

How to Calculate a Filtered Sum in Power BI: A Step-by-Step Guide

Authored by
stats writer

January 28, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Calculate a Filtered Sum in Power BI: A Step-by-Step Guide*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=128155>

Calculating the sum based on conditional criteria--often referred to as conditional aggregation--is a core analytical requirement in modern business intelligence platforms like Power BI. This powerful technique allows analysts to determine the total value of a specific metric (like sales or points) only for the subset of data that meets predetermined conditions, such as grouping by a region, date range, or product type. By carefully defining the filter criteria, users can generate highly precise calculations, leading to robust and tailored analytical results essential for effective decision-making.

To execute this complex task within Power BI, we rely heavily on the Data Analysis Expressions (DAX) language, specifically employing the versatile CALCULATE function. This function is paramount because it allows us to override the standard filter context applied to a visual, enabling us to calculate an aggregate like SUM while simultaneously applying one or more specific filters directly within the definition of a new analytical measure.

Power BI: Calculate Sum with Filter

Understanding Conditional Summation in DAX

Conditional summation in Power BI is typically achieved through the definition of a custom measure using DAX. Unlike implicitly filtered sums that rely on slicers or visual interactions, these custom measures contain their own internal filtering logic, ensuring the calculation always returns the correct value regardless of external report filters.

The core principle involves the CALCULATE function, which is arguably the most powerful function in the entire DAX library. CALCULATE takes an expression (in this case, the simple SUM of a column) and then evaluates it under a new or modified filter context defined by the subsequent arguments. This capability is essential for defining nuanced business logic.

You can use the following syntax methods in DAX to calculate a sum with a filter in Power BI. We will explore scenarios involving both single and multiple filter conditions, demonstrating how to precisely control the data being aggregated.

Method 1: Calculating Sum with a Single Filter Condition

When your requirement involves aggregating data based on a single, straightforward condition--such as finding the total sales for a specific product category--you can utilize the simplest form of the CALCULATE function. This method is highly efficient for targeted aggregation tasks where the filtering criteria are fixed.

The standard pattern requires defining the aggregation expression first, followed immediately by the filter expression that restricts the rows considered for the calculation. This filter acts as a temporary override to the existing context for the duration of the calculation. The resulting measure

then holds the conditional sum.

Here is the formula structure used to implement a sum with a single filter, focusing on the team "Mavs" in our example data set:

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

This formula successfully creates a new measure named **Sum Points**. This measure calculates the sum of values found in the **Points** column, but it exclusively considers rows in the `my_data` table where the value in the **Team** column is precisely equal to "Mavs." Rows not meeting this condition are temporarily excluded from the summation.

Method 2: Calculating Sum with Multiple Filter Conditions (Logical AND)

Often, analytical requirements demand aggregation based on more complex criteria involving two or more conditions that must simultaneously be true. When multiple filter arguments are passed to the `CALCULATE` function, they are implicitly treated as a logical AND condition. That is, a row must satisfy **all** specified filter conditions to be included in the final aggregation.

This is extremely useful for drilling down into specific, narrow segments of your data, such as counting points scored by a specific team AND only for players who achieved a minimum number of assists. It provides unparalleled precision in defining calculated metrics.

The following structure illustrates how to implement this using two distinct filter arguments within `CALCULATE`:

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

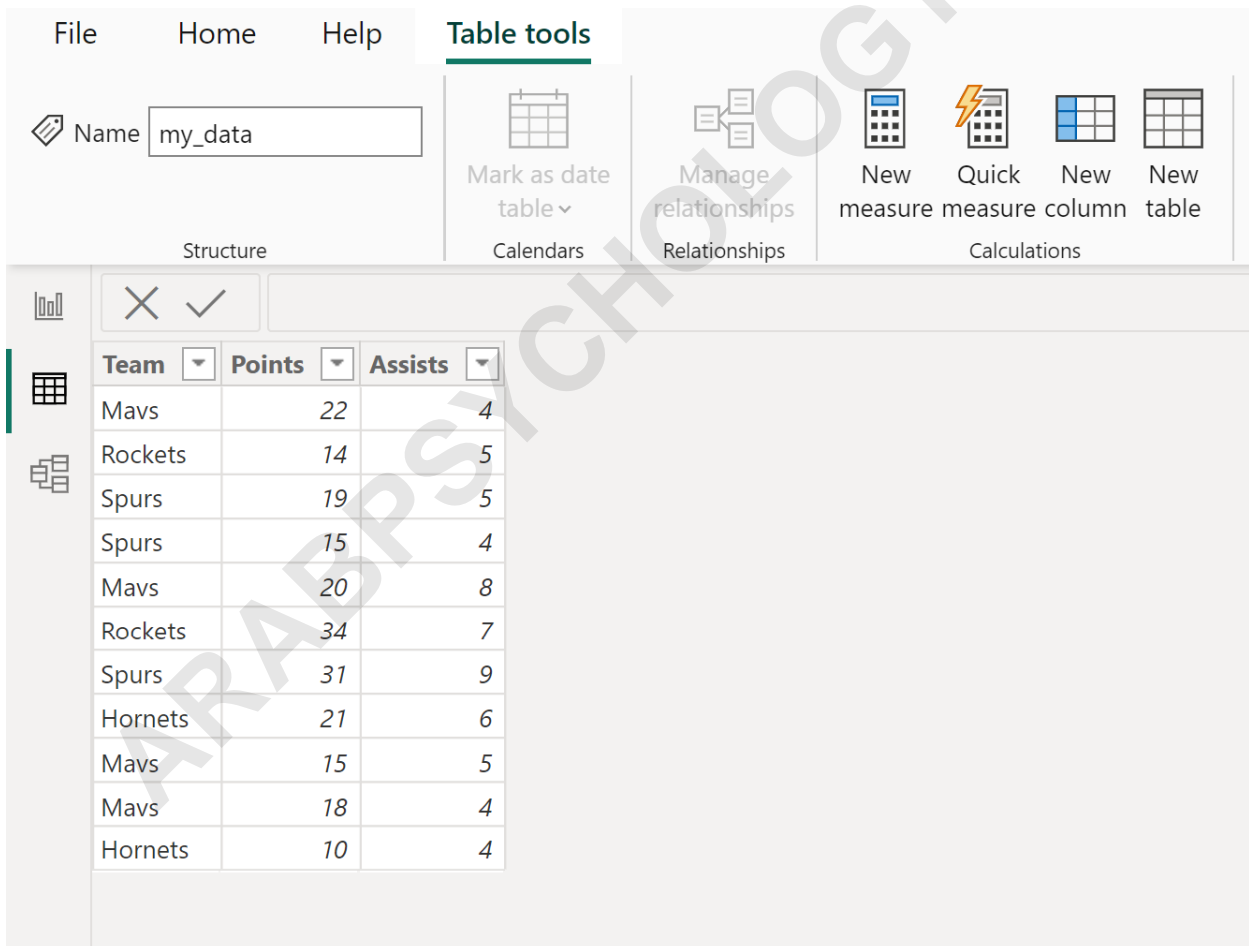
This robust formula defines the measure **Sum Points**. It calculates the aggregate sum of the **Points** column only for those rows where the **Team** column equals "Mavs" *and* where the value in the **Assists** column is strictly greater than 4. If either condition is not met for a given row, that row is excluded from the final sum.

Setting Up the Data Model for Examples

To illustrate these concepts practically, we will use a sample data table named **my_data** loaded into Power BI. This table contains performance information relating to various basketball players and their respective teams. Understanding the structure of this source data is crucial before defining the conditional sums.

The table contains key columns such as the player's name, their team, total points scored, and total assists recorded. Our goal is to derive specific totals from the **Points** column based on filters applied to the **Team** and **Assists** columns.

The following visual representation displays the structure of the **my_data** table we will be referencing throughout these examples:



The screenshot shows the Power BI interface with the 'Table tools' ribbon active. The 'Name' field is set to 'my_data'. The ribbon includes options for 'Mark as date table', 'Manage relationships', and 'Calculations' (New measure, Quick measure, New column, New table). Below the ribbon, a data table is displayed with the following structure:

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

Example 1: Procedural Guide for a Single Filter

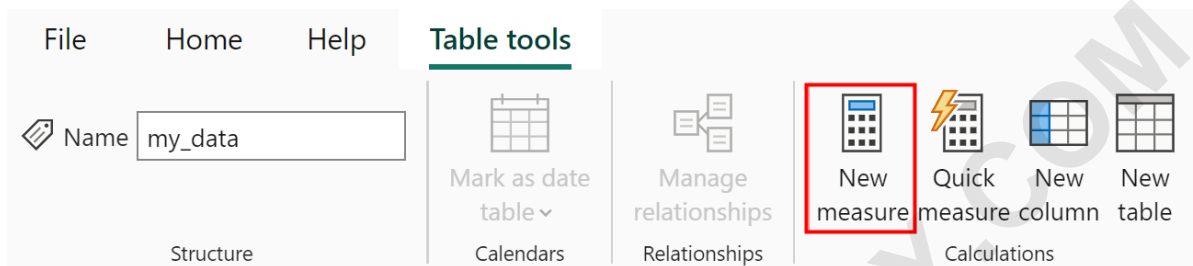
Our first goal is to define a new analytical metric that calculates the total points scored exclusively by players belonging to the "Mavs" team. This requires implementing Method 1 by creating a new

calculated measure.

Follow these procedural steps within the Power BI Desktop interface:

Navigate to the **Table tools** tab in the ribbon menu.

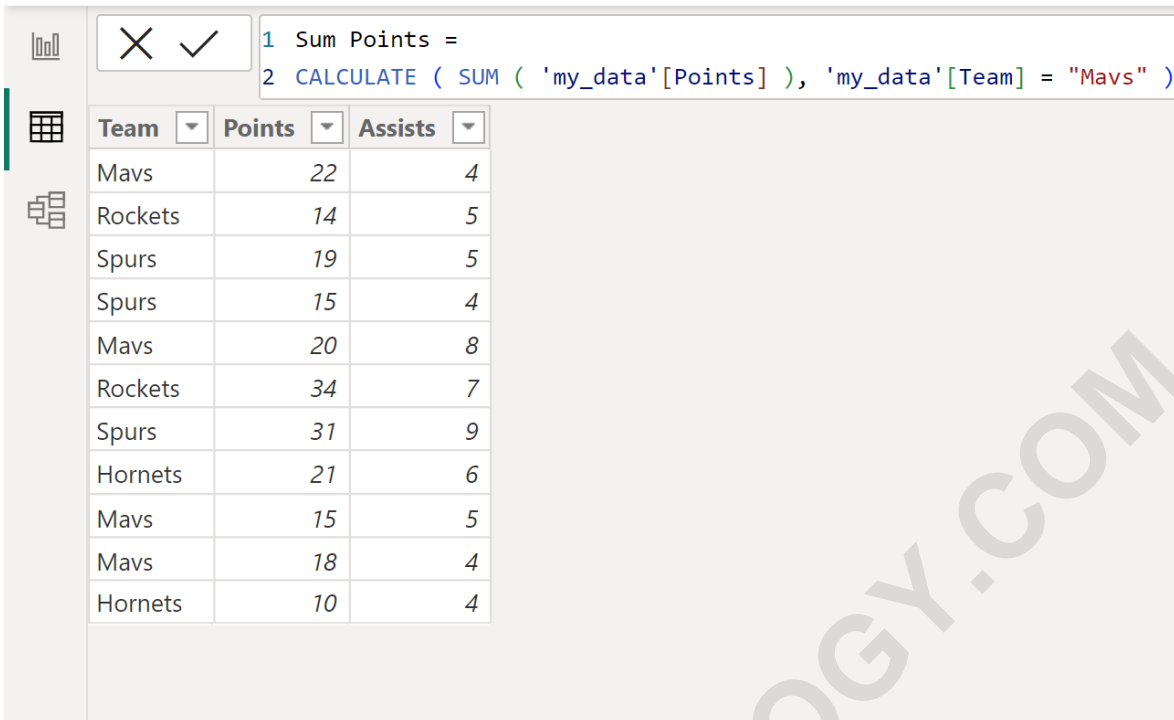
Click the **New measure** icon. This action opens the formula bar, allowing you to define the DAX expression for the new metric.



Next, input the formula that utilizes the CALCULATE and SUM functions, applying the filter criterion that specifies `'my_data' = "Mavs"`:

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

Upon committing the formula, a new measure called **Sum Points** is successfully created. It will reside within the data model and contains the calculated aggregate of points scored only by the players identified as being on the Mavs team. We can observe the measure appearing in the Fields pane, ready for use in visualizations.



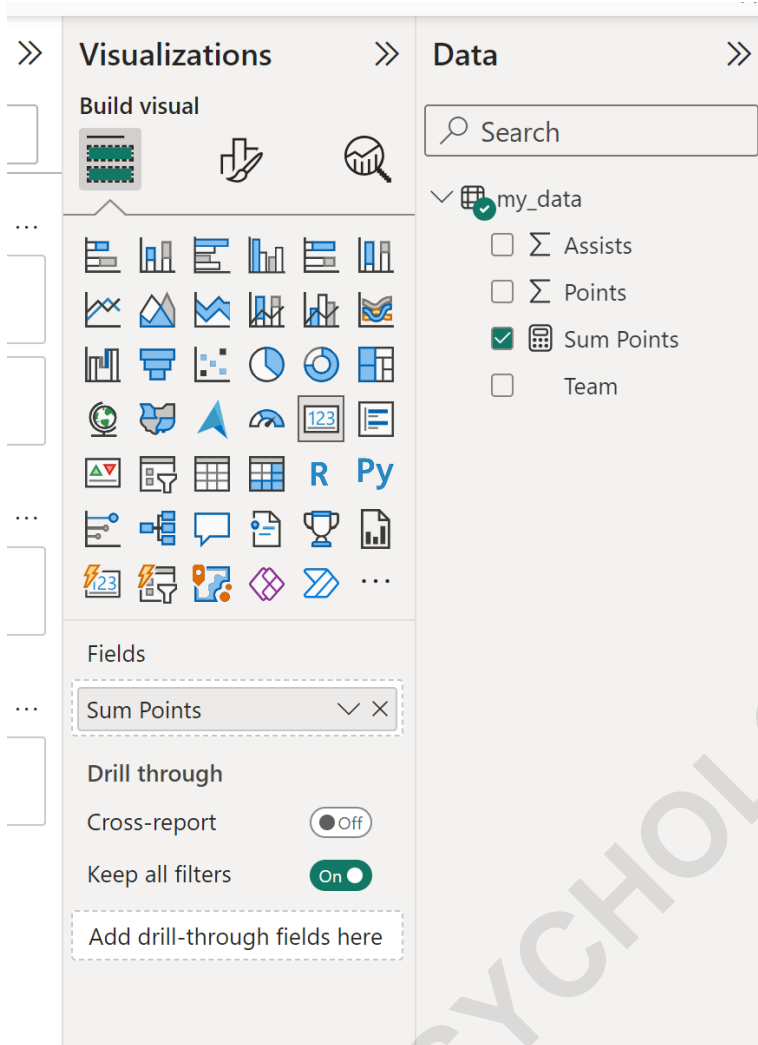
The screenshot shows the DAX editor in Power BI. The formula bar contains the following DAX code:

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

Below the formula bar, a data table is displayed with the following columns: Team, Points, and Assists. The table contains 12 rows of data:

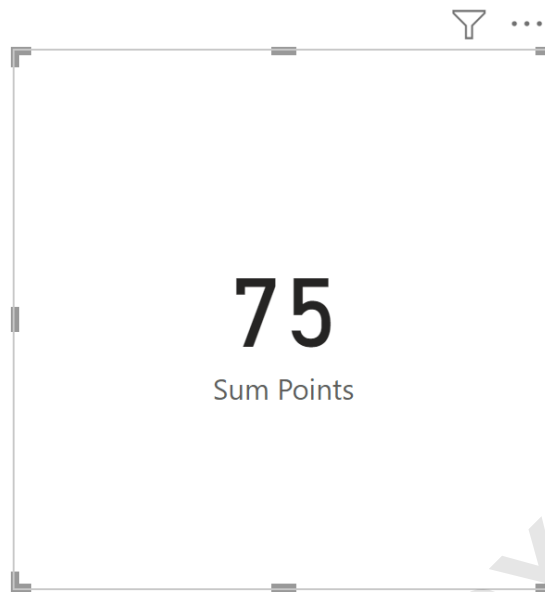
Team	Points	Assists
Mavs	22	4
Rockets	14	5
Spurs	19	5
Spurs	15	4
Mavs	20	8
Rockets	34	7
Spurs	31	9
Hornets	21	6
Mavs	15	5
Mavs	18	4
Hornets	10	4

To display this conditional sum, we can use a Card visual in the Report View. By dragging the newly created **Sum Points** measure onto the canvas and placing it into the Card's field well, the result of the calculation is rendered. The Card visual is ideal for displaying single aggregate values clearly.



The resulting visual confirmation demonstrates the effectiveness of the DAX expression, showing the filtered total:

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



As confirmed by the output, the players associated with the Mavs team scored a cumulative total of **75** points. This value is derived only from the rows that satisfied the single filter condition defined within the CALCULATE function.

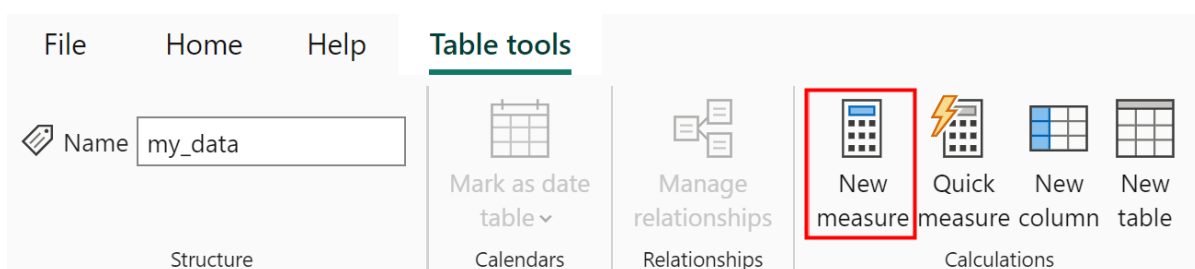
Example 2: Procedural Guide for Multiple Filters

Our second scenario increases complexity by requiring two distinct conditions to be met simultaneously: we want the total points scored only by players on the "Mavs" team **AND** who also recorded more than 4 assists. This demands the use of Method 2, leveraging multiple filter arguments within CALCULATE.

The implementation steps are nearly identical to the single-filter example, beginning with the creation of a new measure:

Click the **Table tools** tab.

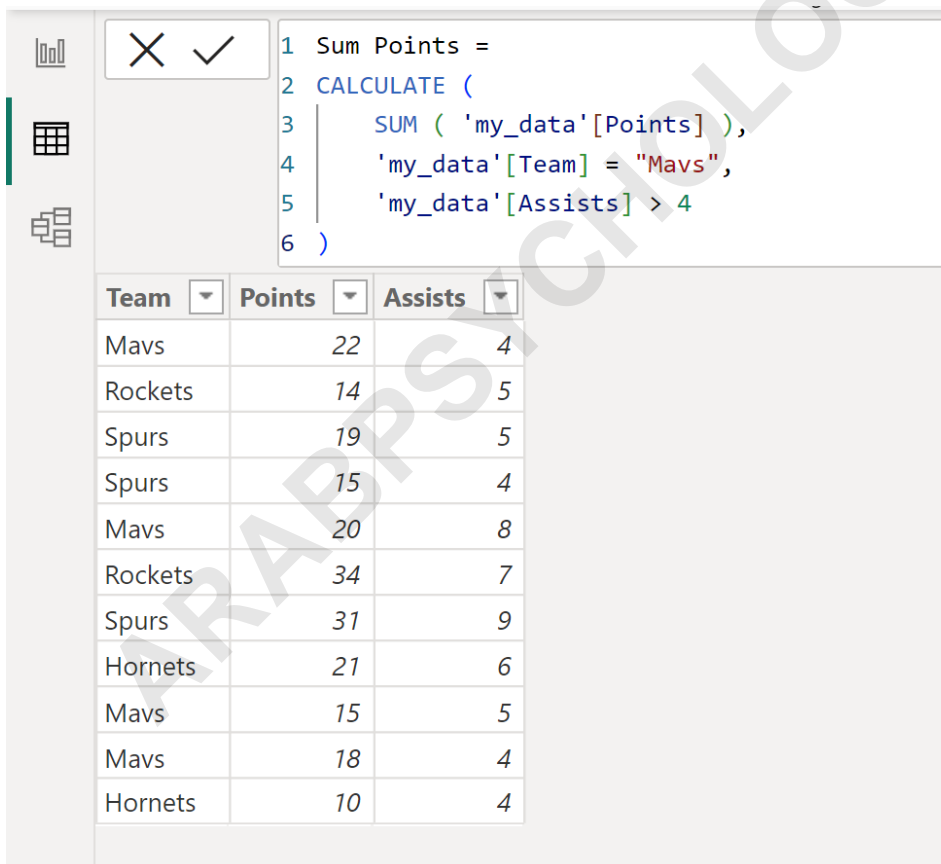
Select the **New measure** icon to open the DAX formula bar.



Now, enter the expanded DAX formula. Note the inclusion of the second condition, `'my_data' > 4`, separated by a comma. DAX interprets these multiple filters as an implicit AND operator:

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

After defining and committing the formula, the **Sum Points** measure is updated or created to reflect the new, narrower criteria. It now contains the sum of points only for the players who satisfy both the team requirement and the minimum assist threshold. The data model panel confirms the new measure's presence.



The screenshot shows the Power BI interface. At the top, the DAX formula for 'Sum Points' is entered in the formula bar:

```
1 Sum Points =  
2 CALCULATE (  
3     SUM ( 'my_data' [Points] ),  
4     'my_data' [Team] = "Mavs",  
5     'my_data' [Assists] > 4  
6 )
```

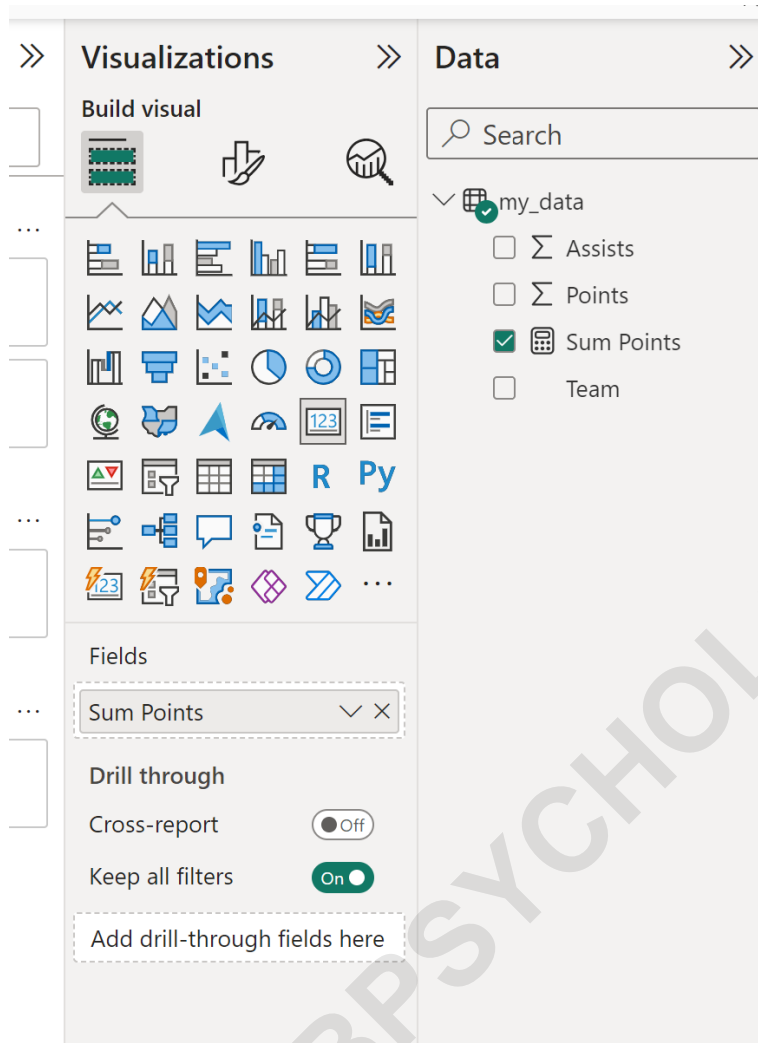
Below the formula bar, a table displays the results of the measure. The table has three columns: Team, Points, and Assists. The data is as follows:

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

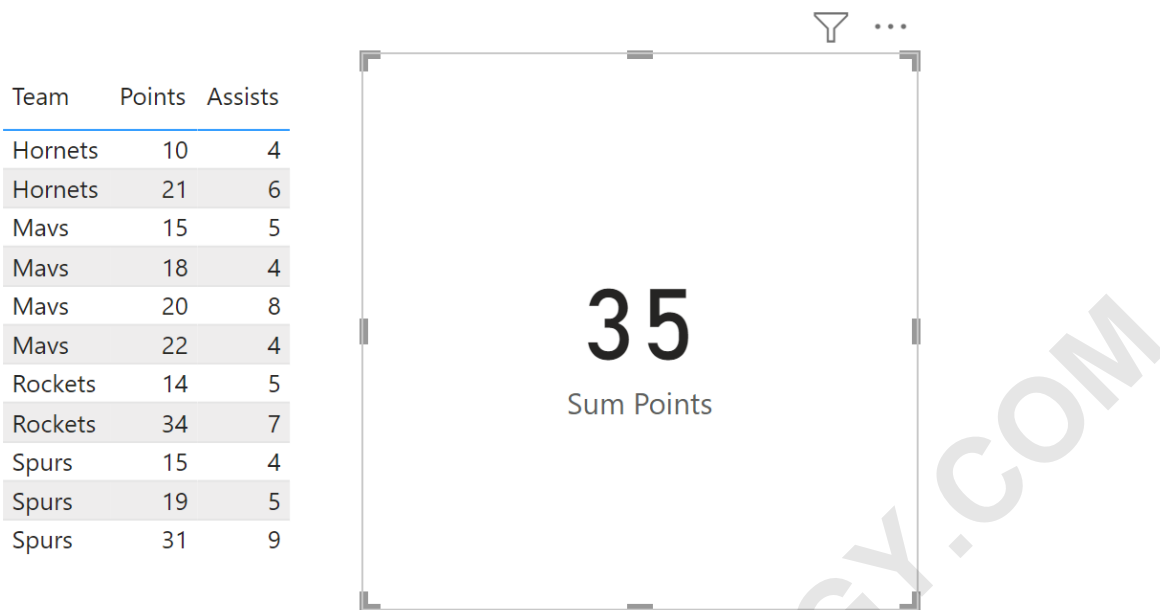
Visualizing and Interpreting the Multi-Filtered Result

To visualize this highly specific result, transition to the **Report View** in Power BI. Under the **Visualizations** tab, select the **Card** icon. Once the visual is on the canvas, drag the **Sum Points**

measure (which now contains the multi-filtered logic) into the **Fields** well of the Card visualization. This action triggers the evaluation of the DAX expression under the defined context.



The Card visual will then display the final aggregate value derived from the conditional summation. Since the filter is much stricter in this example, we expect the resulting total to be lower than the total calculated in Example 1, as fewer rows meet the criteria.



The resulting output clearly shows that the players on the Mavs team who registered more than 4 assists scored a combined total of only **35** points. This detailed calculation demonstrates the power of using the CALCULATE function to define and enforce complex analytical rules directly within a Power BI measure.

Further Advanced Applications of Filter Context

While we focused on simple equality and inequality filters, the principles demonstrated here extend to much more complex scenarios. The filter arguments within CALCULATE can include advanced DAX functions like `FILTER`, `ALL`, or `ALLEXCEPT`, allowing for dynamic modifications to the filter context, time intelligence calculations, and comparisons against total values. Mastering the core concept of filtering within the `SUM` expression is the foundational step toward building sophisticated business logic in Power BI.

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