How to Perform an Inner Join in Power BI: A Step-by-Step Guide

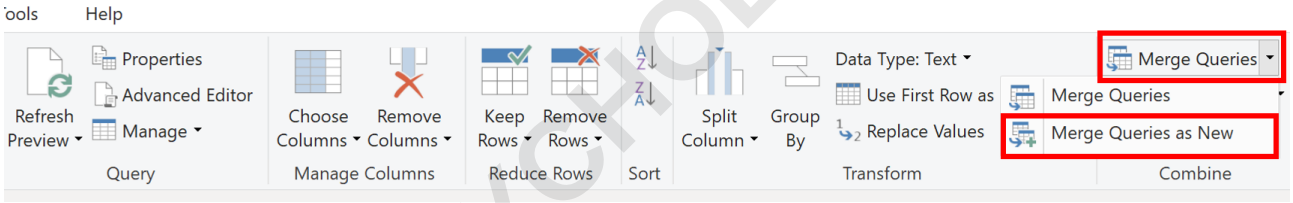
Authored by

mohammed looti

January 9, 2026

The process of performing an Inner Join within Power BI requires two primary components: having at least two distinct tables that share a common key or column, and utilizing the robust functionality of the Query Editor. Once these prerequisites are met, the user initiates the "Merge Queries" function, carefully selecting the two data sources intended for integration. The critical step involves specifying the join type--in this particular instance, an Inner Join--and accurately identifying the common column that facilitates the row matching. The successful execution of this operation yields a new, consolidated table containing only those rows where matching values existed in the specified key column across both original sources. This precise combination of data is fundamental for subsequent data analysis and visualization efforts within the Power BI environment.

The most straightforward and efficient methodology for executing an inner join between two relational tables in Power BI is through the utilization of the dedicated **Merge Queries** feature, which resides within the comprehensive environment of the **Power Query Editor**. This function is essential for combining data sets derived from disparate sources into a unified structure, enabling more complex and holistic reporting capabilities.



To illustrate the practical application of this powerful feature, the following detailed example provides a step-by-step walkthrough, ensuring clarity for users aiming to master data integration techniques in the platform.

Understanding the Inner Join Concept

Before diving into the procedural steps within Power BI, it is critical to firmly grasp the theoretical foundation of an Inner Join. In the context of database management and relational data integration, an Inner Join is a fundamental operation designed to combine rows from two or more tables based on a shared, specified column value. The defining characteristic of this join type is its strict adherence to matching records; only rows that possess corresponding values in the key columns of both the left (primary) and right (secondary) tables are included in the final, resulting data set. This effectively filters out all records unique to either source, yielding a precise intersection of the data.

The resulting table, created after executing the Inner Join, contains all columns from both original

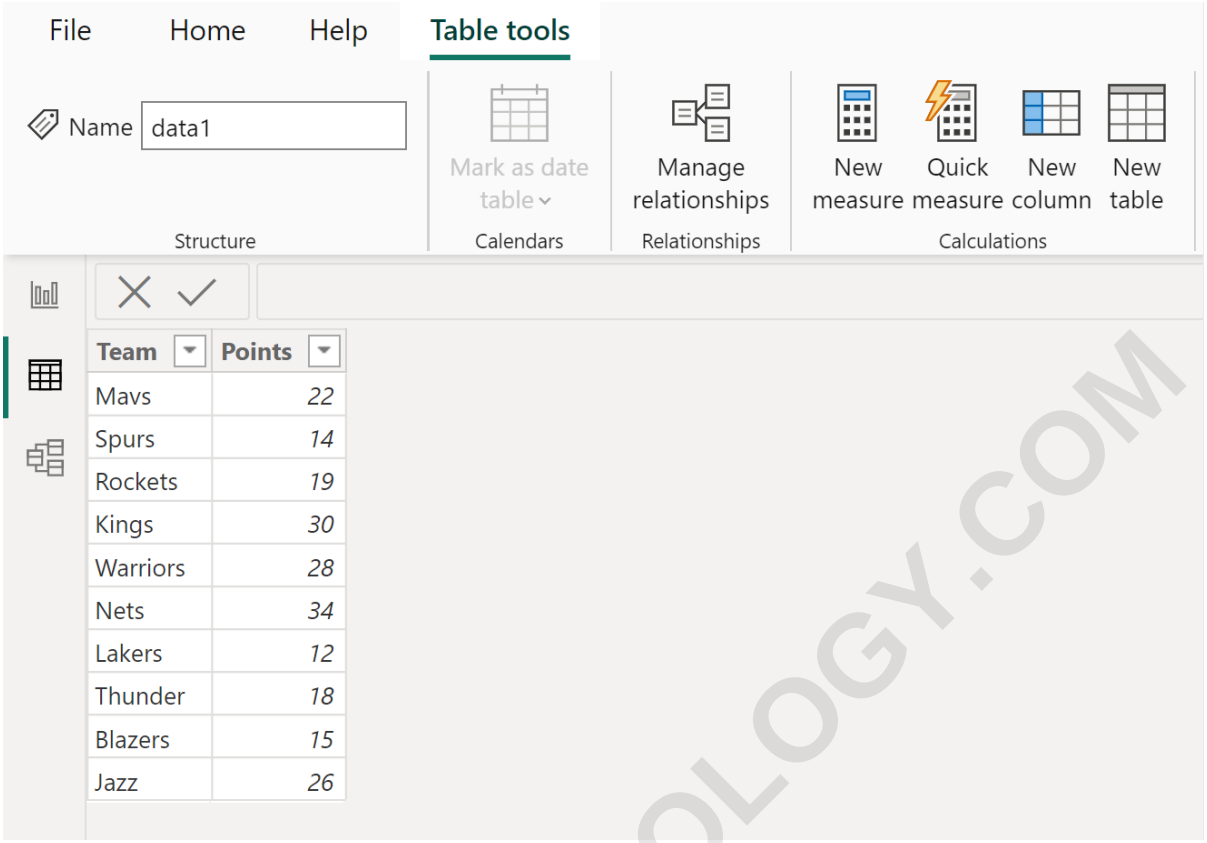
tables, but its rows are strictly limited to the matched entries. This is particularly useful when performing data analysis where completeness and synchronization across data sources are paramount. For instance, if one table holds sales records (SalesID, CustomerID, Amount) and another holds customer demographics (CustomerID, Region, Age), an Inner Join ensures that the final dataset only includes sales records for customers who actually exist in the demographics table, and vice versa. This precision is vital for maintaining data integrity and focusing analysis on truly correlated observations.

Understanding the distinction between an Inner Join and other join types--such as Left Outer, Right Outer, or Full Outer Joins--is essential for accurate data modeling. Unlike outer joins, which preserve non-matching records from one or both sides (filling missing values with nulls), the Inner Join acts as a strict filter, demanding symmetry in the linking column. Mastery of the Inner Join allows data professionals to accurately subset and harmonize complex data structures, forming the bedrock for effective reporting and dashboard creation.

Prerequisites: Setting Up the Data Environment

To follow this guide, we must first establish the sample data tables necessary for the join operation. We will work with a hypothetical scenario involving basketball statistics, where data is split across two distinct tables. The first table, named **data1**, contains core information about players, specifically detailing the team name and the points they have scored. This table serves as our primary, or "left," table in the merge operation.

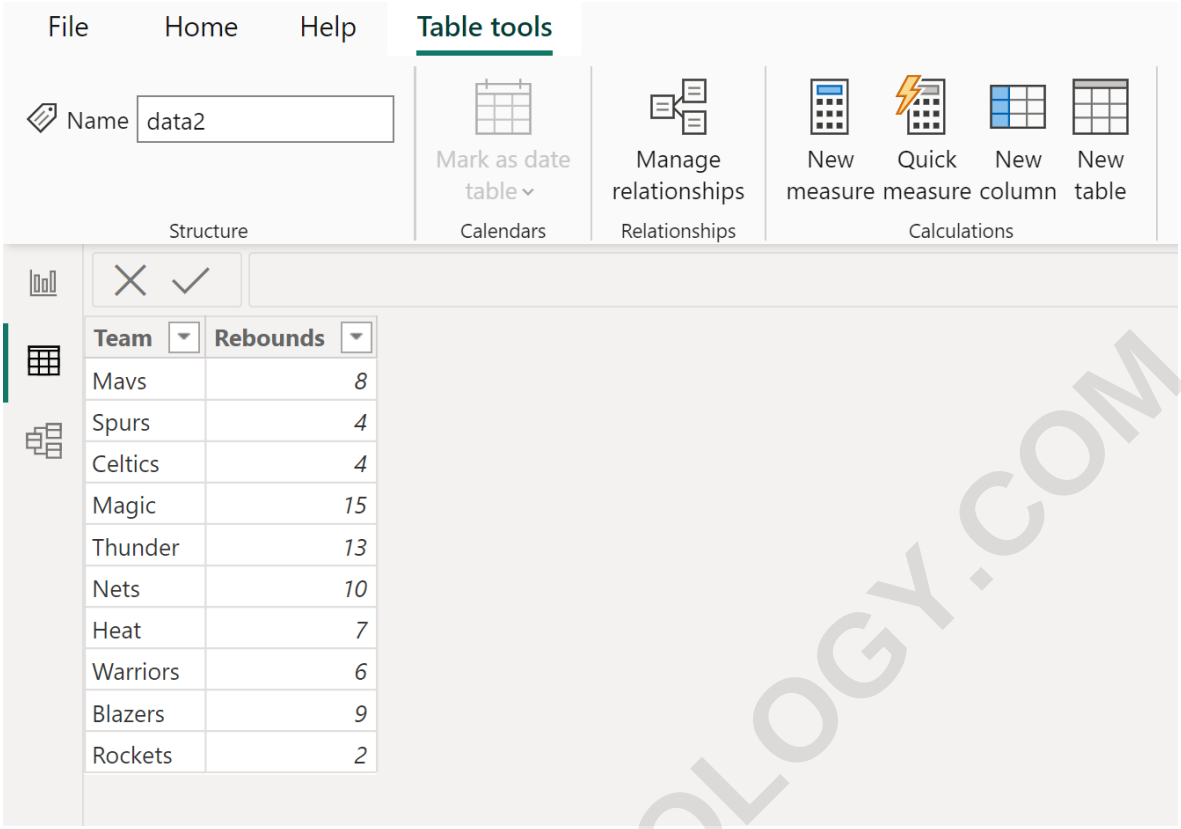
Suppose we have the following table in Power BI named **data1** that contains information about the team name and points scored for various basketball players:



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

Furthermore, we require a secondary data source to supplement the information in **data1**. Let us assume we have a second table, named **data2**, which contains additional performance metrics, specifically information detailing the team name and the number of rebounds achieved by the players. This table will function as our "right" table. Crucially, both tables share a common field: the **Team** column. This shared column is the linchpin that will facilitate the accurate merging of records, as it provides the unique identifier linking related rows across the two datasets.

And suppose that we have another table named **data2** that contains information about the team name and rebounds for various basketball players:



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

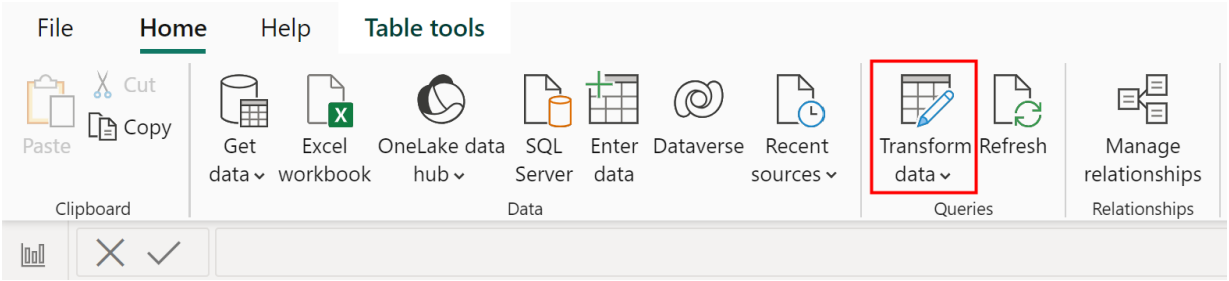
Our specific goal for this operation is to perform an inner join. We aim to strictly preserve only those rows from both tables that possess identical, corresponding values in the **Team** columns. Any team that appears in only one table will be systematically excluded from the final consolidated result, ensuring that we only analyze data where both points and rebounds statistics are available for the matching teams.

Step 1: Accessing the Power Query Editor

The initial and most crucial step in any data transformation or merging process within Power BI Desktop is to access the dedicated **Power Query Editor**. This environment is where all advanced data preparation, shaping, and combining activities occur. To initiate this process, users must navigate to the primary ribbon interface of Power BI Desktop.

To launch the editor, locate and click the **Home** tab situated along the top ribbon menu. Within the options presented, you will find a dedicated group of tools for data transformation. Click the **Transform data** icon (often represented by an icon indicating data modification or structure change). This action initiates the transition from the report view of Power BI Desktop into the powerful Power Query Editor interface, where the merging operation will be executed.

To do so, click the **Home** tab along the top ribbon, then click the **Transform data** icon:



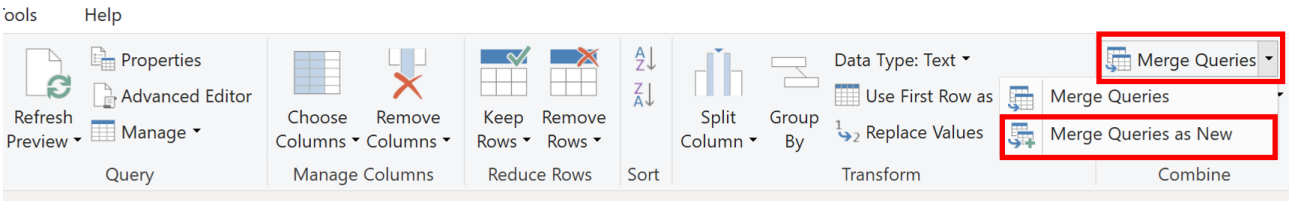
This action successfully transports the user into the dedicated **Power Query Editor** environment, which is necessary for performing the next steps of data merging and cleaning. If the editor is already open, you may skip this preliminary step, but ensuring the editor is active is the prerequisite for proceeding with data integration tasks.

Step 2: Initiating the Merge Queries Process

Once inside the Power Query Editor, the next step involves locating and activating the feature specifically designed for combining tables: **Merge Queries**. This tool is typically found within the **Combine** group, located on the active **Home** tab of the Query Editor's ribbon interface. The choice of how to merge--either replacing an existing table or creating a new one--is crucial for workflow management and preserving source data integrity.

For best practices, especially when dealing with complex data models or requiring the original data sources to remain unmodified, it is highly recommended to create a new merged table. Therefore, after clicking the **Merge Queries** icon, the user should select the option **Merge Queries as New** from the resulting dropdown menu. This ensures that the original tables, **data1** and **data2**, remain untouched by the merging operation, providing a non-destructive transformation path and allowing for easier auditing or rollback if necessary.

Next, click the **Merge Queries** icon in the **Combine** group of the **Home** tab. Then click **Merge Queries as New** from the dropdown menu:



The selection of **Merge Queries as New** immediately prompts a new configuration dialog box, which serves as the interface for defining the parameters of the join operation. It is within this

dialog that the user specifies which tables to combine, identifies the common key columns, and crucially, defines the type of join required, which in our case is the strict Inner Join.

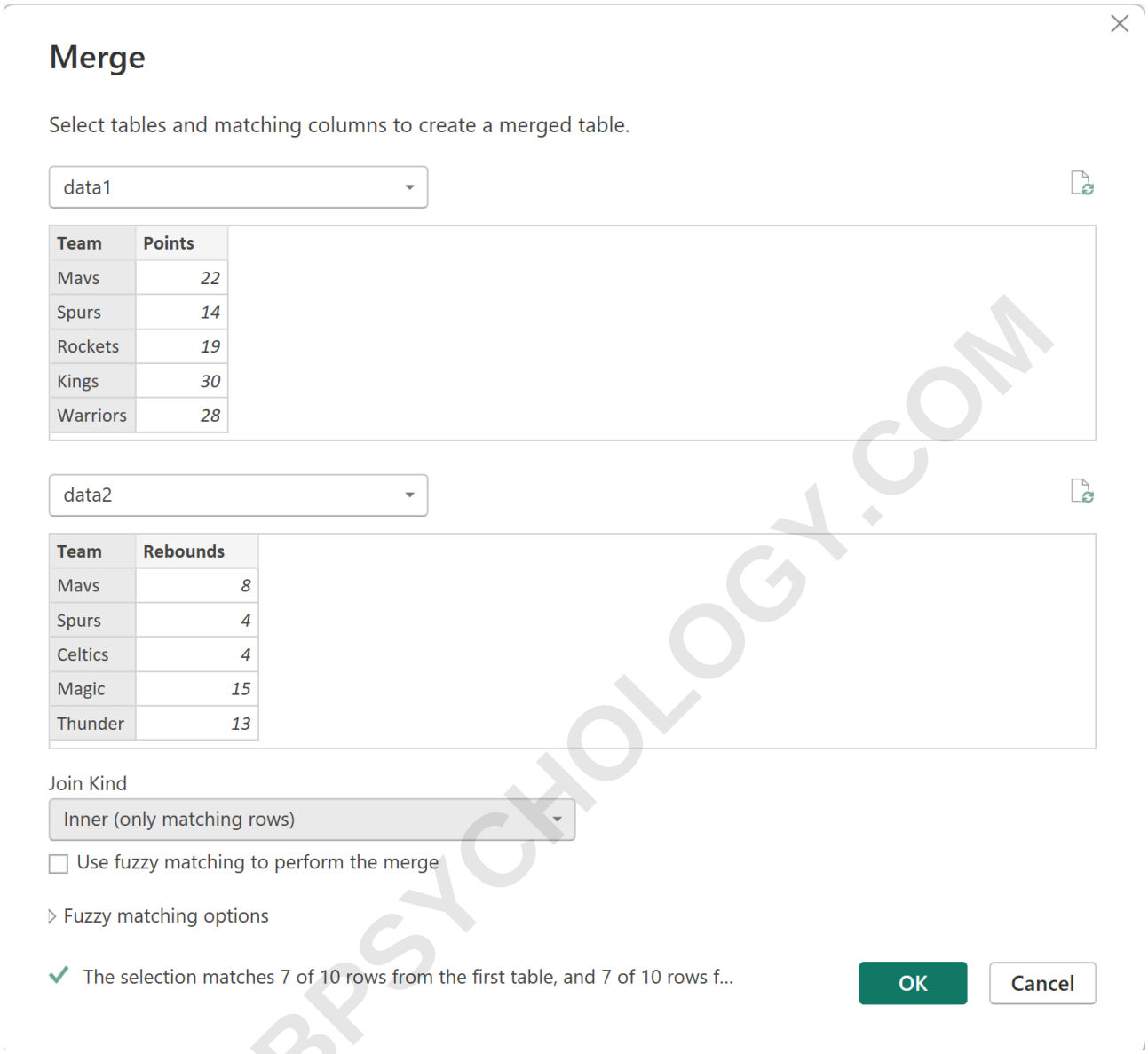
Step 3: Defining the Join Parameters

The Merge dialogue window is the control panel for specifying how the two datasets will interact. Within this interface, the user must meticulously define the primary table, the secondary table, the common linkage column, and the specific join type. For our example, in the upper dropdown selector, choose **data1**, designating it as the primary or left table. In the secondary dropdown selector below it, select **data2**, marking it as the right table in the operation.

The most critical definition is the common column. Click on the header of the **Team** column in both the **data1** preview and the **data2** preview simultaneously. The Power Query Editor will visually confirm this selection, typically by highlighting the column headers, indicating that the merge operation will use these fields to find matching records. This linkage establishes the integrity of the combined data set, ensuring that only related basketball teams' statistics are paired together.

Finally, utilize the **Join Kind** dropdown menu to explicitly select the **Inner (only matching rows)** option. Selecting **Inner** mandates that the resulting table, which will be temporarily named 'Merge1,' will only contain rows where the selected **Team** value exists in **both** the **data1** and **data2** tables. After ensuring all parameters are correctly configured, click **OK** to execute the join operation and display the initial results in the editor.

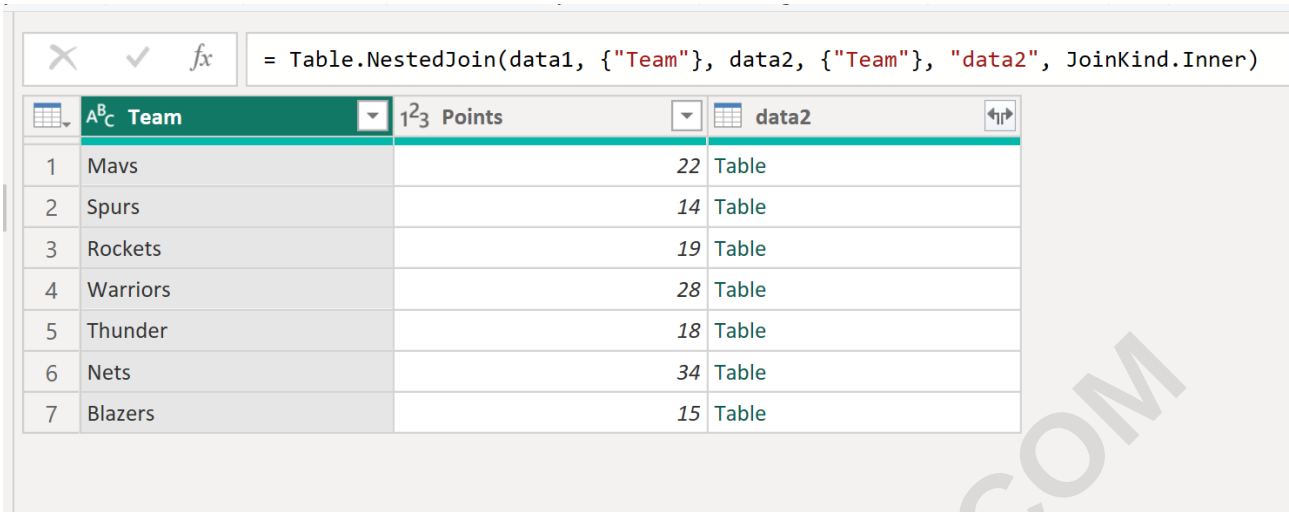
In the new window that appears, choose **data1** as the first table, choose **data2** as the second table, and choose **Inner** as the **Join Kind**. Note that you must click the **Team** column in both previews to establish the join key.



Step 4: Expanding and Finalizing the Merged Data

Upon clicking **OK**, the inner join is initially executed. The result displayed in the **Power Query Editor** will show the columns from the primary table (**data1**) along with a single new column, typically named after the secondary table (**data2**). This new column contains Table objects, representing the matched rows from the secondary source. This structure is a temporary placeholder and requires expansion to expose the necessary data fields.

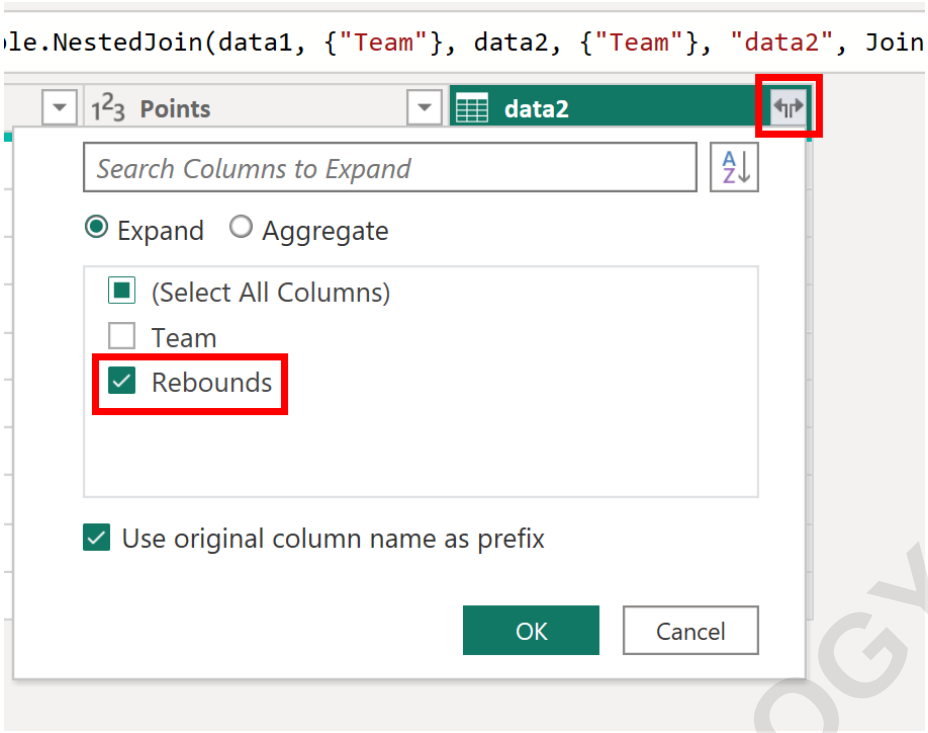
Once you click **OK**, the inner join will be performed:



	Team	Points	data2
1	Mavs	22	Table
2	Spurs	14	Table
3	Rockets	19	Table
4	Warriors	28	Table
5	Thunder	18	Table
6	Nets	34	Table
7	Blazers	15	Table

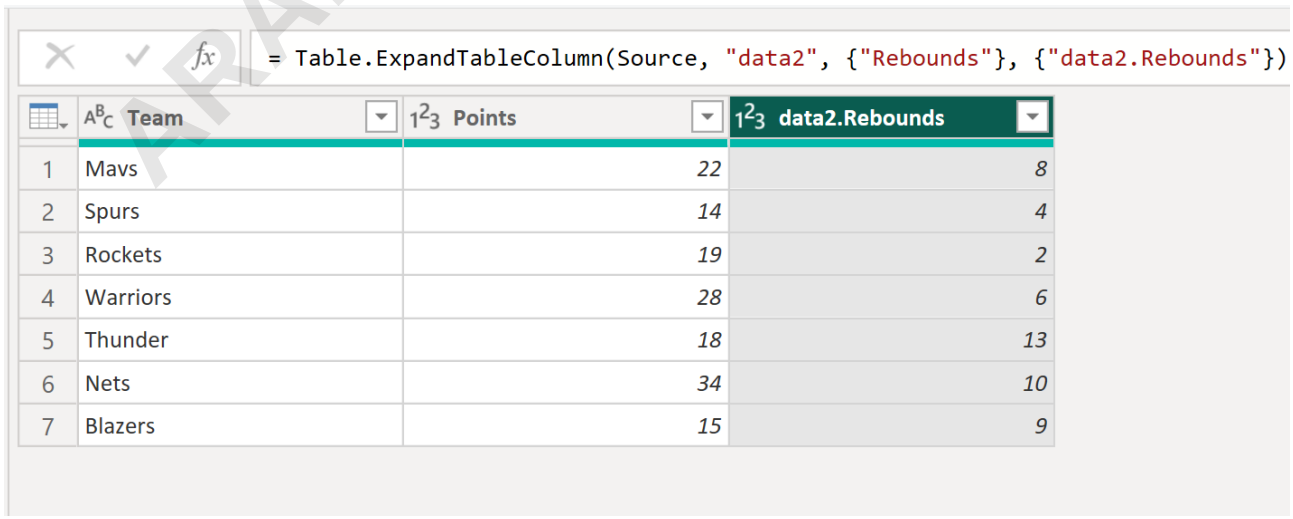
To integrate the required columns from **data2**, locate the header of the new column (**data2**) and click the icon featuring left and right arrows (the Expand icon). A small dialogue box will appear, allowing the user to select which specific columns from the merged table should be extracted and appended to the final result. Since **data1** already contains the **Team** column, we only need the supplementary metric. Therefore, uncheck all options except for **Rebounds**, and ensure the checkbox labeled "Use original column name as prefix" is deselected if you prefer a cleaner column header (or leave it checked if ambiguity might exist later).

Next, click the left and right arrows on the header of the **data2** column. Then check the box next to **Rebounds** to indicate that this column should be the only one included from **data2** in the final merged table:



Executing the expansion step transforms the single 'Table' column into the desired data field. Once you click **OK** after selecting the expansion options, the **Rebounds** column will be successfully displayed, now integrated alongside the data from **data1**. This combined table, usually named **Merge1** by default, now holds a harmonized dataset containing Player, Team, Points, and Rebounds, but only for teams present in both original sources. The resulting table is ready for loading into the Power BI data model.

Once you click **OK**, the **Rebounds** column will be shown from the **data2** table:



	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

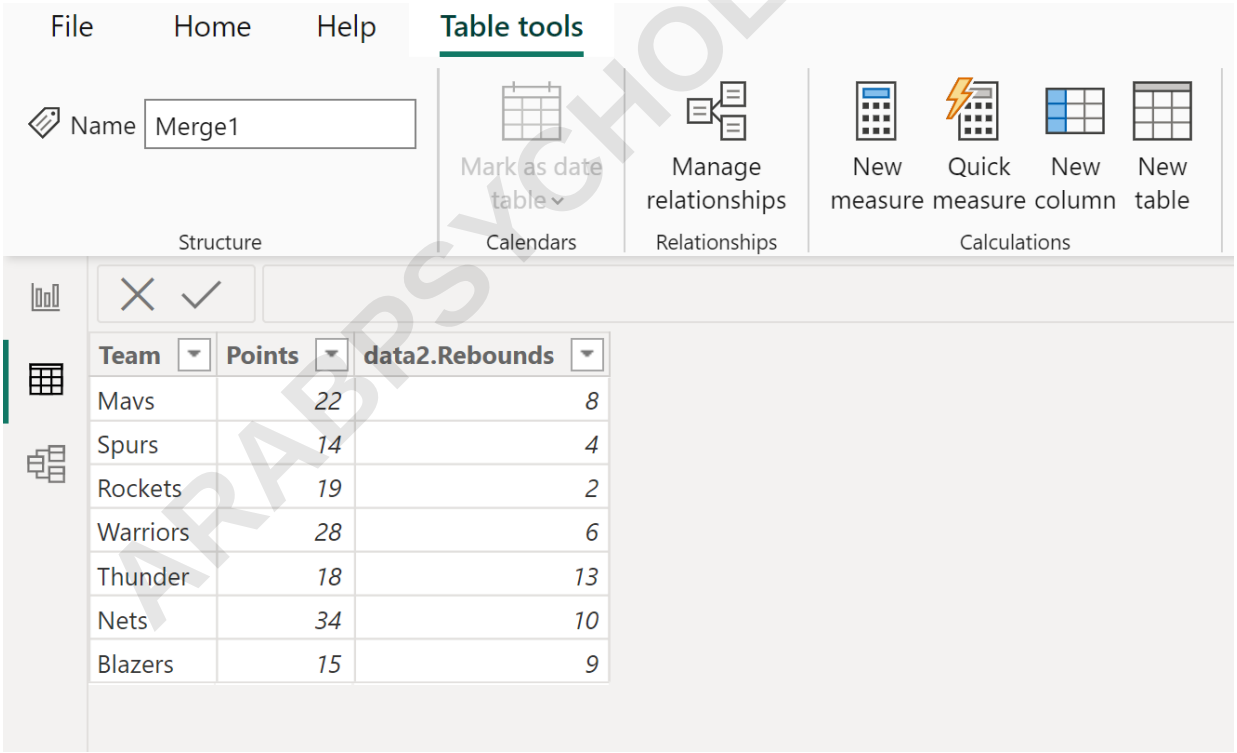
Conclusion and Application of Changes

With the Inner Join successfully performed and the new columns expanded within the **Power Query Editor**, the data transformation phase is complete. The final step involves applying these changes to the main Power BI data model. Exit the **Power Query Editor** by selecting **Close & Apply** on the Home tab of the editor. If you simply close the window, a message box will appear prompting confirmation to apply the pending changes to the underlying model.

Click **Yes** to confirm the application of the steps taken in the editor. Power BI will then execute the M code generated by the Merge Queries operation, loading the new table, **Merge1**, into the data model. You will then be able to locate and view this new, harmonized data set in the Table view within Power BI Desktop, ready for visualization and reporting.

Once you exit out of the **Power Query Editor**, 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:



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

Observe the structure of the resulting table: **Merge1** strictly adheres to the Inner Join rule. Only the rows in which the value from the **Team** column appeared identically in both the **data1** and **data2** tables are included in this final merged table. This ensures that your subsequent data analysis is based solely on the intersecting records, providing highly reliable and correlated metrics.

Note: If the resulting column header is named **data2.Rebounds** due to the prefix setting, you have the option to right-click on this header within the Power Query Editor and use the "Rename" function to change the column name simply to **Rebounds**, improving clarity for report consumption.

Further Data Integration Techniques

Mastering the Inner Join is just one component of effective data modeling in Power BI. Depending on your analytical requirements, you may need to explore alternative join methods or other transformation techniques to prepare your data fully for visualization. For example, if you needed to retain all rows from the primary table regardless of a match in the secondary table, a Left Outer Join would be the appropriate choice instead of the restrictive Inner Join.

Understanding when to use the powerful **Merge Queries** feature versus the simpler **Append Queries** function is also vital. While Merge Queries (joins) combine tables horizontally based on a relationship key, Append Queries stack tables vertically, assuming they have identical column structures. These tools, accessible via the **Power Query Editor**, offer comprehensive flexibility for nearly any data integration challenge encountered during advanced data analysis projects.

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

How to Add Index Column to Table in Power BI