

# How to Summarize Filtered Data in Power BI

Authored by  
**stats writer**

January 26, 2026

## RECOMMENDED CITATION

stats writer (2026). *How to Summarize Filtered Data in Power BI*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=127782>

The synergy between the SUMMARIZE function and the FILTER function in Power BI is fundamental for advanced data modeling. While SUMMARIZE is designed to return a new table by grouping data based on specified columns and calculating measures, the FILTER function allows for precise refinement of the resulting dataset. By wrapping the output of SUMMARIZE within the FILTER function, developers gain unparalleled control to apply specific row-level constraints immediately after aggregation. This powerful technique is essential for focused data analysis and generating highly targeted reports within the Power BI environment.

Mastering this combination is a cornerstone of effective DAX implementation. This guide will walk through the required syntax, practical application, and advanced filtering techniques to leverage these functions efficiently.

## Power BI: How to Use SUMMARIZE with FILTER for Targeted Data Aggregation

### Understanding the DAX Syntax for Summary Tables

To begin constructing a summary table based on filtered conditions, we must employ a specific sequence within DAX (Data Analysis Expressions). The core principle involves generating the potential summary first using SUMMARIZE, and then applying the restrictive criteria using FILTER to limit the resulting rows.

The standard syntax pattern uses FILTER as the external wrapper, taking the output table from SUMMARIZE as its first argument, followed by the filtering expression. This structure ensures that the filter condition is applied only to the aggregated rows defined by the SUMMARIZE operation.

The following DAX formula demonstrates how to define a new table that summarizes specific columns and subsequently filters the results:

```
Summary Table = FILTER(  
SUMMARIZE(my_data, my_data, my_data, my_data),  
my_data = "A")
```

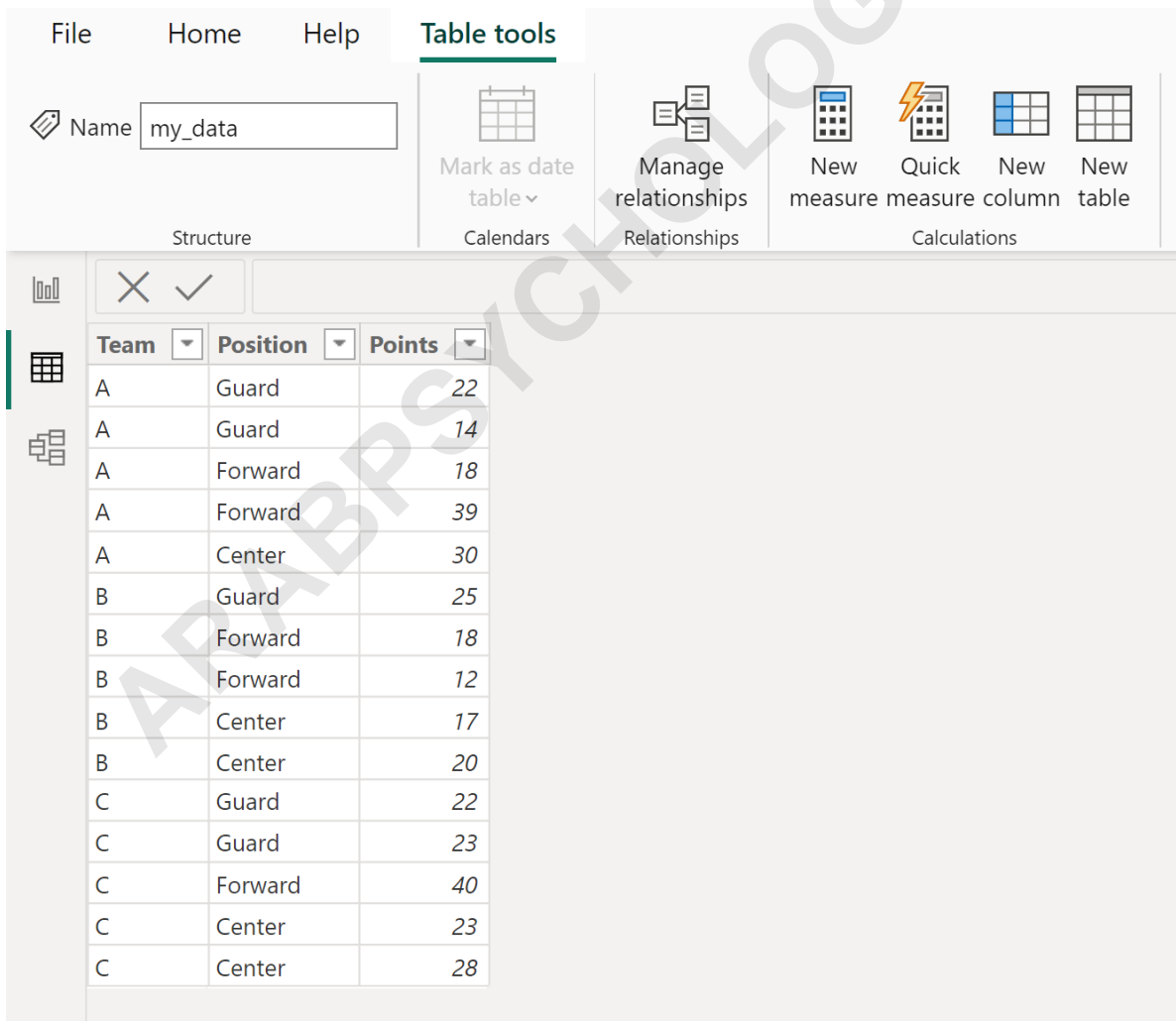
This particular DAX expression creates a new table, designated as **Summary Table**. The internal SUMMARIZE component defines the table structure, including the columns **Team**, **Points**, and **Position** drawn from the source table named **my\_data**. Crucially, the external FILTER function ensures that this new table only includes rows where the value in the **Team** column is exactly equal to "A".

## Practical Example: Defining the Data Scenario

To illustrate the practical application of this powerful DAX combination, consider a common scenario involving performance metrics. Suppose we are working with a data model in Power BI that contains comprehensive data on basketball players. This source table, which we will refer to as **my\_data**, tracks various attributes for individual players across different teams.

The objective is often to isolate specific subsets of data for focused review or comparison, a task perfectly suited for the combined **SUMMARIZE** and **FILTER** approach. Our goal is to extract player details--specifically **Team**, **Position**, and **Points**--but only for those players associated with a particular team, such as Team A.

Review the structure of the original data table, **my\_data**, shown below. Notice the diversity of teams and the various player statistics recorded:



The screenshot displays the Power BI interface with the 'Table tools' ribbon active. The table name 'my\_data' is shown in the 'Name' field. 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 visible with columns for Team, Position, and 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

This dataset serves as the foundation upon which we will build our aggregated and filtered view.

We aim to use the SUMMARIZE and FILTER functions to derive a simplified table that only contains the details necessary for analyzing Team A's composition.

## Step-by-Step Guide to Creating the Filtered Summary Table

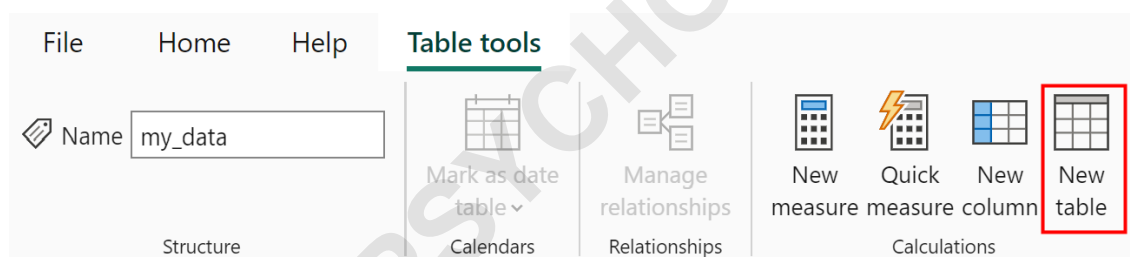
The process of implementing this DAX logic in Power BI Desktop involves creating a calculated table directly within the model view. This ensures the resulting summary table is static and available for use across various reports and measures.

Follow these precise steps to execute the operation:

Navigate to the **Modeling** tab (or **Table tools**, depending on your current view context) within the Power BI interface.

Click the **New table** icon. This action opens the DAX formula bar, prompting you to define the calculated table.

The user interface prompt typically looks like the image below, indicating where the table creation process begins:



Once the formula bar is active, enter the following comprehensive DAX expression. This formula explicitly defines the new table, applying the filter criterion immediately after defining the summarized columns:

```
Summary Table = FILTER(  
SUMMARIZE(my_data, my_data, my_data, my_data),  
my_data = "A")
```

Upon committing the formula, Power BI processes the expression. It first uses SUMMARIZE to virtually create a table comprising the specified columns, and then applies the condition defined by FILTER, isolating only the rows pertaining to Team A.

## Analyzing the Resulting Summary Table

The execution of the DAX formula results in the creation of a definitive new table named **Summary Table** within your data model. This table contains only the filtered subset of the original data, making subsequent analysis highly efficient.

As anticipated, the **Summary Table** exclusively displays records where the **Team** column value is "A". This demonstrates the precise control achieved by nesting SUMMARIZE within FILTER, effectively focusing the aggregated output exactly where required.

The resulting table structure, derived from the original **my\_data** table but filtered to show only Team A players, is displayed below:

The screenshot shows the Power BI interface with the 'Table tools' ribbon active. The ribbon includes options like 'Name' (set to 'Summary Table'), 'Mark as date table', 'Manage relationships', and 'Calculations' (with sub-options for 'New measure', 'Quick measure', 'New column', and 'New table'). Below the ribbon, the DAX formula bar shows the following formula:

```
1 Summary Table = FILTER(
2     SUMMARIZE(my_data, my_data[Team], my_data[Points], my_data[Position]),
3     my_data[Team] = "A")
```

Below the formula bar, a table view displays the resulting data:

Team	Points	Position
A	14	Guard
A	18	Forward
A	22	Guard
A	30	Center
A	39	Forward

Notice that all columns specified in the SUMMARIZE function (Team, Position, Points) are present, but the rows are strictly limited according to the filter condition (Team = "A"). This technique is particularly valuable when the original table is massive, and you only need a small, defined subset for a specific report visual.

## Extending the Filter Logic: Using Multiple Conditions

The true power of the FILTER function is its capacity to handle complex boolean logic, allowing users to impose multiple criteria simultaneously on the summarized table. You are not restricted to filtering based on a single column or a single equality check.

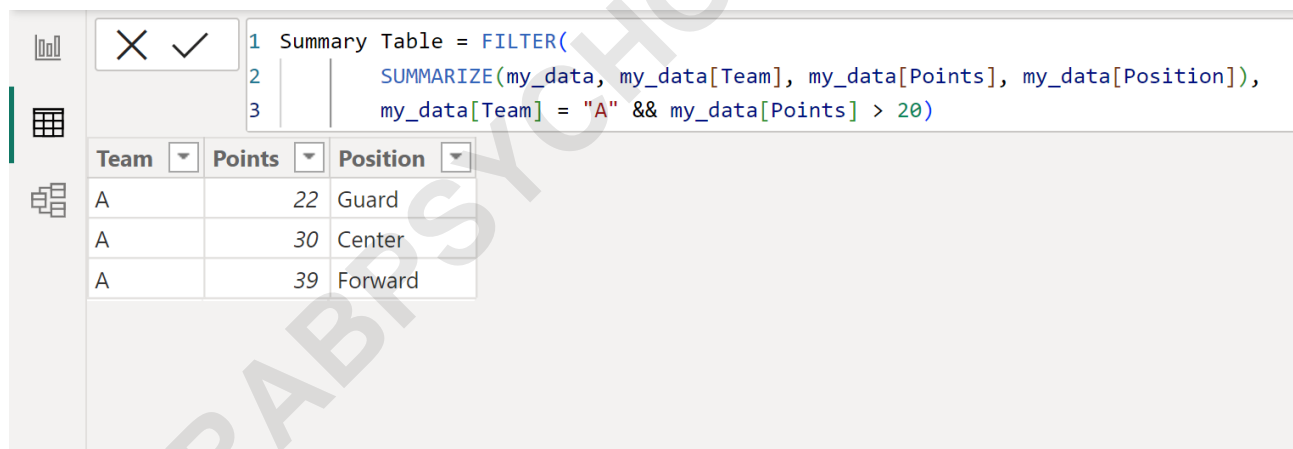
To filter based on multiple conditions, logical operators such as the logical AND (&&) or logical OR (||) must be introduced into the filter argument of the `FILTER` function. For instance, we might want to narrow our focus not just to Team A, but specifically to players on Team A who have achieved high scores, such as accumulating more than 20 points.

To implement this dual condition, the following refined `DAX` syntax is required, linking the two conditions using the logical AND operator:

```
Summary Table = FILTER(
SUMMARIZE(my_data, my_data, my_data, my_data),
my_data = "A" && my_data > 20)
```

This expression instructs `Power BI` to retain only those rows resulting from the `SUMMARIZE` operation where both the **Team** is "A" and the **Points** value is strictly greater than 20. The ability to chain conditions allows for highly granular control over the data set.

The output following the application of this combined filter demonstrates the effectiveness of the refined logic, showing only the players meeting both high-score and team affiliation criteria:



Team	Points	Position
A	22	Guard
A	30	Center
A	39	Forward

Developers are encouraged to utilize as many conditions as necessary, adapting the logical operators (&&, ||) to meet complex business requirements for focused data analysis. The flexibility inherent in the `FILTER` function makes it a cornerstone of conditional table generation in `Power BI`.

## Final Considerations and Further Learning

While the combination of `SUMMARIZE` and `FILTER` is highly effective, it is important to remember the context transitions and efficiency implications of `DAX` calculated tables. Since these tables are evaluated at data refresh time, carefully defining the required columns and minimizing unnecessary calculations are key to model performance.

For those looking to deepen their expertise in [DAX](#) functions, detailed official documentation is an invaluable resource. Understanding the nuances of aggregation and row context is crucial for mastering advanced data manipulation techniques.

**Note:** You can find the complete official documentation for the **SUMMARIZE** function in [DAX](#) on the Microsoft Learn platform. Additionally, exploring the documentation for other table manipulation functions will further enhance your data modeling capabilities.

## Related Power BI Tutorials

To continue building proficiency in [Power BI](#) data modeling and [DAX](#) application, consider exploring tutorials on related common tasks. These resources often provide valuable insights into optimizing performance and solving various [data analysis](#) challenges:

Tutorials explaining the difference between [SUMMARIZE](#) and [SUMMARIZECOLUMNS](#).

Guides on creating dynamic calculated measures using `CALCULATE` and context modification functions.

How to handle time intelligence functions for complex reporting needs.

The following tutorials explain how to perform other common tasks in [Power BI](#):