

How to Write IF Statements with Multiple Conditions in Power BI

Authored by
stats writer

January 29, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Write IF Statements with Multiple Conditions in Power BI*.
PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=128491>

The ability to implement complex conditional logic is fundamental to advanced data analysis. In Power BI, users rely on Data Analysis Expressions (DAX) to define custom calculations, measurements, and columns. A core component of DAX is the IF function, which enables the creation of logical flows that evaluate criteria and return results based on those outcomes. While a simple IF statement handles one condition, real-world data scenarios often require assessing multiple criteria simultaneously.

This comprehensive guide details how to construct robust IF statements in Power BI that incorporate multiple conditions. This is achieved by nesting logical functions like AND or OR within the primary IF function. By mastering these techniques, you can transform raw data into insightful, decision-ready metrics, allowing for sophisticated segmentation and accurate classification based on various data points.

Power BI: Write an IF Statement with Multiple Conditions

Understanding Conditional Logic and the IF Function in DAX

The foundation of decision-making within DAX is the IF function. This function operates similarly to conditional statements found in many programming languages. It requires three arguments: a logical test, a value returned if the test is true, and a value returned if the test is false. The syntax is straightforward: **IF(Logical_Test, Value_if_True, Value_if_False)**. However, when multiple criteria must be checked, simply relying on a single logical test is insufficient, necessitating the use of logical operators to combine these criteria.

In scenarios requiring complex classification--for instance, determining a product's priority based on both inventory levels and sales volume--you must aggregate the results of several tests into a single Boolean outcome for the IF function to evaluate. This aggregation is precisely where the DAX functions **AND** and **OR** become indispensable tools. These functions allow the IF statement to handle sophisticated evaluations that mirror real-world business rules, vastly increasing the analytical power of your Power BI reports.

Utilizing Logical Operators for Multi-Condition IF Statements

When constructing IF statements with more than one condition, you must define the relationship between those conditions. **Logical operators** dictate how the individual conditions interact to determine the final true/false result. The two primary operators used in this context are **OR** and **AND**. Understanding the precise behavior of each operator is crucial for writing accurate DAX calculations.

The **OR** function returns **TRUE** if at least one of the provided conditions is met. If you need a calculation to pass based on fulfilling Criterion A **or** Criterion B, OR is the appropriate operator. This is used for broad categorization where multiple paths lead to the same result.

The **AND** function returns **TRUE** only if **all** of the provided conditions are met. If a calculation requires satisfying both Criterion A **and** Criterion B simultaneously, AND must be used. This operator enforces stricter criteria and narrows down the resulting dataset.

In DAX, these operators are nested within the first argument (the **Logical_Test**) of the main **IF function**. The structure involves wrapping all individual conditions inside the chosen logical operator function, which then provides a single TRUE or FALSE value back to the IF function for final evaluation.

Method 1: Combining Conditions with the OR Operator

The **OR** function is ideal for situations where a positive outcome (Value_if_True) is warranted if any one of several defined criteria is satisfied. For example, if we are evaluating a player's performance, they may be classified as "Good" if they score high on Points or if they score high on Assists. They do not need to excel in both metrics simultaneously.

The following syntax demonstrates how to structure an IF statement in **DAX** using the **OR** function to evaluate two distinct conditions within the same calculation. We create a new calculated column called **Rating** in the table named **my_data**:

Method 1: Write an IF Statement with OR Condition

```
Rating =  
IF(  
OR(  
'my_data' > 20,  
'my_data' > 4  
),  
"Good",  
"Bad"  
)
```

This specific formula creates a new column named **Rating**. It evaluates whether the value in the **Points** column is greater than 20 **OR** the value in the **Assists** column is greater than 4. If either or both of these conditions are true, the function returns "Good." Conversely, if neither condition is met--meaning both **Points** are 20 or less **and** **Assists** are 4 or less--the function defaults to returning "Bad" as the result. This approach is highly effective for broad, inclusive classification

tasks.

Method 2: Enforcing Strict Criteria with the AND Operator

In contrast to OR, the **AND** function is utilized when the desired outcome requires simultaneous fulfillment of all specified criteria. If we use the same performance example, classifying a player as "Good" only happens if they achieve a high score in Points **and** a high score in Assists. This operator implements a much stricter requirement for success.

When embedding the **AND** function within the IF function, you ensure that the logical test only resolves to **TRUE** when every specified condition holds true. Failure of even a single condition results in the entire AND expression resolving to **FALSE**, triggering the Value_if_False output of the IF statement.

The DAX syntax below illustrates how to use the **AND** function to define a combined condition. This example requires both the Points score to be above 20 and the Assists score to be above 4 for the rating to be considered "Good":

Method 2: Write an IF Statement with AND Condition

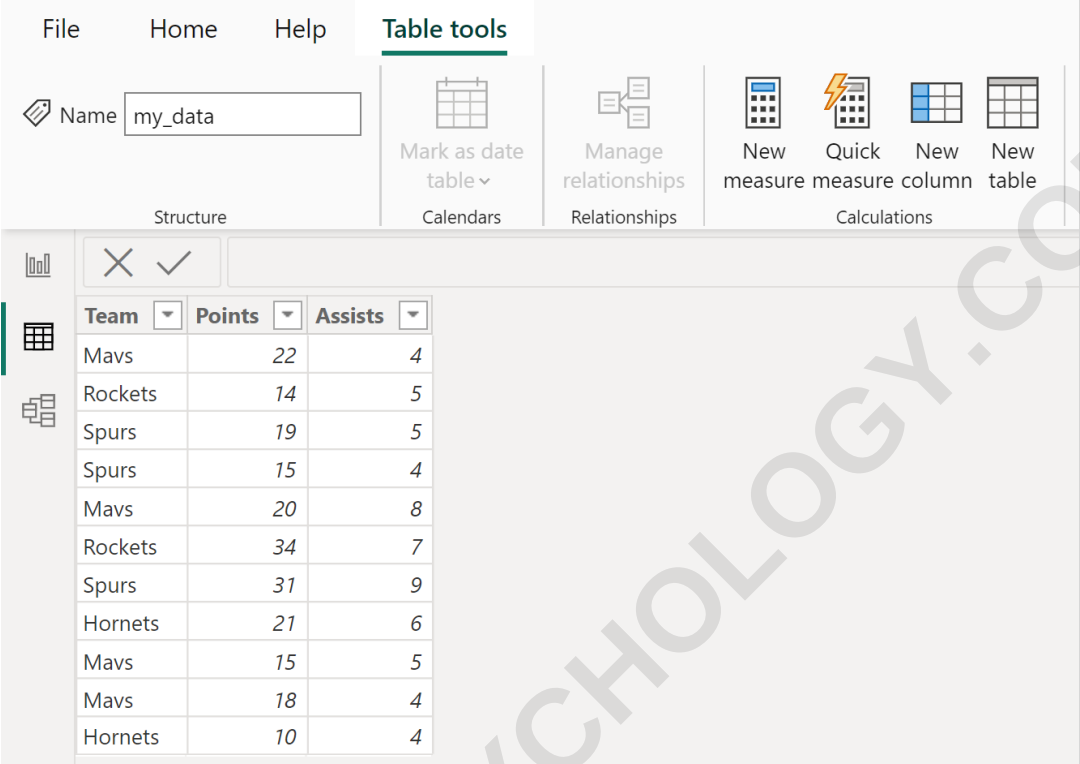
```
Rating =  
IF(  
  AND(  
    'my_data' > 20,  
    'my_data' > 4  
  ),  
  "Good",  
  "Bad"  
)
```

This formula creates a new column named **Rating** that returns "Good" only if the value in the **Points** column is greater than 20 **and** the value in the **Assists** column is greater than 4. If one of these conditions fails, or if both fail, the function immediately returns "Bad." This methodology is essential for precise data filtering and classification where complex thresholds must be met.

Setting Up the Example Data Model in Power BI

To fully appreciate the application of these logical functions, we will demonstrate both the OR and AND methods using a foundational dataset. We will assume the existence of a table within the Power BI data model named **my_data**. This table contains performance metrics, specifically focusing on the columns **Points** and **Assists**.

The following visualization depicts the structure and sample records of the **my_data** table, which serves as the base for our subsequent column calculations. Notice the varying combinations of Points and Assists values, which will yield different results depending on whether the OR or AND logic is applied:



The screenshot shows the Power BI Desktop interface. The 'Table tools' ribbon is active, displaying options like 'Mark as date table', 'Manage relationships', and 'New measure', 'Quick measure', 'New column', and 'New table' under the 'Calculations' group. Below the ribbon, a data table named 'my_data' is visible. The table has three columns: 'Team', 'Points', and 'Assists'. The data is as follows:

Team	Points	Assists
Mavs	22	4
Rockets	14	5
Spurs	19	5
Spurs	15	4
Mavs	20	8
Rockets	34	7
Spurs	31	9
Hornets	21	6
Mavs	15	5
Mavs	18	4
Hornets	10	4

Our goal is to create a new calculated column for "Rating" based on the logic derived from these two columns. Calculated columns, as opposed to measures, evaluate the formula for every row in the table and store the result persistently, making them ideal for row-by-row conditional classification like this example.

Example 1: Write an IF Statement with OR Condition in Power BI

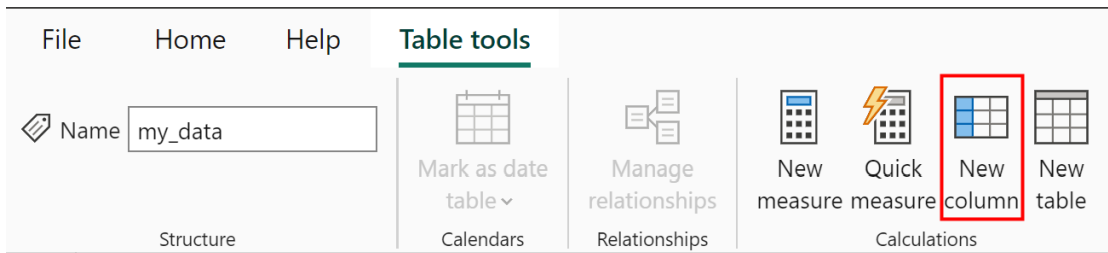
Let us begin with the OR condition scenario. Our business rule dictates that a record receives a "Good" rating if the **Points** score exceeds 20 **or** the **Assists** count exceeds 4. We want to apply this rule to our existing **my_data** table by adding a new calculated column.

The procedure for implementing this logic begins in the Power BI Desktop application:

Navigate to the Data View where the **my_data** table is visible.

On the Table Tools ribbon, click the **New column** icon. This action opens the formula bar, allowing us to input our DAX expression.

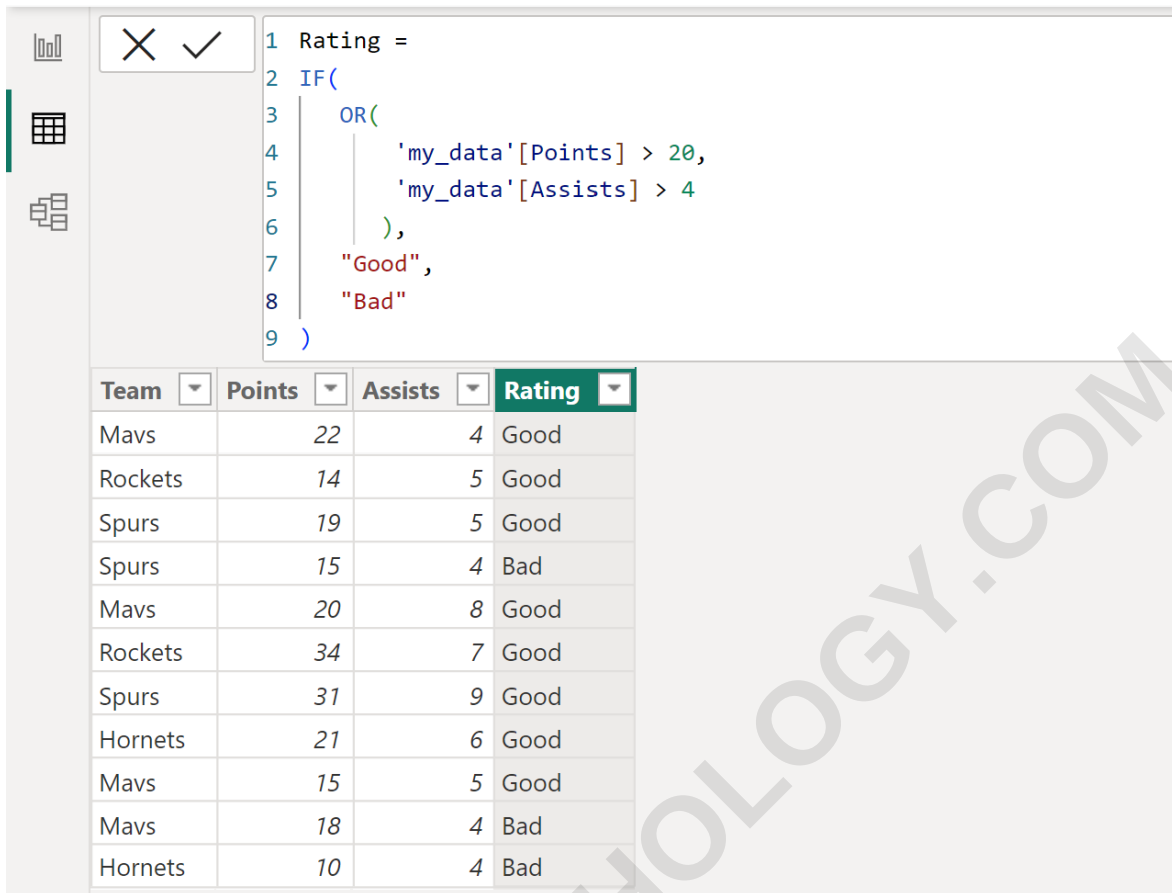
To do so, click the **New column** icon:



Then type in the following formula into the formula bar:

```
Rating =  
IF(  
OR(  
'my_data' > 20,  
'my_data' > 4  
),  
"Good",  
"Bad"  
)
```

Upon committing the formula, the new **Rating** column is populated. Observe how rows that satisfy either the Points condition (e.g., Row 1: Points=25) or the Assists condition (e.g., Row 4: Assists=5) are classified as "Good." Only rows that fail both conditions (e.g., Row 5: Points=18, Assists=2) receive the "Bad" classification:



```

1 Rating =
2 IF(
3     OR(
4         'my_data'[Points] > 20,
5         'my_data'[Assists] > 4
6     ),
7     "Good",
8     "Bad"
9 )

```

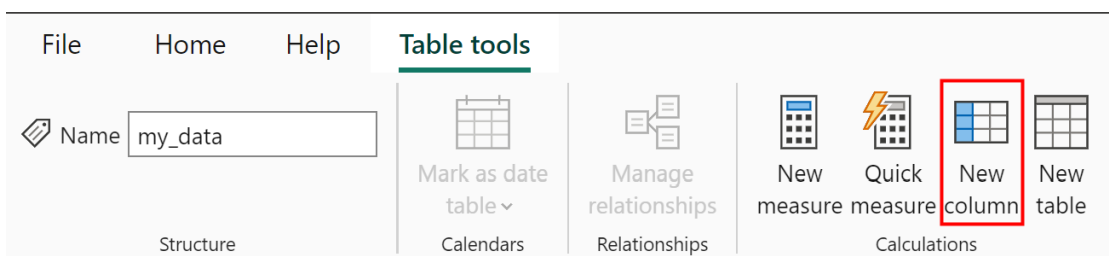
Team	Points	Assists	Rating
Mavs	22	4	Good
Rockets	14	5	Good
Spurs	19	5	Good
Spurs	15	4	Bad
Mavs	20	8	Good
Rockets	34	7	Good
Spurs	31	9	Good
Hornets	21	6	Good
Mavs	15	5	Good
Mavs	18	4	Bad
Hornets	10	4	Bad

Example 2: Write an IF Statement with AND Condition in Power BI

Next, we pivot to the more restrictive **AND** condition. Here, the rule states that a record must receive a "Good" rating only if **both** the **Points** score is greater than 20 **and** the **Assists** count is greater than 4. This strict requirement will result in fewer "Good" classifications compared to the previous OR example.

We follow the same steps to create a new calculated column, clicking the **New column** icon to initiate the process:

To do so, click the **New column** icon:



The image shows the Power BI ribbon with the 'Table tools' tab selected. In the 'Calculations' group, the 'New column' icon (a grid with a blue square) is highlighted with a red rectangle. Other icons in the group include 'New measure', 'Quick measure', and 'New table'.

Then, we input the formula utilizing the **AND** function into the formula bar:

Rating =

IF(

AND(

'my_data' > 20,

'my_data' > 4

),

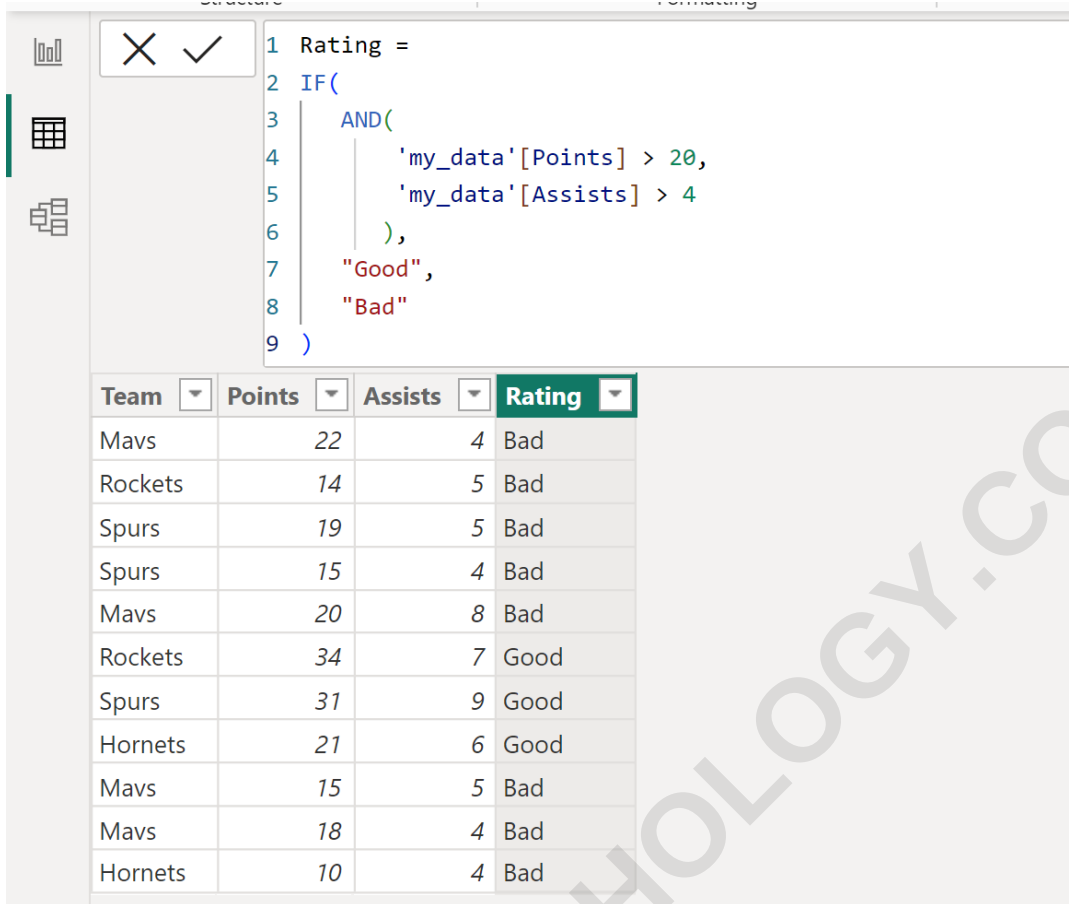
"Good",

"Bad"

)

This will create a new column named **Rating** that contains the value "Good" or "Bad" based on the corresponding values in the **Points** and **Assists** columns:

Observe the results: Only Row 2 (Points=22, Assists=6) and Row 3 (Points=28, Assists=8) satisfy both stringent conditions and are rated "Good." Other rows that met only one condition in the previous example (like Row 1 or Row 4) are now correctly classified as "Bad" because they failed the joint requirement enforced by the **AND** logical operator:



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

```
1 Rating =  
2 IF(  
3     AND(  
4         'my_data'[Points] > 20,  
5         'my_data'[Assists] > 4  
6     ),  
7     "Good",  
8     "Bad"  
9 )
```

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

Team	Points	Assists	Rating
Mavs	22	4	Bad
Rockets	14	5	Bad
Spurs	19	5	Bad
Spurs	15	4	Bad
Mavs	20	8	Bad
Rockets	34	7	Good
Spurs	31	9	Good
Hornets	21	6	Good
Mavs	15	5	Bad
Mavs	18	4	Bad
Hornets	10	4	Bad

Advanced Conditional Logic and Next Steps

While the basic combination of IF with AND or OR covers most scenarios, Power BI allows for far more complex conditional structures. It is possible to nest multiple logical operators, such as combining an AND condition with an OR condition, to address highly specific business logic. For example, you might classify a transaction as "High Priority" if (Sales > 1000 AND Margin > 0.15) OR (Client Status = "VIP"). This requires careful structuring and parenthetical discipline in DAX.

For scenarios involving more than two or three outcomes, relying solely on nested IF statements can become cumbersome and difficult to debug. In such cases, experts often recommend using the **SWITCH(TRUE(), ...)** pattern in DAX. The SWITCH function evaluates a series of logical tests sequentially, providing a cleaner and more scalable approach for multi-branch conditional logic than extensive IF function nesting.

We highly recommend reviewing the official Microsoft documentation for a complete understanding of the intricacies of the IF function and related logical operators within the DAX language. Mastering these foundational concepts is key to generating powerful, dynamic, and accurate calculations in your Power BI data models.

Note: You can find the complete documentation for the **IF** function in DAX .

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

ARABPSYCHOLOGY.COM