

Understanding Inner Joins in Excel: A Practical Tutorial

Authored by
Mohammed looti

November 14, 2025

RECOMMENDED CITATION

Mohammed looti (2025). *Understanding Inner Joins in Excel: A Practical Tutorial*.
PSYCHOLOGICAL STATISTICS. Retrieved from
<https://statistics.arabpsychology.com/?p=1648>

The concept of an [inner join](#) represents a fundamental and precise technique in the field of data management, serving as the essential method for synthesizing two or more disparate datasets based exclusively on shared, related column values. When successfully executed, the resulting output dataset is a meticulous subset containing only those rows where a designated key column possesses perfectly matching values across all tables involved in the merge operation. This technique is immensely valuable for data intersection, as it systematically excludes any data unique to a single table--that is, any row lacking a corresponding match in the partner table--thereby ensuring the final result is perfectly correlated and highly accurate.

While powerful data manipulation methods like this are traditionally associated with dedicated database systems utilizing declarative languages such as [SQL](#) (Structured Query Language), identical functionality can be replicated with remarkable efficiency within [Microsoft Excel](#). This advanced capability is unlocked by harnessing the robust, built-in functionality known as [Power Query](#) (often found under the "Get & Transform Data" heading). This comprehensive, step-by-step tutorial is designed to expertly guide you through leveraging Power Query to execute a flawless inner join operation between two distinct tables in Excel, using a clear and practical real-world data scenario.

Preparing and Structuring Your Source Data

Successful data integration workflows depend entirely on the meticulous preparation of the source material. Before initiating any merge or transformation operations, it is imperative to ensure your raw data is correctly structured, cleanly formatted, and readily accessible. For the purposes of this detailed demonstration, we will begin by populating a standard Excel [worksheet](#) with two distinct, hypothetical datasets. These datasets represent the information we intend to consolidate and filter using the precise method of the inner join.

A crucial preliminary step involves clearly identifying the common link, or "key," that exists between the two datasets. In our scenario, we are working with two tables containing hypothetical team statistics. Both tables must share an identical column, explicitly named **"Team."** This shared column will function as the primary linking key, providing the definitive instruction to the inner join operation regarding precisely which rows belong together. Only rows where the value in the **"Team"** column matches exactly across both source tables will be retained and combined in the final joined output.

	A	B	C	D	E	F	
1	team	points		team	rebounds		
2	Mavericks	22		Mavericks	4		
3	Spurs	29		Hornets	7		
4	Rockets	34		Kings	7		
5	Nets	30		Wizards	12		
6	Hornets	17		Celtics	10		
7	Blazers	15					
8	Kings	14					
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							

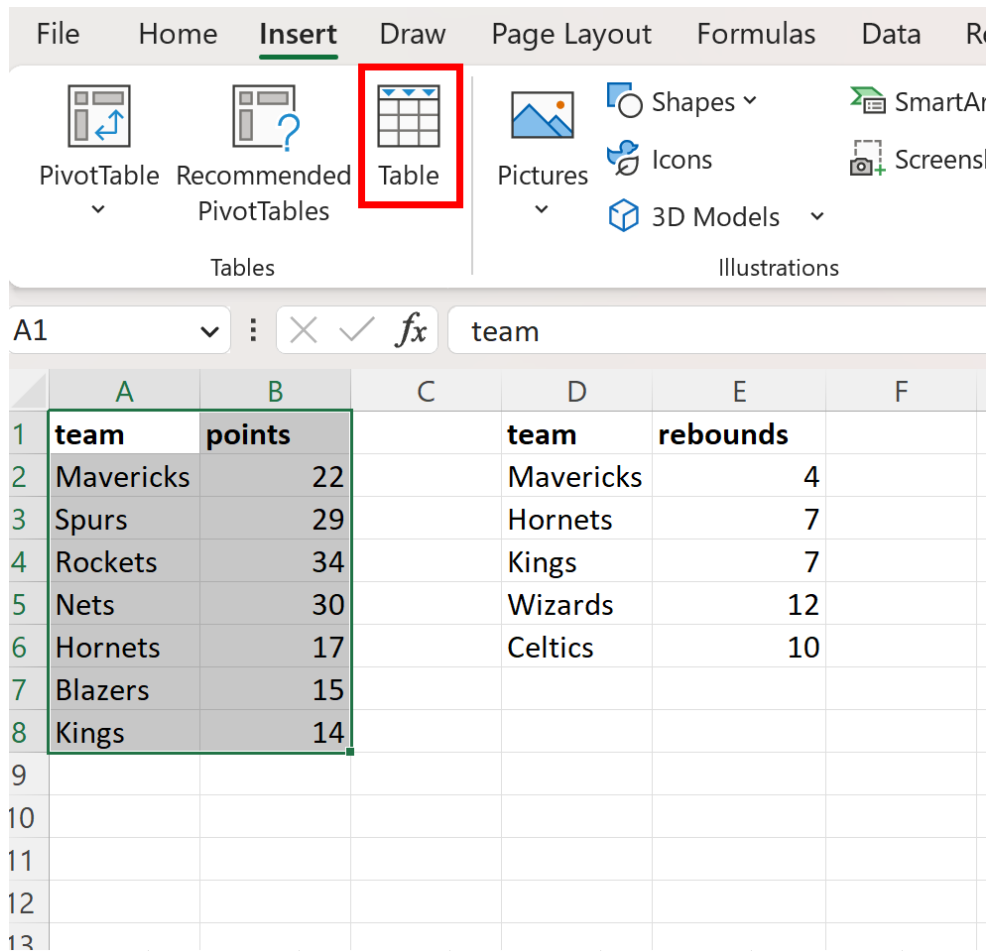
As illustrated above, our objective is to perform an [inner join](#) that accurately intersects these two lists. We anticipate that the resulting combined table will only contain data pertaining to teams that appear both in the first dataset and the second dataset, thereby systematically filtering out any unique entries that lack a counterpart.

Converting Data Ranges to Structured Excel Tables

To correctly leverage the sophisticated capabilities offered by [Power Query](#), it is absolutely mandatory to convert your raw, conventional data ranges into formal, structured [Excel Tables](#). Standard Excel ranges inherently lack the necessary dynamic structure and automatic referencing required for seamless integration within the Power Query Editor environment. Conversely, Excel Tables provide dynamic range names, guarantee robust data structure, and ensure automatic updating, all of which are absolutely essential for constructing reliable and scalable data manipulation workflows.

Initiate this critical transformation process by carefully selecting the entire cell range encompassing your first dataset, which spans cells **A1:B8**. Once the selection is confirmed, navigate immediately to the **Insert** tab located prominently on the Excel ribbon. Within this tab, locate and click the **Table** icon (typically situated within the Tables group). This action will instantaneously open the "Create Table" dialog box. It is vital to ensure that the option labeled "My table has headers" is checked, as

our example data includes clear header rows for both columns. Confirm the action by clicking **OK**.



The screenshot shows the Excel ribbon with the 'Insert' tab selected. The 'Table' button is highlighted with a red rectangle. Below the ribbon, a worksheet is visible with two tables. The first table is in columns A and B, with headers 'team' and 'points'. The second table is in columns D and E, with headers 'team' and 'rebounds'. The data rows for both tables are highlighted with green borders.

	A	B
1	team	points
2	Mavericks	22
3	Spurs	29
4	Rockets	34
5	Nets	30
6	Hornets	17
7	Blazers	15
8	Kings	14

	D	E
	team	rebounds
	Mavericks	4
	Hornets	7
	Kings	7
	Wizards	12
	Celtics	10

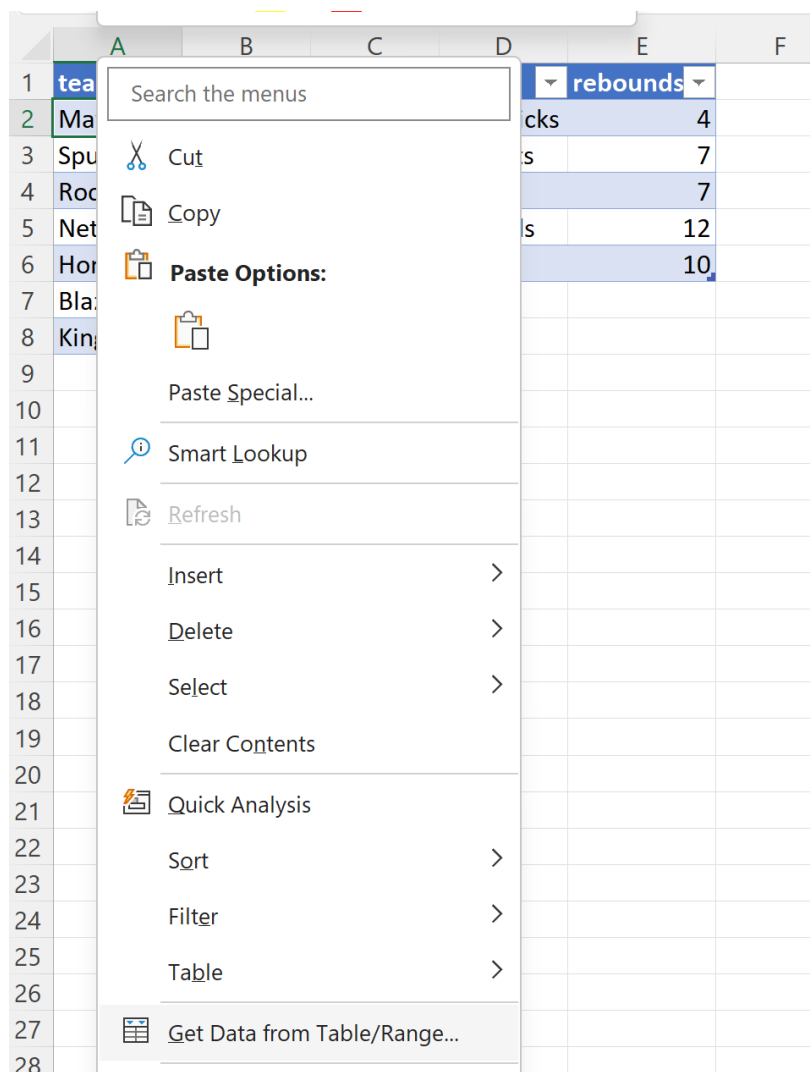
You must then meticulously replicate this entire procedure for your second dataset, which spans the cell range **D1:E6**. After successfully converting both raw data ranges, your Excel sheet will visibly reflect the standardized structured table formatting, often including visual cues like alternating row colors and distinct table design elements. This transformation step is indispensable, as it correctly prepares the data for immediate and reliable ingestion into the Power Query environment.

	A	B	C	D	E	F
1	team ▼	points ▼		team ▼	rebounds ▼	
2	Mavericks	22		Mavericks	4	
3	Spurs	29		Hornets	7	
4	Rockets	34		Kings	7	
5	Nets	30		Wizards	12	
6	Hornets	17		Celtics	10	
7	Blazers	15				
8	Kings	14				
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						

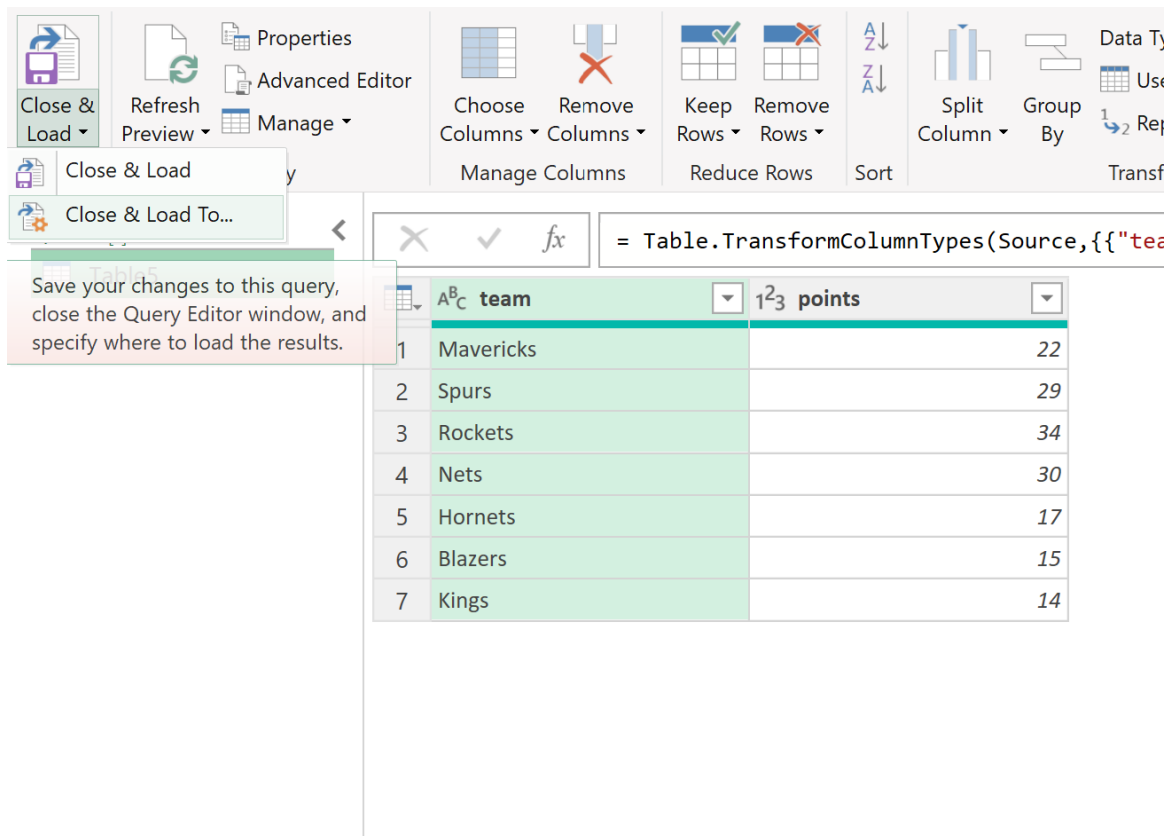
Establishing Power Query Connections for Staging

The subsequent phase involves the critical step of importing these newly created Excel Tables into the [Power Query Editor](#). This editor serves as the dedicated and non-destructive staging area where all data transformation, cleansing, and merging operations will occur, crucially without making any permanent alterations to your original source data.

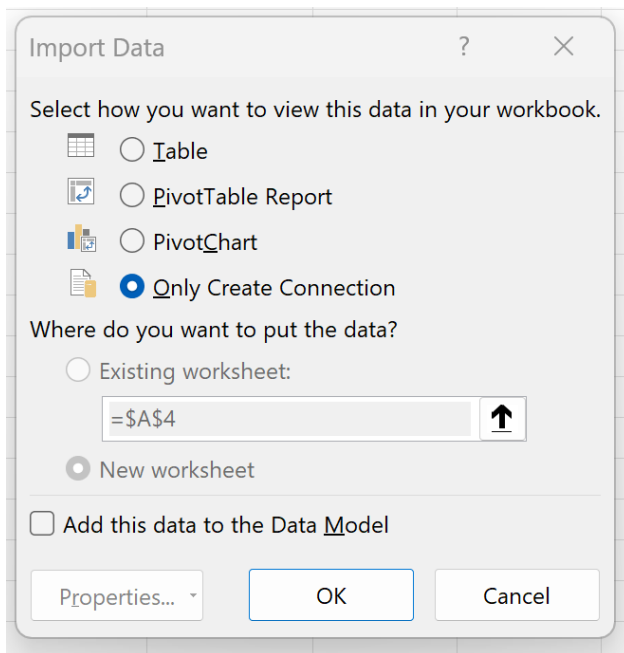
To initiate the connection for the first table (Table1), right-click on any cell located within the table's defined boundaries. A contextual menu will immediately appear; from this menu, select the **Get Data from Table/Range** option. This command instantly launches the Power Query Editor interface, loading your first table as an active query ready for manipulation.



Once the data is clearly visible in the Power Query Editor, we must ensure that it is loaded back to Excel only as a background connection, and not as a visible table output that clutters the sheet. Navigate to the **Home** tab situated within the Power Query Editor ribbon. Click on the **Close & Load** dropdown icon, and then precisely select the **Close & Load To** option.



Within the resulting "Import Data" dialogue box, it is absolutely essential to check the radio button labeled **Only Create Connection**. This critical step ensures that the query remains active in the background, fully prepared for merging operations, without unnecessarily cluttering your Excel [worksheet](#) with intermediate data views. Click **OK** to confirm this setting and close the editor. You must repeat this entire connection process--from right-clicking the table to selecting **Only Create Connection**--for the second table (Table2). Upon completion, both datasets will be available solely as query connections, prominently visible in the "Queries & Connections" pane on the right side of your Excel window.



Executing the Inner Join Operation via Merge Queries

With both source tables successfully staged and loaded as background connections, we are now fully prepared to perform the core [inner join](#) operation. Begin by re-opening the [Power Query Editor](#) interface, which can be done by double-clicking on either Table1 or Table2 within the "Queries & Connections" sidebar. Once inside the editor, navigate directly to the **Home** tab. In the "Combine" section of the ribbon, select the [Merge Queries](#) icon. This action immediately initiates the dialogue box necessary to combine the two prepared datasets.

The "Merge" window requires precise configuration to guarantee a technically accurate inner join. Follow these configuration steps sequentially and carefully:

For the primary table (designated by the top dropdown menu), select **Table1**.

For the secondary table (designated by the bottom dropdown menu), select **Table2**.

Crucially, click the header of the **Team** column in both Table1 and Table2 simultaneously. This action highlights the common key column upon which the entire merge operation will be based.

Under the **Join Kind** dropdown menu, select **Inner (only matching rows)**. This specific selection is essential, as it guarantees that only records present in both tables are included in the final result set, achieving the precise intersection required by an inner join.

After verifying these critical settings, click **OK** to execute the inner join operation and process the data merge.

Merge

Select a table and matching columns to create a merged table.

Table1

team	points
Mavericks	22
Spurs	29
Rockets	34
Nets	30
Hornets	17

Table2

team	rebounds
Mavericks	4
Hornets	7
Kings	7
Wizards	12
Celtics	10

Join Kind

Inner (only matching rows)

☐ Use fuzzy matching to perform the merge

▸ Fuzzy matching options

✓ The selection matches 3 of 7 rows from the first table, and 3 of 5 rows fro...

OK Cancel

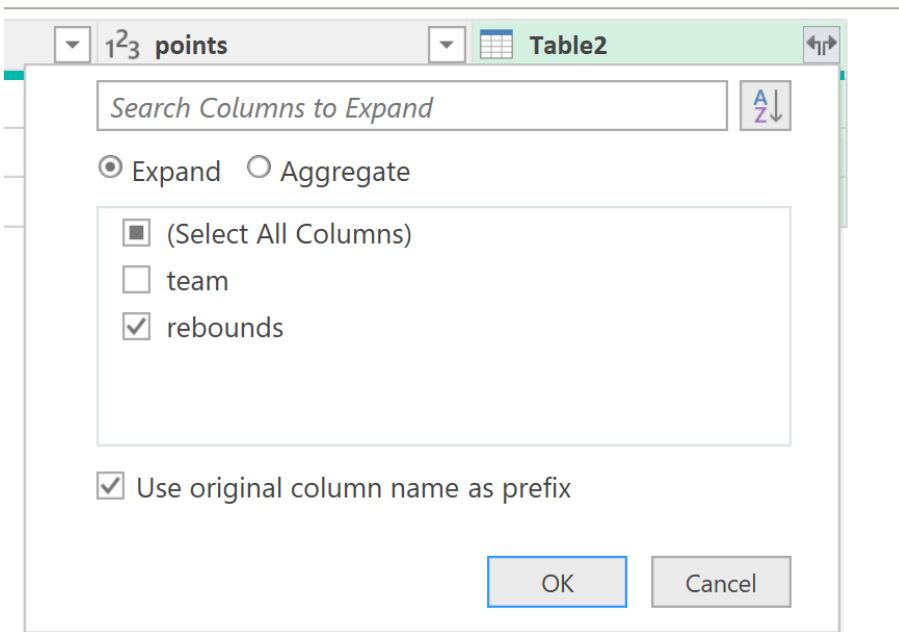
Upon successful execution, the [Power Query](#) Editor will display the merged result set. You will immediately notice the addition of a new column, typically named after the second table ("Table2"), which contains embedded tables. This representation is Power Query's way of confirming that the join has occurred, effectively linking the matching rows from the second table to the corresponding rows in the first.

= Table.NestedJoin("#Changed Type", {"team"}, Table2, {"team"}).			
	team	points	Table2
1	Mavericks	22	Table
2	Hornets	17	Table
3	Kings	14	Table

Expanding and Loading the Final Dataset to Excel

Although the [inner join](#) operation is technically complete, the data sourced from the secondary table remains encapsulated within that nested column. Before this data can be fully utilized and analyzed within the standard Excel environment, it must first be expanded into distinct, usable columns.

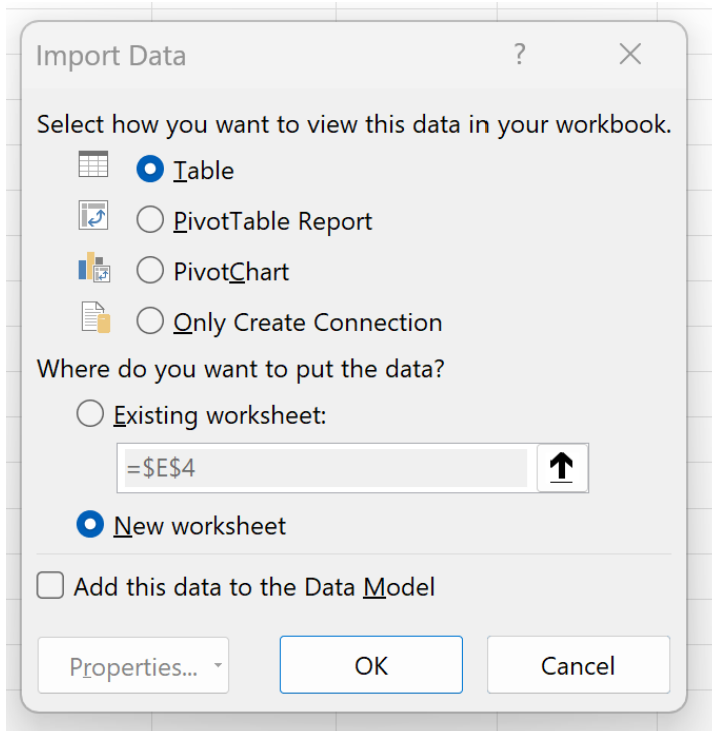
To expand the nested data, locate the column header for **Table2** and click the expansion icon (which visually resembles two opposing arrows) found in the header's top right corner. A subsequent dialogue box will prompt you to select the specific columns from the merged table that you wish to integrate into your final output. In the context of our example, we are specifically interested in adding the "**rebounds**" column. Ensure you deselect all other columns if you only require the "rebounds" data, and confirm your selection by clicking **OK**.



The Power Query Editor will now display the fully merged dataset, with the "**rebounds**" column seamlessly integrated alongside the columns originating from the initial Table1. This finalized view represents the successful consolidation and filtering achieved through the precise criteria of the inner join.

	A ^B _C team	1 ² ₃ points	1 ² ₃ rebounds
1	Mavericks	22	4
2	Hornets	17	7
3	Kings	14	7

The definitive final step is to export this refined, merged dataset back to your working Excel [worksheet](#). Return once more to the **Home** tab in the Power Query Editor and select [Close & Load To](#). In the subsequent "Import Data" window, ensure you select **Table** as the load destination format, and critically, choose **New worksheet** to maintain data cleanliness and prevent accidental modification of your original source tables. Click **OK** to generate the final result.



The resulting table, which is displayed in a brand-new worksheet, showcases the true power of the [inner join](#). It contains only those rows for which the "**Team**" column had a perfect, verified match in both source tables, having successfully performed a precise intersection of the two datasets.

	A	B	C	D	E
1	team	rebounds			
2	Mavericks	4			
3	Hornets	7			
4	Kings	7			
5	Wizards	12			
6	Celtics	10			
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

As this demonstration clearly illustrates, this final, consolidated table provides highly precise information, demonstrating the utility and efficiency of using Power Query to execute complex relational database operations directly within the familiar environment of Excel.

Conclusion and Recommendations for Further Study

Mastering advanced data manipulation techniques in [Excel](#)--specifically through the use of [Power Query](#)--significantly elevates your capability for sophisticated data analysis, robust reporting, and efficient data preparation. The ability to perform an inner join is recognized as a cornerstone of relational data management, enabling users to filter, combine, and clean data with exceptional precision based on defined matching criteria.

We have successfully demonstrated how Excel, when paired with the transformative capabilities of Power Query, can effectively replicate complex database functionality, providing analysts and data professionals with a powerful, accessible, and intuitive toolset. We strongly encourage you to continue developing your proficiency in data transformation and leveraging the full potential of the Power Query Editor.

To further enhance your understanding of common data tasks and more advanced Excel features, consider exploring these related resources and concepts:

Tutorials focusing on performing other essential join types (such as Left Outer, Right Outer, and Full Outer) using the Merge Queries feature in Power Query.

Comprehensive guides for data cleansing, pivoting complex data structures, and unpivoting tables efficiently within the Power Query Editor interface.

Resources detailing the effective and reliable use of structured references within Excel Tables for creating dynamic formulas and automated calculations.