

Learning Inner Joins in Power BI: A Comprehensive Tutorial

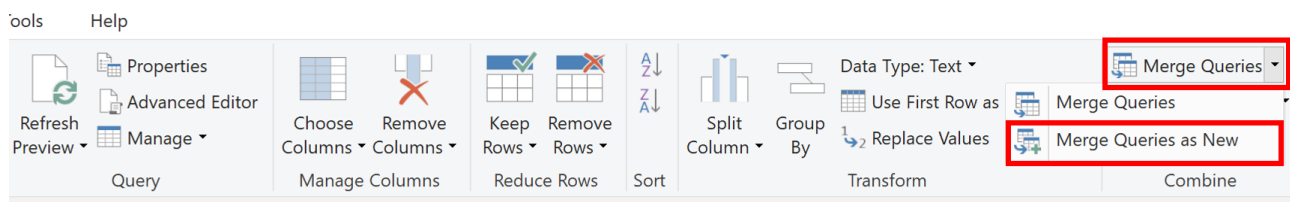
Authored by
Mohammed loot

November 12, 2025

RECOMMENDED CITATION

Mohammed loot (2025). *Learning Inner Joins in Power BI: A Comprehensive Tutorial*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=17908>

Data integration stands as a fundamental requirement in modern business intelligence workflows. The powerful analytical tool, [Power BI](#), provides developers with robust mechanisms designed specifically for combining disparate datasets effectively. When managing complex data models sourced from multiple tables, the most frequent operation required is merging these sources based on shared identifier keys. The most precise and efficient technique for performing an [inner join](#) between two tables within the Power BI environment involves utilizing the specialized [Power Query Editor](#) and its versatile **Merge Queries** feature. This carefully controlled process guarantees that only rows exhibiting matching values across the designated primary and secondary tables are retained, yielding a clean, highly focused resulting dataset perfectly suited for advanced analysis and visualization tasks.



This comprehensive guide is designed to walk you step-by-step through the practical execution of an inner join operation in Power BI Desktop. We will offer detailed explanations and accompanying visuals to ensure maximum clarity throughout the process. A strong understanding of how to correctly merge data sources is absolutely critical for building accurate, reliable, and meaningful data models within the [Power BI](#) ecosystem. To effectively illustrate the process, we will utilize a specific and highly relatable scenario involving basketball performance statistics.

Grasping the Core Concept of the Inner Join

Before diving into the procedural steps necessary for data transformation, it is essential to establish a clear understanding of the core functionality and implications of an [inner join](#). This join type differs significantly from its counterparts (such as left, right, or full outer joins) because it is inherently restrictive in its logic. The primary and sole objective of an inner join is to produce a new combined table that contains only those records for which matching values exist in the specified key column(s) of **both** tables being linked.

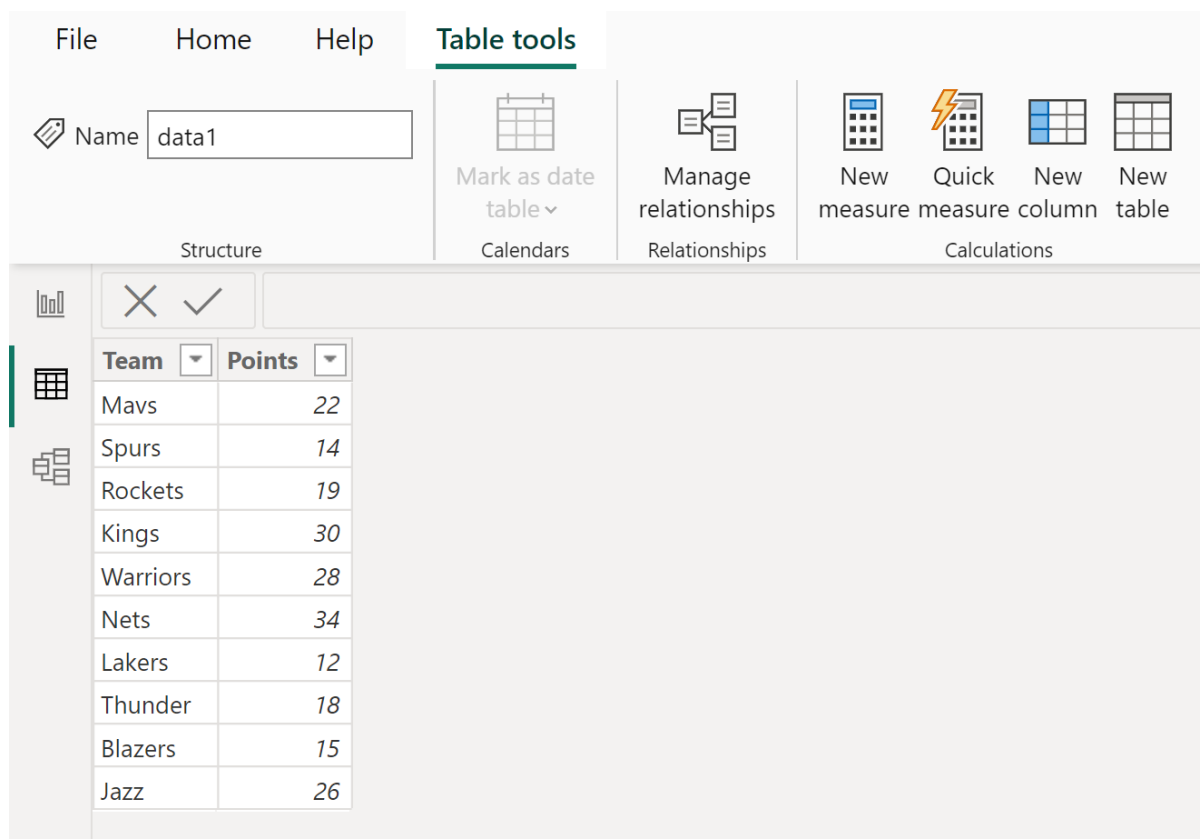
If a record is present in the first table but lacks a corresponding match in the second table, or vice versa, that particular record is systematically and completely excluded from the final output dataset. This precise filtering mechanism makes the inner join an indispensable tool for scenarios requiring the exclusion of incomplete or inconsistent data, ensuring strict alignment between two datasets where consistency across a shared unique identifier is absolutely mandatory for analysis integrity.

In the broader context of relational databases and modern data analysis, various join operations are considered fundamental components of [data transformation](#). Power BI manages these complex transformations using the sophisticated M language engine, which is housed within the user-friendly interface of the [Power Query Editor](#). By leveraging the intuitive **Merge Queries** feature, users are able to define the specific join type (in this case, Inner) and designate the common column(s) that will serve as the linkage points between the primary and secondary tables. This graphical, structured approach is generally preferable to attempting to construct complex DAX formulas for straightforward data merging requirements.

Preparing the Illustrative Data Scenario

To provide a practical demonstration of the inner join process, we will establish a straightforward scenario centered around combining basketball performance statistics. We begin with two distinct data tables that have already been successfully imported into our [Power BI](#) report. These tables share a common, crucial identifier: the **Team** name, but they measure different statistical categories, specifically Points and Rebounds.

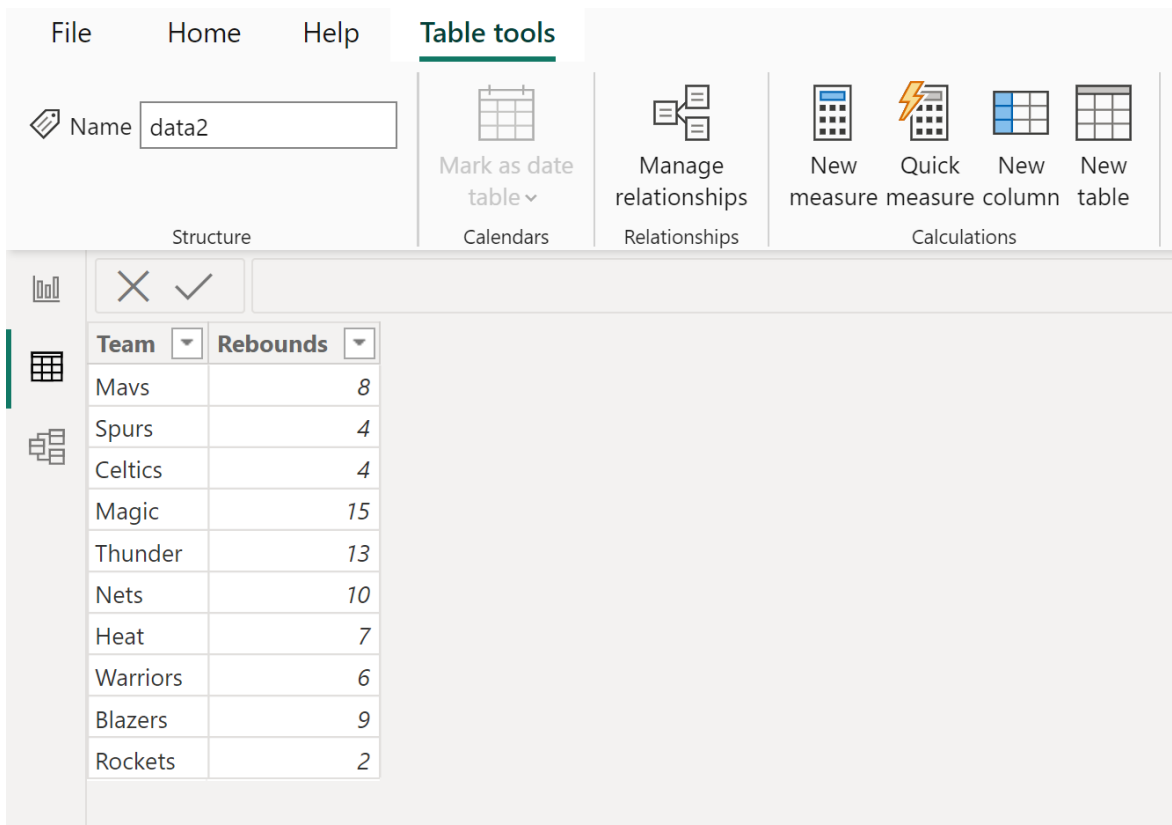
Our first table, logically named **data1**, provides essential information regarding player names, their affiliated team names, and the corresponding points scored by those players:



The screenshot displays the Power BI interface with the 'Table tools' ribbon selected. The ribbon includes options for 'Mark as date table', 'Manage relationships', and 'Calculations' (New measure, Quick measure, New column, New table). Below the ribbon, a data table is visible with the following data:

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

The second table, designated **data2**, complements the first by detailing information about the team name and the rebounds achieved by the players. It is critically important to verify that for a successful inner join, the data types of the joining columns--in this case, the **Team** column--must be identical across both source tables. Our ultimate objective is to combine these two tables, ensuring that we only retain player statistics for teams that are present in both **data1** and **data2** simultaneously.



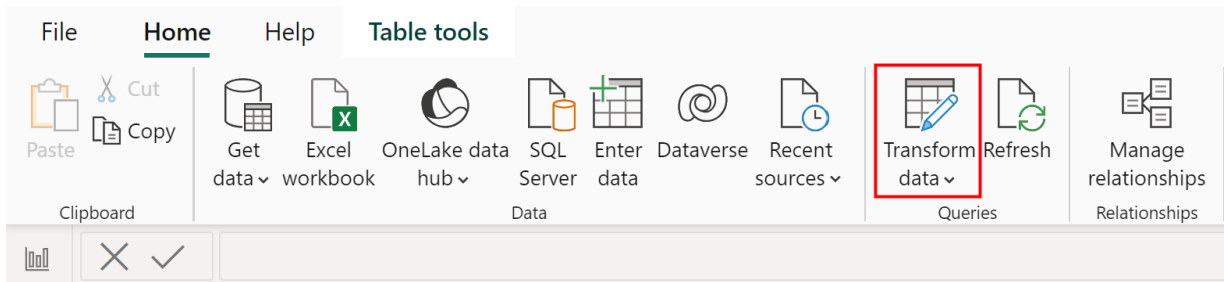
Team	Rebounds
Mavs	8
Spurs	4
Celtics	4
Magic	15
Thunder	13
Nets	10
Heat	7
Warriors	6
Blazers	9
Rockets	2

The goal of the operation is precise: we will execute an **inner join** where we explicitly choose to keep only those rows from both tables that possess matching values exclusively within the designated **Team** columns. Any team listed in one source table but lacking a corresponding entry in the other will be systematically filtered out and excluded from the final merged output.

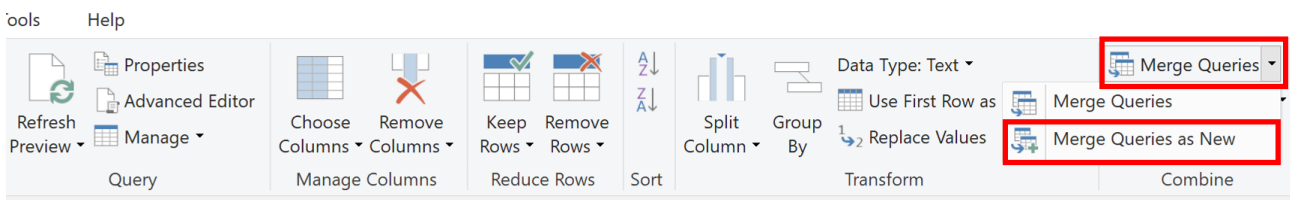
Accessing the Power Query Editor for Transformation

The initial mandatory step for executing any significant data restructuring or merging operation within Power BI requires users to transition away from the standard Report or Data view interfaces and into the specialized data preparation environment. To begin this process, locate the **Home** tab positioned along the top ribbon interface of Power BI Desktop. Within this tab, you must identify and click the critical **Transform data** icon. This action immediately launches the dedicated **Power Query Editor** window, which serves as the central hub for all data preparation, shaping, and

cleaning activities before the final data model is constructed.



Upon successfully entering the [Power Query Editor](#), you gain access to an extensive array of tools engineered specifically for cleaning, shaping, and effectively combining various data sources. To proceed with our join, ensure you remain on the **Home** tab within the Editor interface. Look towards the right-hand side of the ribbon for the **Combine** group of functions. Within this grouping, you will find the **Merge Queries** icon. Since our intention is to generate a new, merged table rather than modifying one of our existing source tables in place, the correct selection is **Merge Queries as New** from the accompanying dropdown menu. Utilizing this option is considered a best practice as it ensures that the integrity of the original source data remains entirely untouched, which facilitates both reproducibility and easier debugging should any issues arise during the transformation sequence.



Defining and Executing the Inner Join Operation

Clicking the **Merge Queries as New** option initializes a dedicated configuration window where all parameters governing the join operation must be clearly defined. This critical window requires three essential inputs to successfully execute the [inner join](#). First, you must explicitly specify the primary table, which is also referred to as the left side of the join; in our current basketball example, we select **data1**. Second, you must select the secondary table (the right side of the join), which is **data2**. Crucially, after selecting the tables, you must click on the common joining column--the **Team** column--in both tables simultaneously to highlight it and formally establish the linkage point between the two datasets.

The final, and arguably most important, configuration step involves setting the **Join Kind**. From the provided dropdown menu, you must select the option labeled **Inner (only matching rows)**. This precise selection explicitly instructs the [Power Query Editor](#) to rigorously apply the restrictive inner join logic, thereby discarding all rows where the identifier value in the **Team** column does not appear in both **data1** and **data2**. Once all of these parameters have been correctly set, the merging configuration window should clearly display the chosen tables, the selected joining columns, and the specified join type, providing a clear visual confirmation of the defined operation.

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

Inner (only matching rows)

Use fuzzy matching to perform the merge

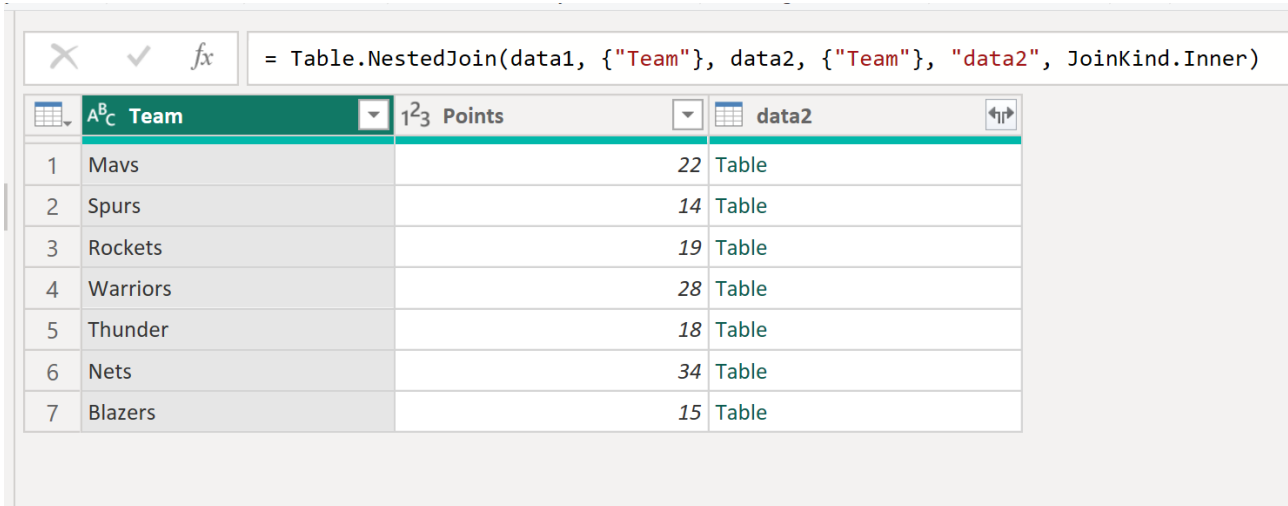
› Fuzzy matching options

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

OK Cancel

After confirming these critical settings and clicking **OK**, the inner join is executed virtually instantaneously by the Power Query engine. [Power BI](#) returns a new table, which is typically labeled **Merge1** by default. This initial result table contains all the original columns from **data1**, plus a single new structured column named **data2**. This structured column holds the complete set of

linked rows retrieved from the secondary table, and it must now be expanded to properly integrate the desired statistical columns, such as Rebounds, into the main merged table structure for analysis.

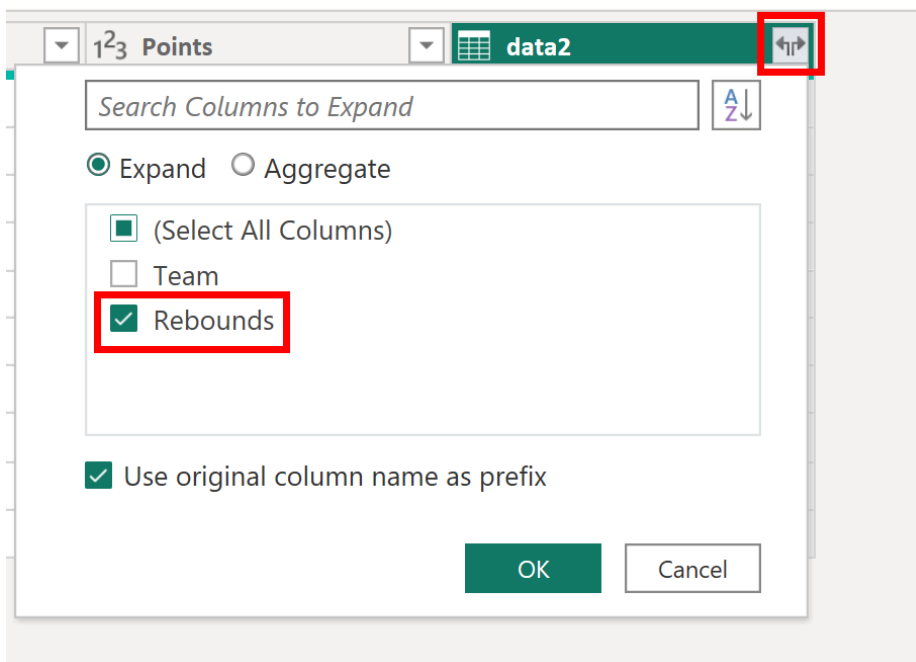


	Team	Points	data2
1	Mavs	22	Table
2	Spurs	14	Table
3	Rockets	19	Table
4	Warriors	28	Table
5	Thunder	18	Table
6	Nets	34	Table
7	Blazers	15	Table

Finalizing the Data Structure and Applying Changes

The next essential step in finalizing the merged table involves extracting the relevant data points hidden within the structured **data2** column. To achieve this necessary expansion, locate the column header for **data2** and click on the small icon containing left and right arrows. This action immediately triggers the column expansion menu. Within this menu, you will be presented with a comprehensive list of available columns sourced from the secondary table. Since our analysis only requires the rebound statistics, ensure that only the checkbox adjacent to **Rebounds** is selected. For the sake of producing cleaner column headers, it is generally recommended to uncheck the option to "Use original column name as prefix."

```
Table.NestedJoin(data1, {"Team"}, data2, {"Team"}, "data2", Join
```



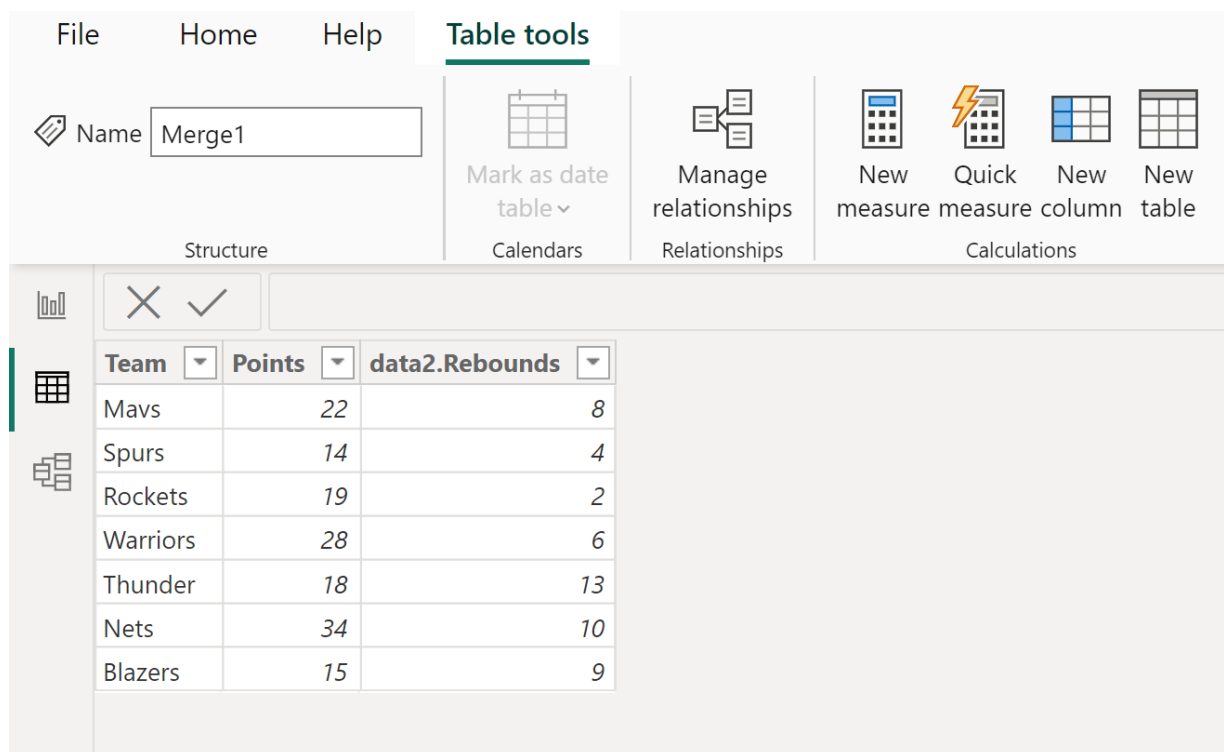
Once you click **OK** after successfully selecting **Rebounds**, the [Power Query Editor](#) executes the expansion step. The new column, now temporarily named **data2.Rebounds**, will be fully integrated into the merged table structure, positioned alongside the original player and points data derived from **data1**. At this point, the core data transformation is complete within the editor environment, and the resulting table contains only the combined datasets that successfully matched across the **Team** column in both original source tables.

`= 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

To integrate this newly transformed table into your main Power BI Data Model, you must exit the **Power Query Editor** environment. Navigate back to the primary Power BI ribbon and click the

command **Close & Apply**. A message box will typically appear, prompting you to confirm your intent to apply the changes made during the data transformation process. Clicking **Yes** confirms the changes, ensuring that the new, structured data is loaded into the Power BI model. You can subsequently view the resulting table, which is usually named **Merge1**, within the Table view of Power BI Desktop.



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', and 'Calculations' (including 'New measure', 'Quick measure', 'New column', and 'New table'). Below the ribbon, a table view shows the merged data with columns 'Team', 'Points', and 'data2.Rebounds'.

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

The final merged table clearly and definitively demonstrates the restrictive power of the [inner join](#): observe that only the rows corresponding to players whose **Team** value appeared successfully in both source tables (data1 and data2) are included in this final result set. This selective retention of only matched records is the defining characteristic of this specific join type, ensuring the highest level of integrity and relevance for the combined analytical data. As an optional aesthetic improvement, you may right-click on the header named **data2.Rebounds** within the **Power Query Editor** and rename the column simply to **Rebounds** before applying the final changes to the data model.

Additional Resources for Advanced Data Transformation

Mastering diverse data manipulation and merging techniques is absolutely crucial for any developer pursuing advanced business intelligence development. The [Power Query Editor](#) offers numerous tools beyond simple merging to handle highly complex data scenarios, including creating index columns, pivoting, and unpivoting data structures. We strongly recommend exploring these additional resources and capabilities to further enhance and optimize your overall

data preparation workflow in Power BI.

Here are some related tutorials that explain how to perform other common **data transformation** tasks in Power BI:

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