**IF Statements in Power Query M (Including Nested IFs, OR, AND)**

### **The IF Function in Power Query**

The IF function is essential in your Power Query toolkit. It enables you to compare a value with a specified condition, leading to two possible results. It’s known as a conditional statement.

The function first checks if a condition is met. If the condition is true, Power Query returns one result. If it’s false, it returns another result.

### **Basic Example**

Let’s start with a straightforward example to grasp the syntax. Imagine you have a list of numbers and want to label them as “positive value” or “negative value”. You could use an IF statement like this:

**if** [Value] > 0 **then** "Positive value" **else** "Non-positive value"

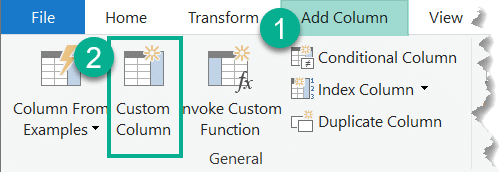
let’s make an IF statement together. Say you have a list of numbers and want to label them as “high” or “low”. You could use an IF statement like this:

**if** [Price] > 10 **then** "High Price" **else** "Low Price"

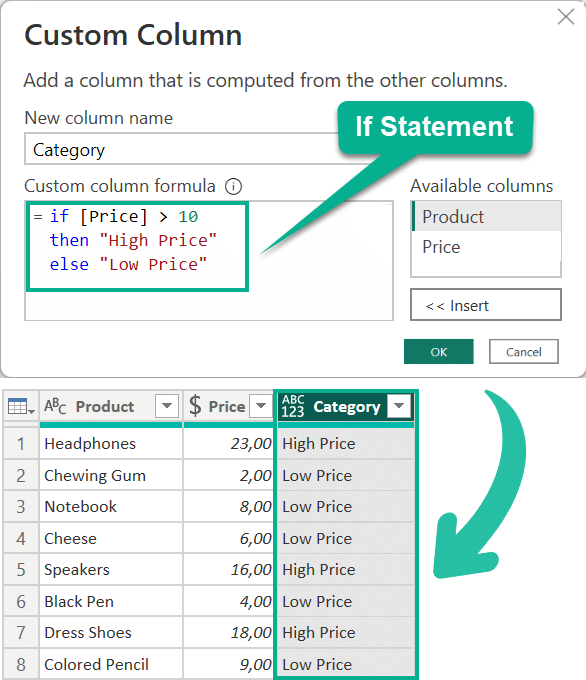
Here’s how to create your IF statement:

* Go to the **Add Column** tab in the ribbon
* Select **Custom Column**
* Provide a Column Name
* Enter your IF statement
* Click OK

After pressing the ‘Custom Column’ button in the ‘Add Column’ tab, a new column will be added to your dataset.

[](https://gorilla.bi/wp-content/uploads/2023/03/Create-Custom-Column-in-Power-Query.png)

The ‘Custom Column’ pop-up will appear, where you can provide both a **Column Name** and a **Column Formula**. Be sure to write “if”, “then”, and “else” in lowercase, and click ‘OK’.

[](https://gorilla.bi/wp-content/uploads/2023/03/Basic-Example-IF-statement-in-Power-Query.png)

Voilà! You now have a table with a newly created column.

Under the hood, Power Query generates this code:

Table.AddColumn(

#"Changed Type",

"Category",

**each** **if** [Price] > 10 **then** "High Price" **else** "Low Price" )

Here’s a quick breakdown:

1. The **Table.AddColumn** function adds a new column to our table, called “Category.
2. The ‘**each’**keyword applies the IF statement to every row in the table.
3. The condition checks if the number is **greater than 10.**
4. If the condition is true, “**High Price**” is returned. If false, “**Low Price**” is returned.

### Comparison Operators

Comparison operators let you compare values within your conditions. Here are the most common operators in Power Query:

| **Operator** | **Description** |
| --- | --- |
| > | Greater than |
| >= | Greater than or equal |
| < | Less than |
| <= | Less than or equal |
| = | Equal |
| <> | Not equal |

For instance, to check if a product’s revenue equals $500, use the equal operator like this:

**if** [Revenue] = 500 **then** ... **else** ...

To classify ages 18 or younger as “**Youth**” and the rest as “**Other**“:

**if** [Age] <= 18 **then** "Youth" **else** "Other"

You get the idea. Now, what if you want to combine multiple conditions?

### Logical Operators

Logical operators allow you to combine multiple conditions. The main logical operators are:

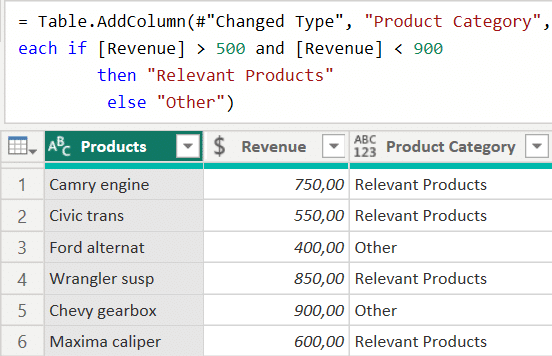
| **Operator** | **Description** |
| --- | --- |
| and | Both conditions must be true |
| or | At least one condition must be true |
| not | Condition must not be true |

## **Applying Operators to IF Statements**

### **IF statement with AND Logic**

It’s also useful to know how to add **if statements with ‘and’ logic** to test multiple conditions. Let’s say you want to find products with revenue greater than $500 and less than $900. To test this, your conditional if statement should include two conditions. You’d use the **and operator** like this:

**if** [Revenue] > 500 **and** [Revenue] < 900 **then** "Relevant Products" **else** "Other"

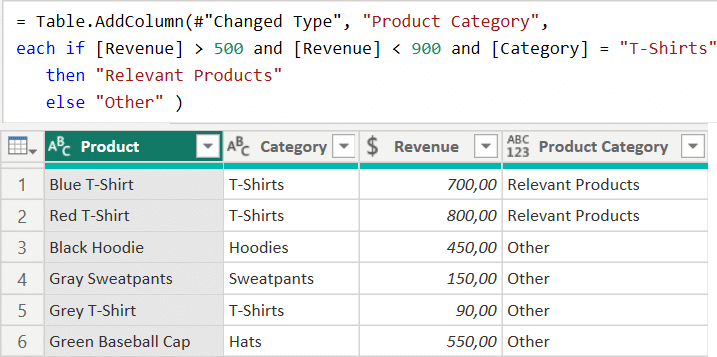
[](https://gorilla.bi/wp-content/uploads/2023/04/If-Statement-with-And-Logic-in-Power-Query.png)

This example only included a single **and**operator. You can add more conditions to the same expression:

**if** [Revenue] > 500 **and** [Revenue] < 900 **and** [Category] = "T-Shirts"

**then** "Relevant Products"

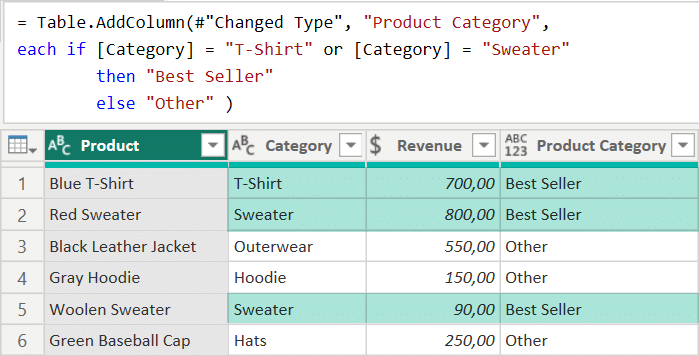
**else** "Other"

[](https://gorilla.bi/wp-content/uploads/2023/04/If-Statement-with-multiple-And-Logic.png)

### **IF statement with OR Logic**

In some cases, you may want to test whether one of multiple conditions is true by combining if with or.  Imagine you are looking for the top 2 selling clothing categories. In this case **T-shirt** or **Sweater**. You could use the **or operator** in this way:

**if** [Category] = "T-Shirt" **or** [Category] = "Sweater" **then** "Best Seller" **else** "Other"

[](https://gorilla.bi/wp-content/uploads/2023/04/If-Statement-with-Or-Logic.png)

### **IF statement with NOT Logic**

Sometimes, you need to test if something is not true, either to exclude a condition or because it’s shorter to write the negative form. For instance, let’s say you want to increase the price of everything except lemons by 10%

You can add the **not** operator right after the word **if**. Just make sure to **put the entire condition between parentheses**.

**if** **not** ( [Food] = "Lemon" ) **then** [Price] \* 1.1 **else** [Price]

### **Nested IF Statements**

Sometimes, you need to check multiple conditions in sequence, and that’s when nested IF statements come in handy. You can include an IF statement inside another IF statement:

**if** [Condition1] **then** [Value1] **else** **if** [Condition2] **then** [Value2] **else** [Value3]

Without formatting, any code is difficult to read. So from now on, I will serve you formatted code. The results are identical, they are simply much easier to read. Here’s the same code but formatted:

**if** [Condition1]

**then** [Value1]

**else** **if** [Condition2]

**then** [Value2]

**else** [Value3]

Let’s say you want to categorize products by revenue as “Low”, “Medium”, or “High”. You could use nested IF statements like this:

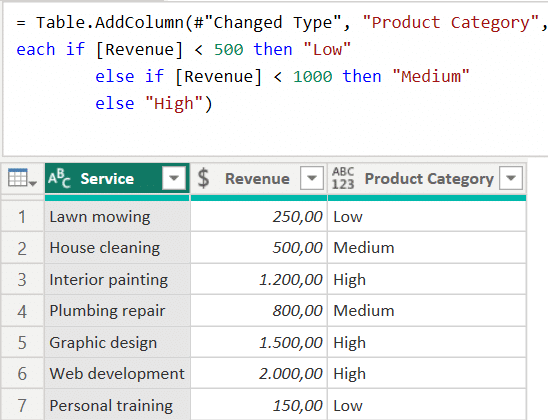
**if** [Revenue] < 500

**then** "Low"

**else** **if** [Revenue] < 1000

**then** "Medium"

**else** "High"

[](https://gorilla.bi/wp-content/uploads/2023/04/Nested-if-statements-in-Power-Query.png)

### **IF Statement with Multiple Conditions**

You can also use logical operators to create more complex conditions. For instance, when you want to find products with revenue between $500 and $1000 and more than 50 units sold. You’d use both **and**and **or**operators like this:

**if** [Revenue] > 500 **and** [Revenue] < 1000 **and** [UnitsSold] > 50

**then** "Match"

**else** "No Match"

### **Using the IF Statement to Categorize Data**

IF statements are great for categorizing data. Let’s say you have a table with student grades, and you want to add a column that shows the grade category (A, B, C, D, or F):

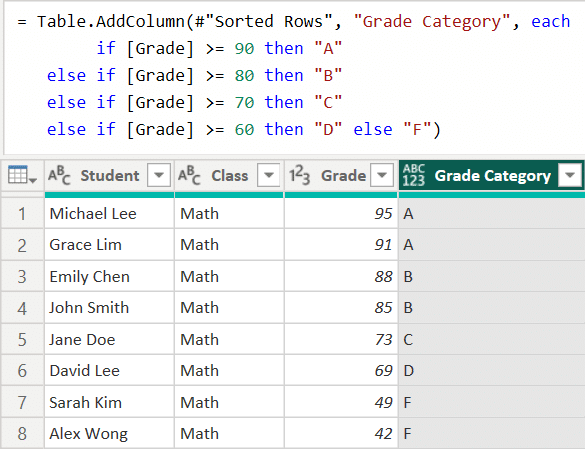
**if** [Grade] >= 90 **then** "A"

**else** **if** [Grade] >= 80 **then** "B"

**else** **if** [Grade] >= 70 **then** "C"

**else** **if** [Grade] >= 60 **then** "D"

**else** "F"

[](https://gorilla.bi/wp-content/uploads/2023/04/Categorizing-data-with-if-statement-in-power-query.png)

## **Working with Different Data Types**

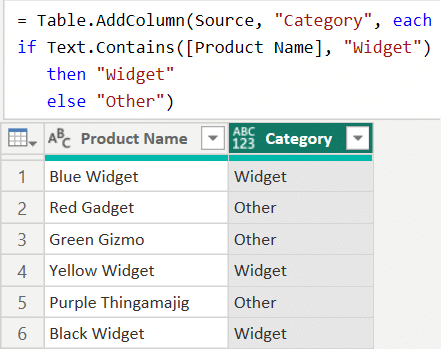
### **Working with Text**

Continuing with our IF statement journey, let’s explore how to work with text values. For instance, to check if a product name contains a specific keyword and categorize it you can use [Text.Contains](https://powerquery.how/text-contains/" \t "_blank):

**if** Text.Contains( [Product Name], "Widget") // Does [ProductName] contain "Widget

**then** "Widget" // if yes, return "Widget"

**else** "Other" // else return "Other"

[](https://gorilla.bi/wp-content/uploads/2023/04/Using-text-functions-in-an-if-statement.png)

Or test if the combination of two text fields matches another field:

**if** [FirstName] & [LastName] = [FullName] **then** true **else** false

You can also return a value whenever a part of a text string matches your test. Imagine that invoices starting with the text “MAR” relate to market revenue:

**if** Text.Start( [InvoiceID], 3 ) = "MAR" **then** "Marketing Revenue" **else** "Other"

### **Working with Dates**

Dates are another common data type you’ll work with in Power Query. You can use IF statements to manipulate and categorize dates. For example, you can check if a date falls within a specific range:

**if** [OrderDate] >= #date(2023, 1, 1) **and** [OrderDate] <= #date(2023, 12, 31)

**then** "2023 Order"

**else** "Other Year"

Or, you could categorize dates by day of the week using [Date.DayOfWeek](https://powerquery.how/date-dayofweek/" \t "_blank):

**if** Date.DayOfWeek([OrderDate]) = 0 **then** "Sunday"

**else** **if** Date.DayOfWeek([OrderDate]) = 6 **then** "Saturday"

**else** "Weekday"