How to Perform a Left Join in Power BI: A Step-by-Step Guide

Authored by

stats writer

January 25, 2026

A Left Join, often referred to as a **Left Outer Join**, is a fundamental data manipulation technique crucial for combining datasets within Power BI. This operation intelligently merges two tables based on shared values in a specified common column. The defining characteristic of a Left Join is that it retains **all rows** from the first table--known as the primary or left table--and includes only the corresponding matching rows from the second table (the right table).

This powerful merging capability is executed using the **Merge Queries** function accessible through the Power Query Editor. When configuring the merge, the user explicitly designates the sequence of tables, thereby establishing which table acts as the primary (left) dataset. The output of this operation is a comprehensive new table that incorporates columns from both sources. Crucially, rows from the primary table are guaranteed to appear in the result, and if a match is found in the secondary table, the corresponding data is appended; otherwise, null values populate the newly added columns. This method provides an invaluable mechanism for comprehensive data comparison and integration, ensuring no data from the primary dataset is lost during the merging process.

Perform a Left Join in Power BI: A Detailed Practical Guide

Understanding the Left Outer Join Mechanism

The concept of the Left Join is rooted deeply in relational database theory and is critical for modern business intelligence workflows. Unlike an Inner Join, which only returns rows where matches exist in both tables, the Left Join prioritizes the integrity and completeness of the primary dataset. If you are performing complex analysis, such as enriching customer order history with product categorization data, using a Left Join ensures that every original order record remains intact, even if the product categorization

table lacks the necessary details for certain products.

In the context of Power BI, performing a join is handled efficiently within the transformation layer, specifically the Power Query Editor. This powerful environment allows for robust data shaping and cleaning before the final data model is loaded for visualization. By leveraging the Merge Queries function, users can specify the exact join condition and the type of join required. For a Left Outer Join, the system systematically checks the linking column in the right table for every single value present in the corresponding column of the left table.

If a value in the linking column of the primary table finds one or more corresponding values in the secondary table, those matched rows are seamlessly included and duplicated as necessary (in the case of one-to-many relationships). If, however, a row in the primary table contains a key value that does not exist in the secondary table's linking column, the columns originating from the secondary table will contain `null` values for that specific row in the final merged output. This characteristic is fundamental for identifying data

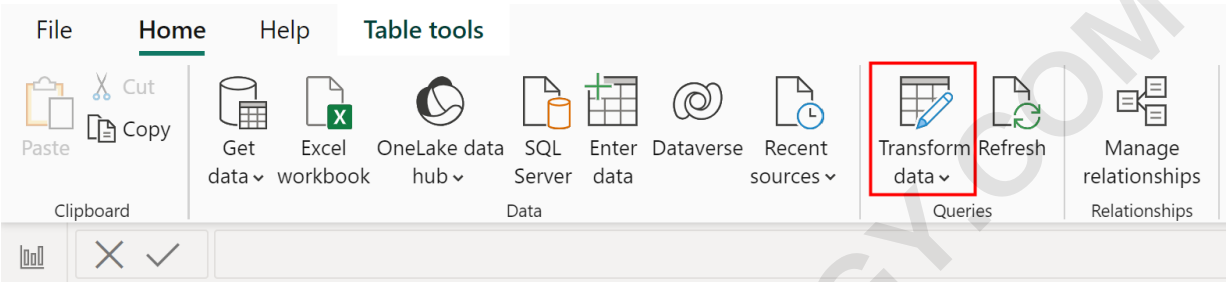
gaps, spotting inconsistencies, and ensuring complete coverage of the primary dataset without discarding any records.

Accessing the Power Query Editor for Data Transformation

The initial and most crucial step in executing any complex data transformation, including merging tables, involves navigating away from the main visualization canvas and into the specialized environment of the Power Query Editor. This editor is effectively the ETL (Extract, Transform, Load) backbone of Power BI Desktop, offering extensive capabilities for data preparation and shaping. The fastest and easiest way to perform a Left Join between two existing tables in Power BI is, therefore, to utilize the dedicated Merge Queries functionality found exclusively within this editor.

To access this critical environment, users typically start from the main Power BI Desktop interface after loading their data sources. Locate the Home tab situated prominently along the top ribbon menu. Within the 'External Data' or 'Queries' group--which manages data connectivity and modification--locate and click the

Transform data icon. Clicking this icon initiates the launch of the separate Power Query Editor window, which provides a feature-rich, dedicated interface for managing and transforming all imported data sources.



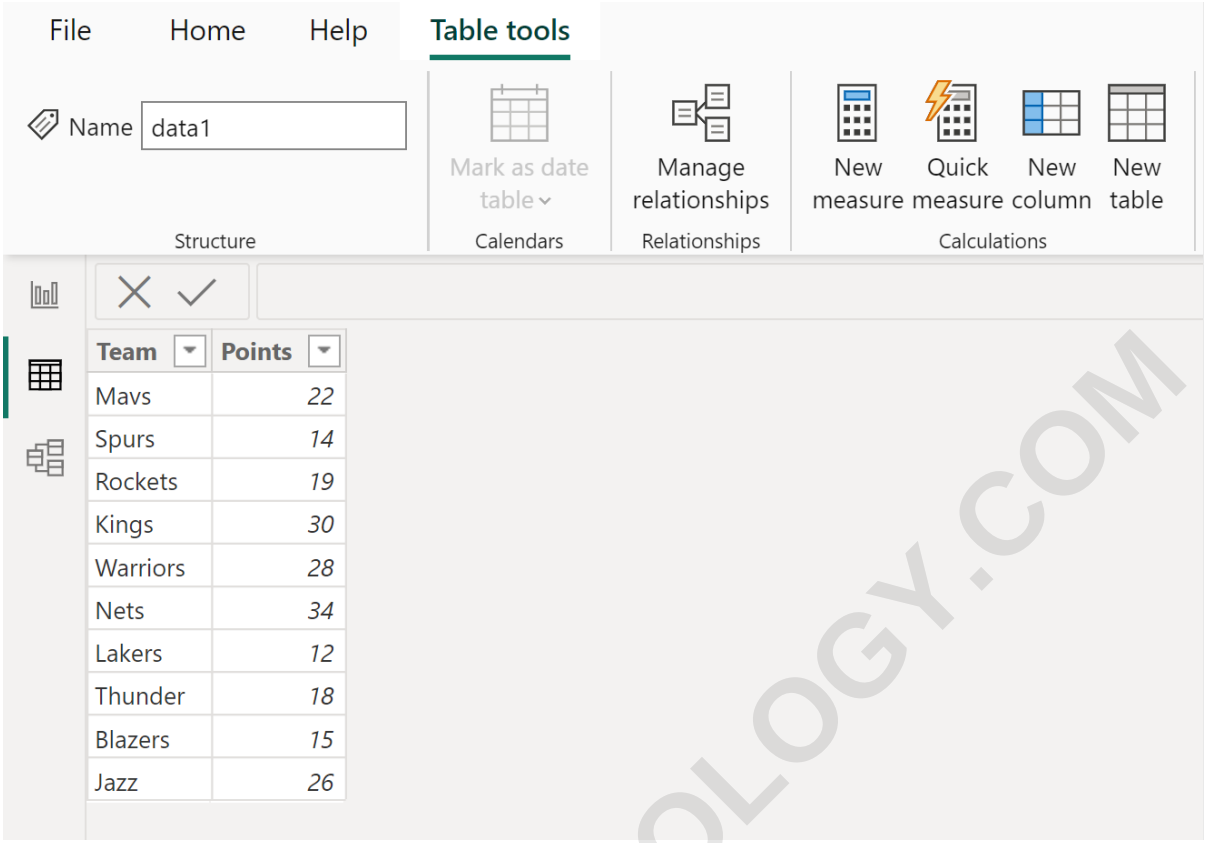
Once inside the editor, you will see a list of all your current queries (which represent your loaded tables) displayed on the left-hand navigation pane. It is absolutely essential to ensure that both tables intended for the join--the left table (primary) and the right table (secondary)--have already been successfully loaded and are visible within the Power Query Editor interface before attempting to proceed with the actual merge operation.

Example Setup: Preparing the Source Data

To demonstrate the practical application and outcome of a Left Join, let us structure a common real-world scenario involving two separate data sources that need

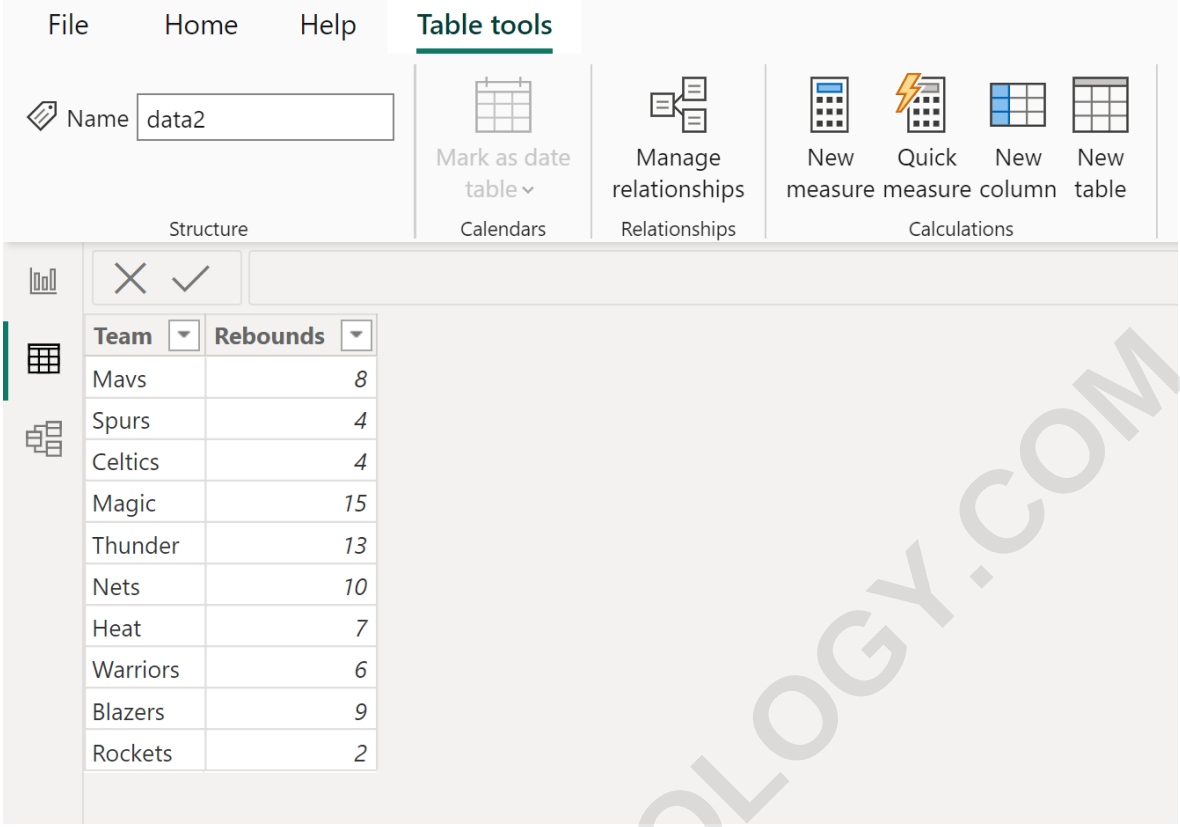
to be unified based on a shared categorical field. Suppose we are managing sports statistics and have two distinct tables that share a common attribute: the team name.

Our first table, named data1, will serve as our primary or left table. This table contains essential information about various basketball players, detailing their individual names, associated team names, and the total points they have scored throughout the season. Because this dataset is the core foundation of our current analysis, we must ensure that all its rows are completely preserved in the final merged output, regardless of secondary matches.



Team	Points
Mavs	22
Spurs	14
Rockets	19
Kings	30
Warriors	28
Nets	34
Lakers	12
Thunder	18
Blazers	15
Jazz	26

Our second table, named data2, acts as the supplementary or right table. This table provides additional statistics at the team level, specifically tracking the team name and the total rebounds achieved by players on those respective teams. Our ultimate goal is to seamlessly augment the records in data1 by appending the corresponding rebound statistics from data2, but the linkage must be strictly governed by matching team names.



Team	Rebounds
Mavs	8
Spurs	4
Celtics	4
Magic	15
Thunder	13
Nets	10
Heat	7
Warriors	6
Blazers	9
Rockets	2

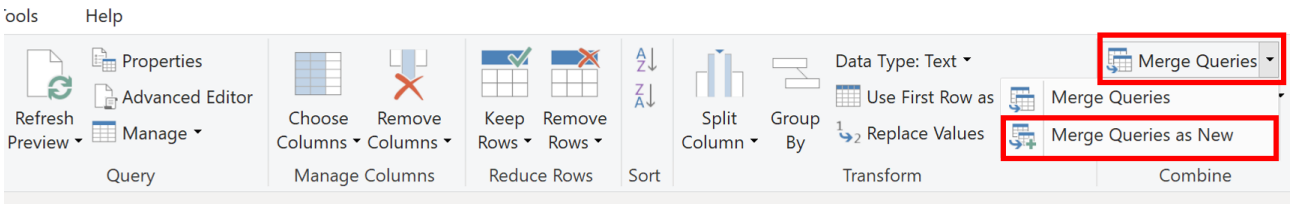
We will proceed to execute a Left Join where every row from the data1 table is retained, while integrating only the matching values found via the Team column in the data2 table. This approach guarantees that even if a team exists in data1 but lacks a corresponding entry in data2, the player's points record (from data1) will still be visible in the result, clearly marked by a blank or null value in the rebounds column, signaling missing data.

Initiating the Merge Queries Process

Once the source data tables are correctly loaded and

you are viewing them within the **Power Query Editor**, the next logical phase is to activate the merge command. Navigate to the Home tab located in the Power Query ribbon interface. Within this tab, focus on the Combine group of functions. This group houses the essential tools dedicated to integrating or combining disparate datasets, including merging and appending operations.

Click the Merge Queries icon. This action triggers a dropdown menu that presents two distinct options: Merge Queries (which modifies the currently selected table in place) or Merge Queries as New (which creates a brand new, independent query/table containing the merged result). For adherence to best practices, particularly for complex transformations, it is highly recommended to select Merge Queries as New. This methodology ensures that the original source tables, data1 and data2, remain untouched, greatly simplifying future verification and debugging processes.



Choosing the 'Merge Queries as New' option automatically launches a specialized dialog box. This is where the core parameters for the join operation are meticulously defined. This dialog is the central configuration hub for executing the Merge Queries process, and precise configuration here is absolutely vital to achieving the desired Left Join outcome.

Configuring the Join Kind and Matching Columns

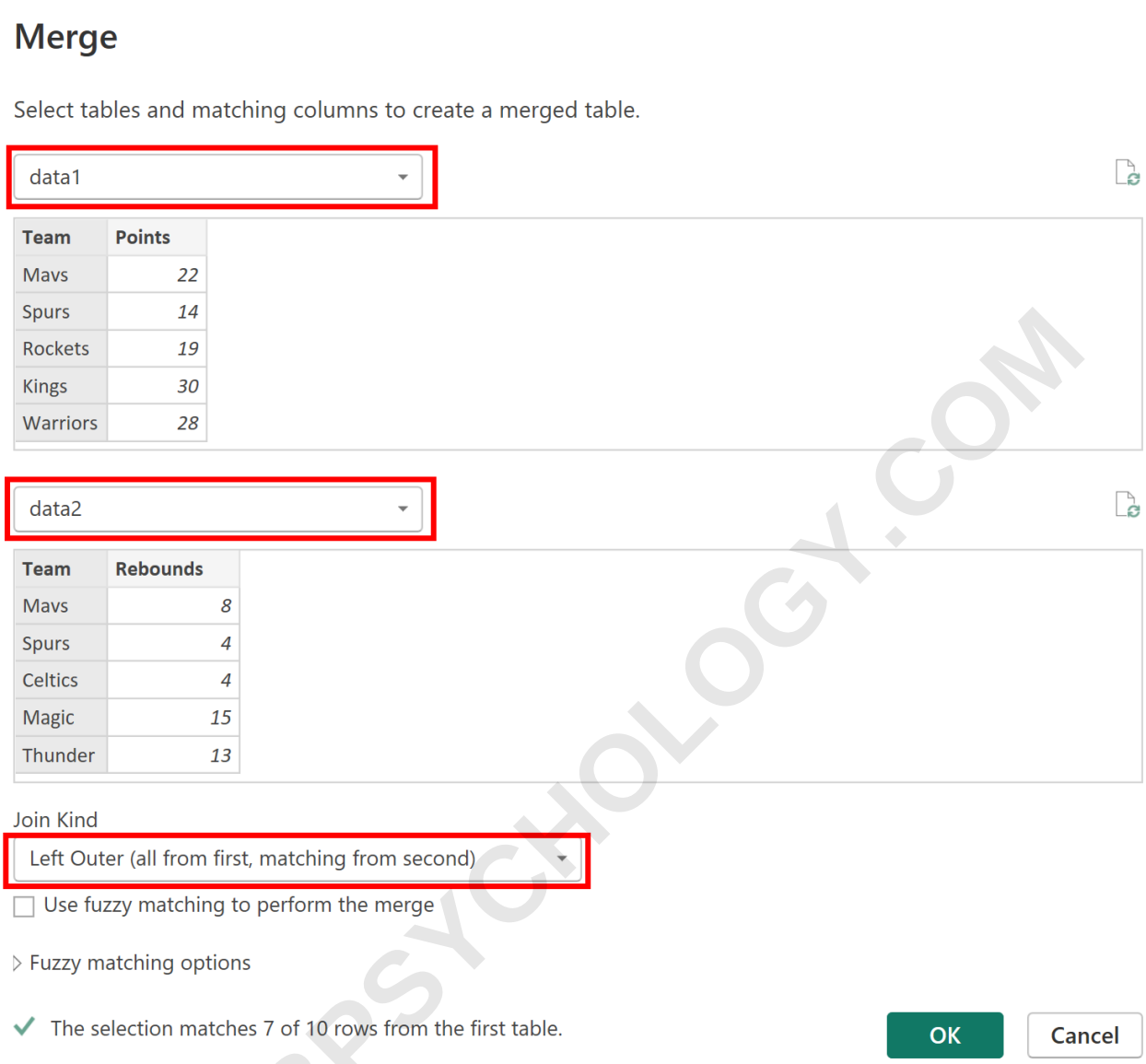
The Merge dialog requires the user to specify three critical components: the primary table, the secondary table, and the join type. These choices fundamentally dictate the structure, data retention policy, and linkage criteria of the resulting merged query.

Selecting the Primary Table: In the top selection field, choose data1. Because we are performing a Left Join, data1 is designated as the 'left' table, confirming that all its records will be retained in the final output.

Selecting the Secondary Table: In the second selection field, choose data2. This is the 'right' table from which supplementary matching data will be sourced and appended.

Defining the Join Kind: Crucially, in the dropdown menu labeled Join Kind, select Left Outer (all from first, matching from second). This explicit selection formally instructs Power BI to apply the precise logic of a Left Join.

The final, non-negotiable step in this configuration window is defining the exact linking mechanism between the two tables. Click the column header for the Team column in the preview of data1, and then, while holding the control key (or equivalent), click the column header for the Team column in the preview of data2. Both columns should become visually highlighted, signaling to Power BI that these specific common columns are the key fields upon which the matching and joining of rows must be executed.



Once you have meticulously reviewed the selected join type and confirmed the designated matching columns, click OK. The Merge Queries operation is immediately executed, and a new query, automatically named Merge1, appears in the list of queries within the Power Query Editor. This new table initially displays all the columns from data1 and a single new column named

data2, which contains nested table values representing the successful (or unsuccessful) matches found.

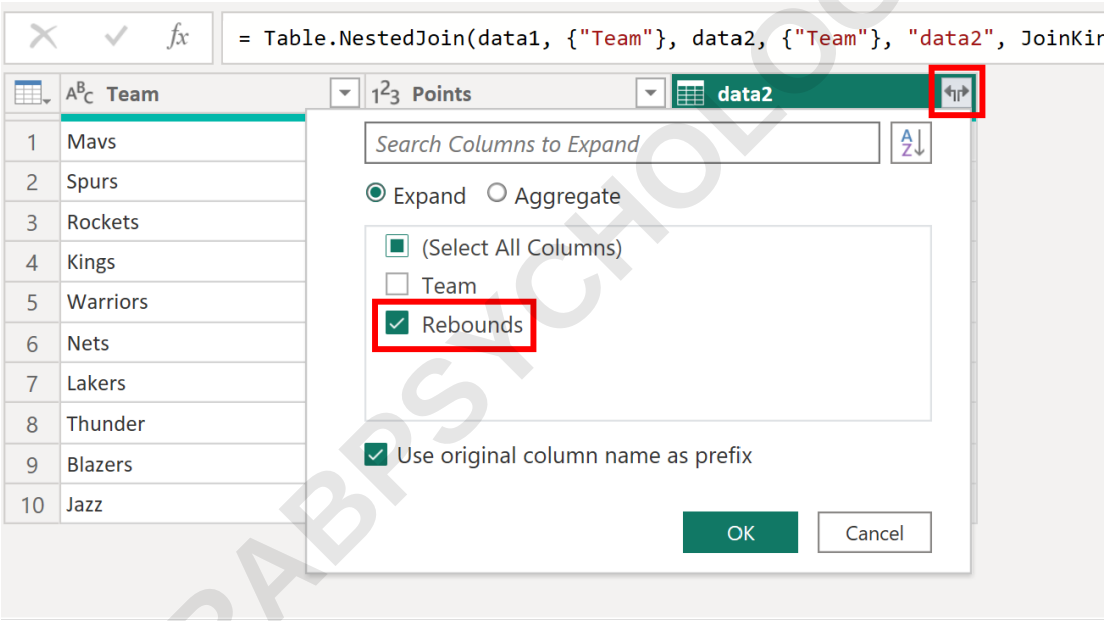
Expanding and Finalizing the Merged Table

The result generated by the initial merge step--which contains nested table objects--is not yet the final, analytical-ready table. The new column, data2, currently holds these nested table structures. We must explicitly instruct Power BI which specific columns from the secondary table (data2) we intend to expand and integrate into the overall structure of the primary table.

Locate the header of the newly created data2 column. You will notice a small but crucial icon consisting of opposing left and right arrows adjacent to the column name. This is the 'Expand' button, signaling that nested data is available. Click this icon to open the expansion menu dialog. This menu presents a comprehensive list of all available columns from the secondary table that can be integrated into the merged result set.

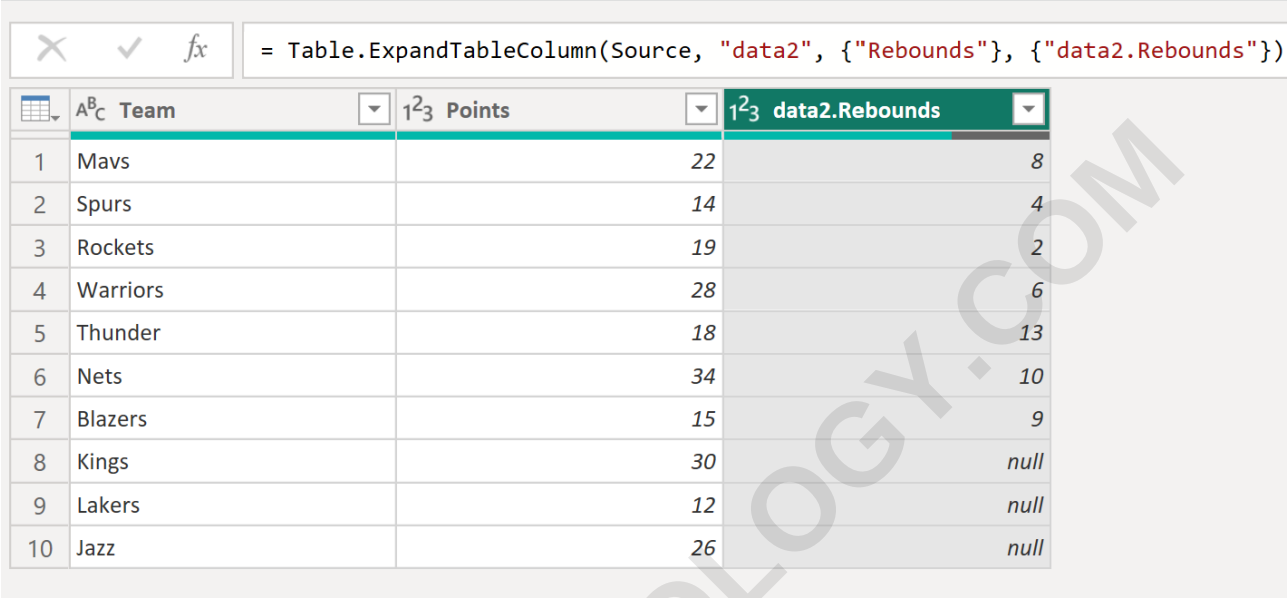
In the context of our example, we are exclusively interested in incorporating the total rebounds statistic. Therefore, ensure you uncheck the primary box labeled

Select All Columns to avoid clutter. Then, selectively check only the box next to the column named Rebounds. For clarity in the data model, you may also choose to uncheck the option to 'Use original column name as prefix' if you prefer a simpler column name like 'Rebounds', although keeping the prefix (e.g., `data2.Rebounds`) is often recommended for tracing data lineage.



Once you click OK, the Rebounds column will be successfully expanded and physically integrated into the column structure of the new query, Merge1. The visual representation of the table will now clearly display all the original columns from data1, alongside

the newly incorporated Rebounds column derived from data2, providing a unified view of the data.



```
= Table.ExpandTableColumn(Source, "data2", {"Rebounds"}, {"data2.Rebounds"})
```

	Team	Points	data2.Rebounds
1	Mavs	22	8
2	Spurs	14	4
3	Rockets	19	2
4	Warriors	28	6
5	Thunder	18	13
6	Nets	34	10
7	Blazers	15	9
8	Kings	30	null
9	Lakers	12	null
10	Jazz	26	null

Analyzing the Output and Applying Changes

Upon reviewing the final merged table in the Power Query Editor, you can immediately verify the successful execution of the Left Join. The core principle holds absolute truth: all rows originating from the left table (data1) are fully maintained within the resulting query.

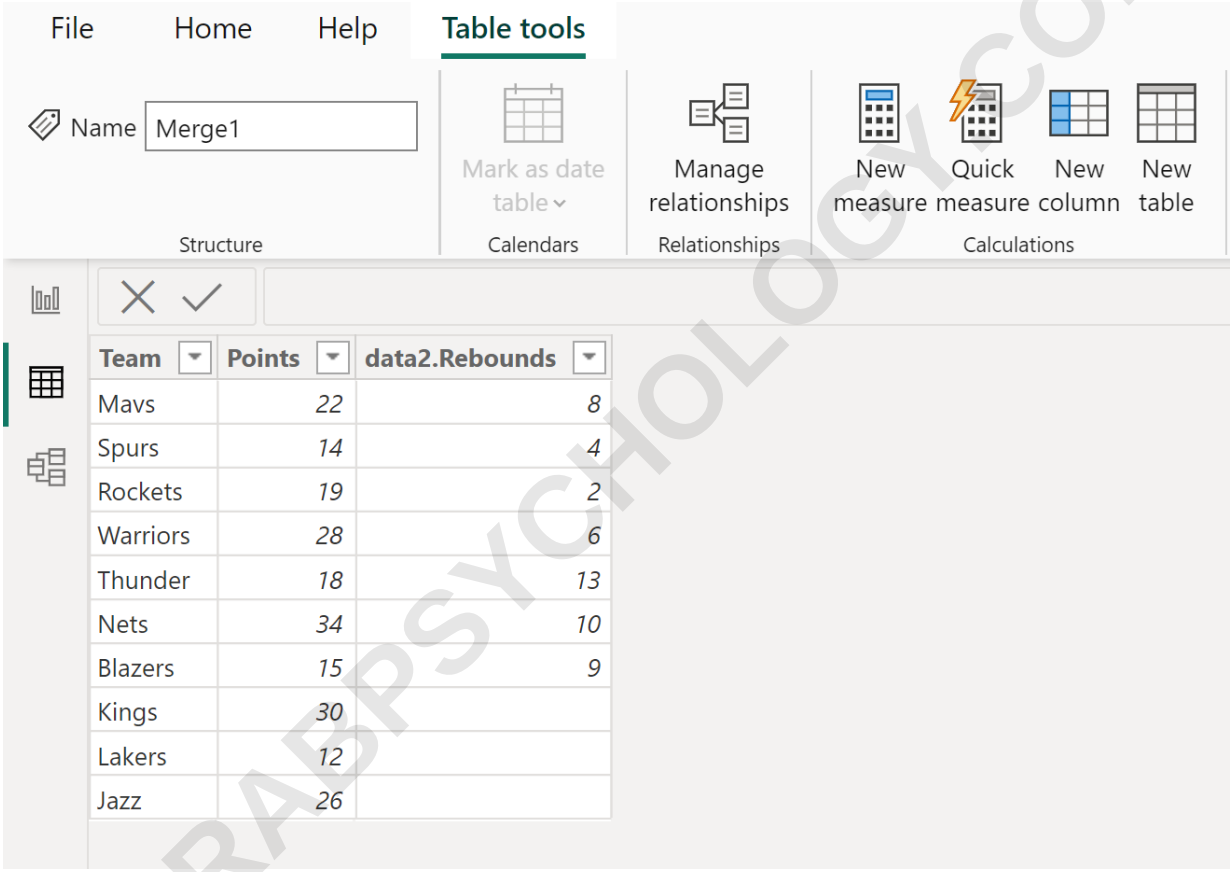
Carefully observe the results for those teams that existed in data1 but did not possess a corresponding matching team name entry in data2. For these specific rows, the original player data (including the Points column) retains its value, but the newly added column,

data2.Rebounds, will conspicuously display `null` or blank values. This behavior is the defining feature of the Left Join, as it allows data analysts to quickly identify missing supplementary data points without incurring any loss of the foundational records from the primary dataset. It is also important to note that any team present in data2 but entirely absent in data1 is, by definition of the Left Join, automatically and intentionally discarded.

A highly recommended post-merge step involves renaming the expanded columns for vastly improved data readability and usability. For instance, you can right-click the header named data2.Rebounds and select the Rename option to change the column title simply to Rebounds. This final clean-up step significantly enhances the clarity and professionalism of the resulting dataset before it is promoted for report creation.

Once you are fully satisfied with all the transformations applied, navigate back to the Home tab in the Power Query Editor interface and click the prominent Close & Apply button. This action begins the loading process.

You will subsequently receive a confirmation message box that asks if you'd like to apply all your staged changes to the Power BI Data Model. Click Yes to finalize and load the newly transformed and merged table, Merge1, into the main Power BI environment.



Team	Points	data2.Rebounds
Mavs	22	8
Spurs	14	4
Rockets	19	2
Warriors	28	6
Thunder	18	13
Nets	34	10
Blazers	15	9
Kings	30	
Lakers	12	
Jazz	26	

The new table, Merge1, will now be fully available in the Data view and Field pane of Power BI Desktop, ready for advanced visualization and reporting, demonstrating a perfectly executed Left Join that successfully combined player points with team rebounds while absolutely

prioritizing the completeness of the initial player data set.

Further Data Transformation Resources

Mastering various data manipulation techniques, such as the Merge Queries function, is absolutely fundamental to building effective and reliable Power BI development workflows. The Left Join is only one of several critical join types available for strategically integrating disparate datasets based on specific analytical requirements. For continuous improvement in your data modeling proficiency, it is highly beneficial to explore and understand the mechanics of the other common transformation tasks.

The following tutorials explain how to perform other common tasks in Power BI:

Inner Join: Explains how to return only those rows that have matching values in both the left and right tables, excluding all unmatched data.

Right Outer Join: The exact inverse of the Left Join, this operation retains all rows from the secondary (right) table and only matching rows from the primary (left)

table.

Full Outer Join: Demonstrates how to combine all rows from both tables, ensuring that unmatched fields from either side are filled with null values.

Anti-Joins: Techniques used to return rows that do not have a match in the other table, useful for identifying exceptions or data quality issues.

By thoroughly understanding and strategically applying these various join methodologies within the confines of the **Power Query Editor**, users can construct robust, accurate, and highly functional data models that are perfectly tailored for complex business intelligence demands.