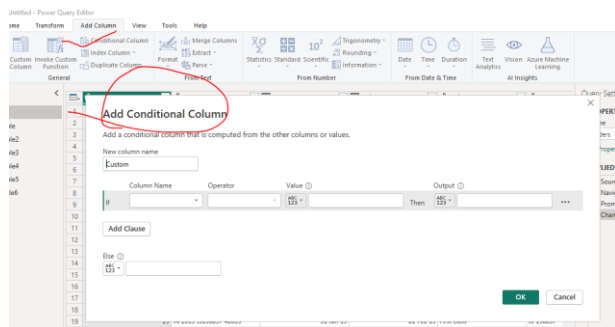


Chapter-20 Creating a Conditional Column in Power BI

A **Conditional Column** in Power BI allows you to create new columns based on specific conditions or rules applied to your existing data. This is useful when you need to categorize, transform, or filter data based on certain criteria. You can use **if-else statements** to create conditions that will modify or create new values in a column.

Steps to Create a Conditional Column in Power BI



Step 1: Open Power Query Editor

1. Open your Power BI Desktop file.
2. In the Home tab, click **Transform Data** to open the **Power Query Editor**.

Step 2: Select the Table and Column

1. In the **Queries** pane on the left, select the table you want to add the conditional column to.
2. Identify the column you want to use for applying the condition.

Step 3: Add a Conditional Column

1. In the Power Query Editor, go to the **Add Column** tab in the ribbon.
2. Click on **Conditional Column**.

Step 4: Define the Condition

1. A dialog box will appear where you can specify the conditions for your new column.
 - **Column Name:** Give your new conditional column a name.

- **Column to Base Condition On:** Select the existing column that you want to use for the condition.
 - **Operator:** Choose the type of condition you want to apply (e.g., **equals**, **greater than**, **less than**, **contains**, etc.).
 - **Value:** Enter the value that you want to compare against.
 - **Output (Result):** Define what the new column will display when the condition is met (e.g., "Yes", "High", etc.).
 - **Else Output:** Define what the new column will display if the condition is not met (e.g., "No", "Low", etc.).
2. You can add more conditions by clicking the **Add Clause** button. This will allow you to create a series of conditional checks (like an "else if" logic).
 - For each new condition, set the **Column**, **Operator**, and **Value**, and define the output for both the condition being met and the condition not being met.

Step 5: Apply the Conditional Column

1. Once you've defined the conditions, click **OK**. The new column will be added to the table.
2. You'll see the new conditional column appear in the Power Query Editor preview.

Step 6: Close and Apply

1. After creating the conditional column, go back to the **Home** tab in the Power Query Editor.
2. Click **Close & Apply** to apply the changes and return to the Power BI report view.

Example of Creating a Conditional Column

Let's say you have a sales table and you want to categorize sales amounts into "High", "Medium", and "Low" based on specific thresholds.

1. **Select the Sales Table** in the Power Query Editor.
2. Go to **Add Column > Conditional Column**.
3. Name the new column "Sales Category".
4. Set the condition:
 - **If Sales Amount** is greater than **1000**, output **"High"**.
 - **Else If Sales Amount** is greater than **500**, output **"Medium"**.
 - **Else**, output **"Low"**.
5. Click **OK**.

Your new column will categorize sales data as "High", "Medium", or "Low" based on the sales amount.

Advanced Conditional Column (Using Multiple Conditions)

You can also create more complex conditional columns using logical operators. For example, if you want to create a column that assigns different categories based on **both Sales Amount** and **Product Category**, you can:

1. Select the **Product Category** column and **Sales Amount** column in the conditional column setup.
 2. Set up conditions such as:
 - **If Product Category = "Electronics" and Sales Amount > 1000**, output **"Premium Electronics"**.
 - **Else If Product Category = "Electronics" and Sales Amount <= 1000**, output **"Standard Electronics"**.
 - **Else**, output **"Other"**.
-

Benefits of Using Conditional Columns

1. **Data Transformation:** Conditional columns help transform raw data into more meaningful categories for better analysis and reporting.
 2. **Custom Logic:** You can create custom logic based on multiple conditions or even combine different fields for more sophisticated rules.
 3. **Improves Visualizations:** By categorizing or grouping data, it makes it easier to visualize and interpret the data in charts, tables, and reports.
-

Best Practices for Conditional Columns

1. **Keep Conditions Simple:** Avoid overly complex conditions in a single column, as it can make the model harder to maintain.
2. **Test Logic:** Ensure that the logic behind the conditional column works correctly by testing with a sample of your data.
3. **Document Your Logic:** It can be helpful to comment or document the conditions you've applied, especially when working with complex logic, so others (or yourself) can understand the reasoning behind the categorization.
4. **Avoid Too Many Conditional Columns:** While useful, too many conditional columns can increase model complexity. If possible, try to consolidate logic into fewer columns.