How to Manually Add a Row to a Table in Power BI

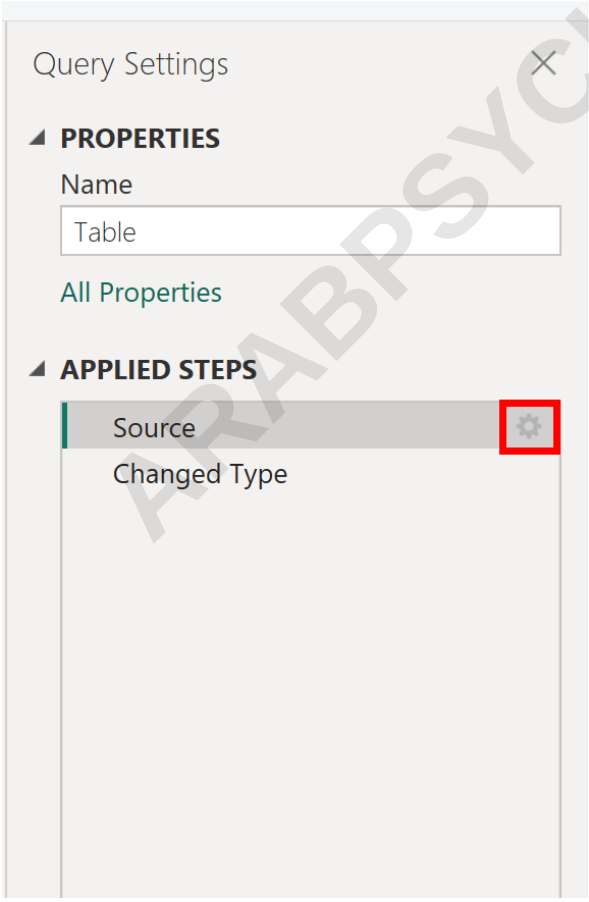
Authored by

mohammed looti

January 10, 2026

A row can be manually added to a table in Power BI by initiating the process through the **"Enter Data"** option, located conveniently in the Home tab. This feature generates a static table within the data model, which is then managed through the Power Query Editor. Unlike typical data import methods that rely on external files, this technique allows users to manually input values for each column directly into a structured grid. For instance, if you maintain a table detailing inventory with columns such as **"Product Name"**, **"SKU"**, and **"Price"**, you can quickly insert a new product entry and its corresponding metrics without the overhead of creating or uploading a separate external data source file. This flexibility is vital for creating small, supplementary datasets or quickly correcting minor data deficiencies.

The most straightforward and sustainable approach to adding a row to a table that was initially created using the **"Enter Data"** functionality in Power BI involves revisiting the original data input screen. This is achieved by clicking the essential **Gear icon** (Settings icon) situated next to the **Source** step, which is meticulously tracked under the **Applied Steps** pane within the dedicated Power Query Editor interface. This action effectively re-opens the data entry dialog box, allowing for direct modification of the raw data contents, including the addition of new rows or columns. This iterative process ensures that the underlying structure remains clean and manageable, while immediately reflecting the changes in the data model.



The following detailed example provides an illustrative, practical guide on how to implement this solution efficiently, ensuring that every step, from initial table creation to final row addition, is clearly understood and executed.

Introduction to Manual Data Entry in Power BI

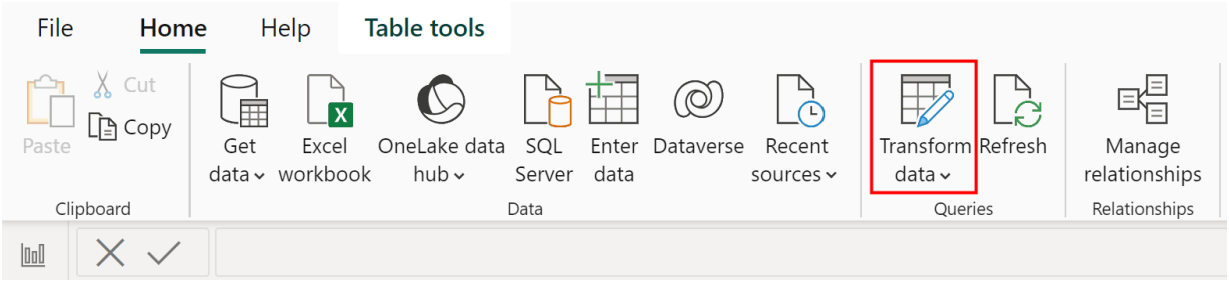
Although Power BI is engineered primarily for connecting to large, dynamic, and external data storage systems--such as SQL databases, cloud services, and flat files--the necessity for manual data entry often arises in specific business intelligence scenarios. These scenarios typically involve creating parameter tables, defining custom hierarchies, or establishing simple translation mappings that do not justify the creation of an entirely new data source. The **"Enter Data"** feature is Microsoft's built-in solution for injecting small volumes of static data directly into the Power BI data model, treating the input as an immediate, self-contained table.

Using this manual input method provides a unique advantage: it simplifies the prototyping phase of report development. Report developers can quickly mock up data structures or add necessary supplementary information, such as budget targets or manual exceptions, without waiting for IT intervention or external data preparation. However, it is paramount to understand that data entered this way is static; unlike connected tables, these manual entries must be updated directly within the Power Query Editor interface whenever changes are required, which is the precise mechanism we will explore for adding subsequent rows.

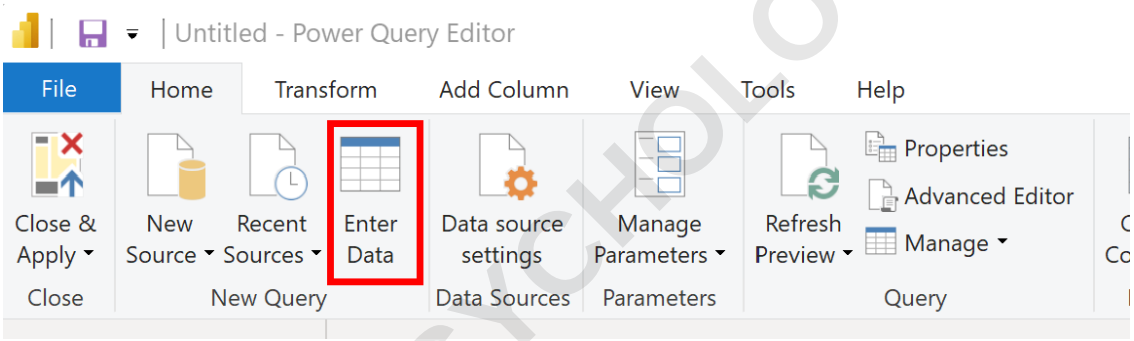
The fundamental constraint of the **"Enter Data"** approach is that the data must be edited by modifying the underlying source configuration, rather than through standard table editing functions found elsewhere in Power BI Desktop. Once the initial table is created, it transforms into an M Language script that is stored within the query definition. To append new rows, we must interact with the user interface that generated that initial script, guaranteeing data integrity and consistency with the data model's structure. We recommend using this approach only for small, highly stable datasets that require infrequent updates.

Initial Table Creation Workflow in Power BI Desktop

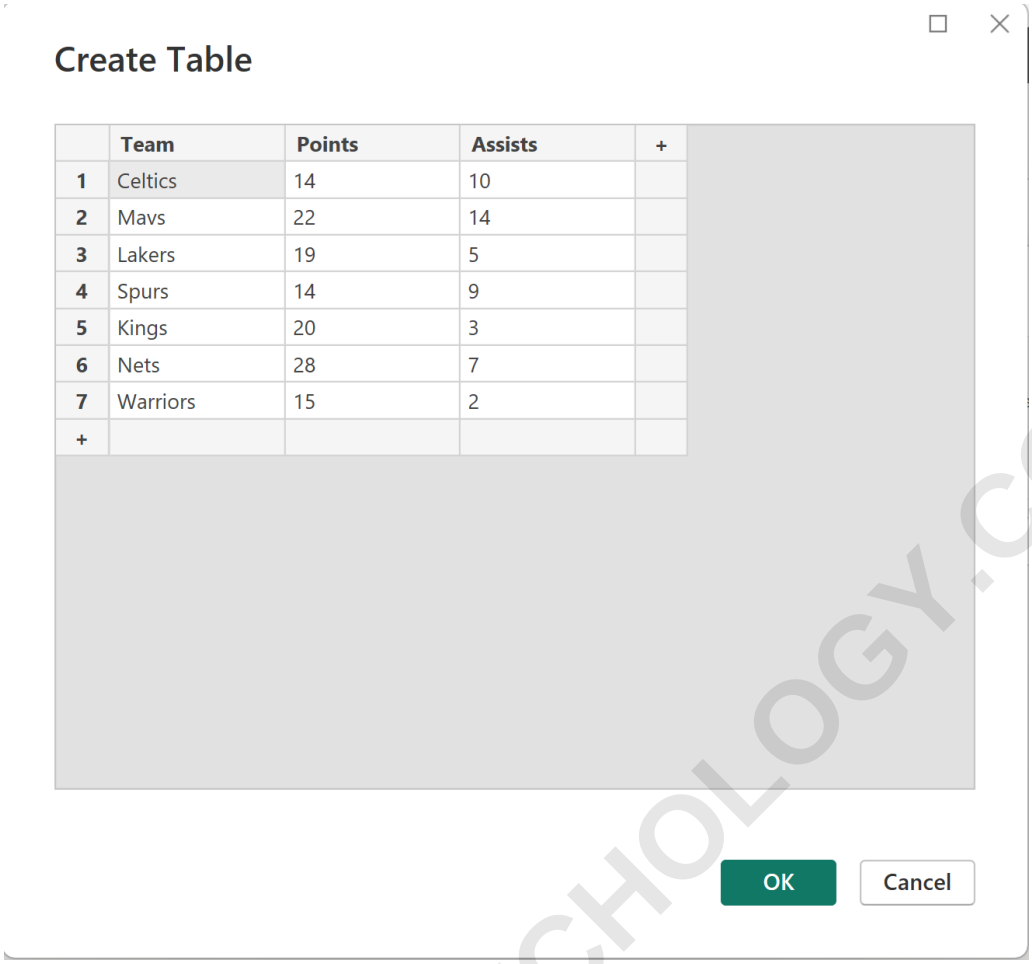
Our goal is to manually create a foundational table in Power BI that will later require manual row additions. This process begins by accessing the data transformation environment. To initiate this process, first click the **Home** tab located along the top ribbon of the Power BI Desktop application. Next, look for and click the **Transform data** icon. Clicking this icon serves as the gateway, launching the highly functional and essential Power Query Editor, which is the environment where all subsequent data cleaning, shaping, and manual input adjustments will take place. This separation of tasks ensures that data preparation is isolated from the report visualization phase.



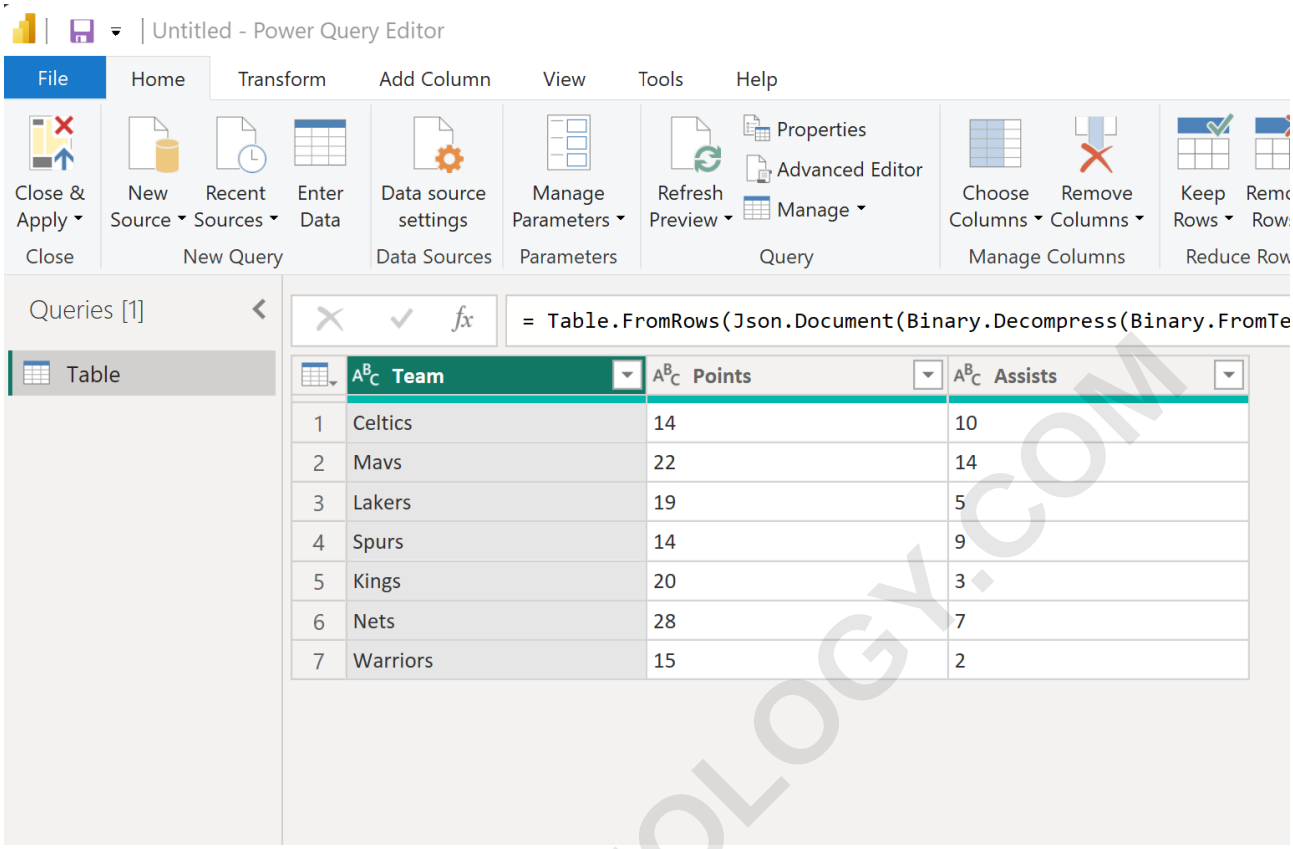
Once the Power Query Editor window is active, the next step is to invoke the manual data input tool. Within the Editor's interface, locate the **New Query** group on the **Home** tab. Click the **Enter Data** icon within this group. This action will immediately open a small dialog box designed for table creation, often referred to as a "Create Table" window, which allows for the manual definition of columns and rows. This interface resembles a simplified spreadsheet, providing a straightforward mechanism for structuring the immediate data requirements.



For the purpose of our demonstration, we will enter specific data relating to various basketball players. We define columns such as Player Name, Team, and Points Scored, populating several initial rows. This setup forms the basis of our manually created table. After inputting all the necessary initial information and ensuring the column headers are correctly named, we finalize the creation by clicking **OK**. This action commits the data and formally creates the table within the Power Query Editor, applying the necessary data type transformations automatically based on the input values.



Upon clicking **OK**, the system generates the new table, which then appears in the Queries pane on the left side of the Power Query Editor interface. The data is immediately displayed in the central data preview window. At this point, the initial structure is complete, and the query history, detailed on the right under **Applied Steps**, will clearly show the first step: **Source**. This initial table, now successfully created and loaded into the editor, serves as the starting point from which we will demonstrate the row addition process.



The Role of Power Query Editor in Data Manipulation

The Power Query Editor is the foundational ETL (Extract, Transform, Load) tool within the Power BI ecosystem, providing a powerful graphical user interface for data shaping. When a table is created using the **"Enter Data"** feature, Power Query translates the manually entered grid into executable code written in the M Language, officially known as the Power Query Formula Language. This M Language script, specifically the function `Table.FromRows` or similar constructs, defines the table's structure and contents. Understanding that the data is stored as code, rather than in a traditional database file, is key to grasping how updates are managed.

Central to the Power Query workflow is the **Applied Steps** pane, which meticulously records every data transformation applied to a query. This pane provides transparency and reproducibility, allowing users to review, reorder, or delete transformations. When we use **"Enter Data"**, the very first step recorded is always the **Source** step. This crucial step contains the M Language code that holds the literal values typed into the initial creation dialog. All subsequent steps, such as changing data types or filtering rows, build upon this initial **Source** step. To add a new row, we must logically return to and modify the contents of this **Source** step.

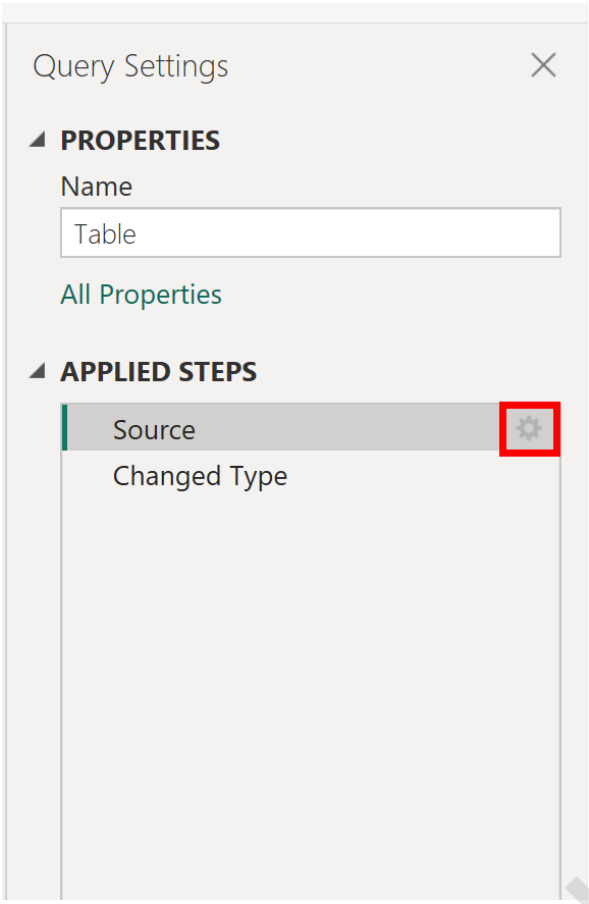
This approach highlights a significant difference between manually created tables and linked external data sources. For external sources, adding a row means updating the source file (e.g.,

adding a row to the Excel sheet). For tables created via **"Enter Data"**, adding a row requires editing the internal M code configuration. Therefore, the task is not to append a row using a transformation step, but rather to redefine the initial static source data set, ensuring the entire table definition is correctly updated to include the new entry. This methodology maintains the integrity of the transformation pipeline defined in the Applied Steps.

Modifying the Source: The Key to Adding New Rows

Now that our initial table containing basketball player data is established, the scenario dictates that we must add new rows to this existing, manually defined table. This is where the power and precision of the Power Query Editor become essential. The method for appending data is not through inserting a new row in the preview pane, but by accessing the underlying definition of the data set. To achieve this, we must navigate to the **Applied Steps** pane on the right side of the Editor and locate the very first step, labeled **Source**.

Next to the **Source** step, there is a small configuration icon--the **Gear icon**. Clicking this icon is the critical action that triggers the re-opening of the original "Create Table" or "Enter Data" dialog box. This mechanism is specifically designed to allow users to modify the parameters of the query's initial data input. By clicking this gear, we effectively instruct Power BI to retrieve and display the textual representation of the data that was saved by the system when the query was first executed, allowing for direct editing of the static data set.



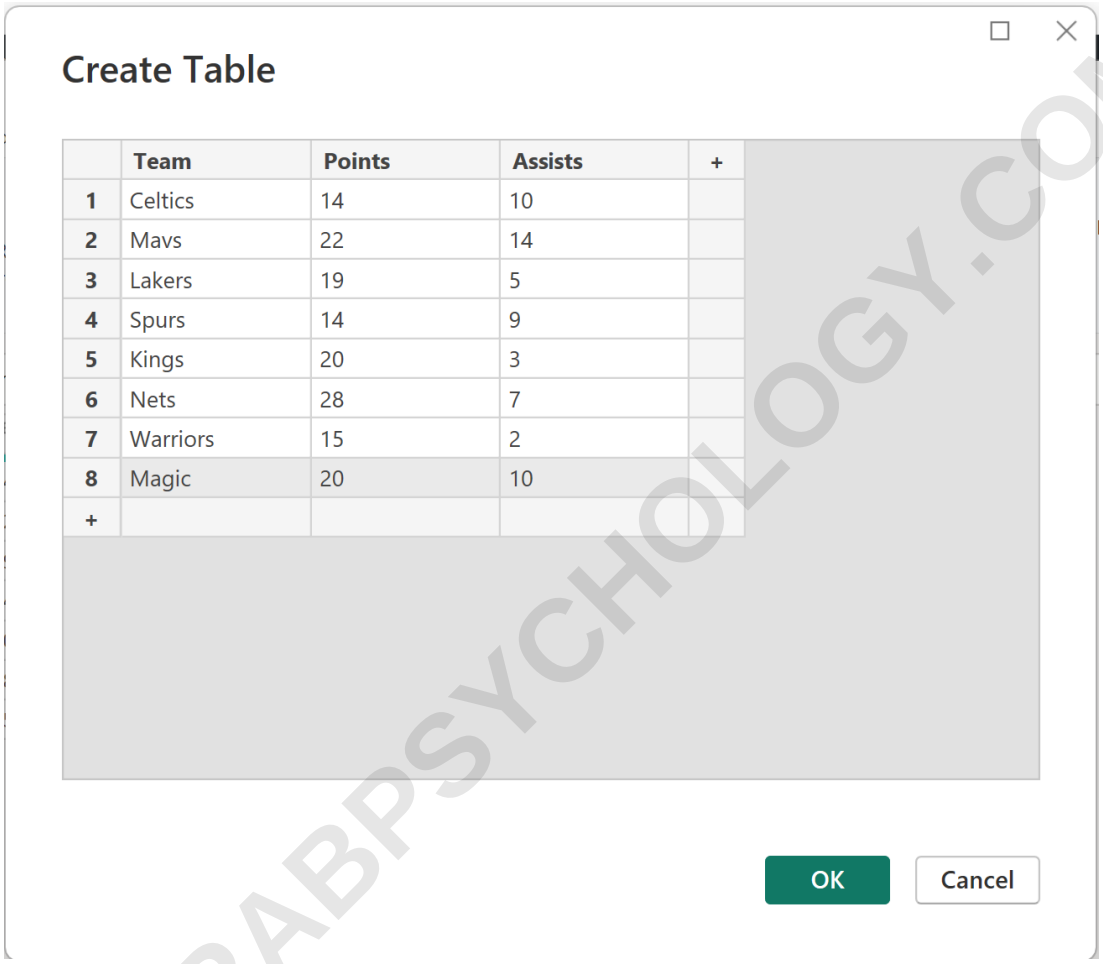
Re-accessing the data entry dialog is the only valid way to add rows to a table created using the **"Enter Data"** feature. If you were to attempt to add rows using subsequent transformation steps (like merging tables or appending queries), you would be creating complex, unnecessary M Language code that hinders maintainability. By modifying the **Source**, you ensure that the table remains a single, cleanly defined static data set. Once the dialog box re-opens, the process of data entry is identical to the initial setup, providing a familiar and intuitive environment for inserting the required new information.

Step-by-Step Guide: Inserting Additional Rows

We are now positioned within the re-opened **"Enter Data"** dialog box, which currently displays all the previously entered player data. Our task is to append information for a new basketball player. This interface functions exactly as it did during the table's creation, allowing us to interact with the grid format. We simply navigate to the bottom of the existing rows and begin typing the data for the new player, ensuring that the values align correctly with the existing column headers (Player Name, Team, and Points Scored).

For this specific example, suppose we decide to incorporate one new row containing critical

information about a player belonging to the Magic team. We would type the player's name, assign them to the "Magic" team, and input their corresponding points total. The dialog dynamically expands to accommodate this new data entry. It is crucial at this stage to double-check that the data types implied by the new values align with the data types established by the initial rows, although Power BI is generally adept at handling minor inconsistencies until the transformation steps are applied.

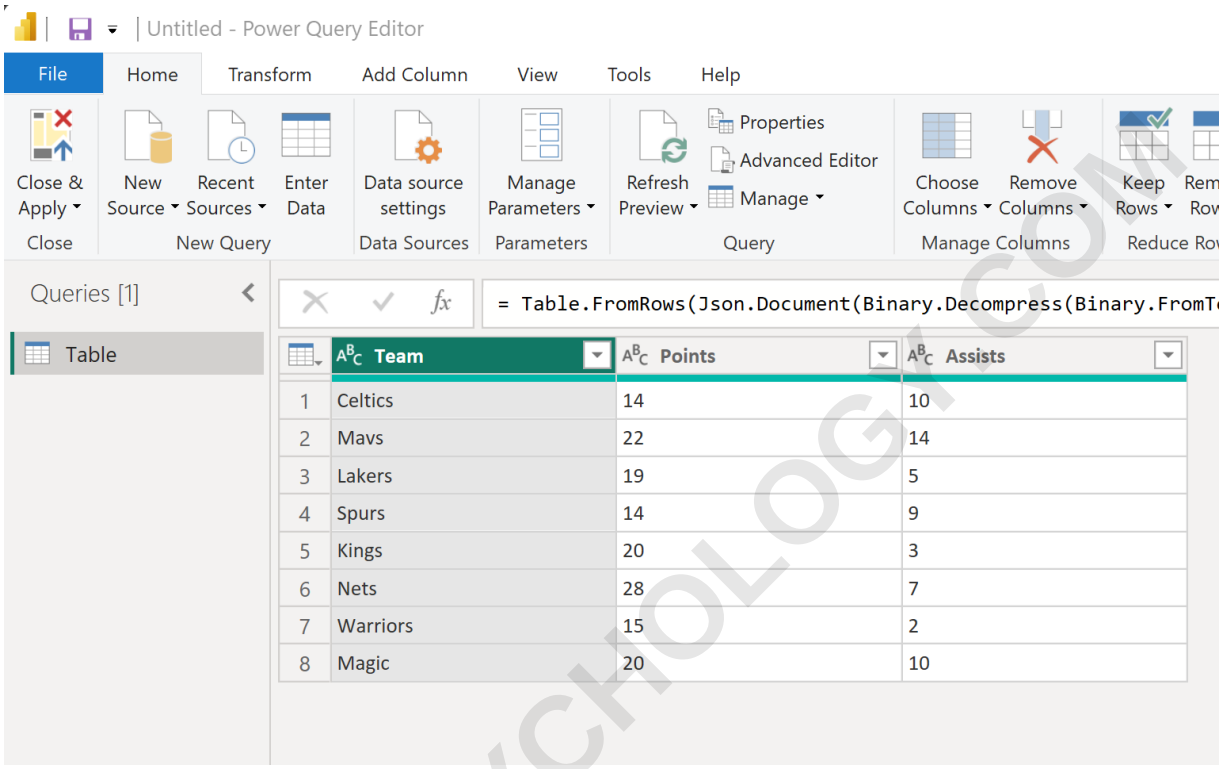


After successfully inputting the new row (or multiple rows, as desired), the process is finalized by clicking **OK** within the dialog box. This action triggers the Power Query Editor to update the underlying M Language script of the **Source** step, embedding the new row directly into the table definition. Since all subsequent transformation steps (e.g., Change Type) in the Applied Steps pane depend on the Source, they will automatically re-execute against the newly expanded data set, guaranteeing that the new row is correctly processed and integrated.

Reviewing the Results and Finalizing Changes

Once **OK** is clicked, the Power Query Editor reloads the query. The central data preview pane will

now display the entire table, including the newly added row. You should observe the data for the Magic player seamlessly integrated with the original data. This confirmation in the preview window is essential, as it validates that the modification to the **Source** step was executed correctly and that the downstream transformation steps did not encounter errors due to the newly introduced data.



With the new data visible and validated in the Power Query environment, the final operational step is to load the modified data set back into the main Power BI data model. This is achieved by clicking the **Close & Apply** button, typically found in the top-left corner of the Power Query Editor Home tab. This crucial step executes all the queries and loads the resulting data tables--including our manually updated table--into the underlying tabular model, making the new row immediately available for use in visualizations, measures, and relationships within the Power BI report.

This method provides a robust and repeatable way to maintain small, manually defined tables within a Power BI project. Users should feel comfortable adding as many new rows as necessary by repeating the process: navigating to the **Applied Steps**, clicking the **Gear icon** next to **Source**, inputting the data, and clicking **OK**. This guarantees that the table remains current and accurate for report generation.

Limitations and Best Practices for Manual Data

While the **"Enter Data"** feature offers exceptional flexibility for supplementary data, it is crucial to recognize its limitations. The primary constraint is scalability; this method is designed exclusively

for small, static datasets. If a table grows beyond a few dozen rows, or if the data requires frequent or automated updates, managing it manually through the Applied Steps configuration becomes cumbersome and highly inefficient. For large or volatile datasets, best practices dictate connecting to an external, automatically refreshed data source such as a CSV file, SharePoint list, or database table.

A significant best practice involves documentation. Since manual data entry is less transparent than a clearly linked source file, report documentation should explicitly note which tables were created using the **"Enter Data"** feature. Furthermore, when adding rows, ensure consistency in data entry. Although Power Query can handle data type conversions, entering inconsistent values (e.g., text in a numeric column) can lead to errors in subsequent transformation steps or within the final report, requiring complex debugging of the underlying M Language code.

Finally, avoid creating multiple query steps just to append data to a manually entered table. For example, creating a second **"Enter Data"** query and then using the **Append Queries** feature is overly complicated for simple additions. The core solution--modifying the original **Source** step via the **Gear icon**--remains the cleanest, most direct, and most maintainable method for managing static, manually generated tables in Power BI Desktop.

Additional Power BI Tutorials

The following tutorials explain how to perform other common tasks in Power BI, leveraging the same powerful capabilities found within the Power Query environment:

How to Add Index Column to Table in Power BI