

How to Remove Characters from a String in Power BI: A Step-by-Step Guide

Authored by
stats writer

January 28, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Remove Characters from a String in Power BI: A Step-by-Step Guide*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=128194>

Data integrity is paramount in the realm of business intelligence. When working with large datasets in Power BI, users frequently encounter instances where text fields, known as strings, contain extraneous characters, prefixes, or suffixes that hinder accurate analysis. The process of removing characters from a string is a fundamental aspect of data cleansing and preparation, ensuring that your visualizations and calculations are based on clean, normalized data.

This detailed guide explores the precise methods available within Power BI to achieve string manipulation, focusing on using the powerful Data Analysis Expressions (DAX) language. We will demonstrate how to leverage built-in functions to systematically eliminate unwanted elements, preparing your data for complex reporting and modeling. Mastering these techniques is essential for any analyst seeking to improve the efficiency and reliability of their Power BI solutions.

Power BI: Remove Characters from String

Understanding String Manipulation and Data Preparation

Effective analysis requires that raw data be transformed into a usable format. In many organizational datasets, data entry inconsistencies or automated system outputs result in text columns containing unwanted fixed strings, such as identifiers, codes, or superfluous labels. For example, a column intended to hold product categories might include a uniform prefix like "CAT-" before every category name. To accurately aggregate or filter these categories, that prefix must be removed.

In the context of Power BI, string manipulation can be accomplished using two primary environments: the Power Query Editor (M language) for transformations applied during data loading, and DAX for calculated columns or measures created after the data model is loaded. While Power Query offers tremendous flexibility, using DAX to create a new calculated column often proves straightforward for targeted, fixed-string removal, especially when the resulting cleaned column is needed directly within the data model for relationships or visualization filters.

For the specific task of removing a fixed, known sequence of characters from a text string, the SUBSTITUTE function in DAX provides a highly efficient and readable solution. This function searches a text string for a specific older text and replaces all occurrences of that text with a new text value, effectively allowing us to replace the unwanted characters with an empty string.

The DAX Approach: Leveraging the SUBSTITUTE Function

The SUBSTITUTE function is the cornerstone of character removal in DAX when dealing with static character sequences. Unlike the REPLACE function, which relies on position and length arguments, SUBSTITUTE simply looks for all instances of a specified substring throughout the

source text. This makes it ideal for cleaning up data where a specific prefix or internal tag needs to be universally scrubbed.

The general syntax for using the `SUBSTITUTE` function is as follows. Notice that the third argument, which represents the new text to insert, is set to an empty string ("") to achieve the removal effect:

NewColumn = SUBSTITUTE(<Text>, <Old_Text>, <New_Text>)

When removing characters, the critical implementation involves specifying the target column, defining the exact characters to be removed (the <Old_Text>), and providing an empty string ("") as the replacement value (<New_Text>). The optional fourth argument, <Instance_Number>, can be ignored if the goal is to remove all occurrences of the unwanted string within the cell, which is typically the case during data cleansing operations.

You can use the following syntax in DAX to remove specific characters from a string:

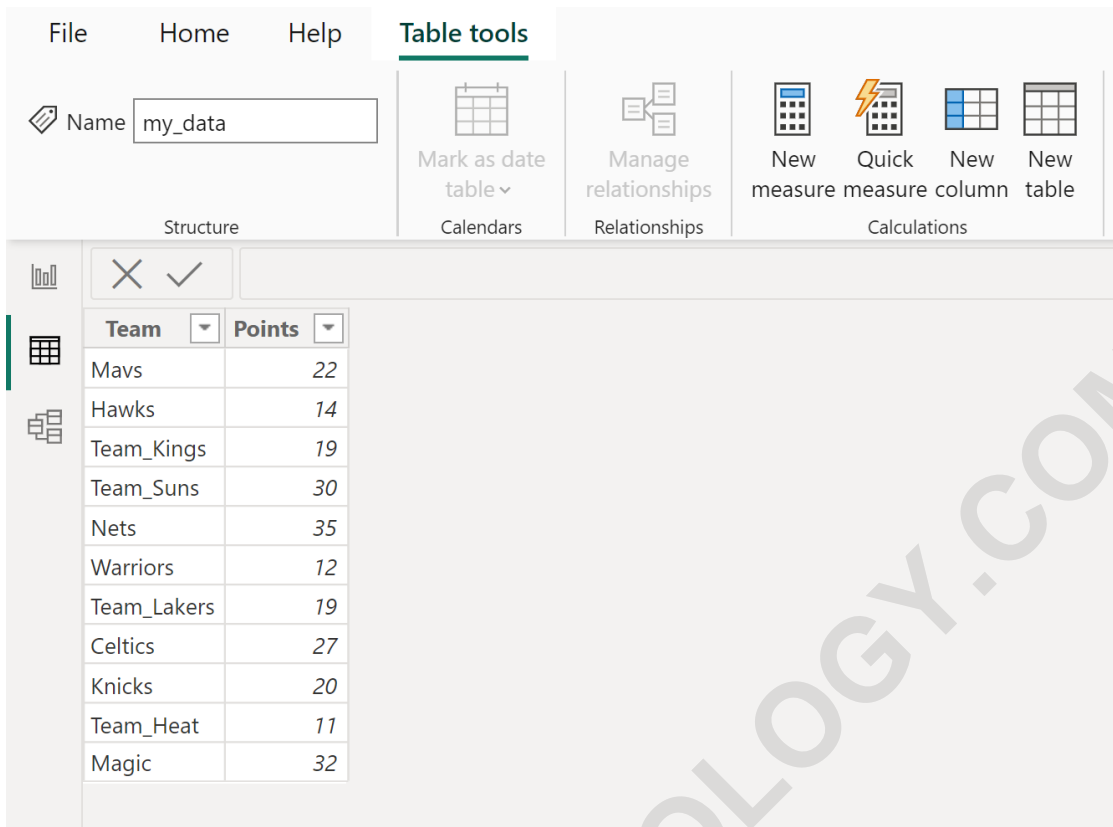
Team_New = SUBSTITUTE('my_data', "Team_", "")

This particular example creates a new column named **Team_New** that removes the literal string "Team_" from each string entry found in the **Team** column of the table named **my_data**. This action is irreversible within the calculated column, producing a clean version ready for downstream analysis.

Practical Example: How to Remove Characters from String in Power BI

To illustrate the efficiency of the SUBSTITUTE function, we will walk through a scenario involving cleaning team names. This is a common requirement where imported data prefixes group identifiers onto the meaningful data label.

Suppose we have the following table in Power BI named **my_data** that contains information about points scored by basketball players on various teams:



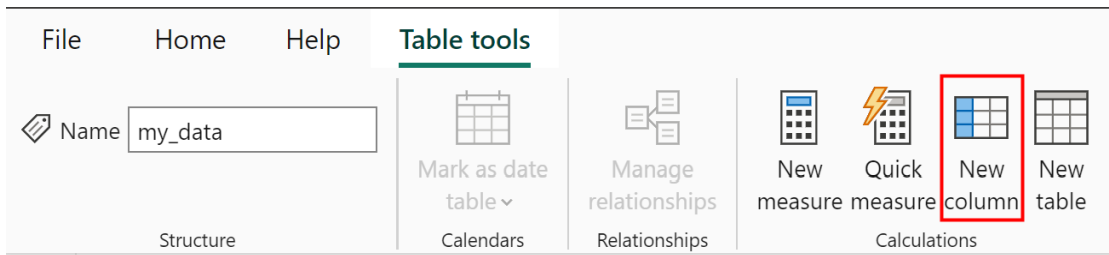
Upon reviewing the table, notice that there are several rows in the **Team** column that contain the fixed string "Team_" within the team name. This prefix is unnecessary for aggregation and potentially confusing for report viewers. Our objective is to create a new, clean column that only contains the actual team identifier, free of the "Team_" prefix.

Suppose that we would like to create a new column that removes "Team_" from each string in the **Team** column. This action is essential if we plan to create a visual filter based on the team name or establish a relationship with a dimension table that uses clean team names.

Step-by-Step Implementation in Power BI Desktop

The following steps detail the precise process required to implement the DAX formula for character removal within the Power BI Desktop environment.

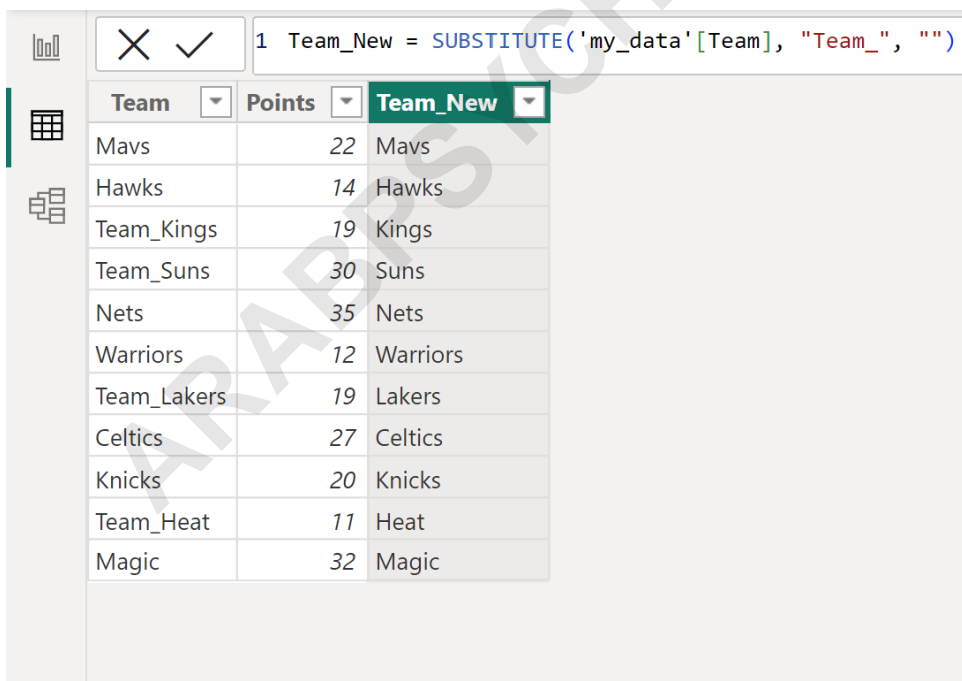
First, navigate to the table view in Power BI Desktop where the **my_data** table is displayed. To initiate the creation of a new calculated column, click the **Table tools** tab along the top ribbon, then click the **New column** icon:



Once the formula bar appears, it allows for the input of the DAX expression. Carefully type the following formula into the formula bar. Ensure that the column and table names are correctly referenced, using single quotes around the table name if it contains spaces or special characters (though generally good practice even without them):

```
Team_New = SUBSTITUTE('my_data', "Team_", "")
```

Upon committing the formula by pressing Enter, Power BI will process the calculation row by row. This will create a new column named **Team_New** that successfully removes "Team_" from each string in the original **Team** column, yielding a standardized and clean set of team names, as shown below:



Team	Points	Team_New
Mavs	22	Mavs
Hawks	14	Hawks
Team_Kings	19	Kings
Team_Suns	30	Suns
Nets	35	Nets
Warriors	12	Warriors
Team_Lakers	19	Lakers
Celtics	27	Celtics
Knicks	20	Knicks
Team_Heat	11	Heat
Magic	32	Magic

Dissecting the SUBSTITUTE Formula Mechanics

Understanding the internal logic of the formula we used is crucial for applying this technique to

different scenarios. Recall the formula:

```
Team_New = SUBSTITUTE('my_data', "Team_", "")
```

This formula relies entirely on the powerful capabilities of the [SUBSTITUTE function](#) within [DAX](#). The function operates on three key arguments provided:

The first argument, `'my_data'`, specifies the column containing the original text that needs modification. This is the source text string.

The second argument, `"Team_"`, defines the exact sequence of characters that the function is instructed to locate and replace. This search is case-sensitive.

The third argument, `""` (an empty [string](#)), determines the replacement text. By using an empty string, the function achieves the effect of deletion or removal, rather than substitution with another set of characters.

In essence, the [SUBSTITUTE function](#) iterates through every cell in the **Team** column. For any cell containing the literal sequence "Team_", it substitutes that specific sequence with nothing, effectively removing it. This method provides robust and reliable [data cleansing](#) for fixed character strings.

Alternative Methods: DAX vs. Power Query (M Language)

While [DAX](#) is highly effective for post-load calculated columns, it is important for expert users to recognize that character removal can often be handled earlier in the data pipeline using [Power BI's](#) Power Query Editor, which utilizes the M language.

In the Power Query Editor, the transformation to remove a specific prefix is achieved using the functions available under the Transform tab, such as "Replace Values" or "Extract/Remove Characters." Using Power Query M language is generally considered best practice for [data cleansing](#) because it reduces the size of the resulting data model and improves query performance, as the transformation occurs before the data is loaded into the model's memory.

However, DAX remains the preferred tool in specific scenarios:

When the transformation must interact dynamically with other calculated elements in the data model.

When the analyst lacks permissions or access to modify the initial Power Query steps.

When the calculation is derived from data already synthesized within the data model (though string removal rarely falls into this category).

For simple, fixed string removal, if it is possible to implement the cleansing step in Power Query, it

is often more resource-efficient to do so. Nonetheless, the DAX [SUBSTITUTE function](#) provides a rapid and powerful way to handle this requirement directly within the tabular model.

Summary of Best Practices for String Cleansing

When approaching character removal in [Power BI](#), adhering to a few key best practices ensures optimal performance and maintainability:

Case Sensitivity: Remember that the [SUBSTITUTE function](#) in [DAX](#) is inherently case-sensitive. If you need to remove "TEAM_" as well as "Team_", you would need to implement two separate SUBSTITUTE calls or use a combination of UPPER/LOWER and SUBSTITUTE functions to normalize the case before substitution, adding complexity to the formula.

Choosing the Right Tool: Prioritize [data cleansing](#) in Power Query (M language) whenever possible for structural transformations like character removal, as this offloads processing from the data model and reduces file size. Use DAX for calculated columns only when the transformation is dependent on post-load model context or when a quick, visible solution is required.

Documentation: Always name your new calculated columns clearly (e.g., **Team_New** or **Team_Cleaned**) and document the logic behind the calculation, especially when using nested string functions, to ensure future maintainers understand the data origin.

The following tutorials explain how to perform other common tasks in [Power BI](#):