

# How to Create a Power BI Measure with Multiple Filters

Authored by  
**stats writer**

January 27, 2026

## RECOMMENDED CITATION

stats writer (2026). *How to Create a Power BI Measure with Multiple Filters*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=128080>

Creating a measure in Power BI that incorporates multiple filter conditions is a fundamental skill for advanced data analysis. This capability empowers users to segment and analyze data based on highly specific, compound criteria, moving beyond simple aggregations to derive truly meaningful insights. By leveraging the power of the DAX language, analysts can define precise formulas that determine which rows of data are included in a calculation, effectively creating custom contexts for visualization.

The ability to apply complex filtering logic--whether through intersection (AND) or union (OR) conditions--is vital for organizations seeking to make data-driven decisions. Complex measures allow businesses to narrow down vast datasets, isolating performance indicators for specific regions, products, or user segments. Understanding how to structure these multi-conditional filters ensures that visualizations accurately reflect the required business logic, providing clarity and depth to the resulting data story within the Power BI environment.

## Power BI: Create Measure with Multiple Filter Conditions

### Decoding Measures, DAX, and the CALCULATE Function

To successfully implement multi-conditional filtering in Power BI, we must rely on the Data Analysis Expressions (DAX) language. Measures are dynamic calculation formulas that are evaluated based on the context of the report visualizations, unlike calculated columns which are computed row-by-row and stored in the model. The cornerstone of virtually all complex measure creation is the CALCULATE function.

The CALCULATE function is arguably the most powerful function in DAX, as it modifies the context in which a calculation is performed. It takes two main arguments: the expression (the calculation to be performed, such as SUM or AVERAGE) and one or more filter arguments. When multiple filter arguments are provided to CALCULATE, they are combined implicitly using the logical AND operator, unless explicitly joined by an OR operator within a single filter argument.

For scenarios requiring multiple conditions within a single filter argument--which is necessary when those conditions must interact logically--we utilize standard logical operators. The two primary operators used for this purpose are the logical AND (`&&`) and the logical OR (`||`). These operators allow us to construct complex Boolean expressions that govern exactly which rows are included in the measure's calculation, ensuring precision in analytical results.

### Applying Logical Operators for Complex Filtering

When filtering data, the choice between the AND operator and the OR operator dictates the resulting subset of data. Understanding the difference between these logical operations is crucial for accurate measure creation. The AND operator requires that **all** specified conditions be true for

a row to be included in the calculation. This results in the intersection of the two filtered datasets, leading to a smaller, more specific result set.

Conversely, the OR operator requires that **at least one** of the specified conditions be true. This results in the union of the two filtered datasets. If a row satisfies condition A or condition B (or both), it is included in the calculation. This generally yields a larger result set compared to the AND operator, as it broadens the criteria for inclusion. Using these operators correctly within the filter context of the CALCULATE function is the key to mastering multi-conditional measures.

The following sections demonstrate the exact DAX syntax required for both intersection (AND) and union (OR) filtering, using the foundational CALCULATE function structure to define the measure.

You can use the following syntax in DAX to create a measure that filters rows based on multiple conditions:

### Method 1: Using the AND Operator ( `&&` ) for Intersection Filters

The AND operator ( `&&` ) is used when you need to ensure that a record meets several criteria simultaneously. When used within the filter argument of the CALCULATE function, it acts as a gatekeeper, only allowing rows that pass all listed tests to contribute to the final aggregated result. This method is perfect for precise targeting, such as calculating sales only for "Product X" in the "Eastern Region."

#### Method 1: Create Measure by Filtering with AND Condition

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

This particular example creates a new measure named **Sum of Points** that calculates the sum of the values in the **Points** column only for the rows where the **Team** column is equal to "A" **and** the **Position** column is equal to "Guard." This filter configuration requires both conditions to be satisfied concurrently. If a player is on Team A but is a Forward, or if they are a Guard but on Team B, they will be excluded from the summation.

### Method 2: Using the OR Operator ( `||` ) for Union Filters

The OR operator ( `||` ) is utilized when you want to include records that satisfy any one of multiple criteria. This broadens the scope of the calculation, including rows that meet condition A, condition

B, or both conditions simultaneously. This approach is valuable when aggregating data across disparate, non-mutually exclusive segments, such as calculating total activity for users who are either "Premium Subscribers" or "Frequent Users."

## Method 2: Create Measure by Filtering with OR Condition

**Sum of Points =**

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

This particular example creates a new measure named **Sum of Points** that calculates the sum of the values in the **Points** column only for the rows where the **Team** column is equal to "A" **or** the **Position** column is equal to "Guard." This union logic means that any row belonging to Team A is counted, and any row designating the position as Guard is also counted. If a row satisfies both criteria, it is included just once in the sum.

## Setting Up the Practical Example Dataset

To illustrate the practical effects of these logical operators, we will use a simple dataset modeled in Power BI. This dataset, named **my\_data**, tracks player statistics, including their team assignment, position, and points scored. This clear, tabular format makes it easy to manually verify the results generated by the DAX measures, thus confirming the accuracy of the filtering logic.

The structure of the data includes columns for **Team** (A or B), **Position** (Guard, Forward, Center), and **Points** (numerical values). By applying our multi-conditional filters to this table, we can clearly observe how the AND operator restricts the results while the OR operator expands them, allowing us to isolate specific player groups for detailed performance analysis.

The following examples show how to use each method in practice with the following table in Power BI named **my\_data**:

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

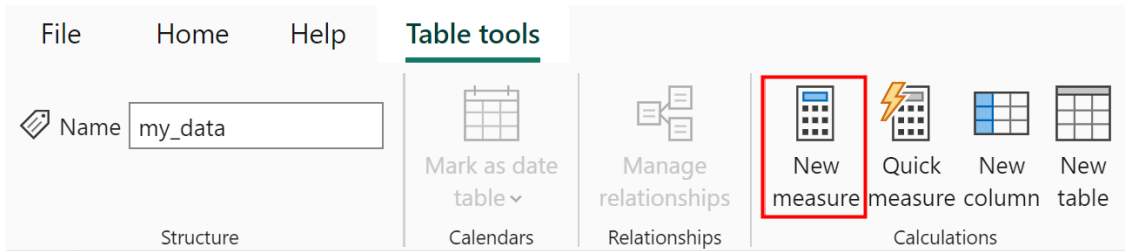
## Practical Walkthrough: Filtering with the AND Condition

In this first scenario, we aim to calculate a total based on an intersection of two conditions. We are specifically interested in the combined score of players who are both members of **Team A** and hold the **Guard** position. This high degree of specificity is essential for targeted performance evaluation, ensuring only players meeting both criteria are included in the aggregation.

Suppose we would like to calculate the sum of values in the **Points** column for the players who are on **Team A** and have a **Position** of Guard.

To implement this measure, navigate to the data modeling interface in Power BI Desktop. Ensure that the 'my\_data' table is selected. You will then initiate the creation of a new measure by using the dedicated tools, preparing the environment for the DAX formula input.

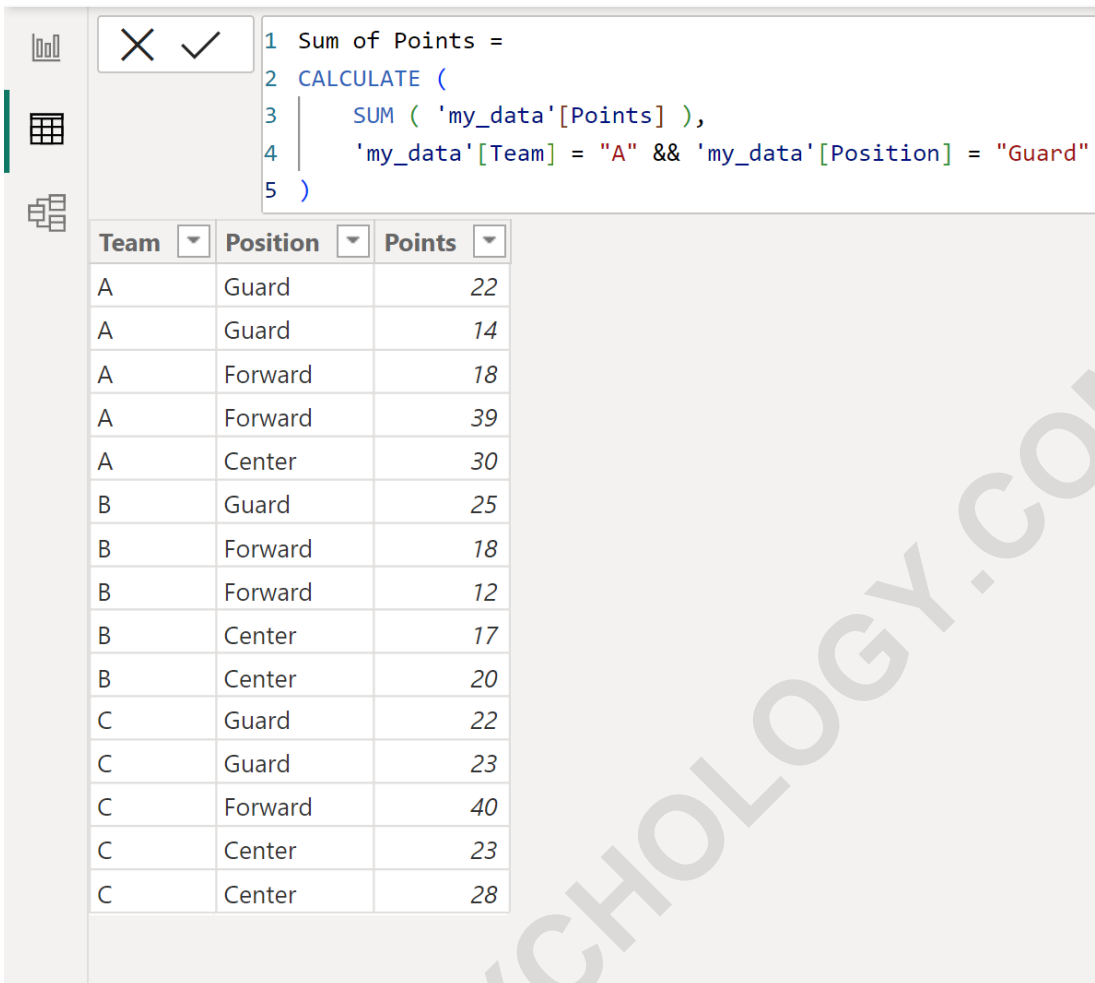
To do so, click the **Table tools** tab and then click the **New measure** icon:



Next, input the DAX formula defined earlier into the formula bar. This formula uses the CALCULATE function to wrap the aggregation (SUM of Points) and applies the two filtering conditions joined by the `&&` operator. Remember that proper syntax, including quoted text values and correct column referencing, is mandatory for the measure to validate successfully.

Then type the following formula into the formula bar:

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



The screenshot shows the Power BI DAX editor interface. At the top, there is a formula bar with a 'X' and a checkmark icon. Below it, the DAX formula for the measure 'Sum of Points' is displayed:

```
1 Sum of Points =  
2 CALCULATE (  
3     SUM ( 'my_data'[Points] ),  
4     'my_data'[Team] = "A" && 'my_data'[Position] = "Guard"  
5 )
```

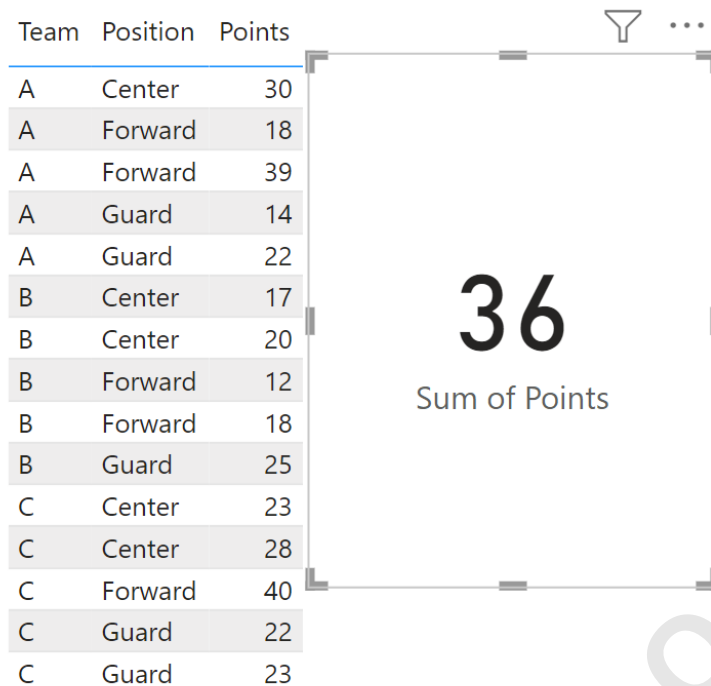
Below the formula bar, a table is displayed with three columns: Team, Position, and Points. The table contains 15 rows of data:

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

Once the measure is created, switch to the Report View. To display the scalar result of the new measure clearly, we insert a Card visualization. Drag the newly created **Sum of Points** measure onto this visualization. The card will dynamically evaluate the formula based on the filters applied within the `CALCULATE` function, providing the precise sum for the intersection of Team A and Guard position players.

We can view this measure by switching to the Report View and inserting a card visualization that displays the value of the 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



By reviewing the source data, we can confirm the result. Only players who satisfy both criteria are Player ID 1 (20 points) and Player ID 6 (16 points). Therefore, the sum of points for the players who are on Team A and have a Position of Guard is **36**. This demonstrates the restrictive power of the AND operator in refining analytical scope.

We can see that the sum of points for the players who are on Team A and have a Position of Guard is **36**.

### Practical Walkthrough: Filtering with the OR Condition

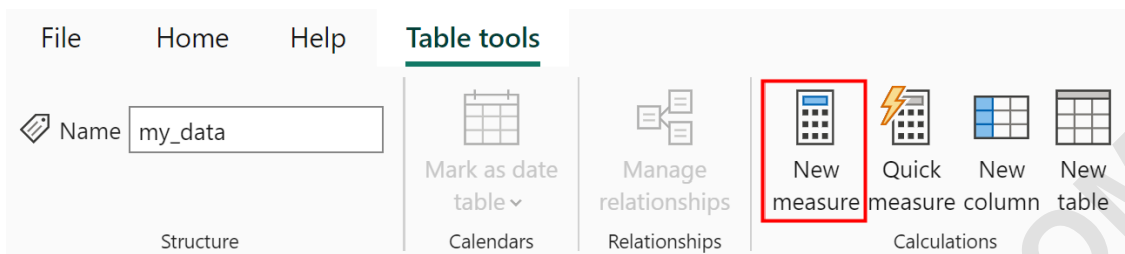
The second scenario explores the union of two conditions using the OR operator. Here, the goal is to calculate the total points scored by any player who meets at least one of the specified conditions: either they are on **Team A**, or they hold the **Guard** position. This calculation provides a broader aggregate, encompassing all players relevant to either category, making it suitable for assessing total group contributions.

Suppose we would like to calculate the sum of values in the **Points** column for the players who are on **Team A** or have a **Position** of Guard.

The process for creating this measure follows the same initial steps: selecting the 'my\_data' table and clicking the 'New measure' icon under the Table tools. The difference lies entirely within the

DAX formula itself, where we substitute the AND operator with the OR operator to change the logical criteria.

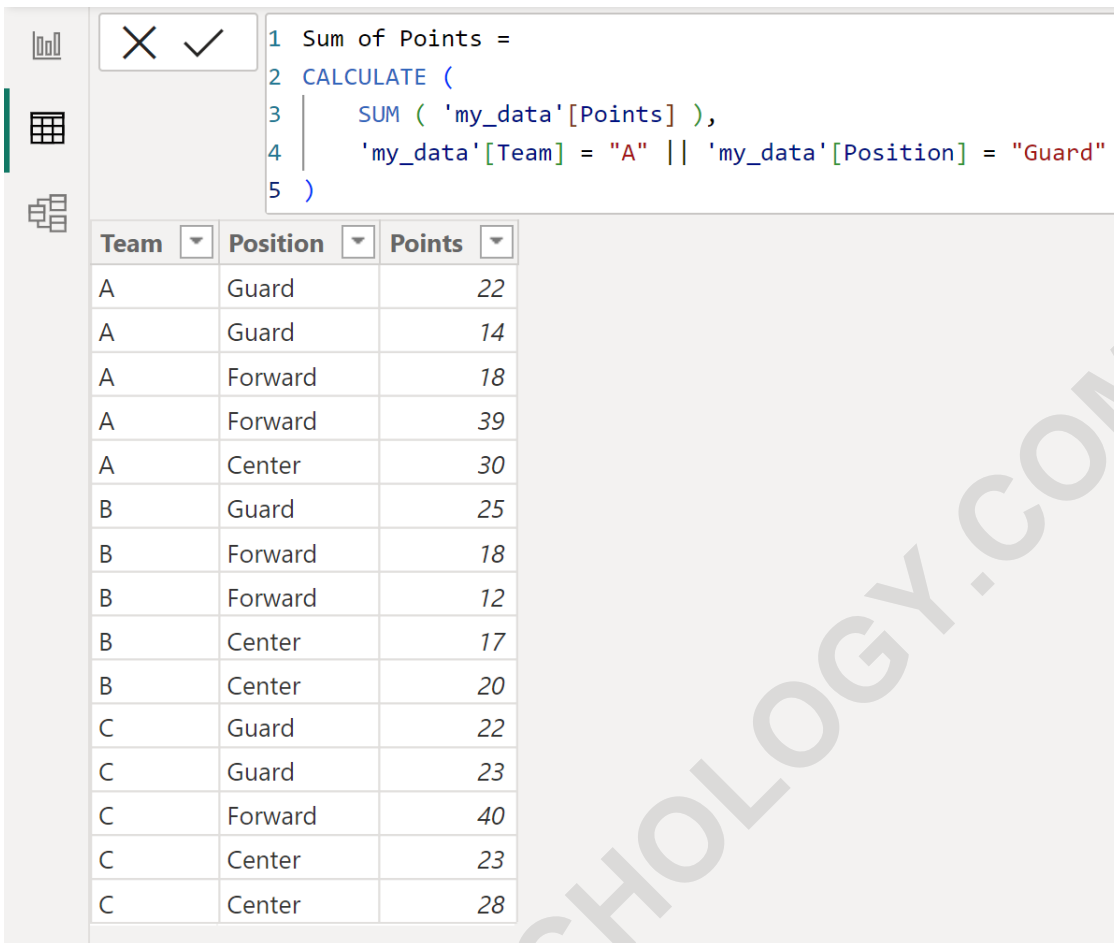
To do so, click the **Table tools** tab and then click the **New measure** icon:



Input the formula below, which utilizes the logical OR operator ('||'). This operator mandates that if a row satisfies the condition 'Team = A' or the condition 'Position = Guard', that row's point total will be included in the overall sum. This significantly increases the number of rows contributing to the final result compared to the previous example.

Then type the following formula into the formula bar:

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



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

```
1 Sum of Points =  
2 CALCULATE (  
3     SUM ( 'my_data'[Points] ),  
4     'my_data'[Team] = "A" || 'my_data'[Position] = "Guard"  
5 )
```

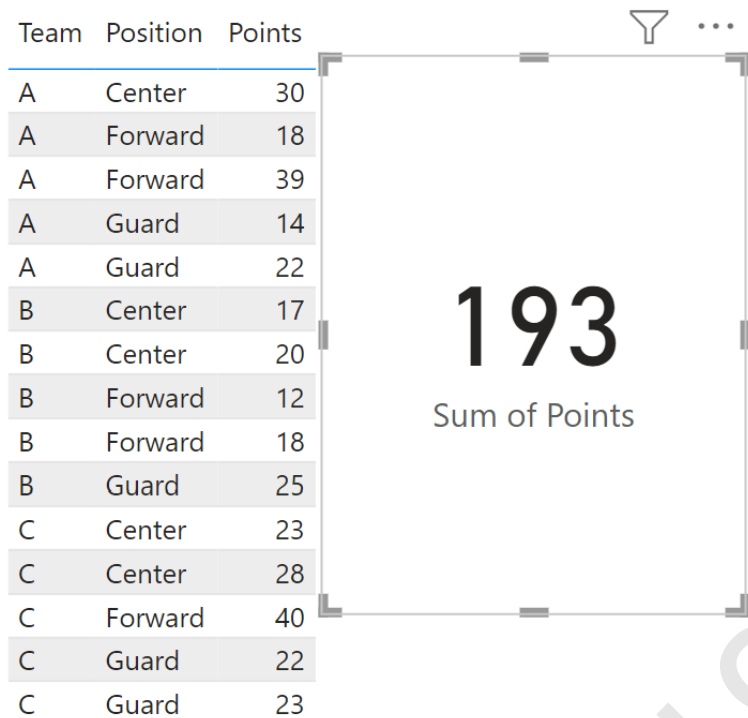
Below the formula bar, a table is displayed with the following data:

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

After defining the measure, return to the Report View and use a Card visualization to display the result. When the OR logic is applied, the measure aggregates the points from all players belonging to Team A (regardless of position) plus any players designated as Guards (regardless of team). This visualization immediately confirms the measure's broader scope, showing a much higher total than the AND calculation.

We can view this measure by switching to the Report View and inserting a card visualization that displays the value of the 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



Verification against the dataset reveals that the included rows are all players on Team A (IDs 1, 2, 6, 7, 10, 11, 14, 15) plus any Guards not on Team A (ID 4). Summing these point totals confirms that the total points for the players who are on Team A or have a Position of Guard is **193**. This result effectively illustrates how the OR operator is used to calculate the union of two distinct segments.

We can see that the sum of points for the players who are on Team A or have a Position of Guard is **193**.

## Summary and Further Steps in Power BI Analysis

Mastering the use of logical operators--AND (`&&`) and OR (`||`)--within the CALCULATE function is essential for creating powerful, custom measures in Power BI. These techniques allow analysts to precisely control the filter context, ensuring that aggregations reflect complex, real-world business rules. The ability to switch between restrictive intersection filters and expansive union filters provides the flexibility required for deep analytical exploration.

As you continue your journey in Power BI and DAX, consider exploring how to nest multiple CALCULATE functions or use filtering functions like FILTER and ALLEXCEPT to handle even more intricate scenarios, such as creating measures based on time intelligence or evaluating performance against dynamic benchmarks. These foundational skills in multi-conditional filtering

pave the way for highly sophisticated data modeling.

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

ARABPSYCHOLOGY.COM