Learning Power BI: Creating Measures with Multiple Filter Conditions Using DAX

Authored by

Mohammed looti

November 12, 2025

Mastering Multi-Condition Filtering in Power BI Measures

When performing advanced data analysis in Power BI, the ability to create dynamic calculations that respond to precise criteria is essential. This is achieved through measures, which utilize the powerful formula language, DAX (Data Analysis Expressions). Often, standard aggregations are insufficient; we require calculations that filter rows based on multiple simultaneous conditions--whether those conditions must all be met (AND logic) or if meeting any one of them is sufficient (OR logic).

The cornerstone of conditional calculation in DAX is the **CALCULATE** function. This function allows us to override or modify the standard Filter Context, enabling highly specific aggregations. By understanding how to properly structure multiple filter arguments within **CALCULATE** using logical operators, you can unlock sophisticated reporting capabilities.

The following sections detail the syntax and practical application for constructing measures that filter data using both the logical **AND** (`&&`) and **OR** (`||`) conditions, providing robust solutions for complex business intelligence scenarios.

Implementing Logical AND (&&) Conditions for Measures

To calculate a measure where all specified criteria must be true for a row to be included in the aggregation, we employ the logical **AND** operator, represented by the double ampersand (`&&`). This method ensures strict compliance with all filter requirements simultaneously.

The general syntax for creating a measure that filters rows based on multiple conditions using the **AND** operator is structured as follows. We are essentially modifying the filter context of an aggregation function, such as SUM, using the capabilities of the CALCULATE function.

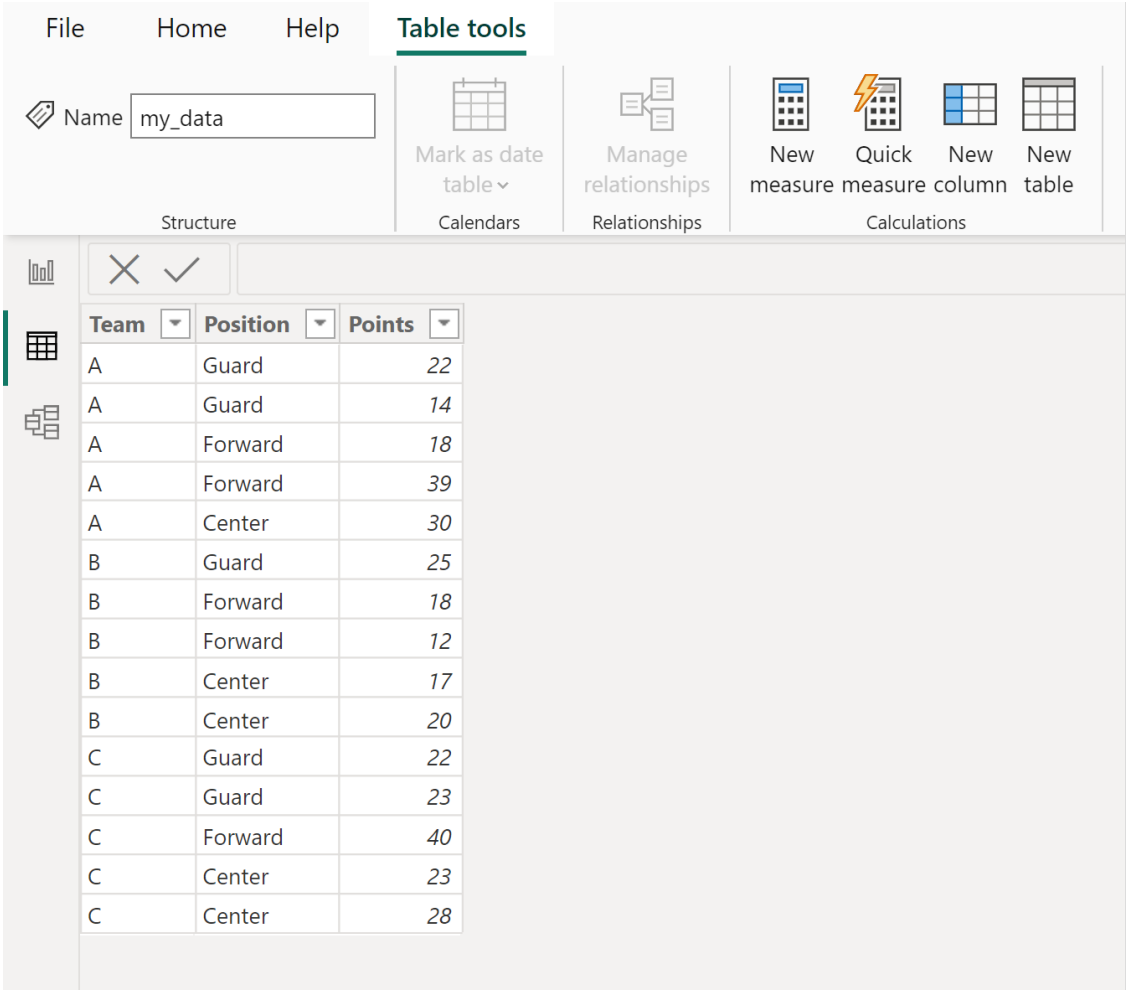
Method 1: Create Measure by Filtering with AND Condition

```
Sum of Points =
CALCULATE (
SUM ( 'my_data' ),
'my_data' = "A" && 'my_data' = "Guard"
)
```

In this specific formula, a new measure named **Sum of Points** is calculated. This calculation aggregates the values found in the **Points** column, but only for those rows where the value in the **Team** column is precisely "A" **AND** the value in the **Position** column is precisely "Guard." If a row satisfies only one of these conditions, it is excluded from the final sum, demonstrating the exclusive nature of the AND operator.

Step-by-Step Walkthrough: Using the AND Condition (Example 1)

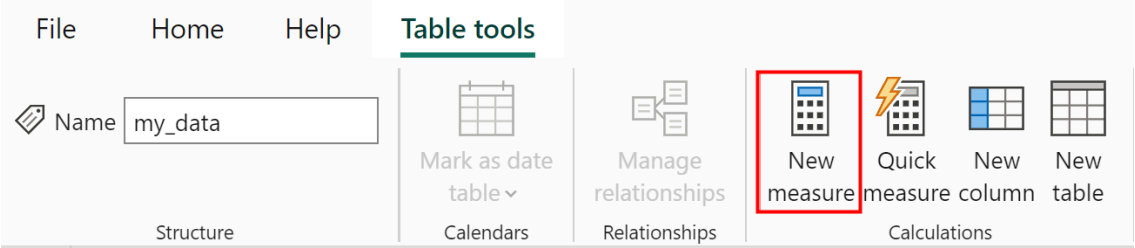
To demonstrate these concepts practically, consider a data table in Power BI named **my_data**, containing player statistics, including their team, position, and accrued points.



Team	Position	Points
A	Guard	22
A	Guard	14
A	Forward	18
A	Forward	39
A	Center	30
B	Guard	25
B	Forward	18
B	Forward	12
B	Center	17
B	Center	20
C	Guard	22
C	Guard	23
C	Forward	40
C	Center	23
C	Center	28

Suppose our requirement is to calculate the total points scored exclusively by players who belong to **Team A AND** whose **Position** is designated as Guard. This requires a precise, filtered aggregation.

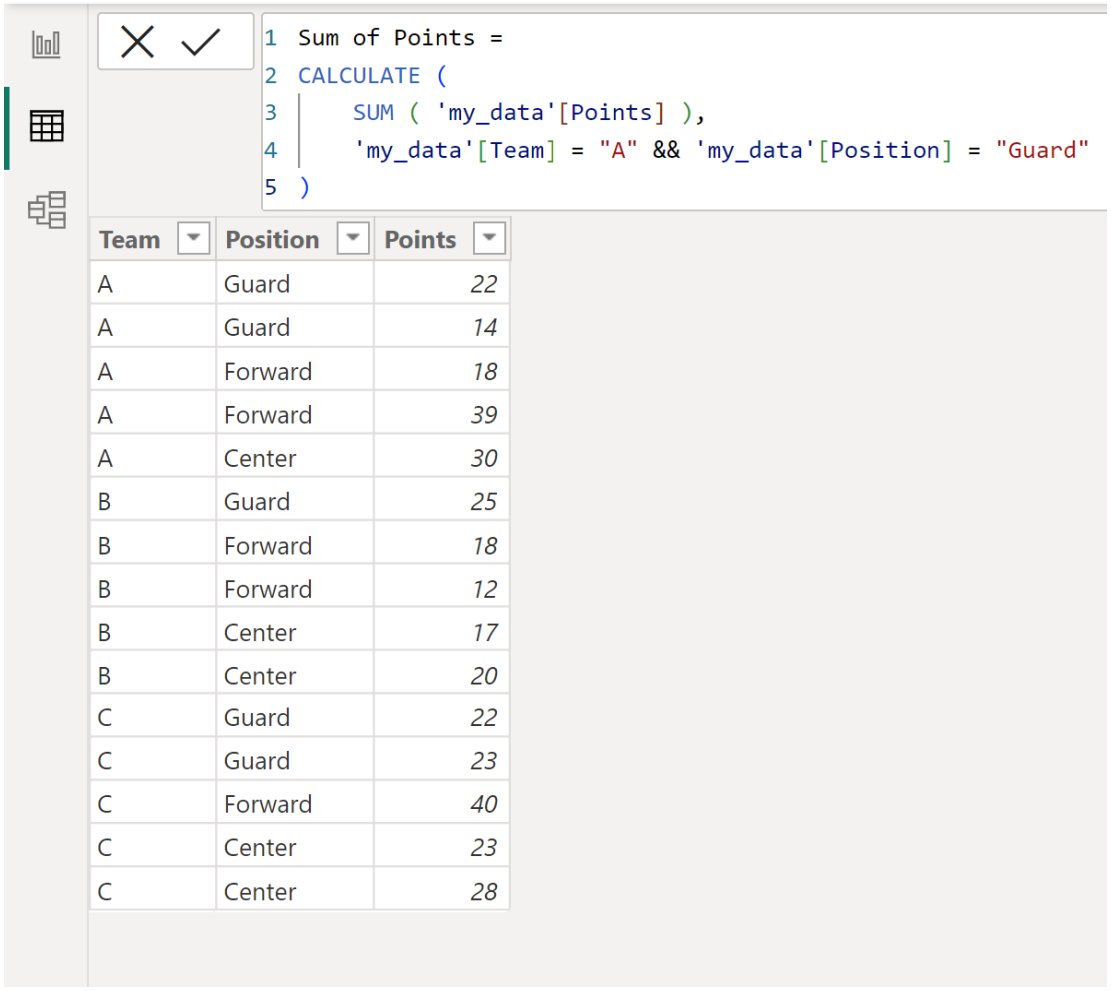
To begin creating this measure within Power BI Desktop, navigate to the **Table tools** tab in the ribbon and select the **New measure** icon. This action opens the formula bar, allowing you to define the DAX expression for the new calculation.



Paste or type the following **AND** formula into the formula bar:

```
Sum of Points =
CALCULATE (
SUM ( 'my_data' ),
'my_data' = "A" && 'my_data' = "Guard"
)
```

Once the measure is defined and committed, it appears in the Fields pane. The measure is now available for use in visualizations. The resulting filtered data set, based on the `&&` condition, isolates only the rows that satisfy both criteria simultaneously.

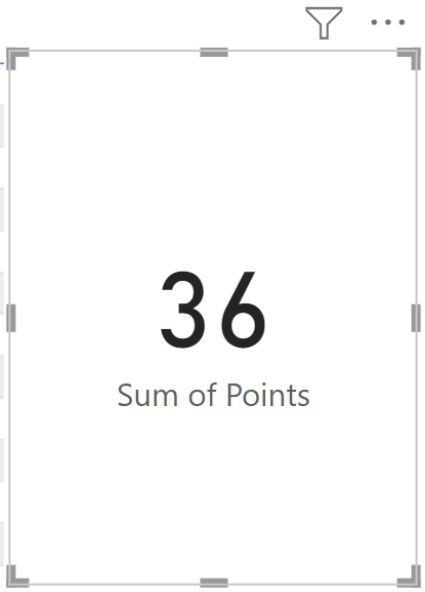


```
Sum of Points =
CALCULATE (
    SUM ( 'my_data'[Points] ),
    'my_data'[Team] = "A" && 'my_data'[Position] = "Guard"
)
```

Team	Position	Points
A	Guard	22
A	Guard	14
A	Forward	18
A	Forward	39
A	Center	30
B	Guard	25
B	Forward	18
B	Forward	12
B	Center	17
B	Center	20
C	Guard	22
C	Guard	23
C	Forward	40
C	Center	23
C	Center	28

To visually confirm the result, switch to the Report View and insert a Card visualization. Drag the newly created **Sum of Points** measure onto the card.

Team	Position	Points
A	Center	30
A	Forward	18
A	Forward	39
A	Guard	14
A	Guard	22
B	Center	17
B	Center	20
B	Forward	12
B	Forward	18
B	Guard	25
C	Center	23
C	Center	28
C	Forward	40
C	Guard	22
C	Guard	23



The resulting calculation confirms that the sum of points for players who are both on Team A and hold the Position of Guard is **36**. This demonstrates the precision achieved by combining filters using the logical **AND** operator in DAX.

Implementing Logical OR (||) Conditions for Measures

In contrast to the strict requirements of the **AND** condition, the logical **OR** operator, represented by the double pipe symbol (`||`), is used when a row should be included in the calculation if it satisfies **any one** of the defined filter conditions. This approach is necessary when calculating totals for members belonging to one group **or** another group.

Using **OR** significantly expands the data set included in the measure, as only one condition needs to be true. This is particularly useful for analyzing groups that may overlap or defining broad categories of interest.

Method 2: Create Measure by Filtering with OR Condition

```
Sum of Points =
CALCULATE (
SUM ( 'my_data' ),
'my_data' = "A" || 'my_data' = "Guard"
```

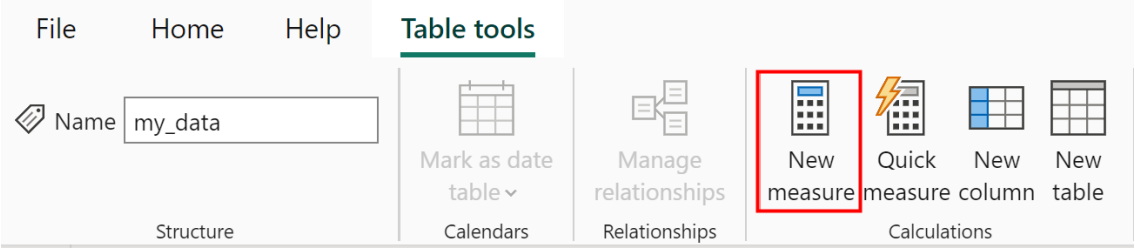
)

Here, the measure **Sum of Points** calculates the total points for any row where the **Team** column equals "A" **OR** the **Position** column equals "Guard." If a player is on Team B but is a Guard, they are included. If a player is on Team A but is a Forward, they are also included. This inclusive logic is key to understanding the difference between `&&` and `||`.

Step-by-Step Walkthrough: Using the OR Condition (Example 2)

Let us now apply the OR logic to the same **my_data** table. Our goal is to calculate the sum of values in the **Points** column for all players who are either members of **Team A OR** hold the **Position** of Guard, regardless of their team.

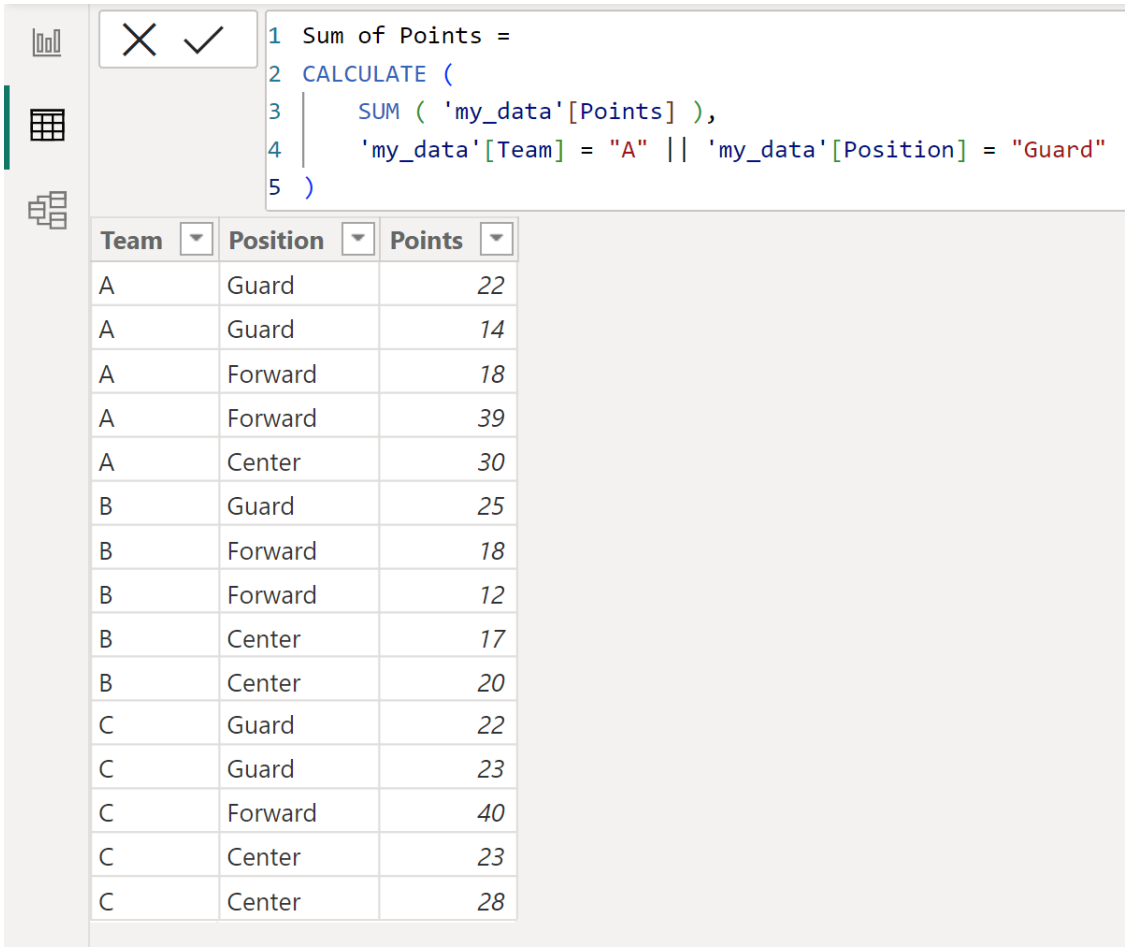
As before, navigate to the **Table tools** tab and click the **New measure** icon to open the DAX formula editor.



Enter the following formula, utilizing the logical **OR** operator (`||`):

```
Sum of Points =
CALCULATE (
SUM ( 'my_data' ),
'my_data' = "A" || 'my_data' = "Guard"
)
```

By implementing this formula, we capture a much broader set of data points compared to the previous example, as the calculation is satisfied as long as one of the two conditions is met.

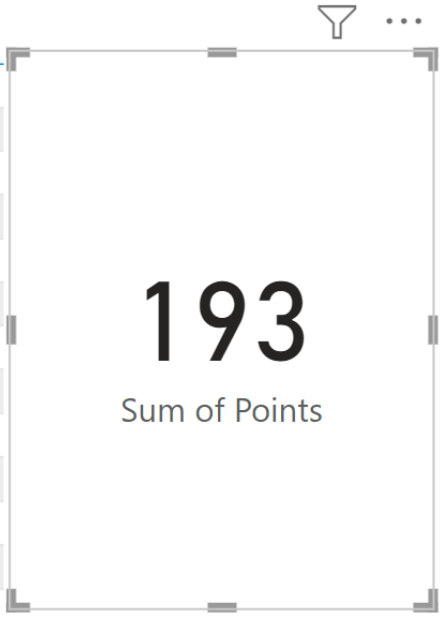


```
Sum of Points =
CALCULATE (
    SUM ( 'my_data'[Points] ),
    'my_data'[Team] = "A" || 'my_data'[Position] = "Guard"
)
```

Team	Position	Points
A	Guard	22
A	Guard	14
A	Forward	18
A	Forward	39
A	Center	30
B	Guard	25
B	Forward	18
B	Forward	12
B	Center	17
B	Center	20
C	Guard	22
C	Guard	23
C	Forward	40
C	Center	23
C	Center	28

To visualize the result, return to the Report View and insert a new Card visualization displaying the value of this new measure.

Team	Position	Points
A	Center	30
A	Forward	18
A	Forward	39
A	Guard	14
A	Guard	22
B	Center	17
B	Center	20
B	Forward	12
B	Forward	18
B	Guard	25
C	Center	23
C	Center	28
C	Forward	40
C	Guard	22
C	Guard	23



As expected, the result is significantly higher: the sum of points for players who are on Team A **or** have a Position of Guard is **193**. This dramatic increase reflects the inclusion of all rows that satisfied either condition independently.

Conclusion and Advanced Filter Management

Effectively utilizing **CALCULATE** with multiple conditions is a fundamental skill in Power BI development. Whether requiring the strict intersection of data using **AND** (`&&`) or the broad union of data using **OR** (`||`), DAX provides intuitive syntax to manage the Filter Context precisely. While direct use of `&&` and `||` is straightforward for simple column filtering, it is worth noting that for more complex scenarios involving related tables or expressions, advanced functions like **FILTER** might be required, though the logic remains fundamentally based on these Boolean operators.

Understanding these core filtering techniques ensures that your measures accurately reflect the specific business logic required for insightful reporting and analysis.

Additional Resources

The following tutorials explain how to perform other common tasks in Power BI, helping you further expand your DAX expertise:

Official Microsoft documentation on the CALCULATE function.
Tutorials detailing the use of the **SUM** function within conditional measures.
Guides for managing Row Context vs. Filter Context in DAX.