

How to Create a Power BI Measure with Multiple Filter Conditions

Authored by
mohammed loot

January 11, 2026

RECOMMENDED CITATION

mohammed loot (2026). *How to Create a Power BI Measure with Multiple Filter Conditions*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=125481>

Creating a robust Measure in Power BI often requires more than just a simple aggregation; it demands precision through complex filtering. When analyzing specific subsets of data--such as sales figures from a particular region sold during a certain quarter, or performance metrics for a specific team member holding a particular role--you need the ability to combine multiple criteria into a single calculation. This advanced filtering mechanism is achieved using DAX (Data Analysis Expressions), the functional language native to Power BI and tabular models.

The core strategy involves utilizing the powerful CALCULATE function, which is the cornerstone for modifying the evaluation context in DAX. By default, measures are evaluated within the existing filter context defined by your report visuals (e.g., slicers or row/column headers). However, CALCULATE allows you to override or add new filter conditions internally. When introducing multiple filter conditions, we use logical operators--specifically, the logical AND (`&&`) or the logical OR (`||`)--to define precisely which rows should be included in the calculation before the aggregation takes place. This approach ensures that your measures provide highly targeted and accurate insights, crucial for advanced business intelligence reporting.

The Role of DAX and Filter Context Modification

The fundamental challenge in defining a precise measure lies in managing the filter context. In simple terms, the filter context dictates which rows of data are visible to the measure calculation at any given moment. To apply multiple conditions, we leverage the CALCULATE function. CALCULATE takes an expression (like SUM or AVERAGEX) as its first argument, and then accepts one or more filter arguments that redefine the context for that calculation only. When multiple filter arguments are passed to CALCULATE, they are inherently treated as an implicit AND condition, meaning all conditions must be true for a row to be included. However, when we want to create complex internal logic that goes beyond simple column comparisons, or implement an OR condition, we must explicitly use logical operators within a single filter argument.

When formulating these complex measures, clear definition is key. You must first identify the raw aggregation you wish to perform (e.g., summing the 'Points' column) and then define the exact criteria that must be met simultaneously. Using embedded logical operators within the CALCULATE function provides a clean, readable, and highly performant way to achieve this targeted filtering. The use of logical operators (`&&` for AND, `||` for OR) allows for efficient row-level evaluation, ensuring that the function returns the desired result set before the aggregation function, such as SUM, is applied.

Implementing Multiple Filter Conditions in DAX Syntax

To construct a measure that filters rows based on several criteria, you will apply the following syntax patterns within DAX. These methods rely on combining filtering expressions using logical

operators inside the CALCULATE function's filter argument. Understanding the distinction between the AND and OR conditions is critical, as they fundamentally alter which rows contribute to the final result. The logical AND requires simultaneous fulfillment of all conditions, while the logical OR requires only one condition to be true.

Method 1: Filtering with the Logical AND Operator (`&&`)

The logical AND operator (`&&`) is used when you need a row to satisfy all specified conditions concurrently. For instance, if you are calculating total sales for "Product A" sold in "Region B," both conditions must be true for the sale to be included. This is the most common use case for targeted metric calculation, ensuring high specificity.

The following syntax demonstrates how to structure a Measure using the AND condition. The calculation will aggregate the values only in rows where the 'Team' column equals "A" **AND** the 'Position' column equals "Guard."

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

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 precise formula ensures that only records meeting both criteria simultaneously contribute to the sum, providing a highly filtered aggregate result.

Method 2: Filtering with the Logical OR Operator (`||`)

The logical OR operator (`||`) is employed when a row should be included in the calculation if it satisfies at least one of the defined conditions. For example, if you want to calculate total inventory for items that are either "Out of Stock" **OR** have been "Discontinued," the OR condition expands the resulting data set. This is useful for grouping related but non-exclusive data points into a single metric.

The following syntax illustrates how to construct the same measure using the OR condition. The

calculation will aggregate the values in rows where the 'Team' column equals "A" **OR** the 'Position' column equals "Guard."

Method 2: Create Measure by Filtering with OR Condition

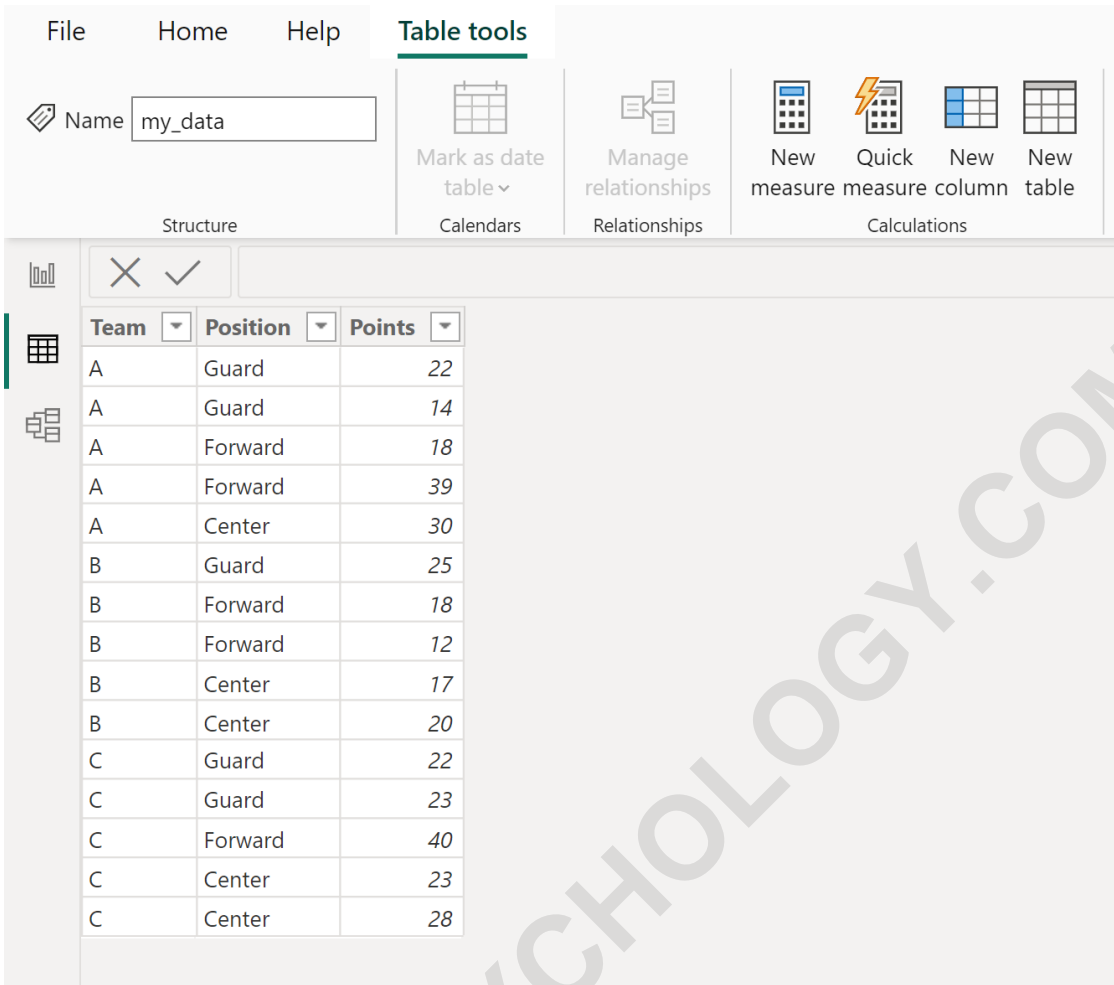
```
Sum of Points =  
CALCULATE (  
SUM ( 'my_data' ),  
'my_data' = "A" || 'my_data' = "Guard"  
)
```

This implementation creates a new measure named **Sum of Points** that calculates the sum of the values in the **Points** column for any row that satisfies either condition: the **Team** column is equal to "A" **or** the **Position** column is equal to "Guard." This significantly broadens the scope of the calculation compared to the AND condition, capturing records that meet either criterion.

Prerequisite Data Table for Examples

To clearly demonstrate the practical difference between the AND and OR conditions, we will use a sample dataset named **my_data** loaded into Power BI. This table contains player statistics, including their team, position, and the points scored. Pay close attention to how the application of the logical operators dramatically changes the resulting aggregated sum based on which rows are permitted to pass the filter context.

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



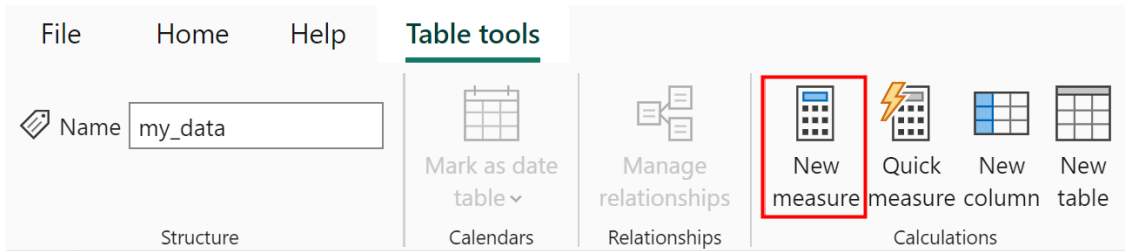
The screenshot shows the Power BI Desktop interface with the 'Table tools' ribbon selected. The ribbon includes options for 'Mark as date table', 'Manage relationships', and 'Calculations'. Under 'Calculations', the 'New measure' icon is highlighted. Below the ribbon, a data table is displayed with the 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 |

Example 1: Practical Application of the AND Condition

For our first practical demonstration, suppose we require a highly specific metric: the total sum of points scored only by players who belong to **Team A AND** whose **Position** is Guard. This calculation demands that both columns match the specified text values. We anticipate a smaller, highly targeted result set, as only the rows that satisfy the intersection of these two criteria will be aggregated by the SUM function within CALCULATE.

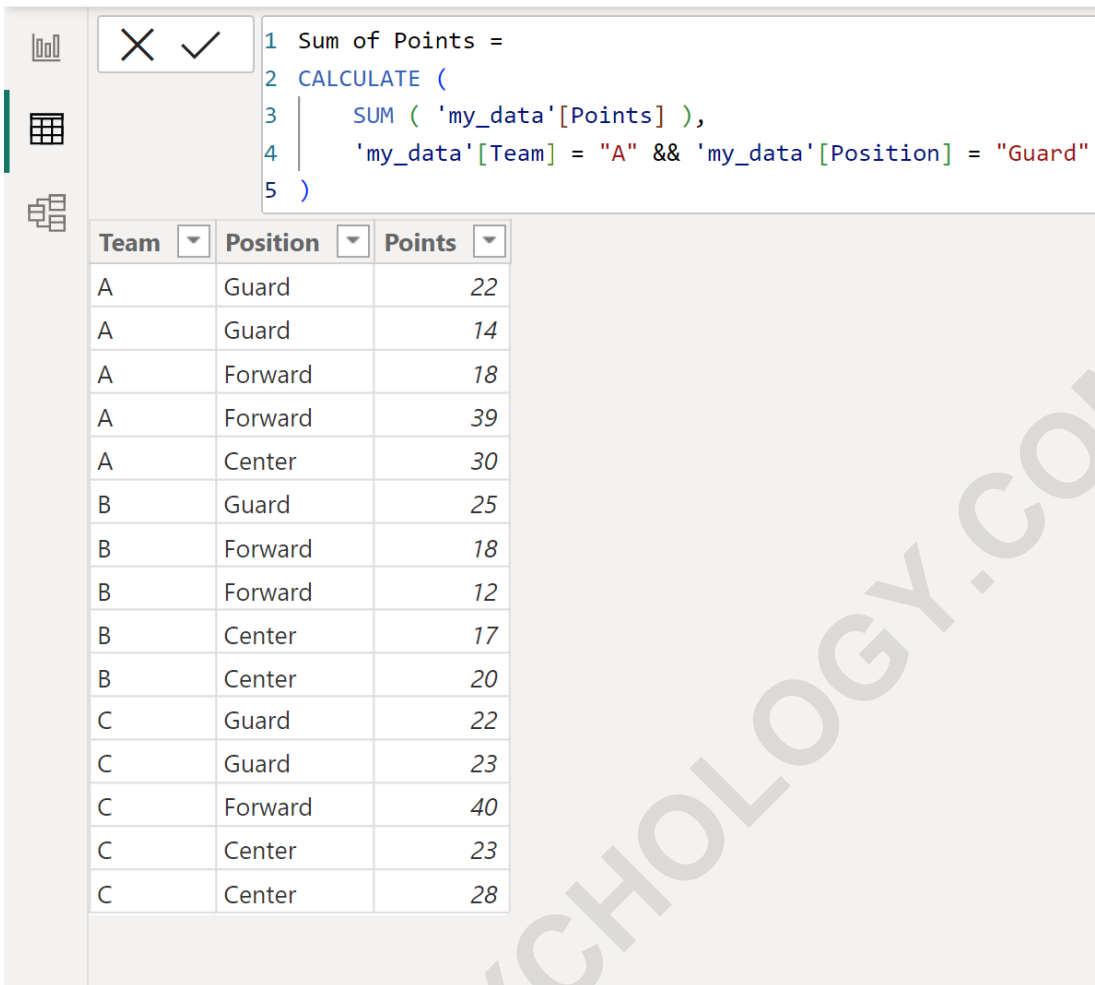
To implement this, you must first navigate to the modeling ribbon in Power BI Desktop. The process starts by creating a new measure associated with your data model. Click the **Table tools** tab, which is where measure definitions are managed, and then initiate the creation process by clicking the **New measure** icon.



Next, input the formula that utilizes the logical AND operator (`&&`) into the formula bar. This concise expression explicitly tells DAX to evaluate the 'my_data' table, filtering out any row where both Team is not "A" or Position is not "Guard," before finally calculating the sum of the remaining 'Points'.

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

Once the measure is saved, it becomes available in your Fields pane. Below shows the resulting data model after the measure has been defined. Note that the calculation itself only runs when placed on a visual.



The screenshot shows the Power BI DAX editor interface. At the top, there is a formula bar with a 'Sum of Points' measure defined as follows:

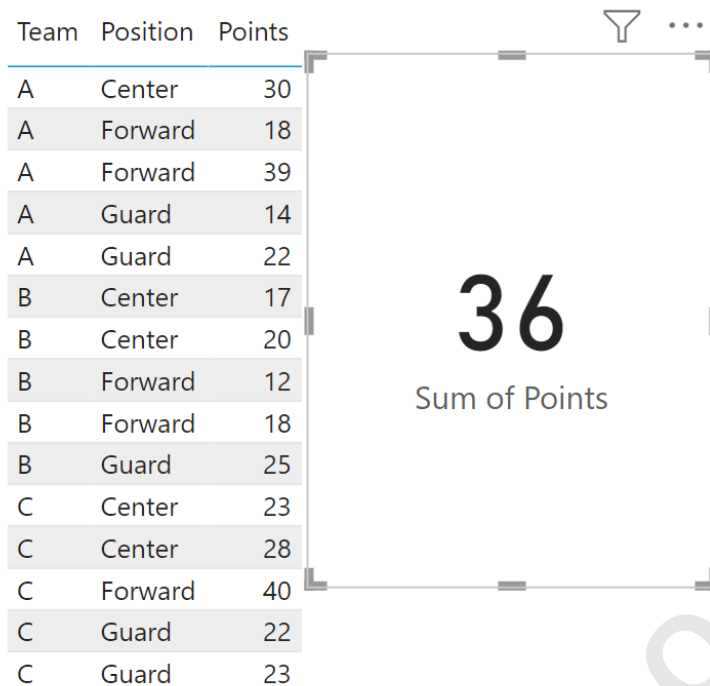
```
1 Sum of Points =  
2 CALCULATE (  
3     SUM ( 'my_data'[Points] ),  
4     'my_data'[Team] = "A" && 'my_data'[Position] = "Guard"  
5 )
```

Below the formula bar, a data table is displayed with three columns: Team, Position, and Points. The table contains 15 rows of data. A large watermark 'ARABPSYCHOLOGY.COM' is overlaid diagonally across the table.

| 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 verify the outcome, switch to the Report View and insert a Card visualization. Configure the Card visualization to display the value of the newly created **Sum of Points** measure. The Card visual will instantly display the calculated result, which is derived only from the rows meeting the strict AND criteria defined in the DAX formula.

| 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 |



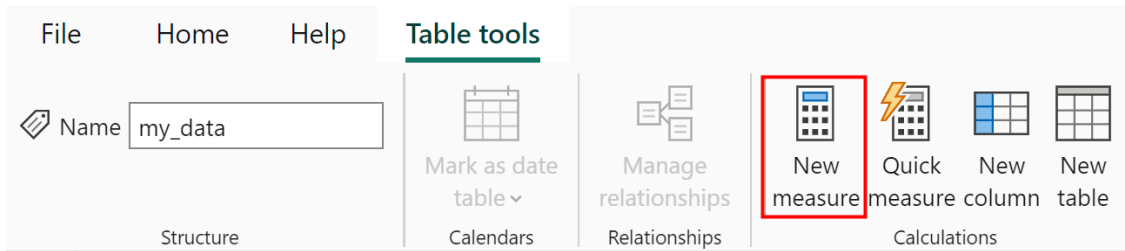
The visualization consists of a table on the left and a card on the right. The table has three columns: Team, Position, and Points. The card displays the number 36 and the text 'Sum of Points'. A filter icon is visible above the card.

Based on our underlying data table, we can confirm that only two rows satisfy both conditions (Team A and Position Guard): the player with 18 points and the player with 18 points. Therefore, the sum of points for the players who are on Team A and have a Position of Guard is **36** (18 + 18). This demonstrates the restrictive power of the logical AND operator in modifying the filter context.

Example 2: Practical Application of the OR Condition

In contrast to the previous example, suppose we now need a broader metric that calculates the sum of values in the **Points** column for any player who is on **Team A OR** holds the **Position** of Guard. This is a significantly expanded requirement, as we are now including players who might be on Team A but not a Guard, or players who are Guards but not on Team A. We expect a much larger aggregated result because the OR condition is inclusive.

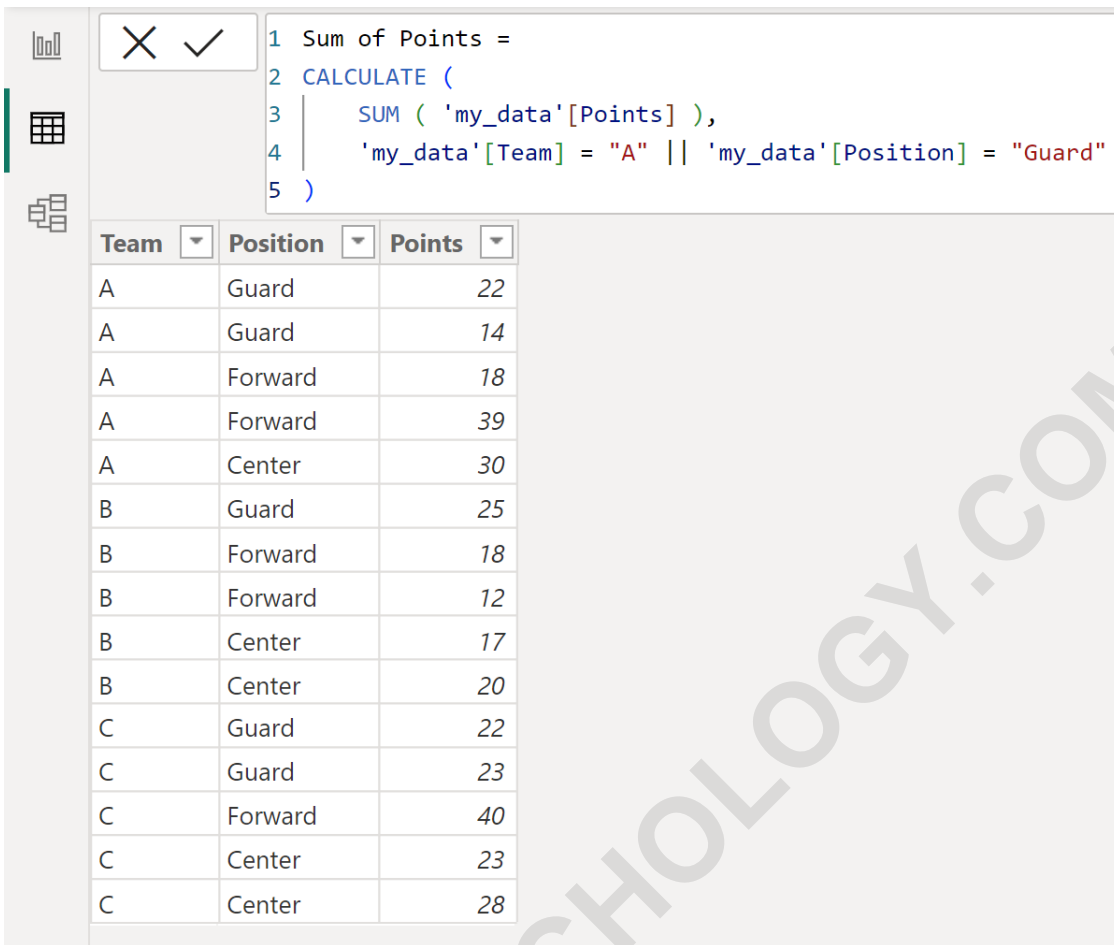
The implementation steps start identically to the AND example. You must navigate to the **Table tools** tab in Power BI Desktop and click the **New measure** icon to begin defining the new calculation. This ensures the measure is correctly registered within the model structure.



Subsequently, type the following formula into the formula bar. This time, we use the logical OR operator (`||`) to combine the filter conditions. The CALCULATE function will now include any row where the Team equals "A" or the Position equals "Guard," ensuring that the SUM aggregation captures all relevant data points based on either criteria.

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

Once defined, the measure is added to the model, as visualized below. It is important to remember that measures defined using the logical OR operator often capture a superset of the data compared to their AND counterparts, which is reflected in the final output.



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

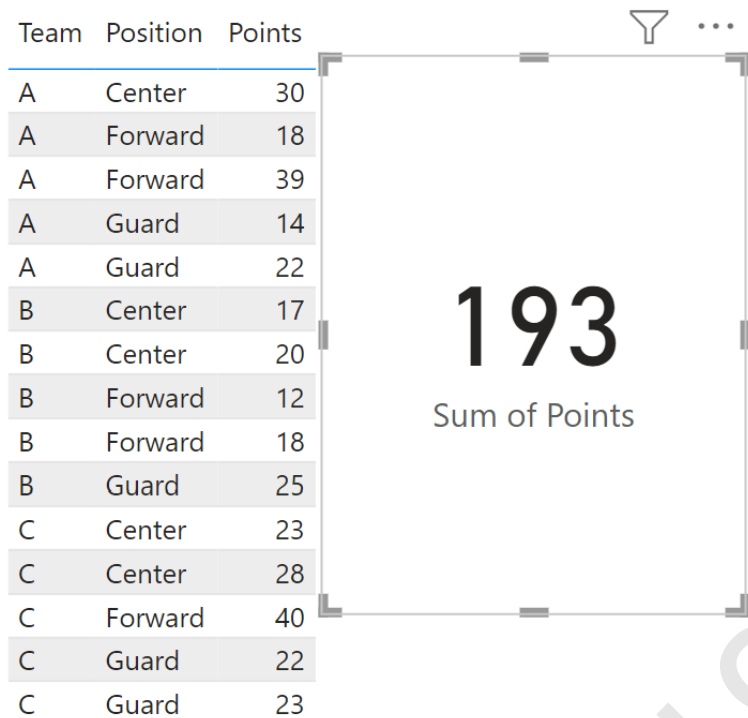
```
1 Sum of Points =  
2 CALCULATE (  
3     SUM ( 'my_data'[Points] ),  
4     'my_data'[Team] = "A" || 'my_data'[Position] = "Guard"  
5 )
```

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 |

To visualize the result, switch back to the Report View and insert a Card visualization configured to display the value of this new measure. The resulting figure will be substantially higher than the AND calculation, reflecting the inclusive nature of the OR logic.

| 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 |



Analyzing the data set, we find that the sum of points for the players who are on Team A or have a Position of Guard is **193**. This result includes all players from Team A (regardless of position) and all players who are Guards (regardless of team), highlighting the expansive nature of the OR condition in context modification. This illustrates how crucial the choice of logical operator is for accurate metric calculation in advanced Power BI reporting.

Further Considerations for Complex Filtering

While direct usage of `&&` and `||` within CALCULATE is the most efficient and common way to handle multiple filter conditions, DAX offers alternative and more complex methods, particularly when dealing with non-contiguous filters or dynamic conditions. For situations involving dynamic table filtering, you might explicitly utilize the FILTER function. The FILTER function allows you to define a virtual table based on complex boolean logic before passing that filtered table to CALCULATE. However, for simple column equality checks shown in the examples above, the implicit filtering provided by `&&` and `||` is generally preferred for performance and readability.

The ability to precisely control the filter context using logical operators is fundamental to mastering Measure creation. By defining the exact row criteria using AND or OR, you ensure that your metrics are accurate, targeted, and highly effective for detailed data analysis within Power BI. Always choose the logical operator that correctly reflects the business requirement: use AND for

intersections of data, and OR for unions of data.

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

ARABPSYCHOLOGY.COM