How to Write IF Statements in Power BI: A Step-by-Step Guide

Authored by

stats writer

January 14, 2026

An IF statement in Power BI is a powerful logical function crucial for advanced data modeling. It enables you to execute specific actions or return certain values based on whether a defined condition is met. The fundamental syntax in Data Analysis Expressions (DAX) is straightforward: `IF(condition, value if true, value if false)`. The engine evaluates the **condition**; if it resolves to true, the first value is returned; otherwise, the second value is returned. For instance, imagine needing to categorize product sales data. If we define a threshold of 1,000, we can use the formula `IF(Sales > 1000, "High", "Low")`. This formula immediately classifies transactions, returning "High" only when sales exceed 1,000 and "Low" in all other scenarios, providing instant segmentation within your reports.

Understanding the DAX Syntax for IF Statements

To effectively implement conditional logic within your datasets, you must utilize the specific syntax provided by DAX (Data Analysis Expressions). This powerful formula language, native to Power BI, supports various methods for conditional evaluation, ranging from simple binary checks to complex nested structures. Mastering the **IF function** is the first step toward creating flexible and dynamic data calculations that drive meaningful insights in your reports.

Method 1: Implementing a Simple IF Statement

The most common usage involves a single conditional check resulting in one of two outcomes. This method is ideal for creating binary classifications or simple flags within your data model, such as labeling customers as active/inactive or products as high-stock/low-stock. The syntax below demonstrates how to assign a rating based on a numeric field.

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

This specific DAX expression creates a new calculated column named **Rating**. This column evaluates the measure contained within the **Points** column of the `my_data` table. If the value in **Points** is strictly greater than 20, the result is "Good"; otherwise, for any value 20 or less, the result defaults to "Bad." This technique provides immediate, clear segmentation based on a single numerical threshold.

Method 2: Constructing a Nested IF Statement

When simple binary logic is insufficient, the Nested IF Statement allows you to evaluate multiple sequential conditions. This structure is essential for multi-tier categorization or grading systems where more than two possible outcomes exist. Although effective, it is important to format nested IFs clearly for readability, as shown in the example below, which evaluates three potential outcomes.

```
Rating =
IF(
'my_data' < 20,
"Bad",
IF(
'my_data' <30,
"Good",
"Great"
)
)
```

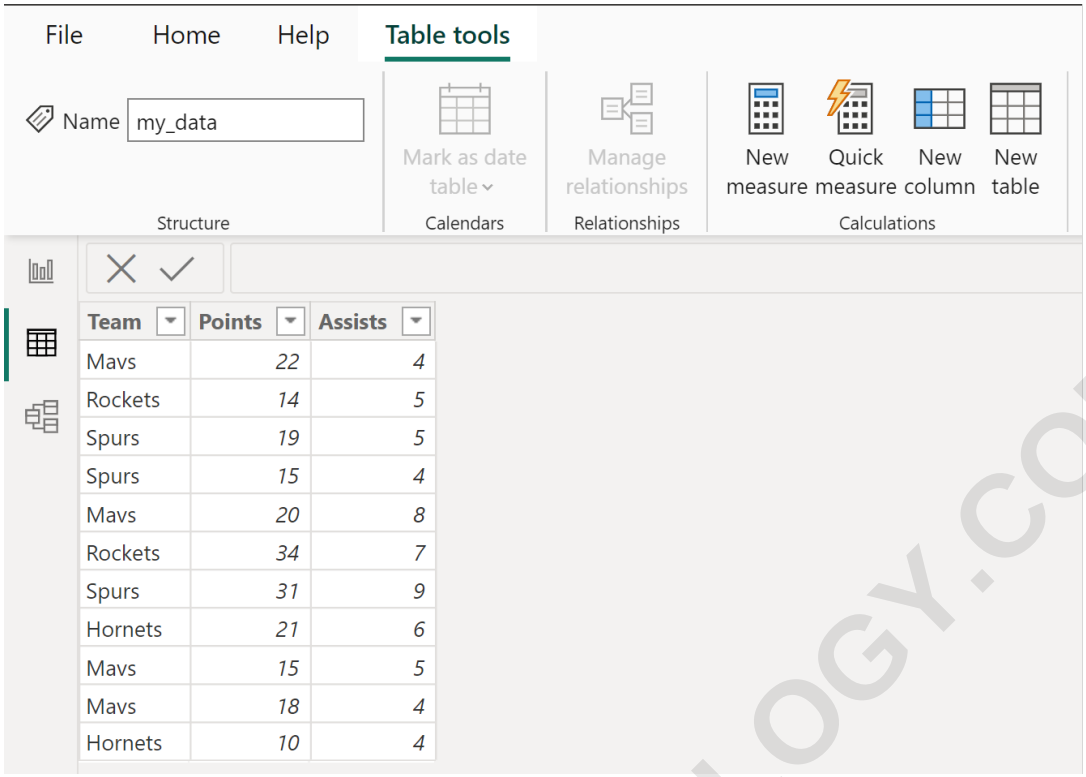
This complex expression also creates a new column named **Rating** but uses a sequential evaluation process. The conditions are checked one after the other, and the first condition that evaluates to true determines the returned value. The final result is the default outcome if all preceding conditions have been evaluated as false.

The logic of the Nested IF Statement is executed in the following order:

"Bad" is returned if the value in the **Points** column is less than 20 (Condition 1).
If Condition 1 was false (i.e., Points is 20 or higher), the second IF statement is evaluated: "Good" is returned if the value in **Points** is less than 30 (Condition 2).
If both Condition 1 and Condition 2 were false (i.e., Points is 30 or higher), the default value of "Great" is returned.

Setting Up the Demonstration Dataset

To illustrate how these formulas operate in a real-world context within Power BI, we will use a sample dataset named **my_data**. This table contains various data points, including a numerical column titled **Points**, which will serve as the basis for our conditional evaluations. Review the source data structure below before proceeding with the examples.

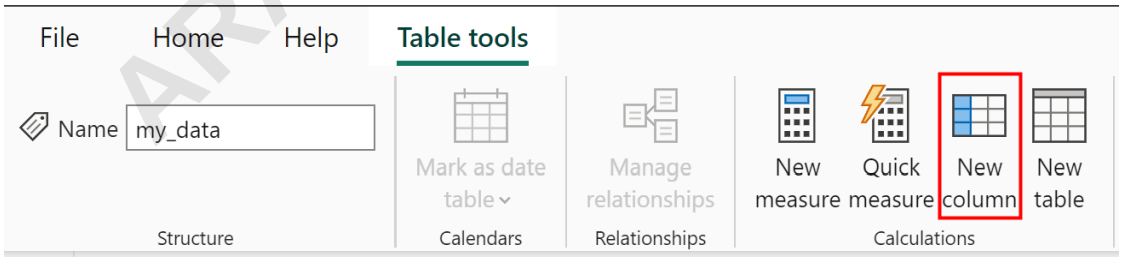


Team	Points	Assists
Mavs	22	4
Rockets	14	5
Spurs	19	5
Spurs	15	4
Mavs	20	8
Rockets	34	7
Spurs	31	9
Hornets	21	6
Mavs	15	5
Mavs	18	4
Hornets	10	4

Example 1: Creating a Simple Rating with IF

In this first practical scenario, our objective is to append a new column to the **my_data** table. This column, labeled **Rating**, must contain the label "Good" if the corresponding value in the **Points** column strictly exceeds 20, and otherwise assign the label "Bad." This requires a straightforward application of the basic IF statement to create a binary classifier.

To begin, navigate to the table tools in the Power BI interface and click the **New column** option:

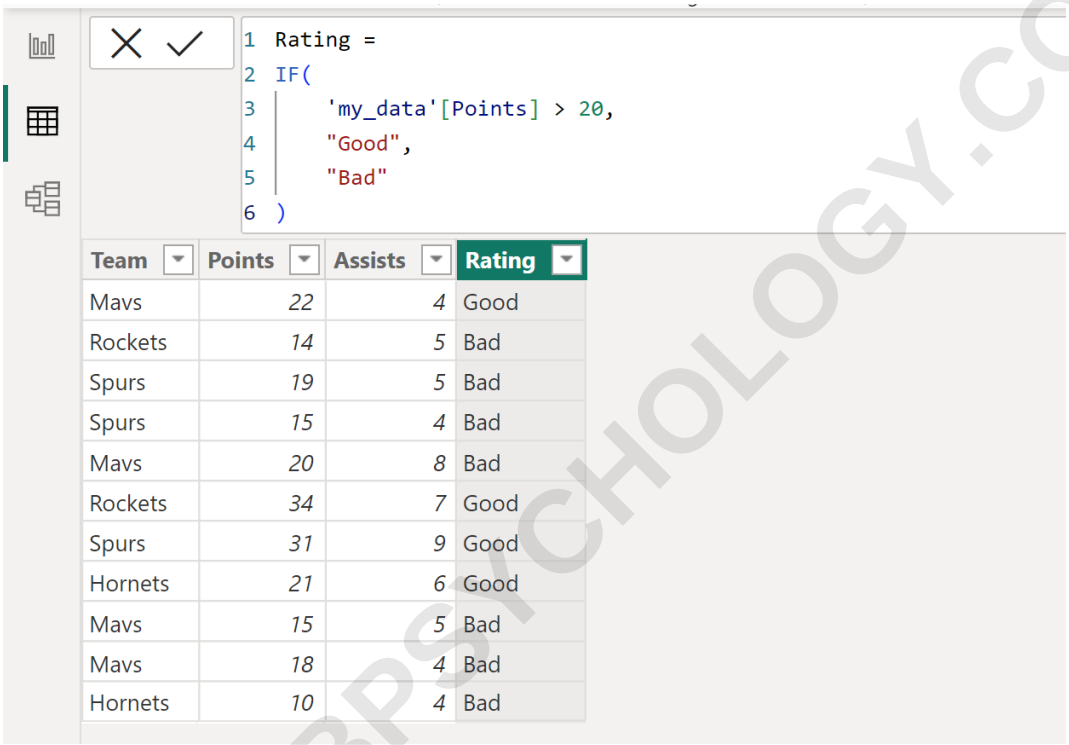


Next, input the following definitive formula into the DAX formula bar. Note how the logical test (`> 20`) dictates the outcome for every row:

Rating =
IF(

'my_data' > 20,
"Good",
"Bad"
)

Executing this formula successfully generates the new calculated column, **Rating**, which clearly reflects the binary classification based on the **Points** threshold of 20, thereby segmenting the data immediately:

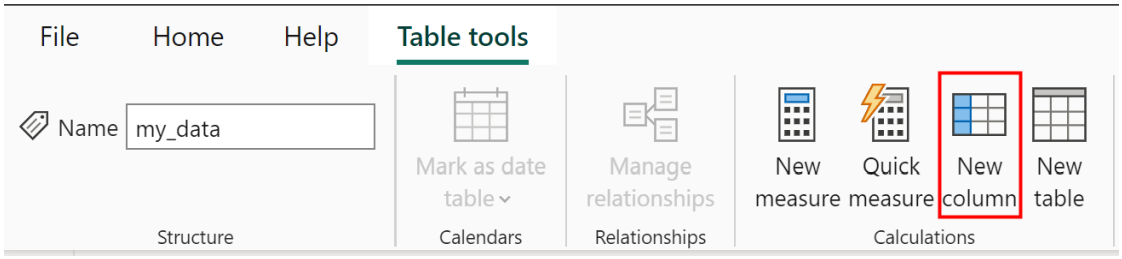


Team	Points	Assists	Rating
Mavs	22	4	Good
Rockets	14	5	Bad
Spurs	19	5	Bad
Spurs	15	4	Bad
Mavs	20	8	Bad
Rockets	34	7	Good
Spurs	31	9	Good
Hornets	21	6	Good
Mavs	15	5	Bad
Mavs	18	4	Bad
Hornets	10	4	Bad

Example 2: Implementing Complex Logic with Nested IF

For situations requiring more granular categorization--such as grading performance across multiple tiers--the Nested IF Statement becomes indispensable. Our goal here is to assign "Bad" for scores below 20, "Good" for scores between 20 and 29, and "Great" for scores 30 and above. This multi-tiered structure requires the use of one IF function embedded within the false result parameter of another.

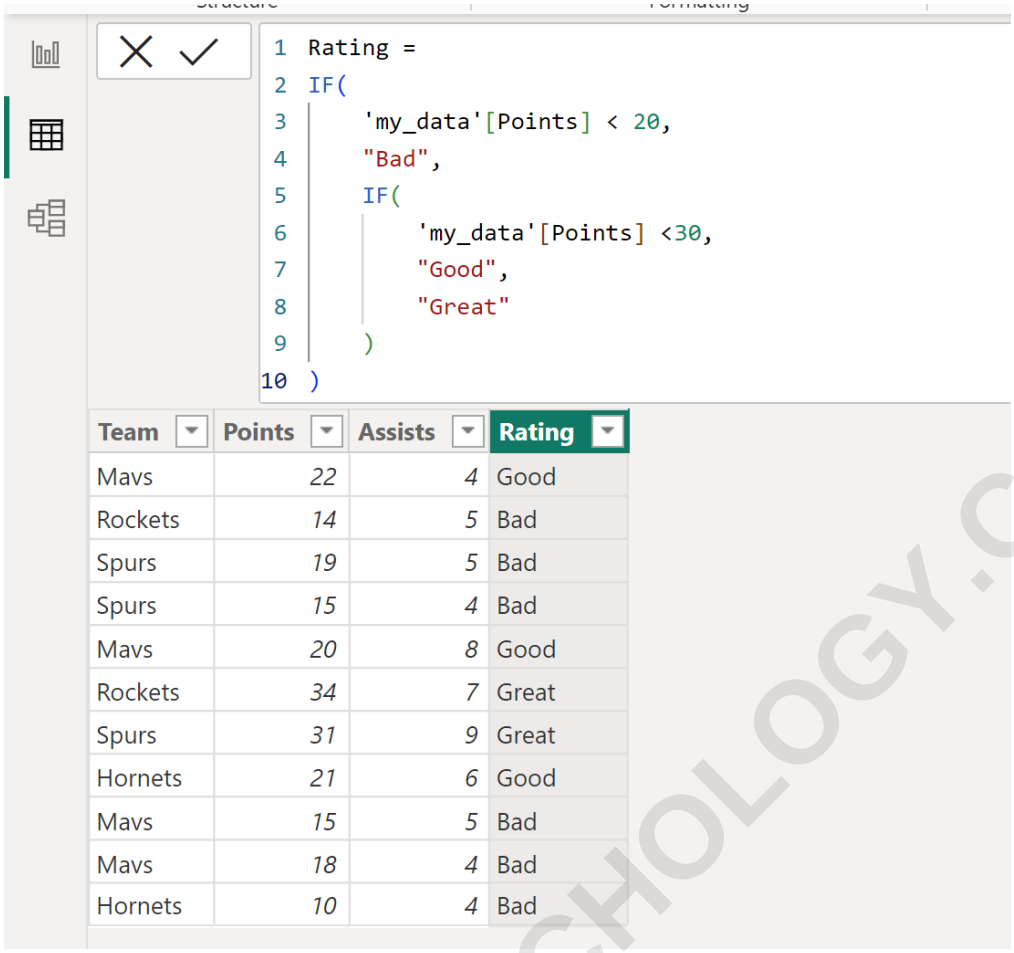
As before, initiate the process by clicking the **New column** icon located in the modeling ribbon of Power BI:



The complexity of the logic necessitates the use of two interconnected IF statements nested within one another. Input this formula carefully into the DAX formula bar, ensuring the syntax aligns precisely with the logic required:

```
Rating =
IF(
'my_data' < 20,
"Bad",
IF(
'my_data' <30,
"Good",
"Great"
)
)
```

Upon execution, the resulting **Rating** column accurately reflects the three-tiered classification, assigning "Bad," "Good," or "Great" based on the sequential evaluation of the **Points** data, demonstrating the power of conditional structuring in data analysis:



```
Rating =
IF(
    'my_data'[Points] < 20,
    "Bad",
    IF(
        'my_data'[Points] <30,
        "Good",
        "Great"
    )
)
```

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

Alternatives and Further DAX Exploration

While the IF function is fundamental for conditional logic, it is important to note that deeply nested IF structures can become cumbersome and difficult to maintain. For scenarios involving three or more possible outcomes, DAX also offers the more efficient **SWITCH** function. The **SWITCH** function evaluates an expression against a list of values, returning a result associated with the first matching value, often leading to cleaner code than multiple nested IF statements. It is highly recommended to consult the official documentation for advanced usage and alternative conditional functions.

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