

# How to Perform an Inner Join in Power BI and Combine Data Effectively

Authored by  
**stats writer**

January 25, 2026

## RECOMMENDED CITATION

stats writer (2026). *How to Perform an Inner Join in Power BI and Combine Data Effectively*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=127629>

An Inner Join is one of the most fundamental and widely used techniques in data integration and analysis, serving as a critical tool for data professionals working with complex datasets. Within the context of Power BI, the Inner Join method is essential for combining information from multiple tables precisely based on common key columns. This join type ensures that only those records possessing matching values in both source tables are included in the resulting output, yielding a focused and highly relevant dataset for analytical purposes.

To effectively execute an Inner Join within Power BI, users rely on the powerful Merge Queries functionality available in the Power Query Editor. This process involves specifying the two tables to be combined and designating the specific column(s) that act as the matching key. For instance, if you maintain separate tables for customer demographic data and transactional sales records, an Inner Join allows you to create a combined view that links each customer to their corresponding sales history. This merged dataset is invaluable for performing detailed analysis, such as modeling customer behavior, calculating lifetime value, and identifying specific purchasing patterns within your sales data.

The following detailed guide will walk through the entire process, demonstrating how to use the Merge Queries feature to perform a perfect Inner Join, utilizing a practical, step-by-step example in the Power Query Editor environment.

## Perform an Inner Join in Power BI (With Example)

### Understanding the Inner Join Concept in Data Analysis

A fundamental principle of relational databases and data analysis is the ability to logically link disparate pieces of information. The Inner Join operation is mathematically defined to return only the rows that have non-null, identical values in the specified join column(s) of both the primary table (left) and the secondary table (right). It acts as an intersection operation, filtering out any records that exist exclusively in one table but not the other.

This specificity is what makes the Inner Join incredibly powerful for cleaning and focusing data models. When dealing with operational data, it is common to encounter data integrity issues or records that simply lack a corresponding match in a related dataset. By using an Inner Join, you are automatically enforcing data quality, ensuring that every record in your resulting table is complete and validated against both sources. This is critical for generating reliable metrics and dashboards in Power BI.

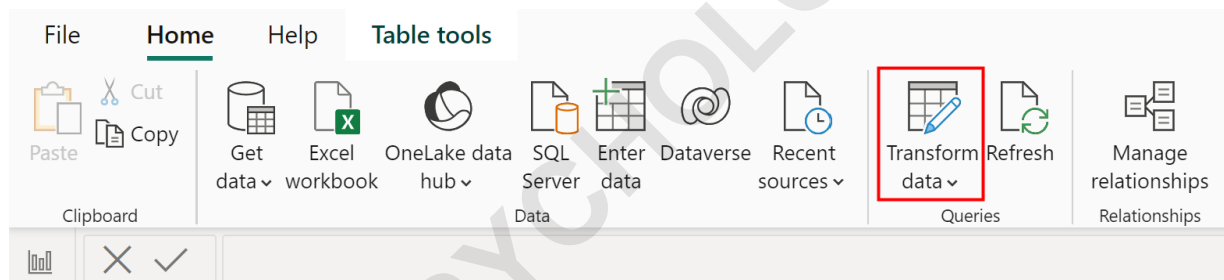
The primary advantage of choosing an Inner Join over other types (like Left Outer or Full Outer) is the resultant reduction in dataset size and the guarantee of referential integrity for the chosen key. For large datasets, reducing the number of rows that Power BI needs to process can significantly

improve performance and report loading times. Furthermore, the simplicity of the intersection logic makes it straightforward to interpret the combined data, as you know every row has a confirmed match.

## Accessing the Power Query Editor for Data Merging

The gateway to performing data transformations and joins in Power BI is the Power Query Editor. This robust, ETL (Extract, Transform, Load) environment provides a graphical interface and extensive M-language capabilities for manipulating and structuring your source data before it is loaded into the final data model. All joining operations, including the Inner Join, are executed within this editor.

To access this editor, you must first load your initial data sources into Power BI Desktop. Once loaded, navigate to the **Home** tab located along the top ribbon of the application. Within this tab, you will find the **Transform data** button, often located within the External Data group. Clicking this button is the necessary first step to launch the dedicated Power Query Editor window.



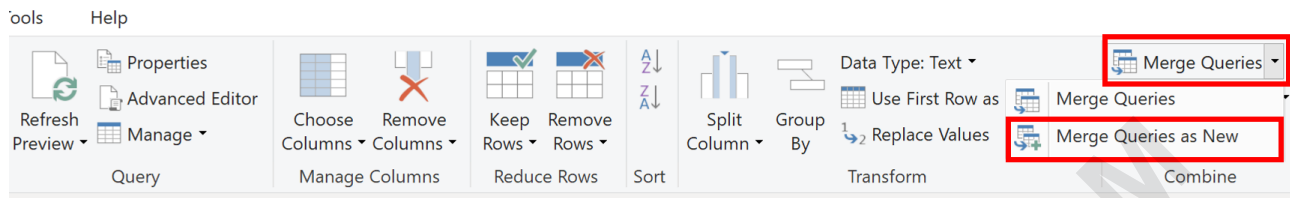
Once inside the Power Query Editor, you will see a list of all loaded tables on the left-hand panel. This environment is where you will define the joining logic using the Merge Queries function, which is specifically designed for combining tables horizontally based on shared columns.

## Step-by-Step Guide: Initiating the Merge Queries Function

The easiest and most common approach to performing an Inner Join between two tables in Power BI involves using the dedicated Merge Queries feature. This function can be accessed directly from the main interface of the Power Query Editor, allowing users to define the join type and matching keys graphically.

Within the Power Query Editor, ensure you are on the **Home** tab. Locate the **Combine** group, which houses the primary functions for data integration. Click the **Merge Queries** icon. At this point, you will be presented with two main options: **Merge Queries** (which modifies the currently selected table) or **Merge Queries as New** (which creates a brand-new table for the merged result).

It is generally recommended to choose **Merge Queries as New** to preserve your original source tables, maintaining a clean transformation history and allowing for easier debugging if issues arise later.

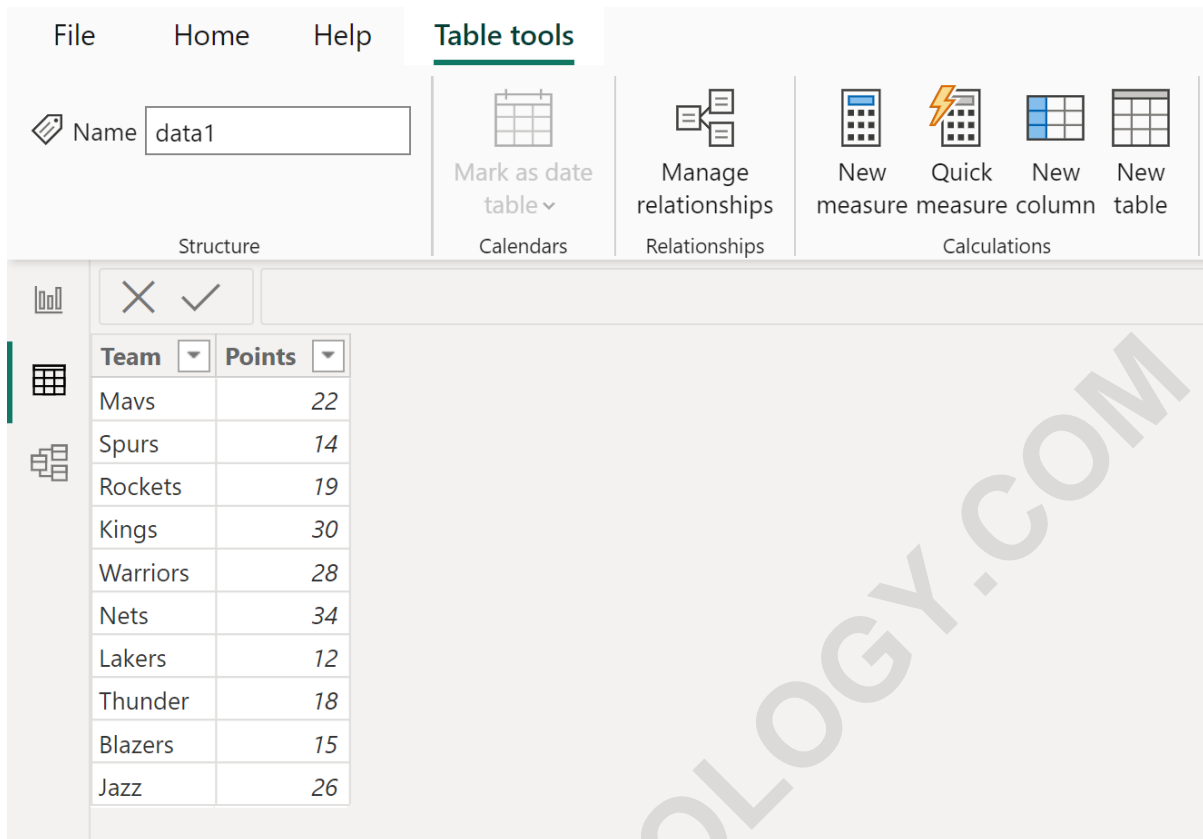


Selecting either option opens the Merge dialog box, which is the crucial interface for setting up the join parameters. This box requires you to define three key elements: the Primary Table (the left side of the join), the Secondary Table (the right side of the join), and the specific columns used for the join condition. This setup ensures a precise application of the intersection logic inherent to the Inner Join type.

### Configuring the Inner Join Parameters (The Example Setup)

To illustrate the practical application of the Merge Queries function, let us consider a common scenario involving sports data. Suppose we have two tables currently loaded into the Power Query Editor.

The first table, named **data1**, contains essential information about basketball players, specifically tracking their team names and the points they have scored over a season.



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, named **data2**, holds complementary data, focusing on the team name and the rebounds achieved by players. Note that the lists of teams and players may not be identical across both tables, which is precisely why the Inner Join is required--to find only the common entries.

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

Our goal is to execute an Inner Join that retains only the rows where the value in the **Team** column is present in both **data1** and **data2**. In the Merge Queries dialog box:

Select **data1** as the first (primary) table.

Select **data2** as the second table to merge.

Crucially, click the **Team** column header in both table previews to establish this column as the join key. (A small numerical indicator will appear next to the header, confirming the selection.)

In the **Join Kind** dropdown menu, select **Inner (only matching rows)**.

## Executing the Inner Join and Viewing the Initial Result

After defining the primary and secondary tables, selecting the common join column (**Team**), and specifying the **Inner** join kind, the Merge Queries dialog box should reflect these choices visually.

The preview section at the bottom of the window provides initial feedback on how many matches were found, which is a useful indicator of the join's success before execution.

## 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

Once you confirm the settings and click **OK**, the [Power Query Editor](#) executes the join operation. The resulting merged table (which will be named **Merge1** by default if you chose "Merge Queries as New") will display all the columns from **data1**, followed by a single column representing the entirety of the merged **data2** table, encapsulated as structured values.

The screenshot shows a DAX formula bar with the following formula: `= Table.NestedJoin(data1, {"Team"}, data2, {"Team"}, "data2", JoinKind.Inner)`. Below the formula bar is a table with three columns: **Team**, **Points**, and **data2**. The table contains seven rows of data:

	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

A key observation here is that the resultant table only includes rows where the **Team** value was present in both source tables. For example, if a team existed only in **data1** but not in **data2**, that entire row is excluded from **Merge1**, fulfilling the strict intersection requirement of the Inner Join.

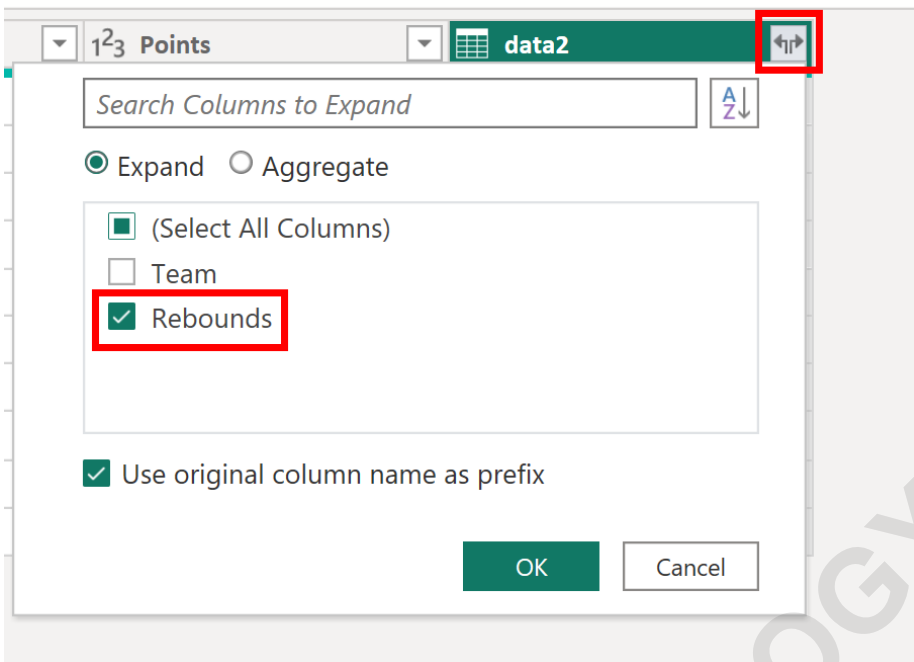
## Expanding and Refining the Merged Table Columns

The current state of the merged table, with the second table aggregated into a single structured column (labeled **data2** in our example), is not yet ready for analysis in Power BI. We need to expand this column to reveal the specific data fields we want to include in our final model, which, in this case, is the **Rebounds** column.

To expand the structured column, click on the expand icon (the two opposing arrows) located at the header of the **data2** column. This action opens a selection menu allowing you to specify which columns from the merged table should be extracted and added to the primary table.

In the expansion menu, carefully check the box next to **Rebounds**. It is important to review the list and deselect any unnecessary columns, especially if the original key column (like **Team**) is already present in **data1**, to avoid redundancy. Also, ensure the option to **Use original column name as prefix** is unchecked if you prefer cleaner column names (though in this example, we keep the prefix initially).

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



Upon clicking **OK**, the structured **data2** column is replaced by the selected fields, resulting in a single, wide table that contains the player's name, points, and their associated rebounds, but only for those players whose teams were present in both original datasets. If the prefix option was kept, the new column will appear as **data2.Rebounds**.

`= 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

### Finalizing and Applying Changes to the Power BI Model

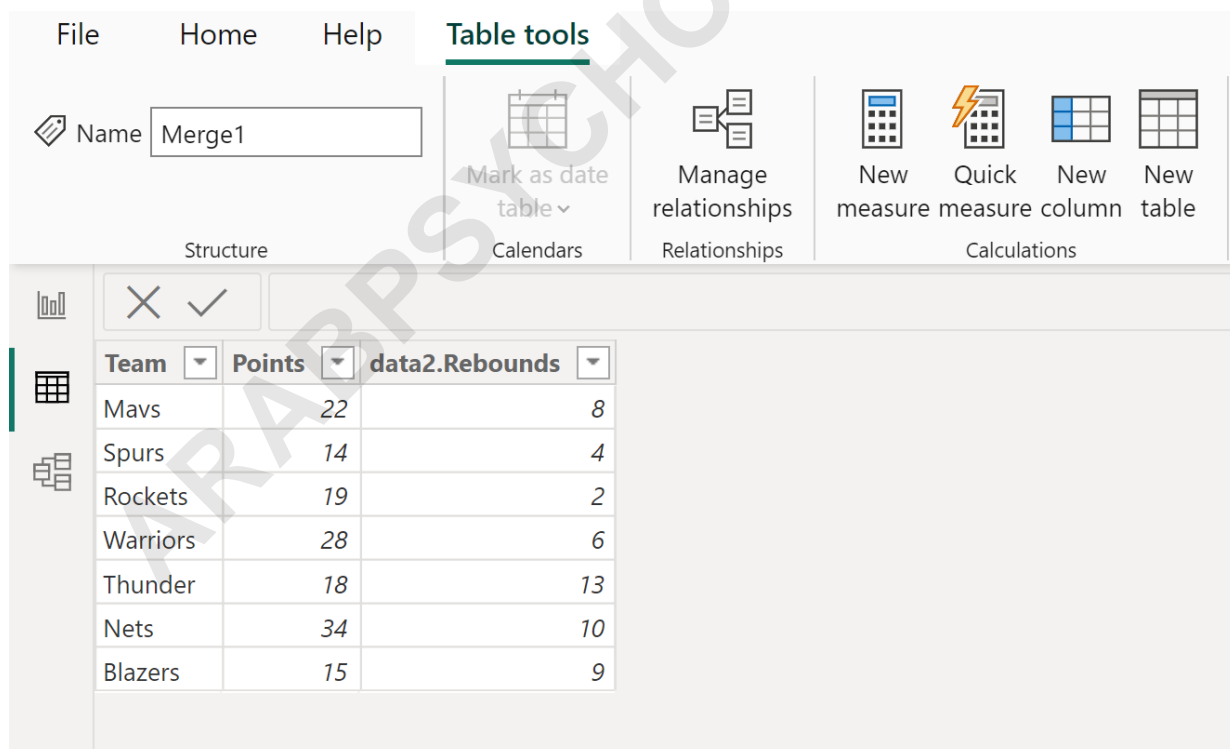
Once the merged table (**Merge1**) is correctly structured and contains only the necessary columns derived from the successful Inner Join, the final step is to apply these transformations and load the

resulting table back into the [Power BI](#) data model environment.

Before leaving the [Power Query Editor](#), you may want to perform basic cleanup. For instance, if the expanded column is named **data2.Rebounds**, you can right-click the header and select **Rename** to simplify it to just **Rebounds**, ensuring clarity for report builders. This is considered a best practice in data modeling.

To finalize the process, click the **Close & Apply** button, located within the **Home** tab of the [Power Query Editor](#). A message box will appear, prompting you to confirm that you wish to load the changes and apply the steps defined in the query history. Click **Yes** to confirm. [Power BI](#) will then process the merged query and load the new table, **Merge1**, into the main data model.

The new table, **Merge1**, is now available in the Table view and Report view of [Power BI](#) Desktop. It serves as a unified source containing data that passed the strict criteria of the [Inner Join](#), ensuring that all records are complete with both Points (from data1) and Rebounds (from data2). Notice that only the rows in which the value from the **Team** column appeared in both tables are kept in the final merged table, confirming the successful execution of the Inner Join operation via the [Merge Queries](#) feature.



The screenshot displays the Power BI interface. At the top, the 'Table tools' ribbon is active, showing options like 'Mark as date table', 'Manage relationships', and 'Calculations'. Below the ribbon, a table is visible 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

## Related Data Transformation Tutorials

Mastering the [Merge Queries](#) function for an [Inner Join](#) is just one step in becoming proficient with

Power BI data preparation. For further skill development in data transformation, consider exploring other common tasks:

Understanding and utilizing different join types (Left Outer, Full Outer, Anti Join).

Creating calculated columns and measures using DAX.

Implementing advanced steps in the Power Query Editor using custom M functions.

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

How to Add Index Column to Table in Power BI

ARABPSYCHOLOGY.COM