

# How to Filter Data in Power BI Using the NOT IN Operator

Authored by  
**mohammed looti**

January 9, 2026

## RECOMMENDED CITATION

mohammed looti (2026). *How to Filter Data in Power BI Using the NOT IN Operator*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=125276>

The ability to precisely control the flow and visibility of data is fundamental to effective business intelligence. In Power BI, achieving targeted filtering often requires excluding specific data points rather than including them. While many programming languages offer a direct `NOT IN` operator, the DAX language utilizes a logical combination of the `NOT` and `IN` operators to achieve this powerful exclusionary effect. This technique allows analysts to define a set of values (a list) and subsequently exclude any rows where a specified column matches any item within that list.

The primary purpose of implementing the "NOT IN" logic is data cleansing and focused analysis. Consider a scenario involving large customer datasets: if an analyst needs to calculate aggregate metrics, such as total revenue, but must ignore data originating from known test accounts or underperforming regions, an exclusionary filter is necessary. Rather than laboriously listing every region or account **to include**, using the "NOT IN" equivalent in DAX provides a concise and scalable solution for removing the specified undesirable data elements. This results in cleaner visualizations and more accurate calculations, directly supporting better decision-making.

Understanding how to correctly synthesize the `NOT` and `IN` logical operators within the DAX environment is essential for advanced data modeling. This functionality is particularly crucial when dealing with dimensions that have a high cardinality, where defining what to exclude is significantly easier than defining what to include. By mastering this syntax, users gain fine-grained control over the evaluation context, ensuring that calculations only operate on the data relevant to the analytical question at hand, thereby improving both the efficiency and interpretability of their Power BI reports.

## DAX Fundamentals: Syntax for the "NOT IN" Equivalent

The core of implementing exclusionary logic in DAX involves utilizing specific functions that modify the filter context, allowing us to specify conditions that must be negated. The most common and robust way to apply this type of filter, especially when creating new tables or measures, is through the use of functions like CALCULATE or CALCULATETABLE. These functions accept filter arguments that can be wrapped in the logical `NOT` operator, which reverses the outcome of the inner condition.

The standard syntax structure relies on defining the exclusion criteria within the filter argument of a table-manipulating function. The `IN` operator is inherently used to check if a value exists within a defined list, often enclosed in curly braces `{}`. By placing the entire `IN` clause inside the `NOT` operator, we instruct DAX to return `TRUE` only for rows where the column value is **not** present in the specified list. This logical inversion is the key to simulating the "NOT IN" functionality that might be familiar from SQL or other query languages.

Here is the exact syntax structure used in Power BI for applying a simple single-column exclusion filter, typically used when defining a new calculated table:

```
filtered_data =  
CALCULATETABLE('my_data', NOT('my_data' IN {"A", "C"}))
```

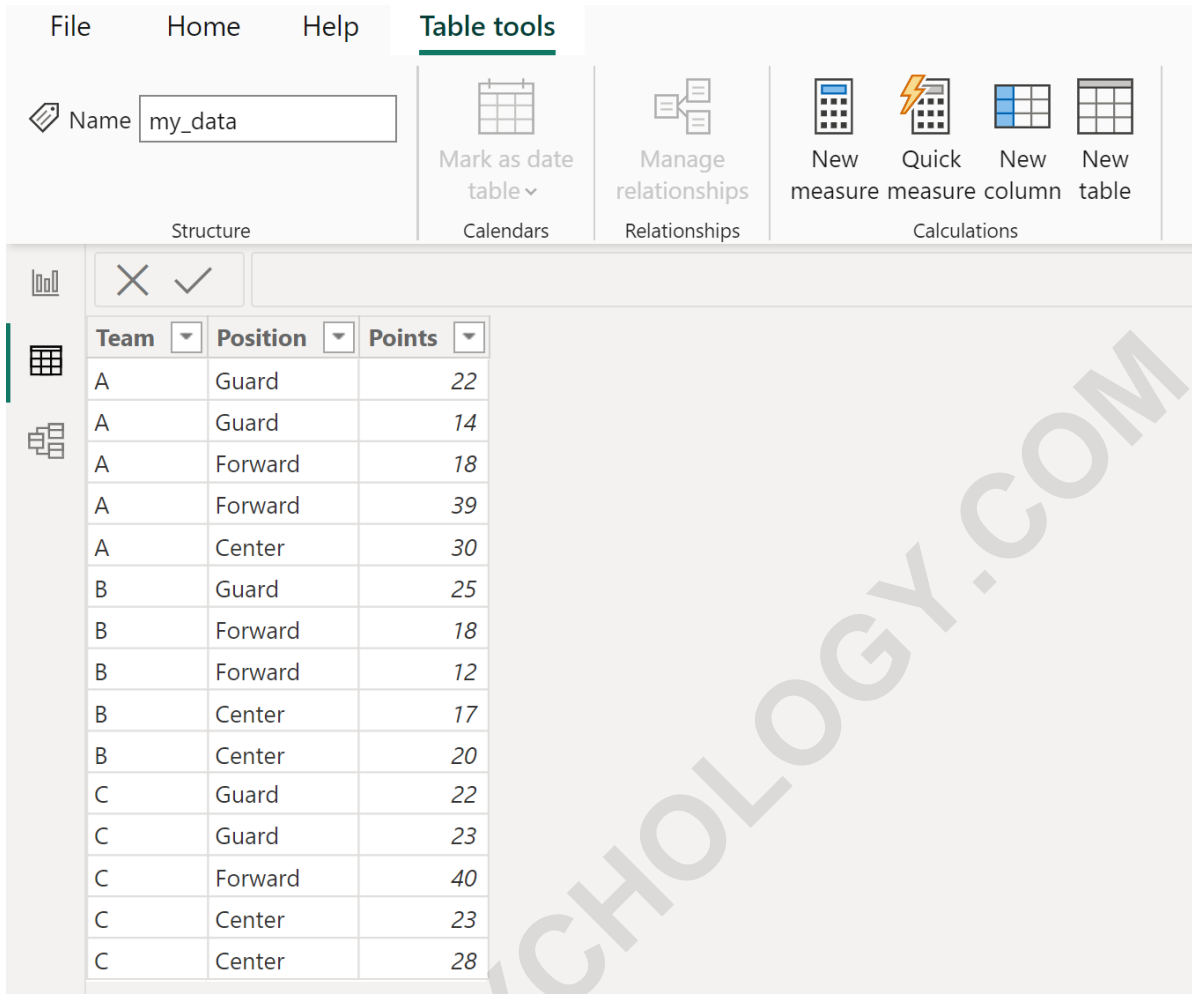
This powerful formula performs a specific task: it generates a new table named **filtered\_data**. This new table is populated only with the rows sourced from the original table, **my\_data**, provided that the value found in the **Team** column is not equal to A or C. This precise application of exclusionary filtering is central to targeted data manipulation within the Power BI data model.

It is important to remember the syntax requirements for the list of excluded items. DAX mandates the use of curly brackets { } when defining a static list or a set of values directly within the formula, as demonstrated above. Each item within the list must also be enclosed in double quotes if it represents a text string. Missing these brackets or quotes will result in a syntax error, preventing the successful creation or modification of the data structure.

### Step-by-Step Example: Excluding Teams from a Dataset

To fully grasp the practical application of the "NOT IN" operator in Power BI, let us walk through a concrete scenario involving athlete data. Suppose we have a source table, aptly named **my\_data**, which comprehensively details various basketball players, including their names, positions, and the teams they belong to. Our analytical objective is to isolate players from specific teams (A and C) to analyze the performance metrics of the remaining teams without any external influence.

The initial dataset, **my\_data**, contains the following structure and values. This table serves as our starting point, containing all records before the exclusionary logic is applied. Note the distribution of players across the various teams:



The screenshot shows the Power BI Desktop interface with the 'Table tools' ribbon selected. The ribbon includes options like 'Name' (set to 'my\_data'), 'Mark as date table', 'Manage relationships', and 'Calculations' (with sub-options: 'New measure', 'Quick measure', 'New column', 'New table'). Below the ribbon, a data table is displayed with the following columns: 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

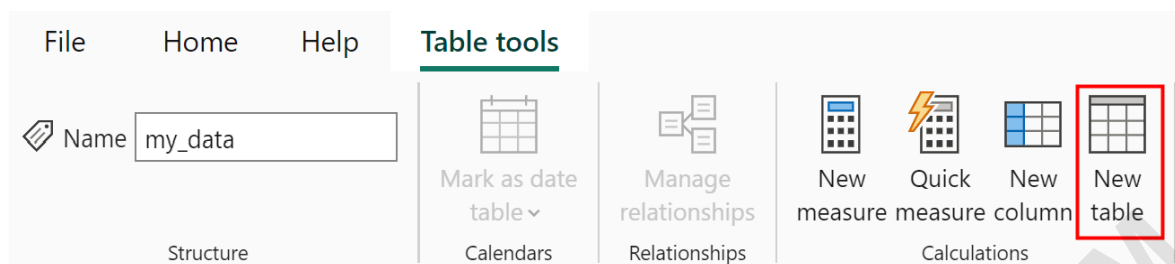
Suppose we would like to create a new table that only contains players who are not on teams A or C. This process requires creating a calculated table using DAX, ensuring that the original data integrity is maintained while providing a focused subset for specialized analysis. The methodology outlined below details the exact steps required within the Power BI desktop environment to achieve this goal.

## Implementing the Single "NOT IN" Filter in Power BI

The implementation begins within the Power BI Desktop interface. Since we are creating a completely new data structure based on the filtering condition, we must utilize the table creation tools. Navigate to the top ribbon menu where the data modeling tools reside. Specifically, click the **Table tools** tab, which exposes the options necessary to introduce new calculated tables into the data model.

Once you are within the **Table tools** context, the critical next step is initiating the calculation environment. Look for and click the **New table** icon. This action immediately opens the formula

bar, prompting you to input the DAX expression that will define the content and structure of your new table. This is where the exclusionary logic is formally defined.



Then, in the formula bar, meticulously enter the following DAX formula. This command defines the new table **filtered\_data** using the CALCULATETABLE function, applying the NOT IN logic to the **Team** column. Remember that the logical NOT reverses the result of the IN operation, ensuring that only records where the team is **not** A or C are preserved:

```
filtered_data =  
CALCULATETABLE('my_data', NOT('my_data' IN {"A", "C"}))
```

This will create a new table named **filtered\_data** that only contains players who are not on teams A or C. A quick inspection of this new table confirms that the exclusionary criteria have been met. Only those players who belong to Team B or Team D remain, demonstrating the effective removal of the unwanted records:

The image shows the Power BI interface with the DAX formula bar and the resulting filtered data table. The formula bar contains the following DAX formula:

```
1 filtered_data =  
2 CALCULATETABLE('my_data', NOT('my_data'[Team] IN {"A", "C"}))
```

The resulting table is displayed below the formula bar:

Team	Points	Position
B	25	Guard
B	18	Forward
B	12	Forward
B	17	Center
B	20	Center

## Handling Complex Requirements: Applying Multiple Exclusionary Conditions

While single column exclusions are straightforward, real-world data analysis often demands the simultaneous application of multiple, distinct filtering rules. For instance, an analyst might need to exclude not only specific teams but also players occupying certain positions, such as guards or centers. DAX facilitates this complex filtering by allowing multiple logical expressions to be chained together using logical operators like `&&` (AND) or `||` (OR).

When chaining multiple exclusionary conditions, it is critical to use the appropriate logical conjunction. If we require a row to satisfy **all** exclusionary rules simultaneously--meaning the row must be excluded if it fails Rule 1 AND Rule 2--we use the `&&` (AND) operator. Each individual exclusionary requirement must be defined using the `NOT(Column IN {Values})` structure, and these structures are then connected to form a compound filter argument within the CALCULATETABLE function.

In the following advanced example, we would like to create a new table where the following two exclusionary conditions are both true for a row to be included in the final result set:

The value in the **Team** column is not A or C.

The value in the **Position** column is not Guard or Center.

This compound requirement ensures highly targeted data isolation. The resulting table will only contain players who meet both criteria simultaneously.

You can use the following syntax to implement these multiple exclusionary filters:

```
filtered_data =  
CALCULATETABLE('my_data', NOT('my_data' IN {"A", "C"}) &&  
NOT('my_data' IN {"Guard", "Center"}))
```

This will create a new table named **filtered\_data** that only contains players who are not on teams A or C and who also do not have a position of Guard or Center. This detailed filtering capability is essential for generating specialized subsets of data necessary for deep-dive analysis:



## Conclusion and Related Resources

Mastering exclusionary logic using the `NOT IN` construct in DAX is a vital skill for anyone working extensively with Power BI. By combining the `NOT` and `IN` operators, analysts can define precise filter contexts that streamline data visualization and ensure that calculations are based only on the required subset of information. This method offers scalability and clarity, making complex data exclusion manageable even in large data models.

**Note:** You can find a complete list of operators you can use in DAX, along with detailed documentation on the logical operators, in the official Microsoft documentation.

The following tutorials provide insight into other common data manipulation tasks within the Power BI environment: