values in the **Points** column only for the rows in the table where the value in the **Team** column is equal to "Mavs" *and* where the value in the **Assists** column is greater than 4. By adding 'my_data' > 4 as a second argument, we refine the filter context significantly, ensuring only data points meeting both criteria are included in the final **SUM** calculation.

Setting Up the Demonstration Data

To illustrate these concepts practically, we will use a sample dataset within **Power BI**. Assume we have a table named **my_data** that contains statistical information about various basketball players, including their team affiliation, points scored, and assists made. This table serves as the basis for calculating our conditional sums.



The screenshot shows the Power BI Desktop interface. The 'Table tools' ribbon is active, displaying options like 'Mark as date table', 'Manage relationships', and 'New measure'. Below the ribbon, a table named 'my_data' is visible, containing columns for 'Team', 'Points', and 'Assists'. The table lists 12 rows of data for various basketball teams and players.

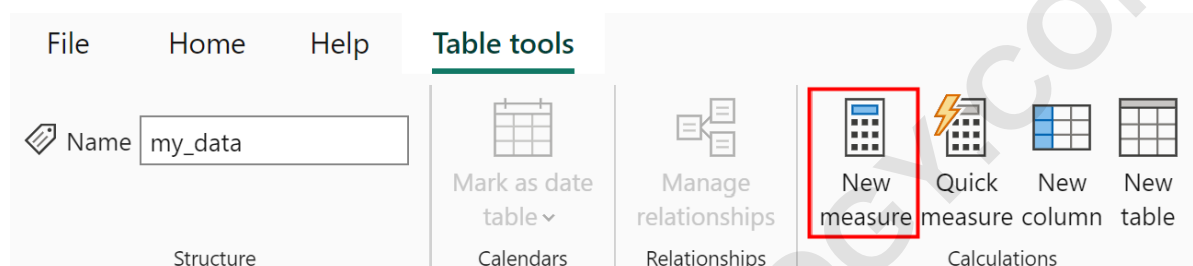
Team	Points	Assists
Mavs	22	4
Rockets	14	5
Spurs	19	5
Spurs	15	4
Mavs	20	8
Rockets	34	7
Spurs	31	9
Hornets	21	6
Mavs	15	5
Mavs	18	4
Hornets	10	4

As shown in the image above, the table contains details for players across three teams (Mavs, Nets, Lakers). Our goal is to derive specific summary statistics for the Mavs team using the explicit filtering capabilities of **DAX Measures**. The process for creating these measures starts by accessing the modeling tools within the **Power BI** Desktop application.

Example 1: Calculating Sum with a Single Filter in Practice

Suppose we would like to create a new measure that precisely shows the sum of points scored exclusively by players on the Mavs team. This requires implementing the first method discussed, utilizing a single filter argument within **CALCULATE**.

To begin, navigate to the **Table tools** tab within the **Power BI** ribbon, and then click the **New measure** icon. This action opens the formula bar, allowing you to define the **DAX** logic for your new metric.



Once the formula bar is active, type the following **DAX** formula. Ensure the table and column references match your data model exactly.

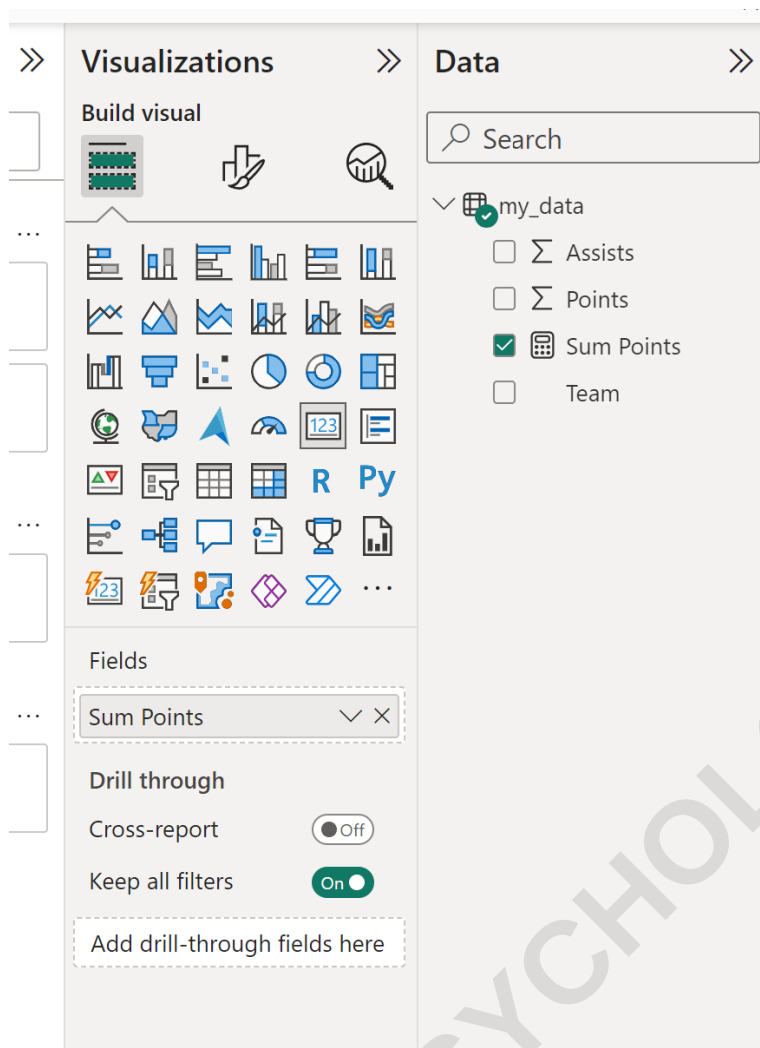
Sum Points =

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

Upon committing the formula, a new measure named **Sum Points** will be created and added to your data model. This measure contains the aggregate sum of points only for the players associated with the Mavs team, overriding any default **Filter Context** that might be present in the report visual.

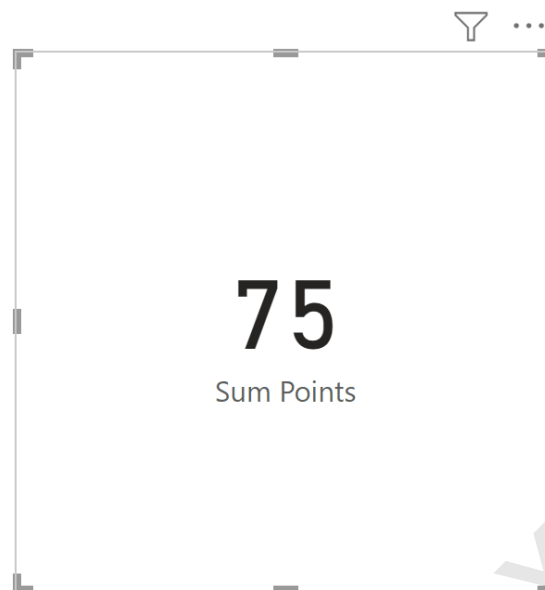
1 Sum Points =
2 CALCULATE (SUM ('my_data'[Points]), 'my_data'[Team] = "Mavs")

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



By placing this new **Sum Points** measure onto a visualization, such as a Card visual, we obtain the filtered result immediately. This is one of the most common applications for calculated **Measures**, providing an unchanging metric value.

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

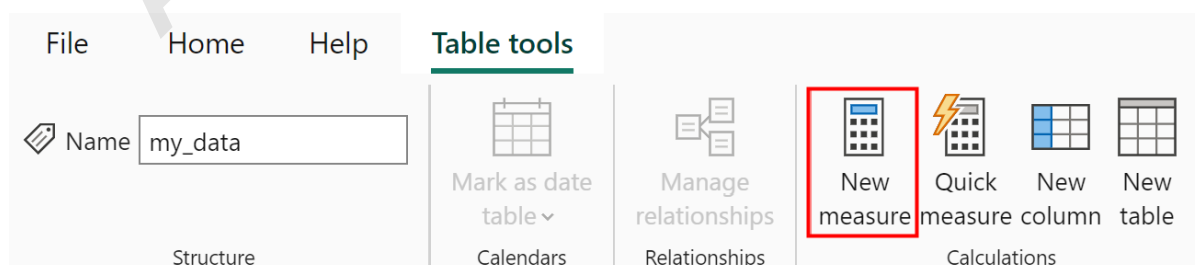


The resulting Card visual confirms that the players on the Mavs team scored a total of **75** points. This value is derived by summing the points (30 + 15 + 30) only for the rows where the team equals "Mavs."

Example 2: Calculating Sum with Multiple Filters in Practice

Building upon the previous example, let us define a more stringent metric: the sum of points scored only by players on the Mavs team who also achieved more than 4 assists. This requires incorporating two distinct filter conditions within the **CALCULATE** statement, demonstrating the power of compound filtering in **DAX**.

The procedure for creating this measure remains identical to the single-filter method. Click the **Table tools** tab, then click the **New measure** icon to open the formula bar.

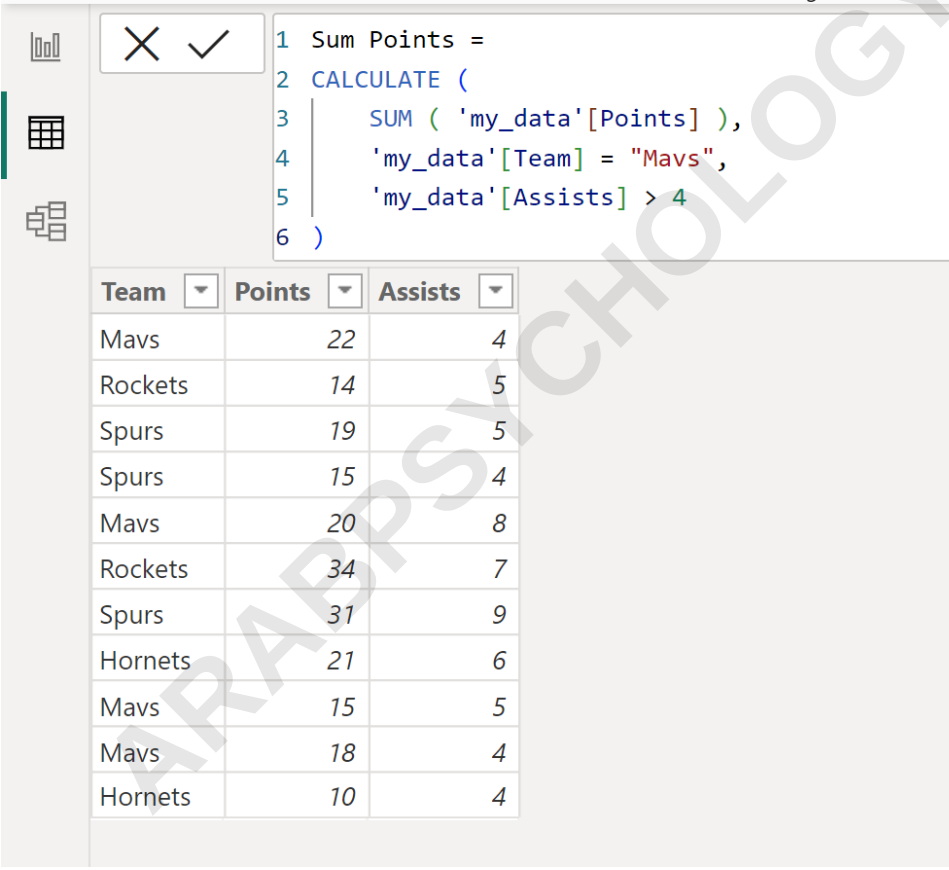


Then, type the following expanded formula into the formula bar, including both the team filter and the assist threshold filter. Note that both filters are applied simultaneously using the comma

separator, creating an implicit **AND** relationship.

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

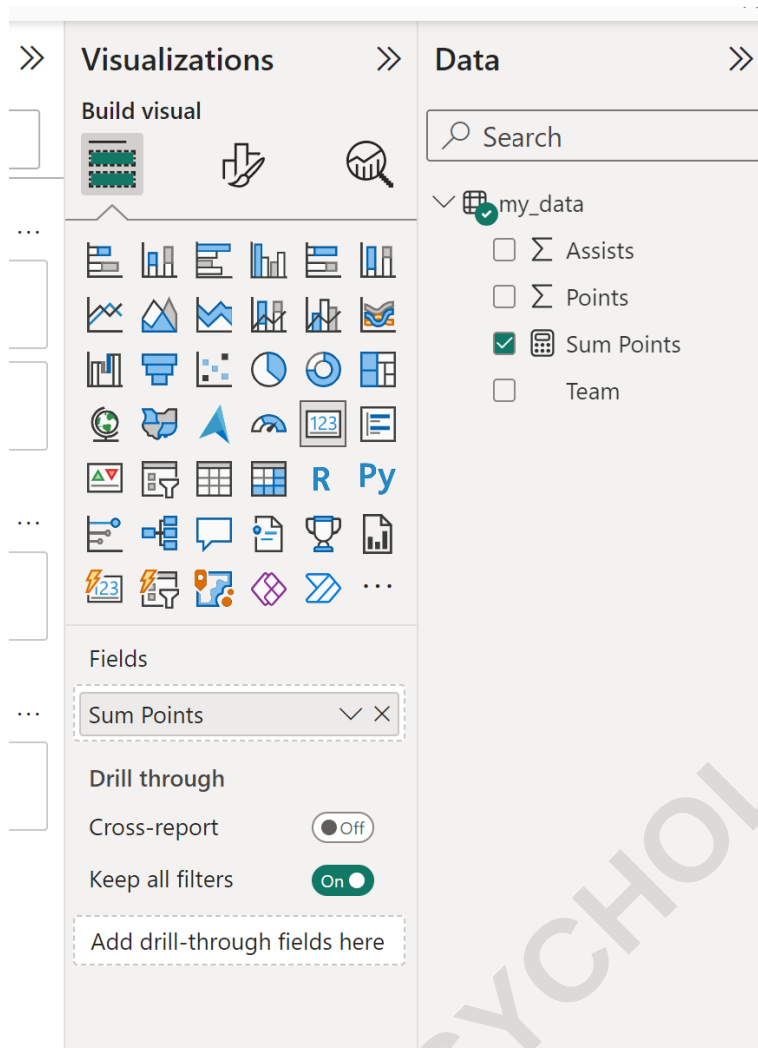
This definition results in a new measure named **Sum Points** that calculates the sum of points only for the players on the Mavs team who had more than 4 assists. This highly specific **Filter Context** ensures that only qualifying rows are aggregated.



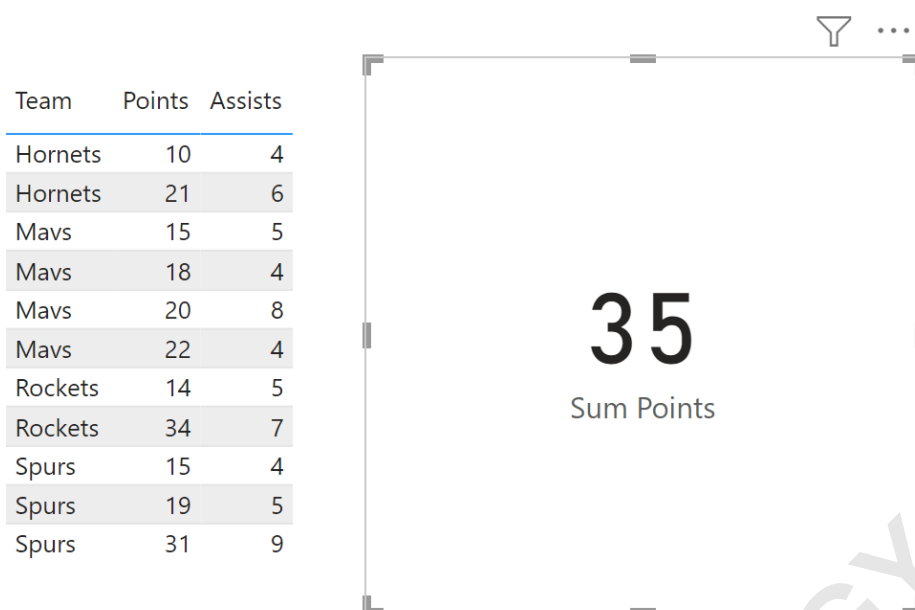
The screenshot shows the Power BI interface. At the top, there is a formula bar with a DAX measure definition. Below it, a table displays the data for the 'my_data' table, filtered by the measure's criteria. The table has three columns: Team, Points, and Assists. The data is filtered to show only rows where the Team is 'Mavs' and the Assists are greater than 4.

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 display this refined value, switch to the **Report View** in **Power BI**, select the **Card** visual under the **Visualizations** tab, and drag the new **Sum Points** measure into the **Fields** well.



The resulting card visual shows the sum of values in the **Points** column that satisfy both the team criteria and the assists criteria. Only two players on the Mavs team meet the assist requirement (Assists > 4), resulting in a reduced total sum.



Based on the output, we confirm that the players on the Mavs team who had more than 4 assists scored a combined total of **35** points. This result demonstrates how precisely **CALCULATE** enables the creation of complex, conditional aggregated metrics, forming the backbone of sophisticated data analysis in **Power BI**.

Summary of DAX Best Practices

When developing conditional sums using **DAX**, it is always recommended to use explicit **Measures** rather than calculated columns for aggregation logic. Measures are calculated dynamically at the time of query execution, optimizing performance and reducing file size, especially when dealing with large datasets. Furthermore, placing filters directly within the **CALCULATE** function is the standard and most readable method for modifying the **Filter Context**, ensuring that the defined metric is always accurate and consistent across the entire report. Always ensure that column references are fully qualified (e.g., 'Table') for clarity and robust formula execution.

The following tutorials explain how to perform other common tasks in **Power BI**: