

How to Perform Fuzzy Matching in Power BI: A Step-by-Step Guide

Authored by
stats writer

January 25, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Perform Fuzzy Matching in Power BI: A Step-by-Step Guide*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=127675>

Perform Fuzzy Matching in Power BI (With Example)

Power BI is recognized globally as an industry-leading business analytics service, essential for transforming raw data into insightful visualizations. A critical challenge often encountered during data preparation is dealing with inconsistencies when attempting to consolidate information from disparate sources. This is where fuzzy matching becomes indispensable.

Fuzzy matching in Power BI is a powerful feature that enables users to compare and join strings of text that are similar but not perfectly identical. This functionality is absolutely essential in real-world scenarios where data quality is imperfect, meaning records may contain typographical errors, inconsistent abbreviations, or variations in nomenclature. Without fuzzy logic, these slight differences would prevent standard merge operations from correctly linking related data points, leading to incomplete or inaccurate analysis.

To effectively implement this feature within Power BI, users utilize the **Fuzzy Merge** capability found within the Power Query Editor. This feature allows analysts to specify source and target columns, define the matching tolerance, and subsequently generate a consolidated table where matched values are correctly aligned based on the defined criteria. For instance, if one dataset lists a company as "John Smith Corp" and another lists it as "Jon Smithe Co.", a fuzzy merge operation executed with an appropriate threshold--such as 80% similarity--will successfully identify and merge these entries. Ultimately, leveraging fuzzy matching significantly enhances data accuracy and operational efficiency by identifying and linking comparable data points that rigid exact-match methods would overlook.

Understanding the Core Concept of Fuzzy Matching

At its core, **fuzzy matching**, often referred to as approximate string matching, goes beyond the binary exact-match logic common in standard database joins. Instead of requiring two strings to be character-for-character identical, fuzzy algorithms measure the degree of similarity between them. This is crucial when integrating data pulled from multiple systems, external vendors, or manual inputs, where human error or system differences inevitably introduce noise. The goal is to maximize the utility of the data by reducing the impact of these minor data quality issues.

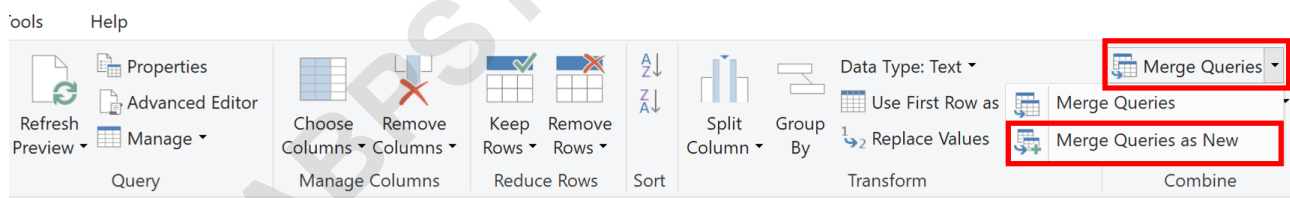
The practical application of fuzzy matching is vast, spanning fields from customer relationship management (CRM) systems--where duplicate contact records need to be reconciled--to supply chain management, where product descriptions might vary slightly across different databases. By identifying these approximate matches, analysts ensure that calculations and reports accurately reflect the true underlying business entity or relationship, preventing double-counting or data fragmentation. This sophisticated capability transforms the traditionally tedious process of data cleaning into an efficient, automated step within the data preparation workflow.

In Power BI, the mechanism employed relies on calculating a similarity score between the text strings, often using algorithms like Jaccard or Levenshtein distance, though the specific method is abstracted away from the user via the **Fuzzy Merge** dialog box. Users primarily interact with the Similarity threshold, which dictates the minimum score required for two strings to be considered a match. Understanding this fundamental concept is the first step toward mastering data integration within the Power BI environment.

The Role of Power Query Editor in Data Transformation

The entire process of performing a fuzzy match is managed within the Power Query Editor (often known as the Query Editor or M language environment). The Query Editor serves as Power BI's dedicated environment for extracting, transforming, and loading (ETL) data before it is presented in the final report canvas. It is within this powerful interface that complex data transformations, including merging, pivoting, unpivoting, and, critically, fuzzy merging, are executed and recorded as steps.

When dealing with the task of joining two tables based on imperfectly matching strings, the primary tool leveraged is the **Merge Queries** function. This function allows users to combine data from two separate queries into a new, consolidated query based on key columns. When standard exact matching fails due to slight variations in data entry, activating the fuzzy matching option during the merge setup provides the necessary flexibility to bridge the gap.



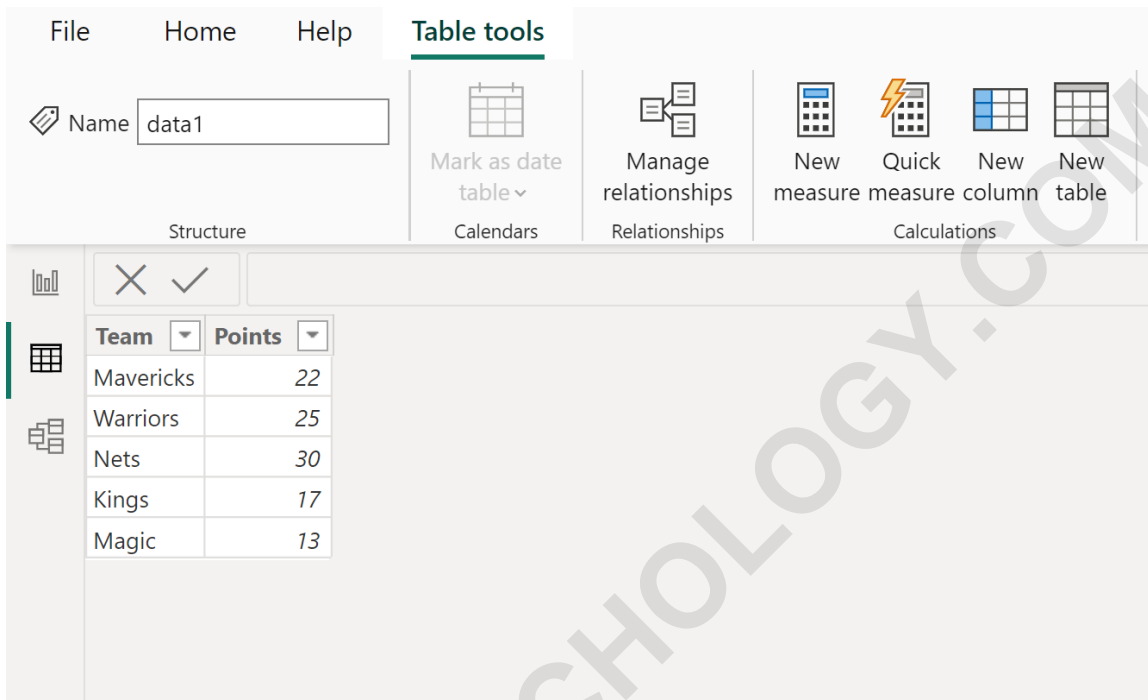
Using the Merge Queries function ensures that the data cleansing and joining steps are documented and repeatable. Every action taken in the Power Query Editor is logged, forming the M code that defines the transformation process. This reproducibility is vital for maintaining data governance and refreshing data sources efficiently, ensuring that the fuzzy matching logic is applied consistently whenever new data is imported.

Step-by-Step Example: Preparing the Data Sets

To illustrate the practical implementation of fuzzy matching, let us consider a scenario involving two tables that contain basketball statistics. We aim to consolidate points and assists data, but the team names--the linking key--are inconsistently formatted across the two sources. This common

issue necessitates the use of a fuzzy merge operation to successfully link the records.

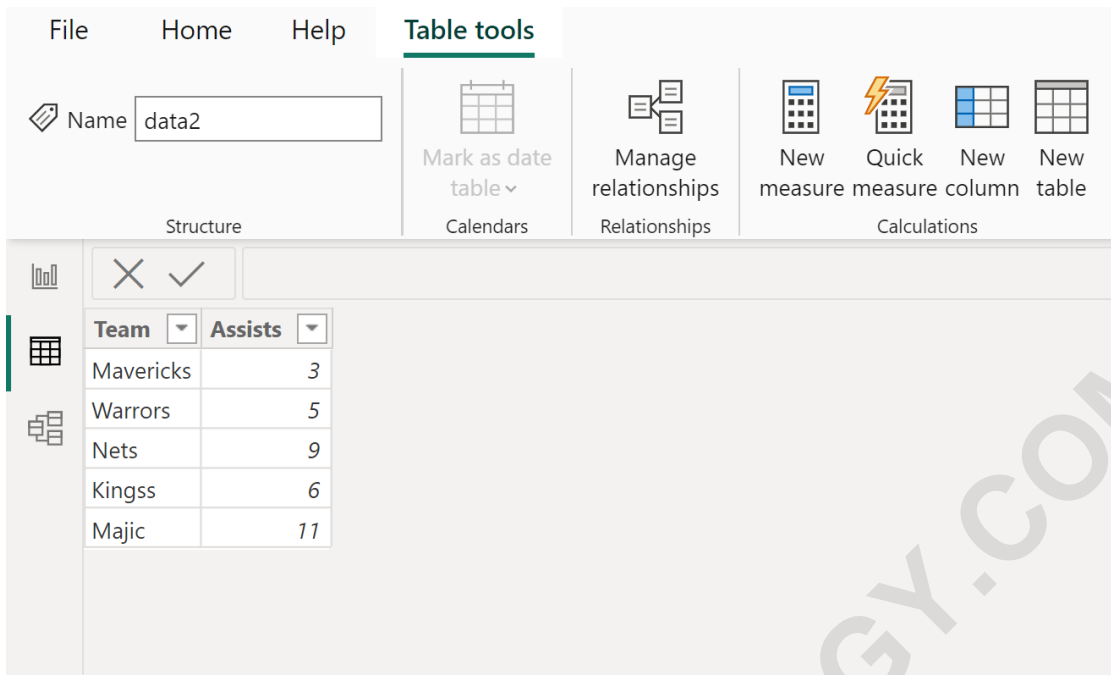
First, we introduce the two hypothetical tables. Suppose we have a table loaded into Power BI named **data1** that contains basic information regarding team names and the total points scored by various players:



The screenshot shows the 'Table tools' ribbon in Power BI. The 'Name' field is set to 'data1'. The ribbon includes sections for 'Calendars', 'Relationships', and 'Calculations'. The 'Calculations' section has options for 'New measure', 'Quick measure', 'New column', and 'New table'. Below the ribbon, a table is displayed with the following data:

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

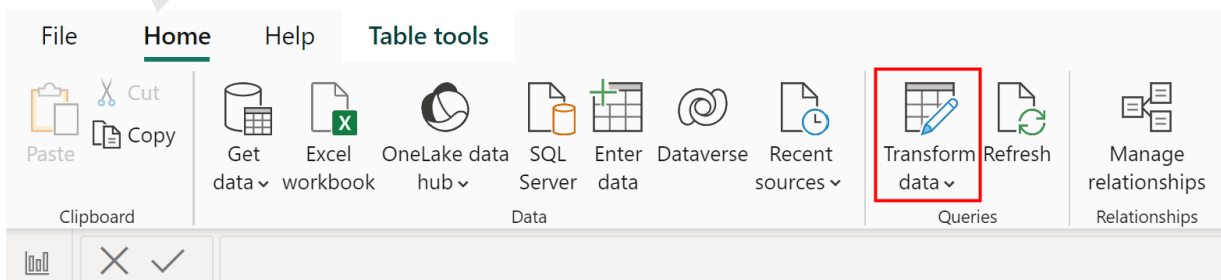
Next, consider a second table, named **data2**, which contains related statistics--specifically, the number of assists--but utilizes slightly different spellings or abbreviations for the team names:



A careful inspection of the "Team" columns reveals the data inconsistency: "Lakers" is listed in one table while "LA Lakers" is in the other; "Celtics" versus "Boston Celtics"; and "Bulls" versus "Chi Bulls". A standard exact match merge operation would fail to link any of these rows, rendering the consolidation effort useless. Our objective is to perform an inner join between these two tables using a fuzzy match on the respective **Team** columns to accurately combine the points and assists data.

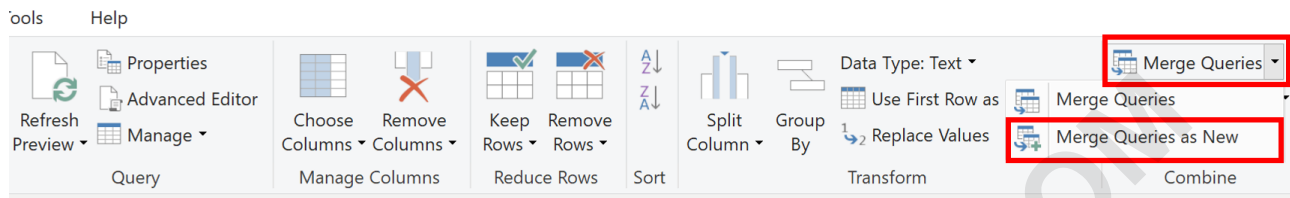
Initiating the Data Transformation Process

The journey begins by accessing the powerful transformation environment. To initiate the merge process, navigate to the **Home** tab located along the top ribbon of the Power BI desktop interface. Within this tab, locate and click the **Transform data** icon. This action launches the dedicated Power Query Editor window, where all subsequent data manipulation steps will occur.



Once the Power Query Editor is open, your focus should shift to the **Combine** group on the **Home**

tab. Here, you will find the Merge Queries icon. For best practice, and to preserve your original data tables, it is recommended to select **Merge Queries as New** from the dropdown menu. This creates a new, independent table resulting from the merge operation, leaving **data1** and **data2** untouched.



Executing the Fuzzy Merge Operation

Selecting the **Merge Queries as New** option prompts a new configuration window to appear. This dialog requires specific inputs to define how the two tables should be combined. First, establish the primary table (the left side of the merge), which in our case is **data1**. Second, select the table to be merged onto the primary table (the right side), which is **data2**. Crucially, select the common key column in both tables--the **Team** column--by clicking on the header of the respective column in each preview pane.

The next essential parameter is the **Join Kind**. Since we are interested only in records where a match exists between the two tables, we select **Inner** as the **Join Kind**. An inner join ensures that only rows with corresponding matches in both tables--as determined by the fuzzy logic--will be included in the resulting merged query.

The defining action for this process is activating the fuzzy logic. This is achieved by checking the box labeled **Use fuzzy matching to perform the merge**. Upon checking this box, additional configuration options related to similarity scoring become available, allowing fine-tuning of the matching process.

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...

Configuring the Similarity Threshold and Advanced Options

The quality of the fuzzy match is largely dictated by the Similarity threshold, which is one of the most important parameters to configure. This value ranges between 0.0 and 1.0. A threshold value of **1.0** is equivalent to an exact match, meaning strings must be perfectly identical to be joined. Conversely, a threshold of **0.0** is extremely lenient, potentially matching almost any string pair, regardless of similarity, which is rarely useful in practice. The default setting is typically **0.8**, providing a reasonable balance between strictness and flexibility.

Similarity threshold optimization often requires iteration and testing, as the ideal value depends heavily on the nature and variability of your source data. For data with minor spelling errors, the default 0.8 is often sufficient. If your data contains significant abbreviations or complex variations, a lower threshold (e.g., 0.6 or 0.7) might be necessary, though this increases the risk of false positives.

Beyond the threshold, advanced options offer further control:

Ignore Case: This crucial setting, enabled by default, ensures that "lakers" matches "LAKERS," preventing case differences from breaking a match.

Maximum number of matches: This controls how many records from the second table can be matched to a single record in the first table. Leaving this blank allows for all potential matches.

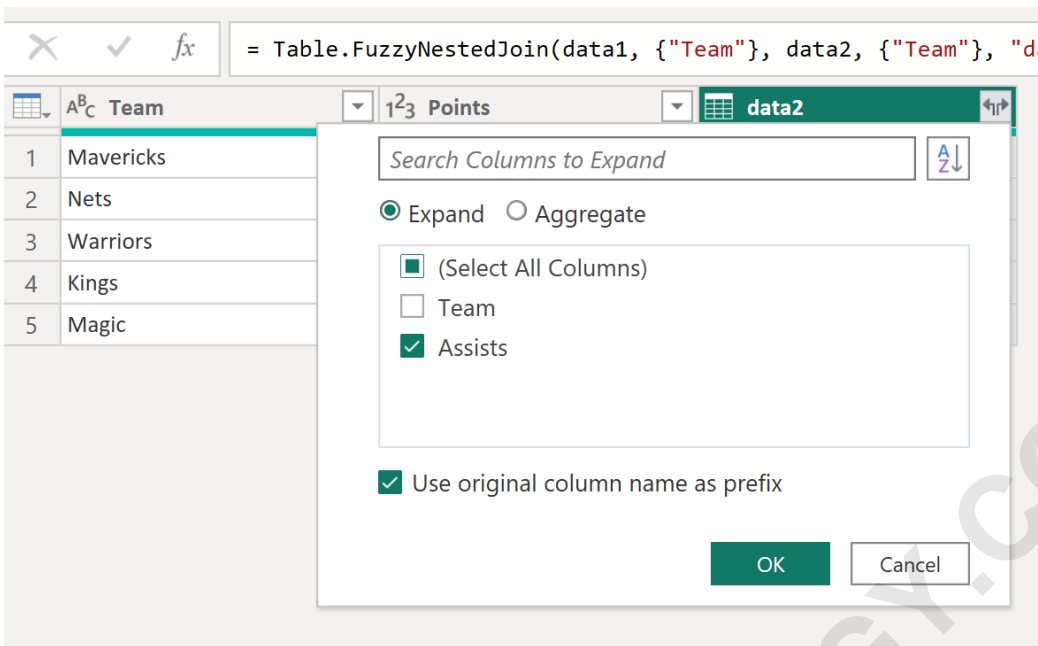
Transformation table: For highly complex scenarios, a custom transformation table can be supplied. This table maps known variations (e.g., "NY" to "New York") explicitly, providing deterministic matches alongside the probabilistic fuzzy logic.

Once the parameters, including the default Similarity threshold of 0.8, are set, clicking **OK** executes the Merge Queries operation, producing the new merged table in the Power Query Editor environment.

Finalizing the Merged Query and Applying Changes

After clicking **OK**, the new query, likely named "Merge1," will appear. This merged table contains all columns from **data1**, and a single column named **data2**, which currently holds structured table data representing the matched rows from the second source. To integrate the desired statistics from **data2** (the Assists column) into the primary view, we must expand this column.

To perform this expansion, click on the expand icon--the left and right arrows located on the header of the **data2** column. This action opens a dialog box that allows you to select which specific columns from the linked table should be included in the final output. In our example, we are only interested in the **Assists** metric.



Ensure that the box next to **Assists** is checked, and typically, uncheck the option that says "Use original column name as prefix." However, in this scenario, if the original prefix is left, the column will be named **data2.Assists**. Once you click **OK**, the table structure is updated, and the consolidated data is ready. The power of the fuzzy merge is now evident, as all corresponding statistics are correctly aligned despite the original spelling variances.

The screenshot shows the final merged table in Power BI. The table has three columns: 'Team', 'Points', and 'data2.Assists'. The data is as follows:

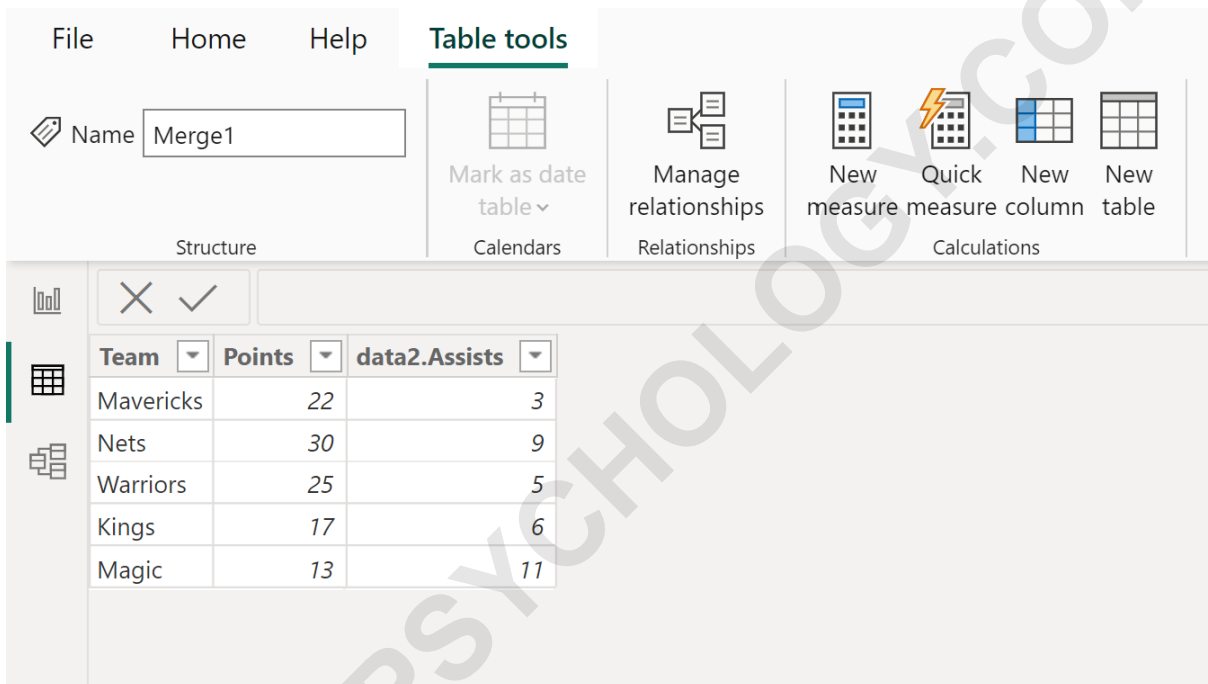
| | Team | Points | data2.Assists |
|---|-----------|--------|---------------|
| 1 | Mavericks | 22 | 3 |
| 2 | Nets | 30 | 9 |
| 3 | Warriors | 25 | 5 |
| 4 | Kings | 17 | 6 |
| 5 | Magic | 13 | 11 |

Notice how this final merged table successfully matched records: "Lakers" (data1) found "LA Lakers" (data2), "Celtics" (data1) found "Boston Celtics" (data2), and "Bulls" (data1) found "Chi Bulls" (data2). This reconciliation, impossible with standard joins, provides a complete dataset ready for visualization. As a final cleaning step, analysts often right-click on the header named **data2.Assists** and rename the column simply to **Assists** for improved readability.

Conclusion: Integrating Fuzzy Logic into the Power BI Workflow

The final step is to save and apply the transformations executed within the Power Query Editor. Upon exiting the editor, a message box will prompt the user to confirm whether they wish to apply the pending changes back to the Power BI data model. Clicking **Yes** loads the newly created, consolidated query (Merge1) into the data model.

You can then observe the new table named **Merge1** in the Table view within Power BI Desktop, confirming the success of the fuzzy matching operation.



The screenshot displays the Power BI Desktop interface. The 'Table tools' ribbon is active, showing 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 data table is visible with the following content:

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

Fuzzy matching in Power BI is not merely a technical trick; it is a fundamental capability for dealing with the inherent messiness of real-world data. By offering adjustable sensitivity via the Similarity threshold and integration directly into the Merge Queries function, Power BI empowers analysts to achieve higher data quality and produce reports that are accurate and comprehensive, regardless of minor textual imperfections in the source data. Mastering this technique is crucial for anyone working with heterogeneous data sources.

Further Recommended Tutorials

To continue advancing your data preparation skills in Power BI, explore these additional resources:

[How to Add Index Column to Table in Power BI](#)