

Learning Left Joins in Power BI: A Comprehensive Tutorial

Authored by
Mohammed loot

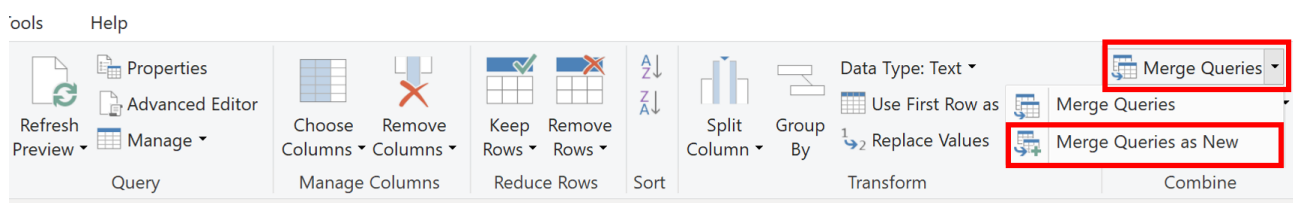
November 12, 2025

RECOMMENDED CITATION

Mohammed loot (2025). *Learning Left Joins in Power BI: A Comprehensive Tutorial*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=17918>

The ability to seamlessly combine data from diverse sources is a cornerstone of effective business intelligence. Within the ecosystem of [Power BI](#), the primary method for integrating datasets is through specialized join operations. Among these, the **left join** stands out as a critical tool, allowing analysts to merge records from two distinct tables while rigorously guaranteeing that every record originating from the primary, or "left," table is fully preserved. This essential operation is executed most efficiently using the [Merge Queries](#) feature, which is intrinsically embedded within the powerful [Power Query Editor](#) environment. This visual interface simplifies traditionally complex data integration tasks, making advanced combining operations accessible even to those unfamiliar with standard SQL syntax.

The illustration below confirms the precise location of the **Merge Queries** functionality within the interface, emphasizing its fundamental role in the overall data preparation and modeling workflow within [Power BI](#).



This step-by-step guide is designed to provide a comprehensive and practical demonstration of how to successfully perform a **left join** in [Power BI](#). By following these instructions, you will ensure that your combined dataset accurately reflects your specific analytical requirements while maintaining data integrity.

Understanding the Core Principle of the Left Join

Before proceeding to the technical implementation, it is vital to establish a clear conceptual understanding of the [left join](#), which is often formally termed a Left Outer Join. Fundamentally, this merging mechanism dictates that the resulting combined table must contain every single row from the first table specified in the operation--the designated "left" table. When records in the left table find corresponding matches based on the predetermined key column within the secondary ("right") table, the relevant columns from the right table are successfully appended.

The most defining characteristic of the **left join** is its data preservation guarantee: if a record exists in the primary dataset but lacks any corresponding match in the secondary dataset, that row from the left table is still fully retained in the final merged output. In scenarios where no match is identified, the newly created columns derived from the right table will be populated with **null** or blank values. This behavior is exceptionally valuable, particularly when the goal is to enrich a

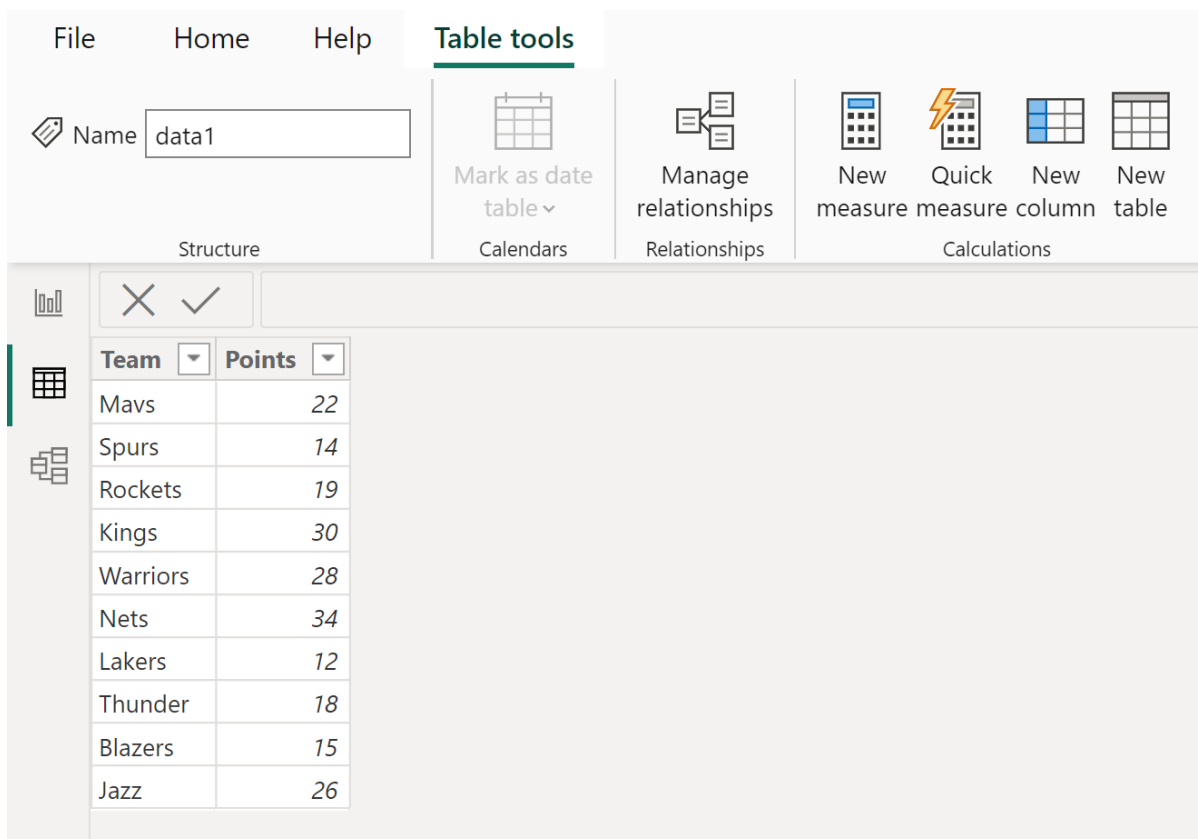
master dataset with supplementary data without risking the loss of any foundational primary records.

Choosing a **left join** over alternative join types, such as inner or full outer joins, depends entirely on the analyst's requirement for data completeness. If the project mandates that the complete set of primary records must be present in the output, regardless of their matching status in the secondary table, then the [Left Outer Join](#) is the definitive optimal choice for maintaining data integrity and ensuring a comprehensive model.

Preparing the Data Model for Merging

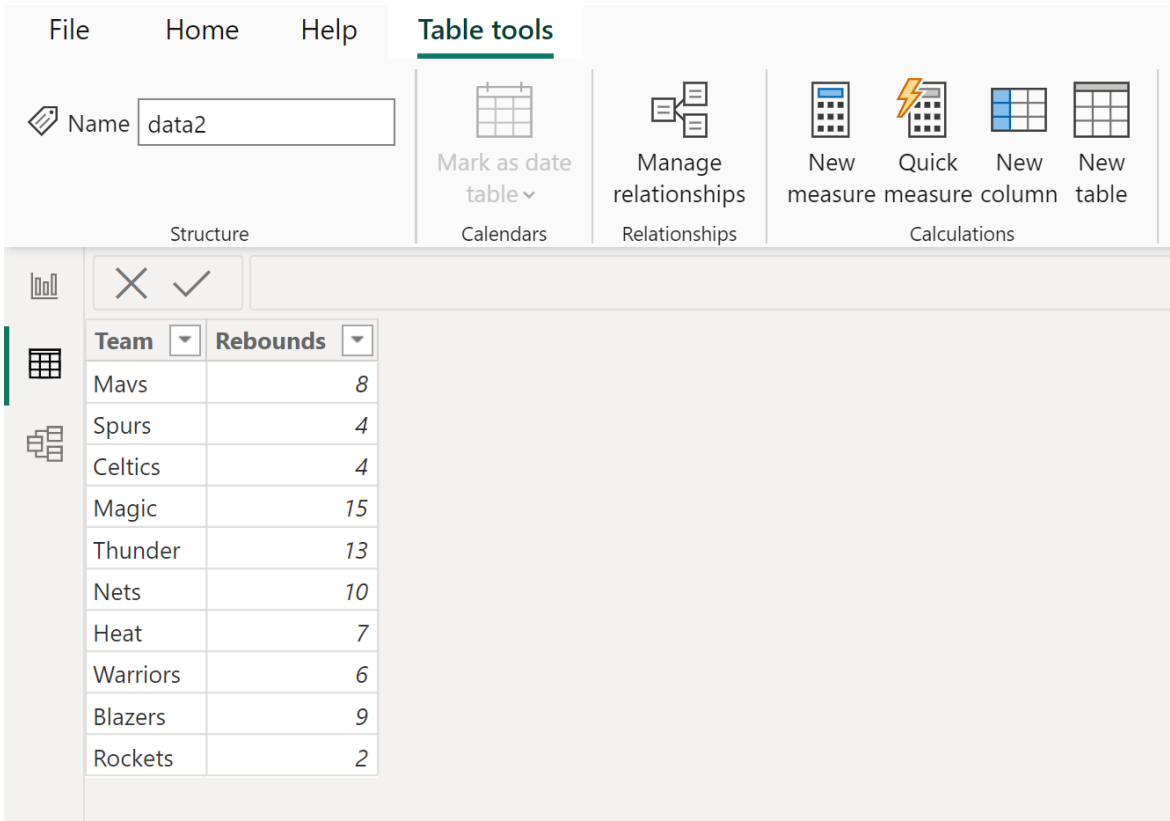
To effectively illustrate the application of the left join functionality, we will use a hypothetical scenario involving two simple datasets focused on basketball performance statistics. We assume that both of these tables have already been successfully loaded into your [Power BI](#) report environment and are ready for transformation.

Our first dataset, designated as **data1**, will serve as our primary, or left, table. This table contains essential information concerning various basketball players, including their designated team affiliations and their total recorded points scored. Crucially, this table includes the vital linking column, **Team**, which will act as the common key for integrating the subsequent dataset.



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

The second dataset, labeled **data2**, functions as the supplementary, or right, table. This table holds complementary performance metrics, specifically rebound counts, which are also categorized and indexed by the **Team** name. Our primary analytical goal is to enrich the **data1** table by seamlessly appending the rebound metrics from **data2**, while simultaneously ensuring that every original player record from **data1** is fully preserved in the final combined output.



The screenshot displays the Power BI interface with the 'Table tools' ribbon active. The ribbon includes options for 'Mark as date table', 'Manage relationships', and 'Calculations'. The 'Calculations' group contains 'New measure', 'Quick measure', 'New column', and 'New table'. Below the ribbon, the 'Structure' pane shows a table with two columns: 'Team' and 'Rebounds'. The table contains the following data:

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

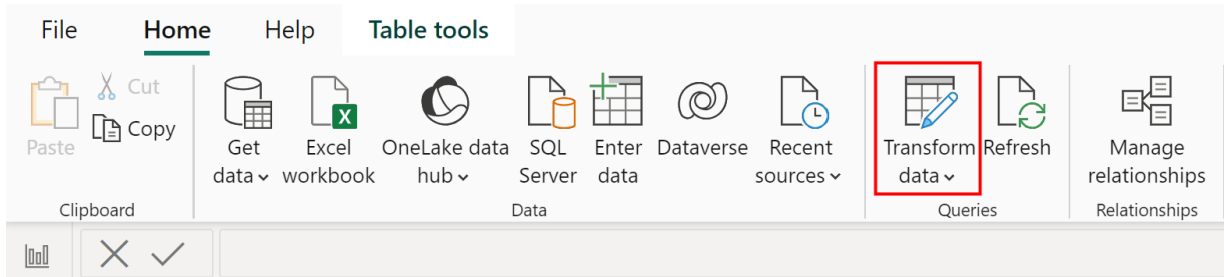
Therefore, we must execute a [left join](#) operation. This approach guarantees that every row originating from **data1** is maintained, attempting to link them with corresponding values found in the **Team** column of **data2**. Any team present in **data1** that does not have a matching entry in **data2** will still appear in the result set, but the associated rebound data will correctly display as blank, adhering strictly to the join logic.

Executing the Merge Operation in Power Query

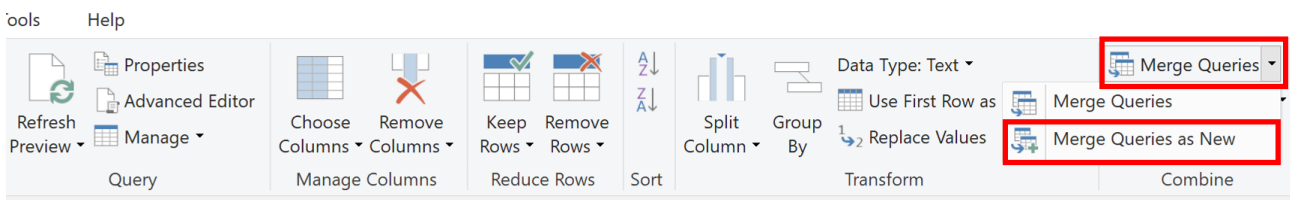
The initial step in combining these tables requires navigating to the [Power Query Editor](#), which serves as the dedicated environment for rigorous data cleaning and [data transformation](#) within Power BI Desktop. To begin, locate the **Home** tab situated on the main ribbon of the application interface.

Under the **Home** tab, click the **Transform data** icon. This action immediately launches the

specialized [Power Query Editor](#) window, where all merging, shaping, and refinement activities are performed. This transition is crucial, moving the user from the visual report design layer into the core underlying data preparation framework.



Once inside the [Power Query Editor](#), the next action is to initiate the merging process. Locate the **Merge Queries** feature within the **Combine** group, which is also located on the Editor's **Home** tab. For complex data integration tasks where the goal is to maintain the integrity of the original source tables, it is highly recommended to select **Merge Queries as New** from the accompanying dropdown menu. Selecting this option generates a brand new, resultant query table, ensuring that the original **data1** table remains entirely unmodified throughout the operation.



Defining the Left Outer Join Parameters

Upon selecting "Merge Queries as New," a dedicated configuration dialog box will appear, requiring precise definitions for the join operation. It is mandatory to specify which tables are to be combined and, most importantly, the exact type of join logic that will govern the combination. Accuracy in this configuration stage is essential for achieving the anticipated merged dataset.

In the configuration window, designate **data1** as the primary table using the top dropdown menu--this establishes it as the critical 'left' table. Subsequently, select **data2** as the secondary table in the bottom dropdown, confirming it as the 'right' table from which the supplementary data will be drawn. Crucially, under the **Join Kind** option, select **Left Outer**. This specific setting enforces the rule that every single row originating from **data1** must be preserved in the resulting query table, regardless of matching status.

To establish the linking key, click on the column header for **Team** in the preview pane of the **data1** table, and then simultaneously click on the column header for **Team** in the preview pane of the **data2** table. Highlighting these common columns instructs Power Query to execute the merge operation by matching values exclusively within these two selected columns.

Merge

Select tables and matching columns to create a merged table.

data1

Team	Points
Mavs	22
Spurs	14
Rockets	19
Kings	30
Warriors	28

data2

Team	Rebounds
Mavs	8
Spurs	4
Celtics	4
Magic	15
Thunder	13

Join Kind

Left Outer (all from first, matching from second)

Use fuzzy matching to perform the merge

‣ Fuzzy matching options

✓ The selection matches 7 of 10 rows from the first table.

OK
Cancel

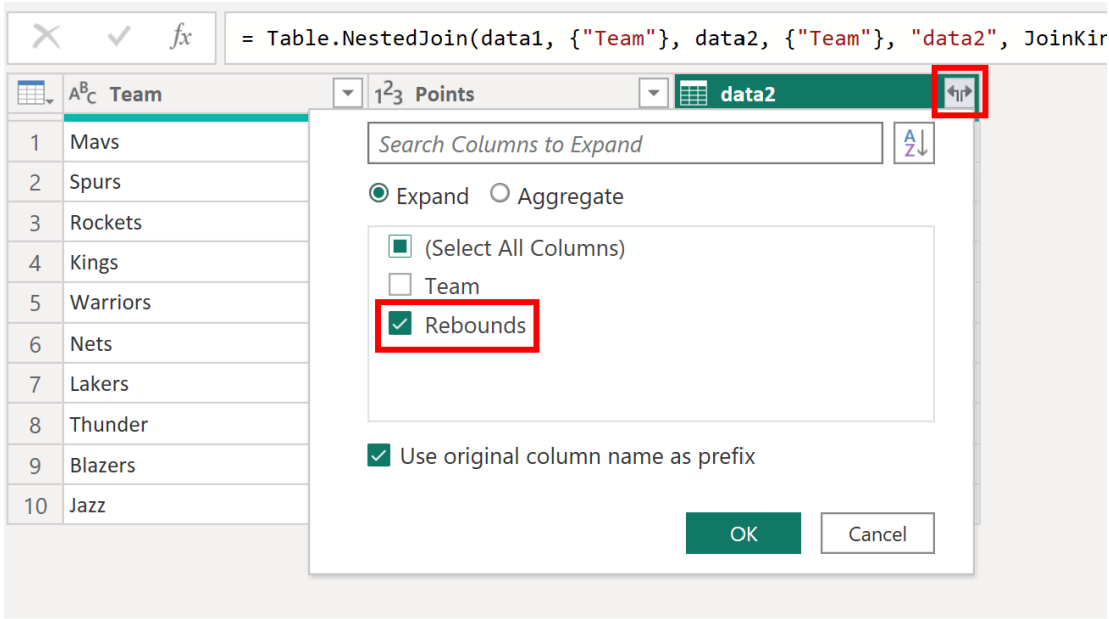
After all parameters--the left table, the right table, the join key, and the **Left Outer** join kind--have been verified and correctly specified, click **OK**. The **Merge Queries** process will execute, resulting in a new query (likely named **Merge1**). Initially, this new table will display the original columns from **data1**, followed by a single column labeled **data2**, which contains structured table values yet to be expanded.

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

Expanding and Applying the Final Merged Query

The newly created **data2** column currently holds the complete table structure corresponding to the linked rows from the right dataset. Since our objective is solely to integrate the **Rebounds** data, the next necessary step is to carefully expand this column to extract the specific metrics required. To initiate this expansion, click on the expand icon (featuring two opposing arrows) located directly on the header of the **data2** column.

A contextual menu will appear, allowing you to select precisely which columns from **data2** should be included in the final merged table. For cleaner field naming, you may deselect the option to use the original column name as a prefix. Most importantly, ensure that only the checkbox adjacent to **Rebounds** is selected. This action explicitly instructs Power Query that **Rebounds** is the only column from the supplementary table that should be appended to the primary dataset.



After confirming the selection by clicking **OK**, the **Rebounds** column will be successfully extracted and positioned alongside the original data from **data1**, thereby completing the composite, merged table structure. At this juncture, the [data transformation](#) is finalized within the [Power Query Editor](#). The new table, typically named **Merge1**, now contains the fully combined and enriched data.

The screenshot shows the final result in the Power Query Editor. The formula bar contains the M code: `= Table.ExpandTableColumn(Source, "data2", {"Rebounds"}, {"data2.Rebounds"})`. The table below has three columns: 'Team', 'Points', and 'data2.Rebounds'. The data is as follows:

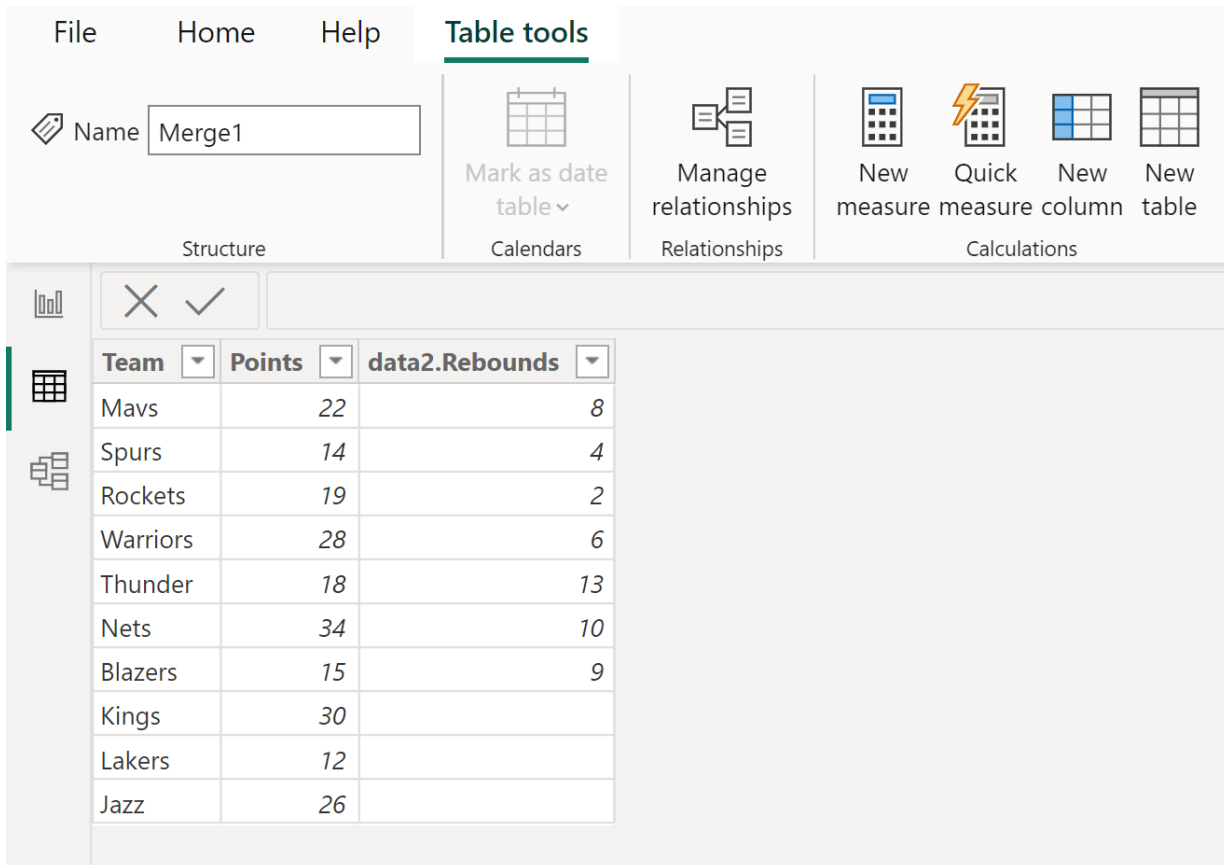
	Team	Points	data2.Rebounds
1	Mavs	22	8
2	Spurs	14	4
3	Rockets	19	2
4	Warriors	28	6
5	Thunder	18	13
6	Nets	34	10
7	Blazers	15	9
8	Kings	30	null
9	Lakers	12	null
10	Jazz	26	null

The final crucial step involves applying these newly performed changes back to the main data model. Exit the [Power Query Editor](#) by clicking **Close & Apply** in the Home tab. A confirmation message box will prompt you to approve the application of all transformations executed during the session. You must click **Yes** to ensure that the newly constructed **Merge1** table is successfully

loaded and becomes fully available for use in your report visualizations and subsequent calculations within the Power BI environment.

Analyzing the Results and Data Integrity

With the changes successfully applied, the analyst can navigate to the Table view within Power BI Desktop. The new merged table, usually labeled **Merge1**, will be visible and ready in the Fields pane. This table serves as tangible evidence of the successful execution of the **left join** operation.



The screenshot displays the Power BI Desktop interface. The 'Table tools' ribbon is active, showing options like 'Name' (Merge1), 'Mark as date table', 'Manage relationships', and 'Calculations'. Below the ribbon, a table view is shown with the following data:

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 primary observation confirming the success of the join is the preservation of records: every row originating from the initial left table (**data1**) has been retained in the merged output. The rebound values have been populated only for those records where the **Team** value found a corresponding match in the right table (**data2**). Critically, for any record from **data1**--such as a specific player or team that had no corresponding rebound entry in **data2**--the resulting **Rebounds** column correctly displays a blank or null value.

This outcome unequivocally confirms the successful implementation of the **Left Outer Join**. It demonstrates how this methodology effectively enriches a primary dataset by appending supplementary information while strictly preserving the integrity and completeness of the source

data model.

As a final refinement step for enhanced model readability, analysts often choose to rename the merged column. You have the option to right-click on the header labeled **data2.Rebounds** directly within the [Power Query Editor](#) interface and rename it to a simpler term, such as **Rebounds**, before applying the final changes to the data model.

Expanding Your Data Integration Skills

Mastery of the [Merge Queries](#) function is not merely a technical skill but a foundational requirement for robust data modeling. For those professionals seeking to advance their expertise beyond the standard left join, the following resources offer valuable tutorials and detailed explanations covering other complex [data transformation](#) and integration techniques used in Power BI.

[How to Perform a Full Outer Join in Power BI](#)

[Understanding the Key Difference Between Append and Merge in Power BI](#)

[How to Unpivot Columns for Optimized Data Analysis in Power BI](#)

[Working Effectively with Advanced \[Data Transformation\]\(#\) Functions in Power Query](#)