

How to Perform Fuzzy Matching in Power BI to Link Similar Data

Authored by
mohammed loot

January 9, 2026

RECOMMENDED CITATION

mohammed loot (2026). *How to Perform Fuzzy Matching in Power BI to Link Similar Data*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=125217>

Defining Fuzzy Matching in the Power BI Environment

Fuzzy matching within the context of Power BI refers to the specialized process of identifying and linking similar, yet not perfectly identical, data points sourced from disparate tables or data sources. Unlike traditional database joins, which demand exact textual or numerical correspondence between key columns, fuzzy matching employs advanced statistical algorithms to calculate the degree of similarity between strings. This capability is absolutely essential when dealing with real-world datasets that often suffer from inconsistencies, misspellings, or variations in nomenclature.

The core mechanism for performing this powerful operation lies within the Merge Queries function accessible through the robust Power Query Editor. Consider a practical scenario: an organization possesses a sales ledger containing customer names and a separate CRM database holding customer demographic details. Due to varying data entry methods, "John Smith Corp." in one table might appear as "J. Smith Corporation" in the other. Standard join operations would fail to link these records, leading to incomplete analysis. Fuzzy matching, however, can intelligently recognize these entries as highly similar and successfully merge them, effectively linking the two tables based on phonetic or textual proximity.

By successfully linking these imperfectly aligned datasets, analysts can create a far more comprehensive and accurate foundation for reporting. Incorporating detailed customer information from one source with sales metrics from another allows for deeper segmentation and pattern identification. This process transforms raw, messy data into structured, actionable intelligence, making fuzzy matching a cornerstone technique for effective data cleaning and preparation in **Power BI** workflows.

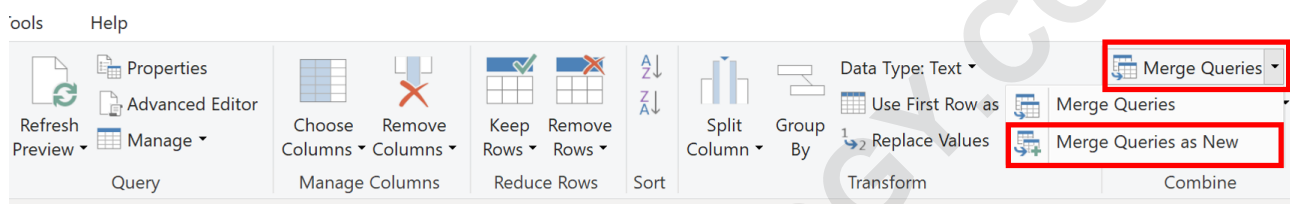
The Critical Need for Imperfect String Joins

In data analysis, particularly when integrating information from multiple operational systems or disparate departments, the necessity often arises to join two or more tables based on strings that do not match exactly. This capability, known technically as fuzzy matching, addresses a fundamental challenge of data management: human error and semantic variation. Standard database joins rely on precise byte-for-byte matches, which instantly breaks down when faced with issues like leading/trailing spaces, case differences, abbreviations, or minor spelling errors.

The easiest and most efficient route to achieving this powerful capability in Power BI is by utilizing the built-in **Merge Queries** function found within the Power Query Editor (also known as the Transform data tool). This feature abstracts the complex statistical methods required for similarity calculation, presenting them through an intuitive user interface. It empowers users to define the degree of tolerance for mismatches, ensuring that relevant data points are correctly associated despite minor inconsistencies in the key fields.

Understanding when to use this feature is key. If you are certain that your linking columns are unique identifiers (like Employee IDs or standardized product SKUs), a standard exact join is appropriate. However, if you are attempting to link using textual descriptors such as names, addresses, product descriptions, or company titles, fuzzy matching becomes indispensable. It significantly reduces the manual effort required for data cleansing prior to merging, speeding up the overall data preparation phase of any business intelligence project.

The following visual illustrates the entry point to this functionality, initiating the powerful data transformation process:

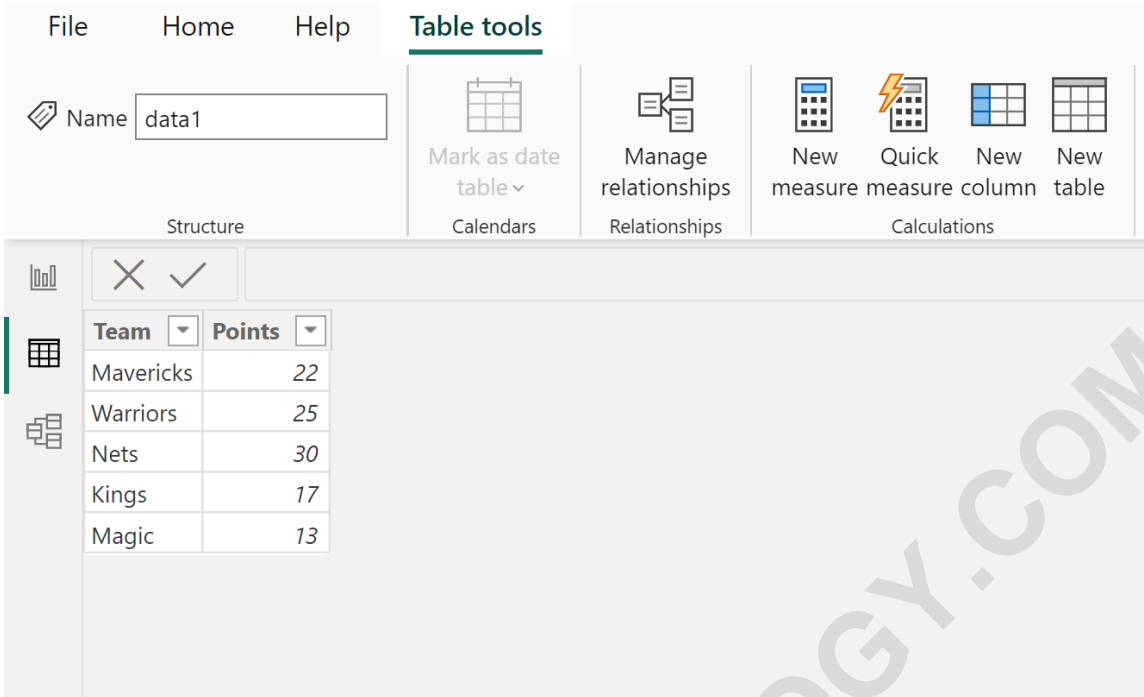


The subsequent detailed example will walk through the practical application of fuzzy matching, demonstrating how this tool handles real-world data imperfections.

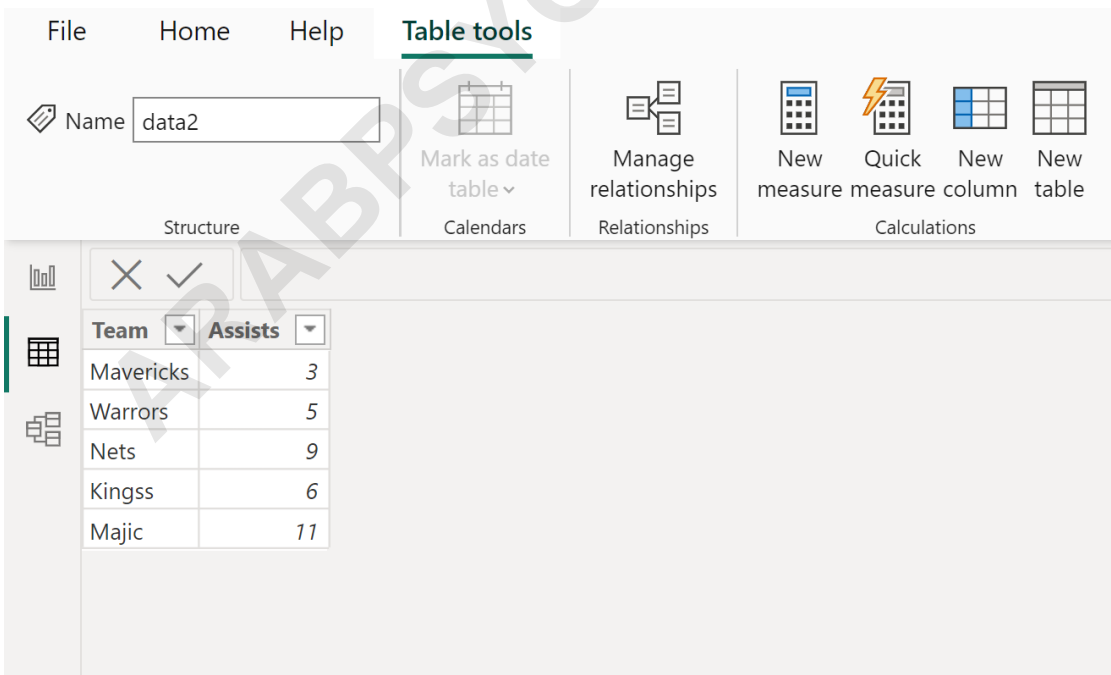
Case Study Setup: Merging Basketball Statistics

To effectively illustrate the application of fuzzy matching, we will utilize a practical scenario involving sports statistics. Suppose we have two distinct tables loaded into Power BI, both relating to basketball team performance, but gathered from different sources, leading to minor inconsistencies in team naming conventions. Our goal is to perform a robust merge queries operation that links the data despite these naming discrepancies.

Our first table, named **data1**, contains essential information regarding the team name and the total points scored for various basketball players:



And suppose that we have another table named **data2** that contains complementary statistics--specifically, the total assists recorded for those same teams. Crucially, observe the slight variations in the 'Team' names here compared to **data1** (e.g., abbreviations or slight rephrasing):



We intend to execute an inner join between these two tables. This means we only want to retain records where a corresponding match is found in both datasets. Since the team names are not

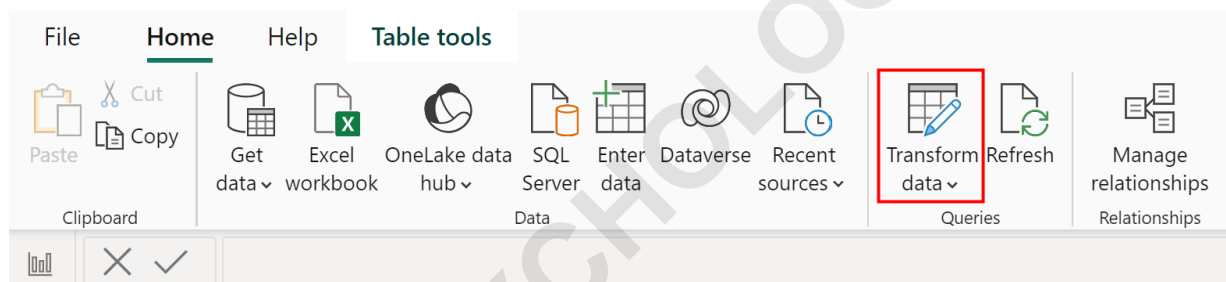
exactly identical strings (e.g., "The Bulls" versus "Bulls Team"), a conventional merge operation would fail to produce the desired combined dataset. This scenario perfectly mandates the use of fuzzy matching on the strings contained within the respective **Team** columns.

Accessing and Initiating the Power Query Editor

The journey to performing a fuzzy merge begins in the data transformation environment of Power BI. Start by navigating to the main ribbon interface of Power BI Desktop. Locate the **Home** tab and subsequently click on the **Transform data** icon.

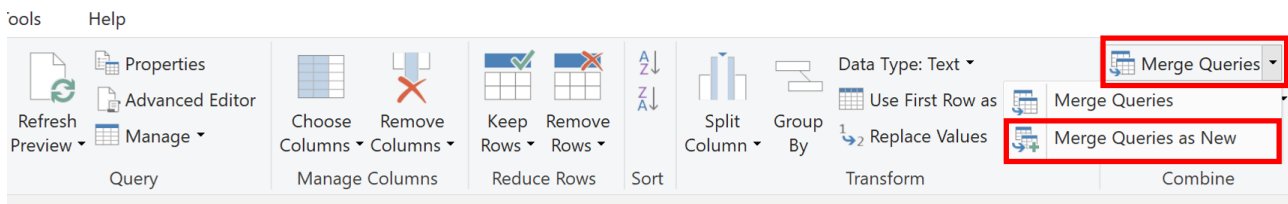
This action immediately launches the dedicated transformation workspace, the Power Query Editor. The Power Query interface is where all significant data shaping, cleansing, and merging operations, including advanced features like fuzzy matching, are meticulously configured.

The visual cue for this initial step is critical for workflow navigation:



Once inside the Power Query Editor, locate the **Home** tab again within this new window. Look towards the **Combine** group, which houses the tools necessary for integrating multiple data sources. Here, you will find the **Merge Queries** icon. Clicking this icon presents two options: "Merge Queries" (to modify an existing query) or "Merge Queries as New" (to create an entirely new, combined table). For best practice, especially when experimenting with complex operations, selecting **Merge Queries as New** is highly recommended to preserve the integrity of the original source tables (data1 and data2).

This selection process directs the user to the precise tool needed for combining datasets based on non-exact criteria:



Configuring the Fuzzy Merge Operation and Parameters

Upon selecting the merge option, a configuration dialog box will appear, prompting the user to define the parameters of the join. This is the crucial stage where we define which tables to combine, the linking columns, the type of join, and, most importantly, enable the fuzzy matching feature.

The required settings are detailed below:

Primary Table: Select **data1** as the first table.

Secondary Table: Select **data2** as the second table.

Linking Column: Click on the **Team** column header in both the **data1** preview and the **data2** preview to designate them as the columns upon which the merge will be performed.

Join Kind: Select **Inner** as the Join Kind. An inner join ensures that only rows that have a corresponding match (based on the fuzzy criteria) in both tables are included in the resulting merged query.

The defining action for this process is activating the fuzzy logic. Below the Join Kind selection, ensure that the checkbox labeled **Use fuzzy matching to perform the merge** is checked. This toggles the advanced similarity calculation algorithms, instructing Power Query to look for relationships beyond exact string equality.

The configuration screen provides a comprehensive overview of the settings before execution:

Merge

Select tables and matching columns to create a merged table.

Team	Points
Mavericks	22
Warriors	25
Nets	30
Kings	17
Magic	13

Team	Assists
Mavericks	3
Warrors	5
Nets	9
Kingss	6
Majic	11

Join Kind

Use fuzzy matching to perform the merge

▾ Fuzzy matching options

Similarity threshold (optional)

 ⓘ

Ignore case

✓ The selection matches 5 of 5 rows from the first table, and 5 of 5 rows fro...

Adjusting the Similarity Threshold for Optimal Results

A key parameter within the fuzzy matching settings is the **Similarity threshold**. This numeric value, ranging from 0.0 to 1.0, dictates how strict the matching criteria will be. This threshold is essentially a confidence score that determines the required textual proximity between two strings:

A value of **1.0** mandates a perfect match, functionally turning off fuzzy logic and reverting to a standard, exact merge queries operation.

A value approaching **0.0** instructs the system to accept virtually any strings as a match, which is

rarely desirable as it leads to false positives and highly inaccurate linkages between data points.

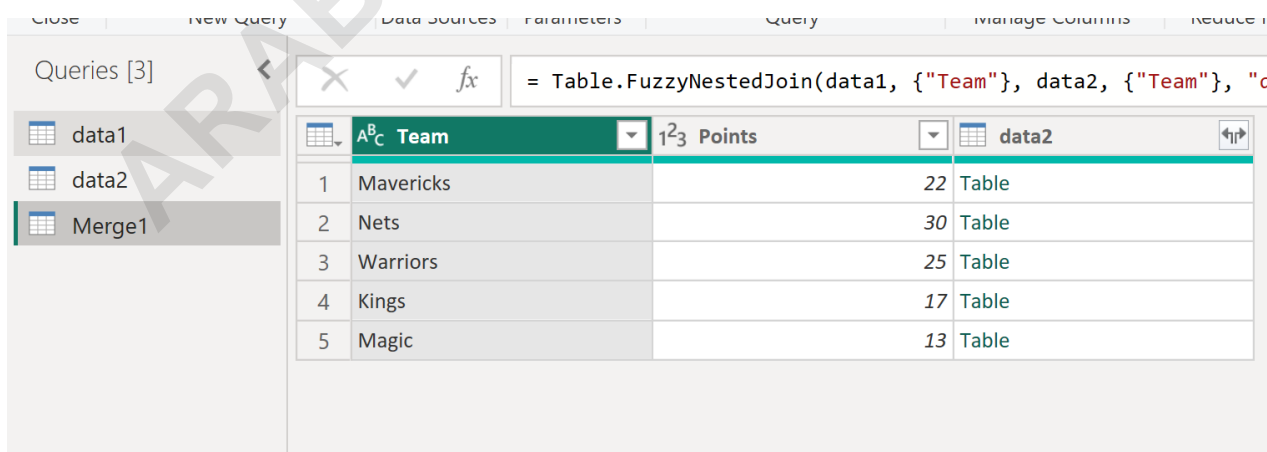
The default threshold is set at **0.8**. This value represents a balance, allowing for common spelling errors, abbreviations, and minor structural variations while preventing wildly dissimilar entries from being matched. For most business data scenarios, 0.8 is a good starting point, but it should be carefully adjusted based on the observed quality and consistency of your specific data sources.

By default, Power BI employs algorithms such as Jaccard similarity or Tversky similarity (often implemented through the Jaro-Winkler distance) to quantify the textual closeness between two strings. Choosing the correct similarity threshold is critical because it directly controls the trade-off between capturing all legitimate matches (high recall) and minimizing incorrect matches (high precision). Experienced analysts often test several threshold values (e.g., 0.75, 0.80, 0.85) to determine the optimal setting for their data quality profile, ensuring the most accurate integration possible.

Finalizing and Expanding the Merged Data Table

Once the configuration is complete, including the selection of the desired similarity threshold, clicking **OK** executes the merge operation. The Power Query Editor processes the two tables, performing the complex fuzzy calculations across the 'Team' columns and generating a new query, which we will call 'Merge1'.

The initial result of the merge operation shows the combined data from **data1**, with the corresponding matched rows from **data2** encapsulated in a single structured column named **data2**. This column holds the secondary table as a nested structure for each successful match found via the fuzzy matching process:

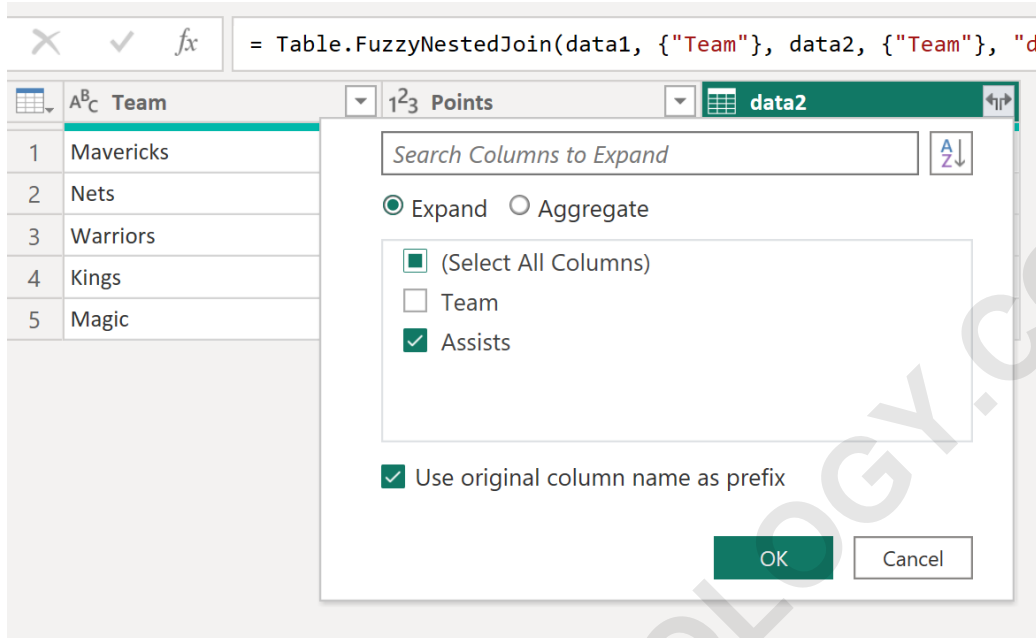


The screenshot shows the Power Query Editor interface. On the left, the 'Queries' pane lists 'data1', 'data2', and 'Merge1'. The main area displays a table with columns 'Team', 'Points', and 'data2'. The 'Team' column has a dropdown menu with 'A B C' and 'Team' options. The 'Points' column has a dropdown menu with '1 2 3' and 'Points' options. The 'data2' column has a dropdown menu with 'data2' and a right-pointing arrow icon. The table contains five rows of data:

	Team	Points	data2
1	Mavericks	22	Table
2	Nets	30	Table
3	Warriors	25	Table
4	Kings	17	Table
5	Magic	13	Table

To integrate the necessary information from **data2** into the main table structure, we must now expand this nested column. Click on the left and right arrows located on the header of the **data2**

column. This action opens the expansion menu. Since our goal is only to incorporate the statistical measure from the secondary table, ensure that only the checkbox next to the **Assists** column is selected.



After clicking **OK** on the expansion menu, the **Assists** column from the **data2** table is successfully added to the new merged query. This final transformation provides the desired unified dataset, where team points and assists are aligned based on the advanced fuzzy comparison, not just exact textual equality:

The screenshot shows the Power Query Editor with the final expanded table. The formula bar contains: `= Table.ExpandTableColumn(Source, "data2", {"Assists"}, {"data2`. The table has three columns: 'Team', 'Points', and 'data2.Assists'. The data rows are:

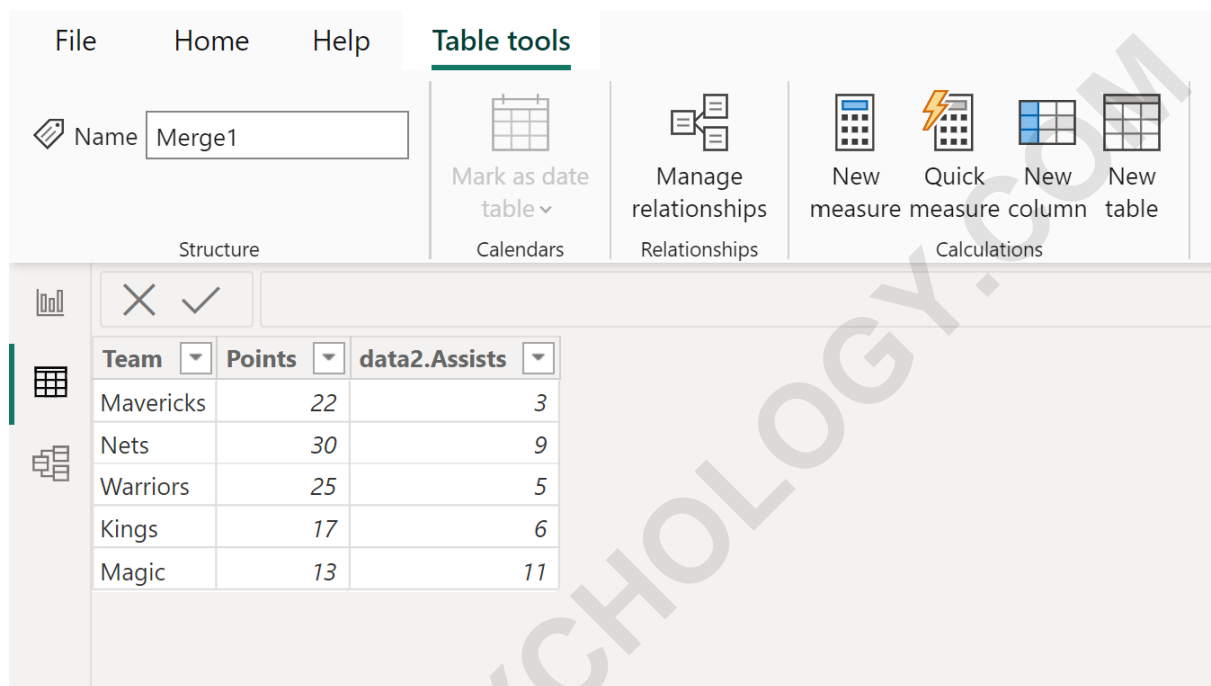
	Team	Points	data2.Assists
1	Mavericks	22	3
2	Nets	30	9
3	Warriors	25	5
4	Kings	17	6
5	Magic	13	11

Applying Changes and Analyzing the Final Dataset

The combined table, now structured and validated within the Power Query Editor, represents the

desired outcome of the fuzzy match. To make this new dataset available for visualization and reporting in Power BI Desktop, the transformations must be applied. Exit the Power Query Editor by navigating to the **Home** tab and selecting **Close & Apply**. A message box will appear that asks if you'd like to apply your changes. Click **Yes**.

You will then be able to see the new table named **Merge1** in the Table view:



The screenshot shows the Power BI Desktop interface with the 'Table tools' ribbon selected. The ribbon includes options like 'Name' (set to 'Merge1'), 'Mark as date table', 'Manage relationships', 'New measure', 'Quick measure', 'New column', and 'New table'. Below the ribbon, a table is displayed with the following data:

Team	Points	data2.Assists
Mavericks	22	3
Nets	30	9
Warriors	25	5
Kings	17	6
Magic	13	11

Notice that this final merged table was able to accurately match each string in the **Team** column of **data1** with a similar, though not identical, string in the **Team** column of **data2** based on fuzzy matching. This confirms that the configuration successfully handled the data inconsistencies inherent in real-world sources.

Note on Column Renaming: If you'd like, you can right click on the header named **data2.Assists** and rename the column to just **Assists**. This provides a cleaner structure for downstream analysis and reporting within the Power BI data model.

Related Data Manipulation Techniques in Power BI

While fuzzy matching is an advanced technique for data integration, mastery of Power BI often requires proficiency in other fundamental data manipulation tasks necessary for comprehensive data preparation. These tasks ensure that every row and column is optimally structured for analytical querying.

The following tutorials explain how to perform other common tasks in Power BI, complementing the

data integration achieved through merge queries:

How to Add Index Column to Table in Power BI

ARABPSYCHOLOGY.COM