

How to Easily Apply Conditional Formatting to Text in Power BI

Authored by
stats writer

January 27, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Easily Apply Conditional Formatting to Text in Power BI*.
PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=128023>

Conditional formatting is an immensely powerful feature integrated into Power BI, designed to empower users to dynamically highlight crucial data points based on a set of predefined, logical conditions. When dealing with quantitative data, applying this formatting is generally straightforward, but applying it effectively to a text column requires a slightly nuanced approach. The core challenge is that Power BI's formatting rules often rely on a numeric scale to determine color gradients or icons. Therefore, to successfully apply visual rules to descriptive text, users must first establish a numeric proxy for those text values. This article will provide a detailed, step-by-step guide on how to implement this technique, enabling you to dramatically enhance the visual clarity and analytical depth of your reports, making it far simpler for stakeholders to quickly grasp critical insights.

Power BI: Applying Conditional Formatting to Text Columns Through Numeric Mapping

Understanding Conditional Formatting in Power BI

Conditional formatting serves as a critical tool for rapid data interpretation within any reporting environment. Instead of forcing the user to manually scan long lists of text, conditional formatting automatically applies visual cues--such as background colors, font changes, or data bars--when certain conditions are met. While features like color scales work seamlessly with numerical measures (e.g., highlighting sales figures based on their value), applying similar robust, rule-based formatting directly to categorical strings within a text column presents a specific technical hurdle in Power BI.

The limitation stems from how Power BI processes formatting rules. When we want to use a color scale, the system expects a continuous or ordinal numerical measure to anchor the visual changes. Text fields, being categorical, do not inherently possess this numerical order. Consequently, if you attempt to apply conditional formatting directly to a text column, you will find that the options available are often limited to basic formatting based on the field value itself, which is inefficient for scalable, rule-based coloring. Our goal is to bypass this constraint by creating a calculated column that translates each text category into a unique, measurable integer, thereby unlocking the full potential of Power BI's conditional formatting engine.

Consider a scenario where you want to highlight different sports teams based on their division or status. You want Team A to be green, Team B to be yellow, and Team C to be red. If you look at the raw data, as illustrated below, you can see the need for immediate visual differentiation. Achieving this outcome requires careful preparation of the data model before the formatting rules can be deployed in the visualization layer, ensuring that the visual outcome is both precise and reflective of the underlying textual categories.

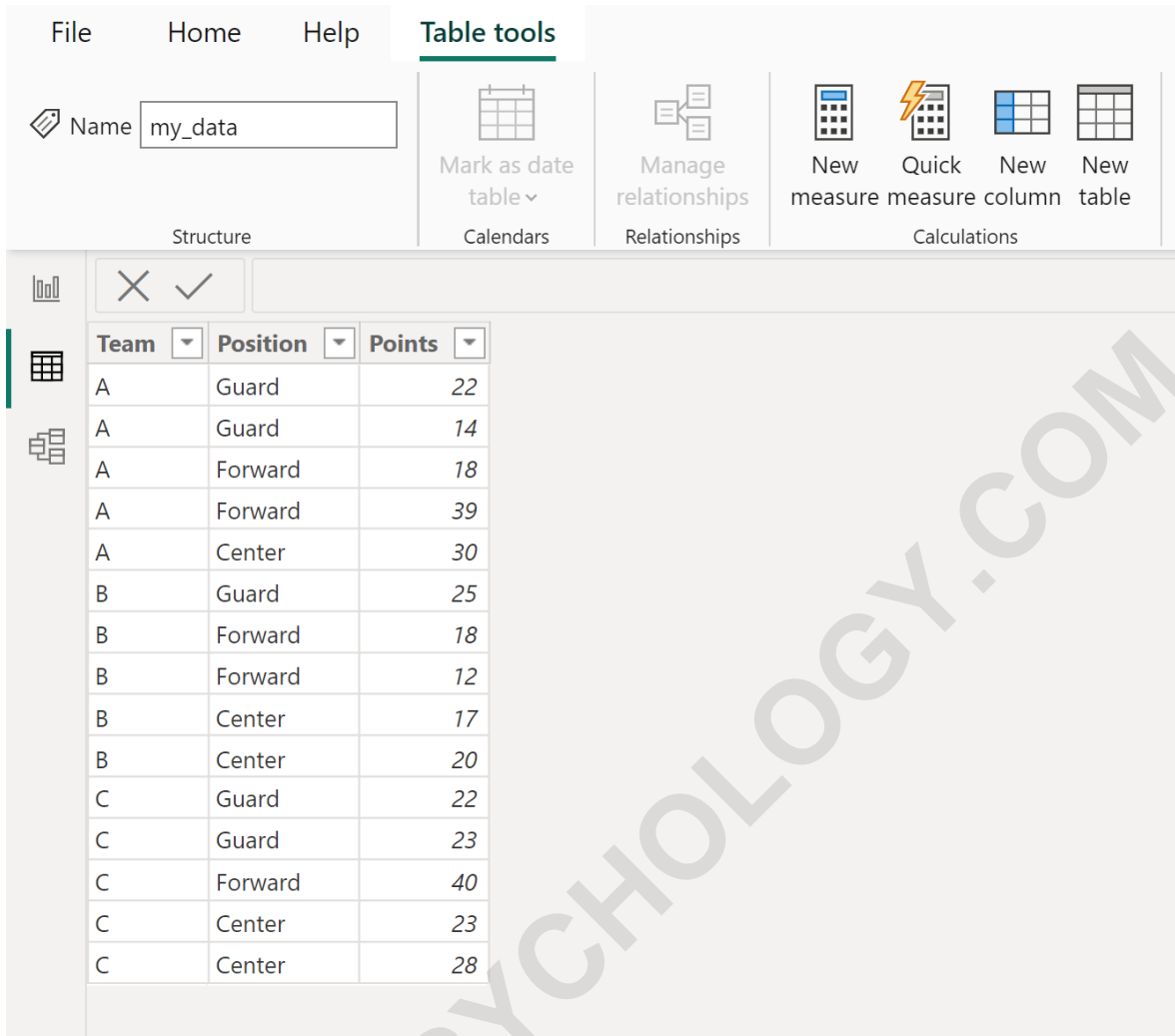
Team	Position	Points
A	Center	30
A	Forward	18
A	Forward	39
A	Guard	14
A	Guard	22
B	Center	17
B	Center	20
B	Forward	12
B	Forward	18
B	Guard	25
C	Center	23
C	Center	28
C	Forward	40
C	Guard	22
C	Guard	23
Total		

Fortunately, this sophisticated visual requirement is achievable through the strategic application of DAX (Data Analysis Expressions) formulas to introduce the necessary numeric mapping. The following detailed steps will walk you through the entire process, starting from data import and moving through the crucial creation of the numeric proxy column, culminating in the final visualization setup.

Step 1: Importing and Preparing the Data Set

The first and most fundamental step in any Power BI project is importing the source data. For this demonstration, we utilize a simple data structure containing information about various basketball players. This example data set includes player statistics and, crucially, a categorical field titled **Team**, which contains the text values we intend to conditionally format. Ensure that your data is correctly loaded into the Power BI Data View and that all columns are appropriately typed.

Upon successful import, your data table should be visible in the Power BI desktop environment, typically looking similar to the structure shown in the image below. Verification of the data quality and structure at this stage is essential, as any errors in the source data or during the import process will inevitably cascade into the subsequent steps, particularly when defining complex DAX logic.



The screenshot shows the Power BI Desktop interface with the 'Table tools' ribbon selected. The ribbon contains several groups of options: 'Name' (my_data), 'Mark as date table', 'Manage relationships', and 'Calculations' (New measure, Quick measure, New column, New table). Below the ribbon is a table with the following data:

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

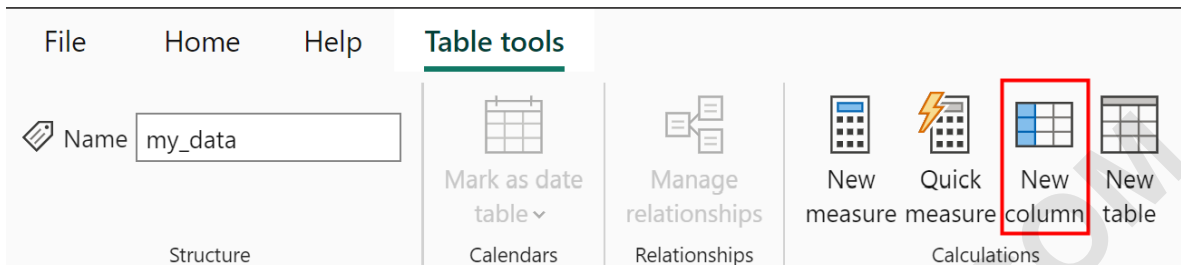
Our primary focus is the **Team** column. Since this column contains the strings 'A', 'B', and 'C', which are purely categorical, we need a mechanism to assign a numerical weight or identifier to each unique category. This numeric identifier will serve as the engine for our conditional formatting rules, allowing us to define precisely what color corresponds to Team 'A' (e.g., 1), Team 'B' (e.g., 2), and Team 'C' (e.g., 3). This preparation sets the foundation for the crucial DAX calculation in the next step.

Step 2: Creating a Numeric Proxy Column using DAX

The inability to apply conditional formatting directly based on specific text values necessitates the creation of an intermediary, numerical column. This new column acts as a proxy, translating the text values of the **Team** column into a quantitative scale. This is achieved by utilizing the powerful capabilities of the DAX language within Power BI.

To begin this process, navigate to the **Table tools** ribbon in Power BI Desktop. Locate and click

the **New column** icon. This action opens the formula bar, allowing you to input the DAX expression that will perform the mapping calculation. It is imperative that this new column is created in the table where the original text data resides, as the DAX expression will reference the existing categorical values row by row.



The specific DAX formula we employ here is the SWITCH function. The **SWITCH** function is highly effective for performing conditional logic by evaluating an expression against a list of values and returning a result corresponding to the first match. In our case, the expression is the value in the **Team** column, and the results are the numeric identifiers we assign. The subsequent section provides a detailed breakdown of this critical formula.

Deconstructing the SWITCH Formula

The formula used to create the numeric proxy column, which we will name **New**, is structured as follows. We are essentially telling Power BI: evaluate the value in the 'my_data' column; if it matches 'A', return 1; if it matches 'B', return 2; and if it matches 'C', return 3. The final, standalone number (4 in this case) acts as the default value if none of the preceding conditions are met, ensuring every row receives a numeric assignment.

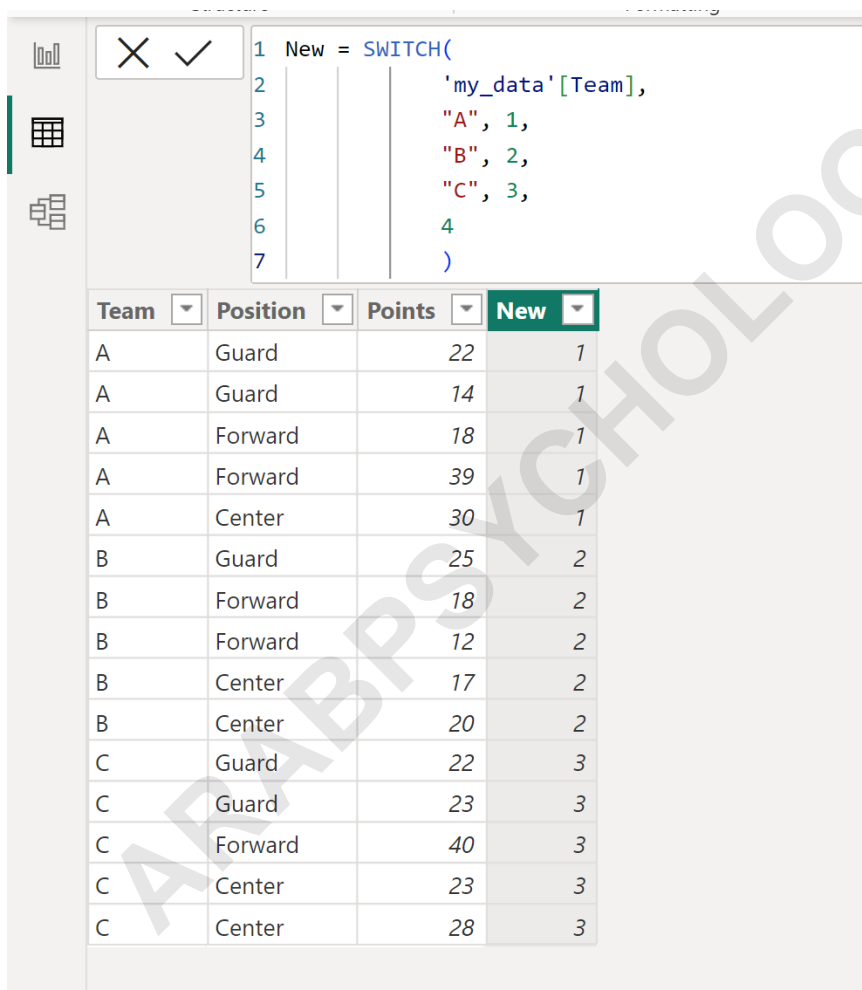
Type the following DAX code meticulously into the formula bar:

```
New = SWITCH(  
'my_data',  
"A", 1,  
"B", 2,  
"C", 3,  
4  
)
```

This formula leverages the logical flexibility of the SWITCH function. The first argument specifies the column whose values are being evaluated ('my_data'). The subsequent pairs of arguments define the mapping: the text value (e.g., "A") followed by the desired numeric outcome (e.g., 1).

This systematic mapping is vital because the numerical result, 1, 2, or 3, is what the conditional formatting engine will use to trigger the corresponding background color rule.

Once the formula is confirmed, Power BI calculates the new column, **New**, automatically populating it with the integers 1, 2, 3, or 4, directly corresponding to the text in the **Team** column. This intermediary column successfully bridges the gap between the categorical nature of the text data and the numerical requirements of Power BI's robust formatting tools. The data model now possesses the necessary components to apply sophisticated conditional formatting rules, as demonstrated by the resulting table below showing the newly calculated column alongside the original text.



```

1 New = SWITCH(
2     'my_data'[Team],
3     "A", 1,
4     "B", 2,
5     "C", 3,
6     4
7 )

```

Team	Position	Points	New
A	Guard	22	1
A	Guard	14	1
A	Forward	18	1
A	Forward	39	1
A	Center	30	1
B	Guard	25	2
B	Forward	18	2
B	Forward	12	2
B	Center	17	2
B	Center	20	2
C	Guard	22	3
C	Guard	23	3
C	Forward	40	3
C	Center	23	3
C	Center	28	3

Step 3: Inserting the Visualization Table

With the numerical proxy column established, the focus shifts to the visualization layer. Navigate to the **Report View** in Power BI Desktop. Here, we need to create a visual container--in this case, a standard Table visual--to display the data and apply the formatting. Insert a new Table visual onto the canvas and add the fields you wish to display. Crucially, ensure that both the original **Team**

column (the text field you want to color) and the calculated **New** column (the numeric field that dictates the color) are present in the visual's fields list.

While the **New** column is required for defining the conditional formatting rules, it does not necessarily need to be visible to the end-user. It can be present in the Columns field well of the visualization but hidden from display if preferred, although for demonstration purposes, it is often useful to keep it visible initially. The primary column displayed should be the **Team** column, as this is the element whose appearance we are targeting for modification.

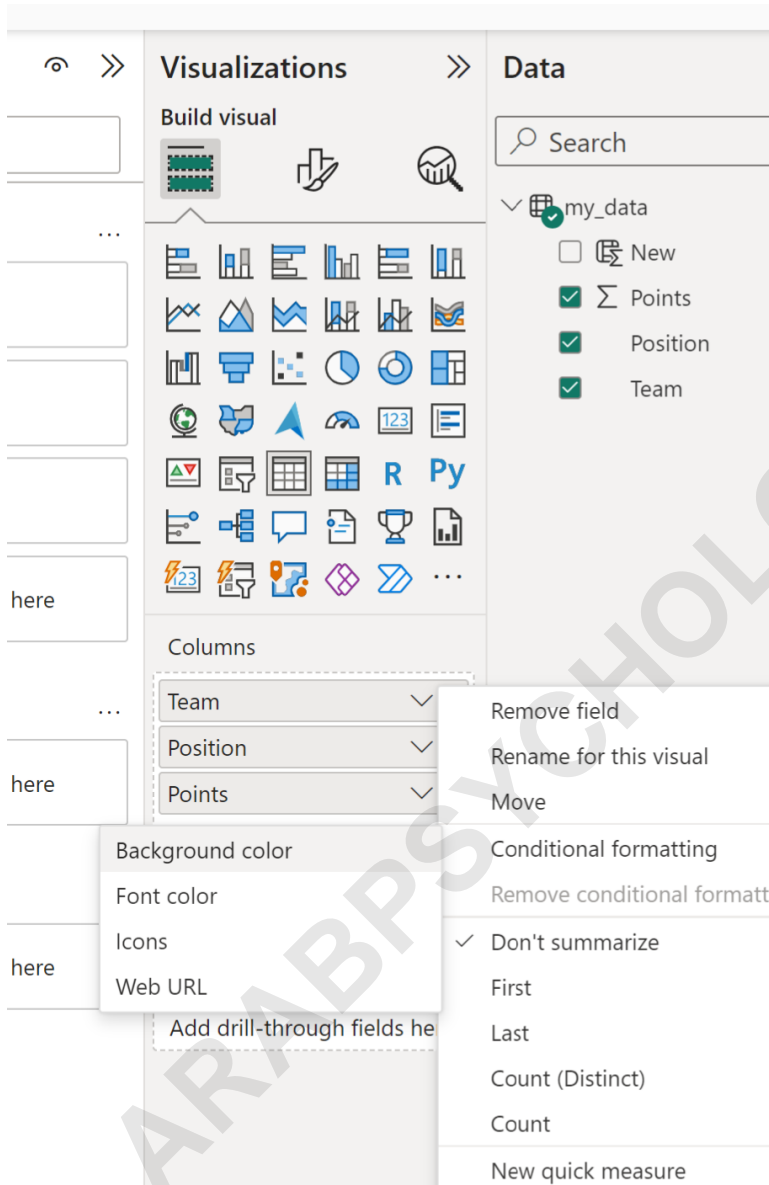
The resulting table visualization, before any formatting is applied, will simply list the player data, including the teams and their corresponding numeric identifiers. This forms the canvas upon which the powerful visual rules will soon be painted, transitioning the raw data into an insightful, color-coded report element.

Team	Position	Points
A	Center	30
A	Forward	18
A	Forward	39
A	Guard	14
A	Guard	22
B	Center	17
B	Center	20
B	Forward	12
B	Forward	18
B	Guard	25
C	Center	23
C	Center	28
C	Forward	40
C	Guard	22
C	Guard	23

Step 4: Applying Conditional Formatting Rules

This is the final and most critical step, where we instruct Power BI to use the values in the **New** numeric column to drive the color changes in the **Team** text column. To initiate the formatting process, locate the Table visual on the canvas. In the Fields pane (or the visualization pane), find the original text field, **Team**, under the **Columns** label.

Click the dropdown arrow next to the **Team** field. Hover over the **Conditional formatting** option in the context menu that appears, and then select **Background color**. This action opens the dedicated conditional formatting dialogue box, allowing precise control over how colors are applied based on data values.



Within the conditional formatting window, meticulous configuration is necessary. First, change the formatting style from 'Color scale' to **Rules**, as we are defining discrete color assignments rather than a gradient. Second, and most importantly, locate the setting **What field should we base this on?** and select our newly created numeric column, **New**, from the dropdown list. Next, set the **Summarization** type to **Minimum**, ensuring the rules are evaluated correctly for each row.

Finally, we must define the specific rules that link the numeric identifier to the desired color. Create

a rule for each categorical value (1, 2, 3, etc.). For instance, Rule 1 might state: If value is greater than or equal to 1 (number) and less than 2 (number), set color to Green. Rule 2: If value is greater than or equal to 2 (number) and less than 3 (number), set color to Yellow. Ensure the boundaries are set precisely to capture only the integer associated with the desired text value. This systematic rule definition ensures that the color chosen corresponds exactly to the original text category.

Background color - Team

Format style: Rules (dropdown) Apply to: Values only (dropdown)

What field should we base this on?: Min of New (dropdown) Summarization: Minimum (dropdown)

Rules: ↑↓ Reverse color order + New rule

If value	=	1	Number	then	Red	↑ ↓ ×
If value	=	2	Number	then	Blue	↑ ↓ ×
If value	=	3	Number	then	Green	↑ ↓ ×
If value	=	4	Percent	then	Grey	↑ ↓ ×

[Learn more about conditional formatting](#) OK Cancel

Once all rules are defined and you click **OK**, the conditional formatting is instantly applied. The background of the text strings in the **Team** column will now dynamically change based on the underlying value of the proxy column, **New**. This completes the process, successfully applying complex, rule-based formatting to a categorical text field, greatly enhancing data comprehension and visual engagement within the Power BI report.

Team	Position	Points
A	Center	30
A	Forward	18
A	Forward	39
A	Guard	14
A	Guard	22
B	Center	17
B	Center	20
B	Forward	12
B	Forward	18
B	Guard	25
C	Center	23
C	Center	28
C	Forward	40
C	Guard	22
C	Guard	23
Total		

Conclusion: Enhancing Power BI Visualizations

The strategy of creating a numeric proxy column using [DAX](#) is an essential workaround for applying sophisticated [conditional formatting](#) to text-based data in [Power BI](#). By mapping categorical strings to measurable integers, report developers gain access to the full suite of formatting tools, enabling them to convey complex information visually and intuitively. This technique is not limited solely to background colors; it can be extended to control font color, icons, and even data bars, providing limitless possibilities for visual storytelling and data analysis within any large [data set](#).

Mastering this method ensures that your reports are not only accurate but also highly accessible and digestible for all audiences. The ability to visually isolate and highlight key performance indicators (KPIs) or category statuses directly within tables or matrices significantly reduces the cognitive load on the user, driving faster decision-making. We have demonstrated that while direct formatting of text values may be restrictive, a structured approach involving DAX manipulation offers a robust and scalable solution for dynamic reporting needs.

For those interested in exploring further enhancements and advanced techniques in Power BI, the following resources provide additional guidance on performing other common data manipulation and visualization tasks, building upon the foundational knowledge presented here.

Related Power BI Tutorials

To continue developing expertise in data visualization and analysis using Power BI, consider exploring tutorials related to advanced DAX calculations, creating custom visual hierarchies, or implementing dynamic security based on data fields.

Advanced DAX Techniques: Learning how to use iterative functions or complex context transitions can unlock deeper analytical capabilities.

Custom Visualizations: Explore how to import and configure custom visuals from the marketplace to present data in unique ways.

Data Modeling Best Practices: Understand the importance of efficient table relationships and star schema designs for optimal report performance.

ARABPSYCHOLOGY.COM