

How to Remove Blank Rows in Power BI: A Step-by-Step Guide

Authored by
stats writer

January 29, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Remove Blank Rows in Power BI: A Step-by-Step Guide*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=128387>

Power BI is widely recognized as a cornerstone platform for sophisticated **data analysis** and visualization, empowering organizations to translate complex datasets into actionable insights through compelling reports and dynamic dashboards. Despite its robust capabilities, analysts frequently encounter challenges stemming from data quality issues, most notably the presence of unwanted blank or null rows. These seemingly minor inconsistencies can significantly compromise the accuracy and reliability of analytical outcomes, leading to faulty conclusions and misinformed business decisions. To uphold the integrity of the resulting reports, it is imperative to employ effective data cleansing methods within the Power BI environment to systematically eliminate these problematic entries.

The initial step in rectifying this issue involves precisely identifying where these blank rows reside within the dataset. While superficial methods such as manual sorting or applying visual filters in the report view might reveal some discrepancies, a more comprehensive approach is typically required when dealing with large volumes of data. Traditionally, users might attempt quick fixes like right-clicking a detected blank row and selecting a "Delete" option, or highlighting an entire row via its number and pressing the Delete key. However, these manual interactions are inefficient, non-reproducible, and generally unsuitable for enterprise-level **data model** preparation.

For a definitive and scalable solution, Power BI provides powerful built-in functionalities accessible through the **Power Query Editor**. This environment is specifically designed for data transformation tasks, offering specialized tools like the "Remove Rows" feature under the "Transform" tab. This functionality enables users to execute bulk operations, such as deleting all blank rows simultaneously, across an entire dataset or restricting the operation to specific columns where blank values are concentrated. Utilizing these advanced features ensures that the foundational data feeding the reports maintains the highest standards of quality and consistency.

In summary, while navigating **Power BI**, achieving pristine data quality is non-negotiable for trustworthy analysis. By mastering the application of the integrated delete and bulk removal options available within the transformation workflow, particularly within the **Power Query Editor**, users can efficiently purge disruptive blank rows. This strategic data cleansing process is fundamental to ensuring the **data integrity** and analytical validity of all subsequent reports and visualizations produced by the platform.

Comprehensive Guide to Removing Blank Rows in Power BI (With Practical Example)

The Definitive Approach: Utilizing the Power Query Editor

The most robust, efficient, and recommended methodology for systematically eliminating blank

rows from any table imported into Power BI involves harnessing the capabilities of the **Power Query Editor**. This specialized environment, often referred to as the ETL (Extract, Transform, Load) engine of Power BI, provides dedicated tools to reshape and refine data before it is loaded into the final **data model**. Specifically, the built-in **Remove Blank Rows** feature automates the tedious process of cleaning datasets that have been poorly sourced or improperly extracted.

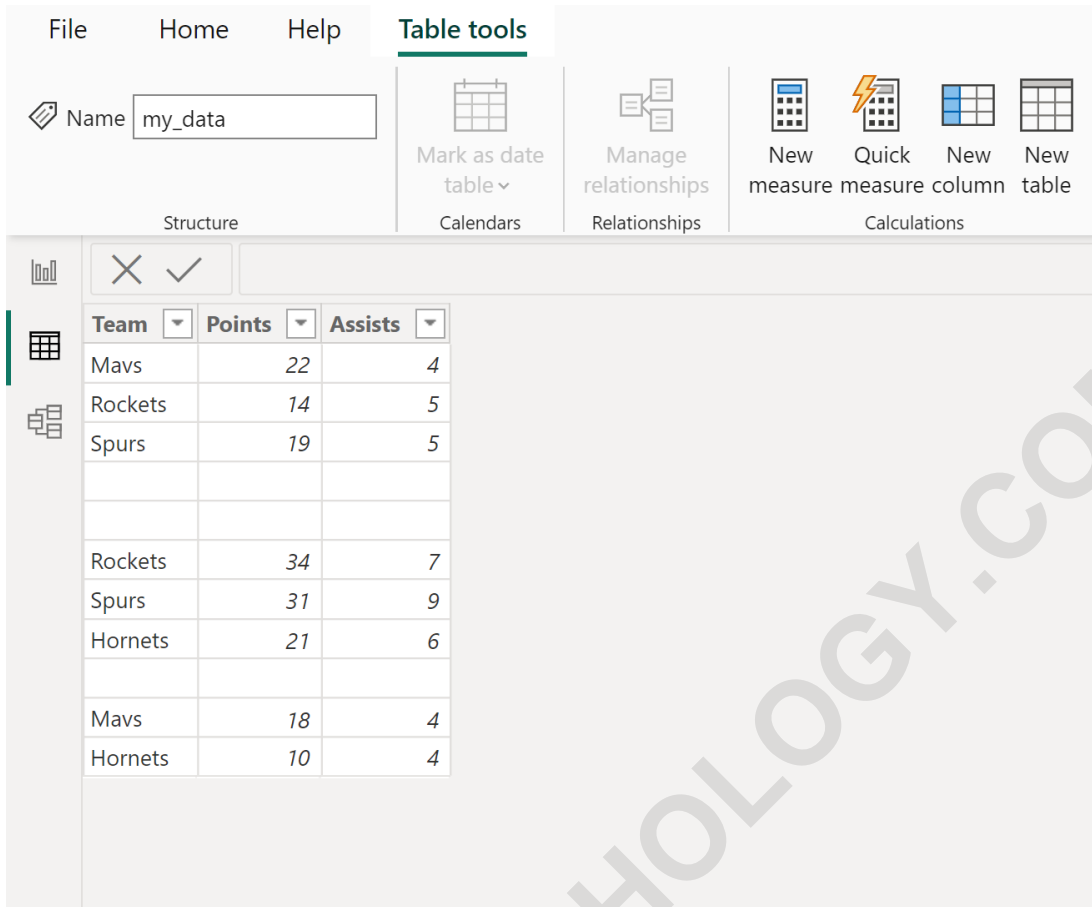
Relying on the **Power Query Editor** ensures that the data cleansing steps are permanently recorded as part of the query definition. This means that every time the data source is refreshed, these transformation steps—including the removal of blank rows—are automatically reapplied. This adherence to best practices guarantees consistency and saves significant time compared to manual deletion within the main Power BI Desktop interface. Furthermore, this method handles true blank rows, where all column values are null, ensuring a thorough cleanup.

The following detailed guide provides a practical, step-by-step demonstration of how to implement this powerful feature, ensuring that your data tables are immaculate and ready for accurate analysis within **Power BI** Desktop. This example illustrates the scenario where a dataset contains multiple rows entirely composed of null values that need to be universally excised.

Practical Example: Identifying Data Quality Issues

For the purpose of this illustration, imagine we have imported a dataset concerning basketball players into Power BI. Upon initial inspection, the table structure looks sound, but closer examination reveals a crucial data quality issue: the presence of multiple rows where entries for player Name, Team, and Points scored are completely missing or contain null values. These rows are essentially placeholders or errors introduced during the data extraction or merging process and must be removed to avoid skewing summary statistics and aggregation functions.

The table below visually represents this raw, uncleaned state of the data. Notice the distinct rows that contain exclusively blank entries across all defined columns. If these rows were to remain, any calculation involving row counts or averages would be inaccurate, undermining the fundamental **data integrity** of the final report.

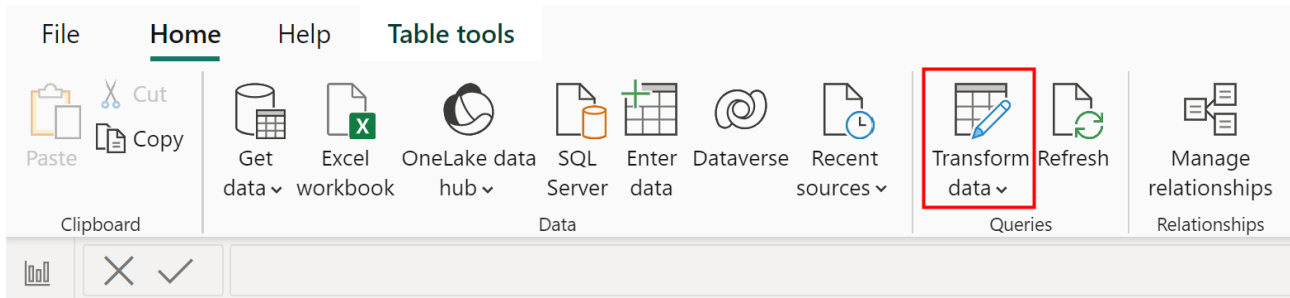


It is crucial to understand that simply hiding these blank rows in the visualization layer is insufficient. The underlying **data model** still retains these extraneous entries, potentially leading to errors in complex DAX calculations. Therefore, the removal process must occur at the transformation stage before the data is formally loaded into the desktop environment.

Accessing the Power Query Transformation Environment

To initiate the cleansing process, we must first access the dedicated environment for data manipulation, the **Power Query Editor**. This is the gateway to all fundamental data shaping operations within **Power BI** Desktop. Regardless of whether your data source is a simple Excel file or a complex SQL database, the steps to reach this editor remain consistent and straightforward.

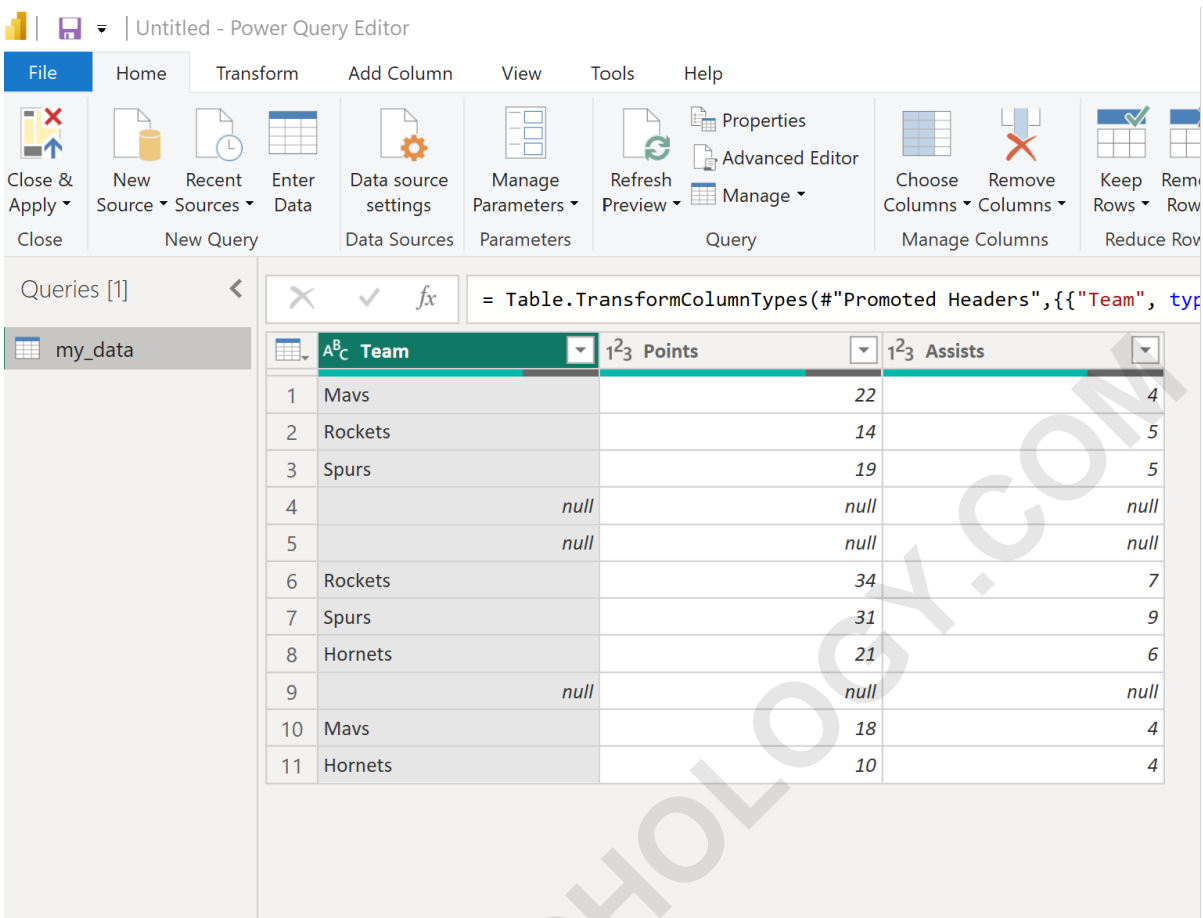
From the main Power BI Desktop interface, navigate to the **Home** tab located along the top ribbon menu. Within this section, locate and click the **Transform data** icon. This action triggers the opening of a new, separate window--the **Power Query Editor**--where all subsequent data preparation tasks will be executed. This step is essential because many advanced data cleaning techniques, including the bulk removal of blank rows, are exclusively available within this specialized interface.



Upon successful execution, the new **Power Query Editor** window will appear, presenting your imported table ready for modification. This interface provides a comprehensive view of your data, along with a list of applied steps on the right-hand pane, offering transparency and control over every transformation performed on the dataset.

Executing the Removal of Blank Rows

Once inside the **Power Query Editor**, the process for removing the unwanted blank rows is remarkably simple and involves utilizing a single, highly effective command designed specifically for this purpose. Ensure that the target table (in this case, our basketball player data) is currently selected in the Queries pane on the left side of the screen, confirming that the transformations are applied to the correct dataset.



With the correct table active, turn your attention to the **Home** tab within the Power Query Editor ribbon. Locate the **Remove Rows** group, which typically contains options for removing top, bottom, alternate, or, crucially, blank rows. Click the main **Remove Rows** icon to reveal the drop-down menu of available row deletion actions. From this menu, select the option labeled **Remove Blank Rows**.

The screenshot shows the Power BI Power Query Editor interface. The 'Remove Rows' dropdown menu is open, and 'Remove Blank Rows' is selected. A tooltip indicates: 'Remove all blank rows from this table.' The data table below shows the following rows:

	Team	Points	
1	Mavs	22	
2	Rockets	14	
3	Spurs		
4		null	null
5		null	null
6	Rockets	34	7
7	Spurs	31	9
8	Hornets	21	6
9		null	null
10	Mavs	18	4
11	Hornets	10	4

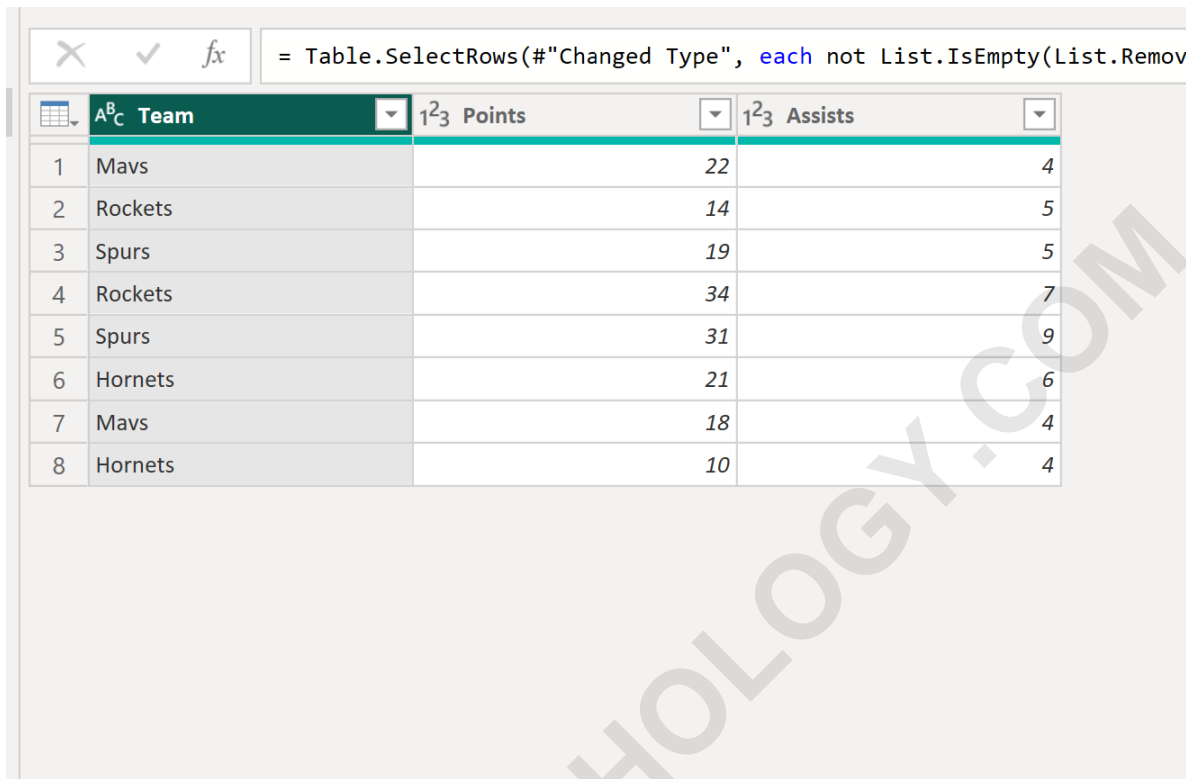
This single click triggers an automated process where the Power Query engine scans the entire table and identifies any row where all non-index columns contain null or empty string values. The transformation is executed immediately, and the resulting, cleaned table is displayed directly within the editor interface, ready for the next stage of preparation or loading. Notice how the applied step, "Removed Blank Rows," is logged in the "Applied Steps" list, guaranteeing reproducibility upon refresh.

Verification and Result Confirmation

Following the execution of the **Remove Blank Rows** step, the table view instantly updates to reflect the cleaned dataset. All rows that previously contained nulls across every column have now been successfully purged from the query. This immediate visual feedback allows for quick verification that the operation achieved its intended goal--a table containing only meaningful, non-empty records.

The resultant table confirms the efficacy of the transformation, now presenting only legitimate records of basketball players, free from the disruptive influence of empty entries. The table below illustrates the final, clean state of the data, which is now optimized for accurate **data analysis**

within **Power BI** Desktop. This rigorous cleansing ensures that all aggregate functions, filtering, and visual representations derived from this data source will be based solely on valid data points.



The screenshot shows the Power BI Desktop interface. At the top, the formula bar contains the DAX formula: `= Table.SelectRows(#"Changed Type", each not List.IsEmpty(List.Remov`. Below the formula bar is a data table with three columns: 'Team', 'Points', and 'Assists'. The table contains 8 rows of data.

	Team	Points	Assists
1	Mavs	22	4
2	Rockets	14	5
3	Spurs	19	5
4	Rockets	34	7
5	Spurs	31	9
6	Hornets	21	6
7	Mavs	18	4
8	Hornets	10	4

It is important to differentiate this global row removal from column-specific null handling. If only a few columns in a row contain nulls while others are populated, the "Remove Blank Rows" feature will typically ignore that row. If the requirement is to remove rows based on nulls in a specific column (e.g., removing any row where 'Player Name' is blank), alternative filtering methods within the **Power Query Editor** must be employed, which provides greater granularity and control.

Applying and Finalizing Changes to the Data Model

The transformations performed within the **Power Query Editor** are temporary until they are explicitly committed and loaded back into the **data model** of Power BI Desktop. This crucial final step ensures that the cleansed data is what is ultimately used for reporting and visualization purposes. To finalize the process, navigate back to the **Home** tab within the Power Query Editor.

In the upper left corner of the ribbon, you will find the **Close & Apply** option. Clicking this button initiates two critical actions: first, the **Power Query Editor** window is closed, and second, the engine executes all defined transformation steps (including "Removed Blank Rows") and loads the resulting, cleaned data structure into Power BI Desktop. This action effectively updates the underlying data source used by all visuals and measures.

If you attempt to close the editor without explicitly clicking **Close & Apply**, the system will prompt a verification dialog asking if you wish to apply the changes made to the original table. It is essential to click **Yes** to ensure the blank rows remain removed in the desktop environment. Once applied, the dataset is permanently stored in its clean state within the Power BI file, dramatically enhancing the performance and reliability of the analytical environment.

Alternative Data Cleansing Techniques for Specific Nulls

While the **Remove Blank Rows** feature is ideal for globally deleting rows where all fields are null, complex data quality issues often require more nuanced approaches. For scenarios where you only want to remove a row if a critical identifying column (such as Customer ID or Product SKU) is blank, filtering techniques are necessary. In the **Power Query Editor**, users can click the filter icon on the header of the specific column, uncheck the 'null' or '(blank)' option, and apply the filter.

Furthermore, users might encounter situations where a row is not entirely blank, but contains a null in a column required for calculation. Instead of deleting the entire row, a better approach might be to replace the null value with a zero (0) or a specific default text value, using the "Replace Values" function. This preserves valuable data in other columns while mitigating calculation errors caused by nulls. These conditional replacements are vital components of maintaining robust **data integrity**.

Understanding when to use global blank removal versus targeted filtering or replacement is fundamental to becoming an expert **data analysis** professional. Always choose the method that ensures maximum data retention while simultaneously eliminating analytical risks.

Conclusion: Maximizing Data Reliability in Power BI

The presence of blank or null rows is a common hurdle in the journey from raw data ingestion to insightful visualization. Fortunately, **Power BI** equips the user with sophisticated tools, primarily housed within the **Power Query Editor**, to manage and mitigate these data quality challenges effectively. By utilizing the dedicated **Remove Blank Rows** function, analysts can swiftly and systematically eradicate completely empty records from their datasets.

This streamlined process--identifying the need for cleansing, accessing the transformation environment, executing the bulk removal, and applying the changes--ensures that the underlying **data model** is optimized for performance and accuracy. Investing the time in data preparation steps, such as removing blanks, is not merely a technical necessity; it is a prerequisite for generating reliable business intelligence and making data-driven decisions based on verifiable facts.

By integrating these robust transformation techniques into their workflow, users ensure continuous

data quality and uphold the analytical reliability of their reports, thereby maximizing the value derived from the **Power BI** platform.

Related Power BI Tutorials for Advanced Data Transformation

To further enhance your skills in data preparation and manipulation within the Power BI ecosystem, consider exploring the following advanced tutorials which explain how to perform other common tasks crucial for building comprehensive data models:

How to handle errors and inconsistent data types.

Techniques for unpivoting complex columns for better modeling.

Methods for merging and appending multiple queries efficiently.

Strategies for creating custom columns using M language formulas.

ARABPSYCHOLOGY.COM