

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

Authored by
mohammed loot

January 12, 2026

RECOMMENDED CITATION

mohammed loot (2026). *How to Remove Duplicate Rows in Power BI: A Step-by-Step Guide*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=125786>

Effective data cleaning is paramount for generating accurate and reliable business insights. One of the most common challenges faced by analysts using Power BI is dealing with redundant or duplicate records within a dataset. Duplicates distort metrics, inflate counts, and compromise the integrity of analytical models. Fortunately, Power BI provides a robust and straightforward mechanism for eliminating these redundant entries: the **Remove Duplicates** function, which is centrally located within the powerful data preparation environment known as the Power Query Editor. Understanding how to leverage this feature is fundamental to preparing high-quality data for visualization and reporting.

The concept of duplicate removal is simple yet powerful: if a table contains multiple rows that are exact copies or copies based on a specific set of identifying columns (like customer ID or transaction key), the system can automatically identify and eliminate the excess copies, leaving only one unique instance. This ensures that analytical processes, such as calculating totals or averages, are based on accurate and non-inflated foundational data. For instance, if a large customer database mistakenly includes three entries for "John Doe" with identical identifying information, applying this function ensures that John Doe is counted only once, leading to a true count of unique customers. This step is a cornerstone of effective data transformation and preparation.

The most efficient and widely used methodology for removing duplicate rows within a table loaded into Power BI involves utilizing the built-in **Remove Duplicates** feature accessible through the dedicated **Power Query Editor** interface. This interface is specifically designed for complex data shaping and cleansing operations, providing granular control over the imported data model before it is loaded into the main report environment.

The following comprehensive example will walk through the exact steps required to implement this critical data cleansing operation in a practical scenario, demonstrating the power and simplicity of this tool.

Understanding Data Duplication in Power BI

Data duplication is a ubiquitous issue stemming from various factors, including flawed data entry processes, integration of multiple data sources, or errors during ETL (Extract, Transform, Load) operations. In the context of Power BI, duplicates can manifest as identical entire rows or, more commonly, as duplicate values across a specific subset of key identifier columns. Recognizing the nature of the redundancy is the first step toward effective mitigation. Exact row duplicates are the easiest to handle, as every cell value matches another row perfectly, while duplicates defined by a key column (e.g., removing duplicate players based only on Team and Position) require focused selection. Improper handling of duplicates fundamentally compromises the fidelity of any subsequent analysis.

The operational impact of failing to address duplication is severe. It leads to inaccurate analytical results, poor resource allocation decisions, and erosion of trust in the data reporting structure. For instance, inventory counts might be artificially inflated if product IDs are duplicated, or sales figures could be misrepresented if transaction records appear more than once. Power BI's strength lies in its ability to aggregate and visualize large volumes of data; therefore, ensuring the foundational data is clean and accurate is not merely a technical step but a core business requirement. The **Power Query Editor** provides the necessary environment to scrutinize and fix these underlying data issues before they affect the final reports, saving significant time and preventing misleading conclusions.

Before proceeding with any removal, analysts must determine the definition of a duplicate for their specific task. Sometimes, two rows containing identical primary keys but different timestamps might need to be evaluated based on the freshness of the data. The "Remove Duplicates" function, by default, removes all subsequent occurrences after the first one it encounters. This behavior necessitates careful planning, particularly when the data order might influence which record is preserved. For maximum accuracy, it is always recommended to sort the data based on a relevant criteria (such as a date column) before invoking the duplicate removal function.

The Role of Power Query Editor in Data Transformation

The Power Query Editor acts as Power BI's primary engine for data connectivity, shaping, and transformation. It is a dedicated, separate environment where complex data manipulations are performed using the M language (often without requiring manual coding), allowing users to define a sequence of repeatable steps applied to the source data. Unlike functions applied in the main Power BI Desktop view (which often rely on DAX measures or calculated columns), transformations conducted in Power Query permanently alter the structure and content of the data set that is ultimately loaded into the data model, making it the authoritative source for all downstream reports.

Removing duplicates is inherently a data shaping task, making Power Query the ideal tool. When duplicates are removed in this editor, the underlying table structure is fundamentally cleansed and optimized. This approach offers significant advantages over merely filtering out duplicates in the visualization layer, which only hides the problem without solving the underlying data redundancy. By cleaning the data at the source preparation stage, we effectively reduce the overall size of the data model, dramatically improve query performance across the board, and ensure that all subsequent calculations and visualizations start from a robust foundation of unique records.

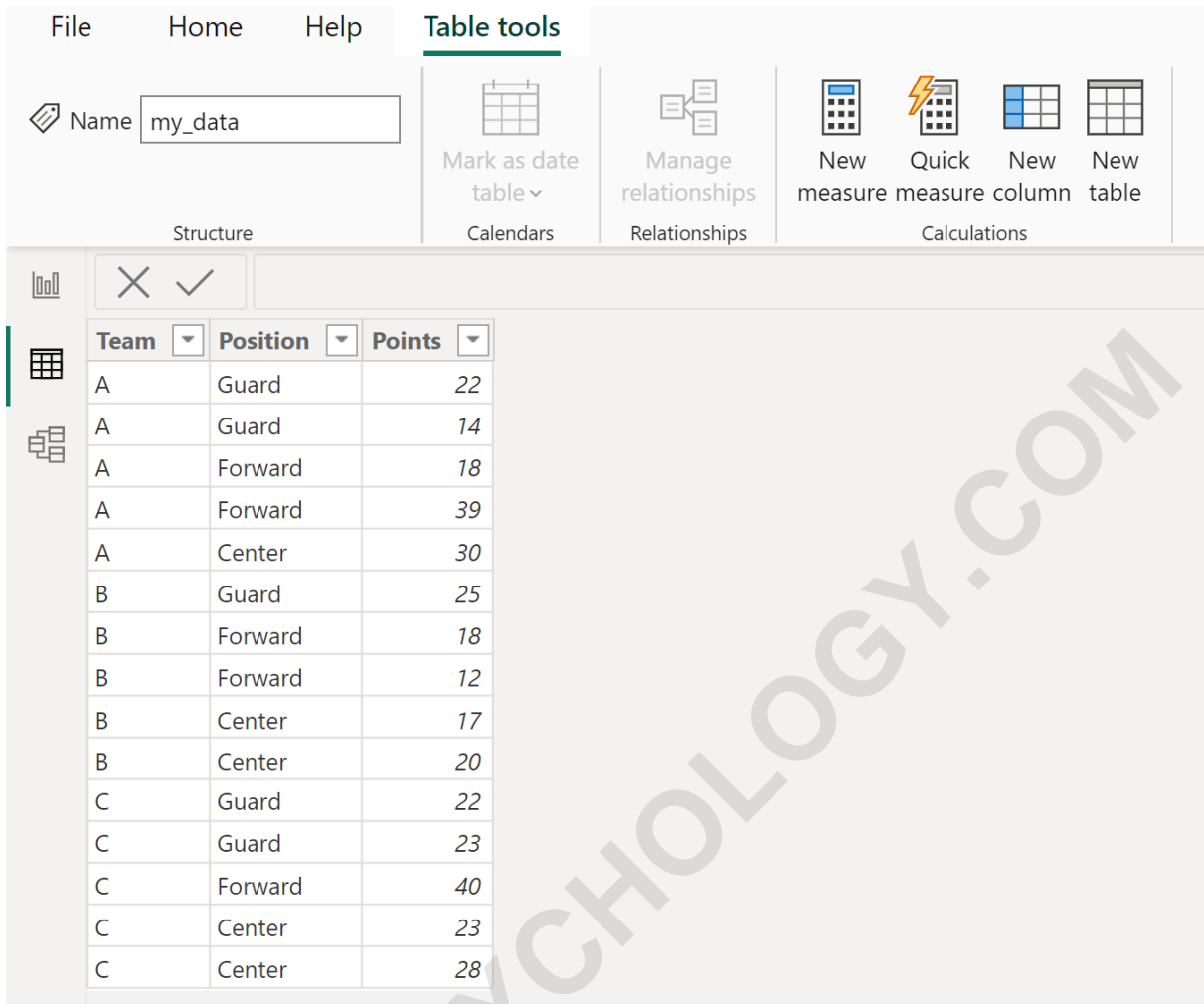
The Power Query Editor guarantees non-destructive editing by meticulously recording all transformation steps in the "Applied Steps" pane. This feature provides an audit trail for the entire data preparation workflow, allowing users to easily review, modify, or revert any transformation,

including the duplicate removal step. Furthermore, because these steps are automatically generated M code, they are consistently applied every time the data source is refreshed, ensuring the transformation pipeline remains consistent and reproducible, which is vital for maintaining data governance standards within an organization.

Detailed Example Setup: Preparing the Dataset

To demonstrate the practical application of duplicate removal, we will work with a sample data table. Assume we have imported a simple dataset into Power BI detailing information about various basketball players. This raw table, as frequently occurs with integrated data sources, contains redundant records. Our goal is to ensure that, for specific combinations of identifying attributes, only one record remains. This process is crucial when we want to analyze unique pairings or aggregate data without counting the same entity multiple times, preventing misinterpretation of team strength or positional distribution.

Suppose our table structure includes columns such as Player Name, Team, and Position. Examination of the raw data reveals several rows where the combination of **Team** and **Position** is repeated, indicating potential redundancy if our analysis requires uniqueness based on these two attributes specifically. If we were only interested in unique players, we would select only the Player Name column. However, since our objective is to identify unique team-position slots, we must select both. The initial view of our data before any manipulation is presented below, clearly illustrating the presence of repeated combinations:



The screenshot shows the Power BI interface with the 'Table tools' ribbon active. The table name is 'my_data'. The ribbon includes options for 'Mark as date table', 'Manage relationships', and 'Calculations' (New measure, Quick measure, New column, New table). The table data is as follows:

Team	Position	Points
A	Guard	22
A	Guard	14
A	Forward	18
A	Forward	39
A	Center	30
B	Guard	25
B	Forward	18
B	Forward	12
B	Center	17
B	Center	20
C	Guard	22
C	Guard	23
C	Forward	40
C	Center	23
C	Center	28

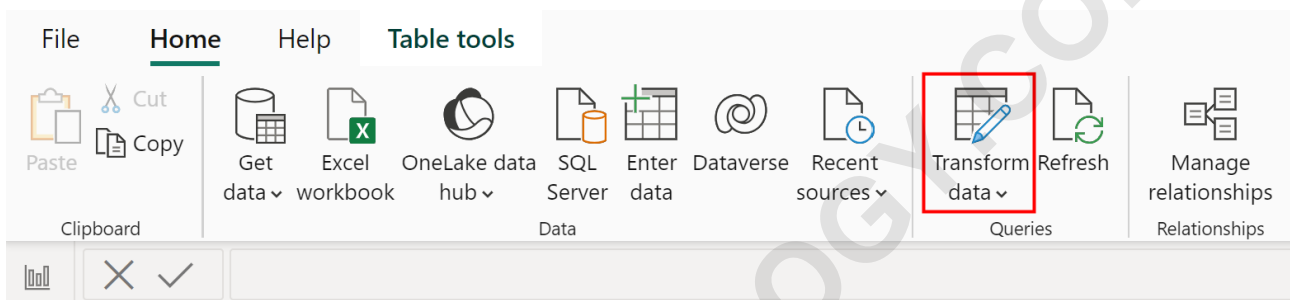
Carefully observe the table image above. Notice that there are several rows that contain the same value when considering both the **Team** and **Position** columns concurrently. For instance, multiple rows might list "Lakers" and "Guard." If our analytical focus is to count the unique team-position pairings, these repetitions must be eliminated. Our objective is to remove all rows that have duplicate values across this chosen subset of columns, leaving only the first occurrence encountered by the system, thereby creating a clean list of unique assignments.

Accessing the Power Query Editor

Before any data transformation can occur, the user must navigate from the standard Power BI Desktop interface into the specialized data shaping environment. This process begins by ensuring you are within the main reporting view of Power BI Desktop, where your data tables are visible in the Fields pane. Accessing the editor is a straightforward procedure designed to transition seamlessly into the data preparation phase. It is essential to remember that any changes made here are preparatory and will only be applied to the data model upon saving and closing the editor, underscoring the importance of verifying results before closing.

To initiate the transformation sequence, locate the primary ribbon menu at the top of the Power BI Desktop application. Click the **Home** tab if it is not already selected. Within the 'External Data' or 'Queries' group (depending on your version layout), you will find the crucial button labeled **Transform data**. Clicking this icon serves as the gateway to launch the separate application window dedicated to the Power Query Editor. This step is the non-negotiable prerequisite for performing advanced data cleansing operations like duplicate removal, as these functions are not available directly within the standard desktop view.

The visual step is demonstrated here, highlighting the path to the transformation environment:



Upon clicking the **Transform data** icon, a new window will appear, presenting the **Power Query Editor** interface. This new environment provides a comprehensive view of the selected table, allowing for detailed manipulation. You will see the table loaded, along with the ribbon of transformation tools, the Queries pane, and the Applied Steps pane--the latter being critical for tracking the cleansing history. The initial state of the editor, showing the loaded table, should resemble the following view, confirming readiness for the next steps and allowing the user to review the entire data structure prior to making changes:

The screenshot displays the Power Query Editor interface. The ribbon includes tabs for File, Home, Transform, Add Column, View, Tools, and Help. The 'Remove Rows' group in the 'Home' tab is active, showing options like 'Keep Rows' and 'Remove Rows'. The 'Queries' pane on the left shows a query named 'my_data'. The main area displays a table with the following data:

	Team	Position	Points
1	A	Guard	22
2	A	Guard	14
3	A	Forward	18
4	A	Forward	39
5	A	Center	30
6	B	Guard	25
7	B	Forward	18
8	B	Forward	12
9	B	Center	17
10	B	Center	20
11	C	Guard	22
12	C	Guard	23
13	C	Forward	40
14	C	Center	23
15	C	Center	28

The formula bar shows the M code: `= Table.TransformColumnTypes("#Promoted Headers",{{"Team", type text}})`

Executing the 'Remove Duplicates' Command

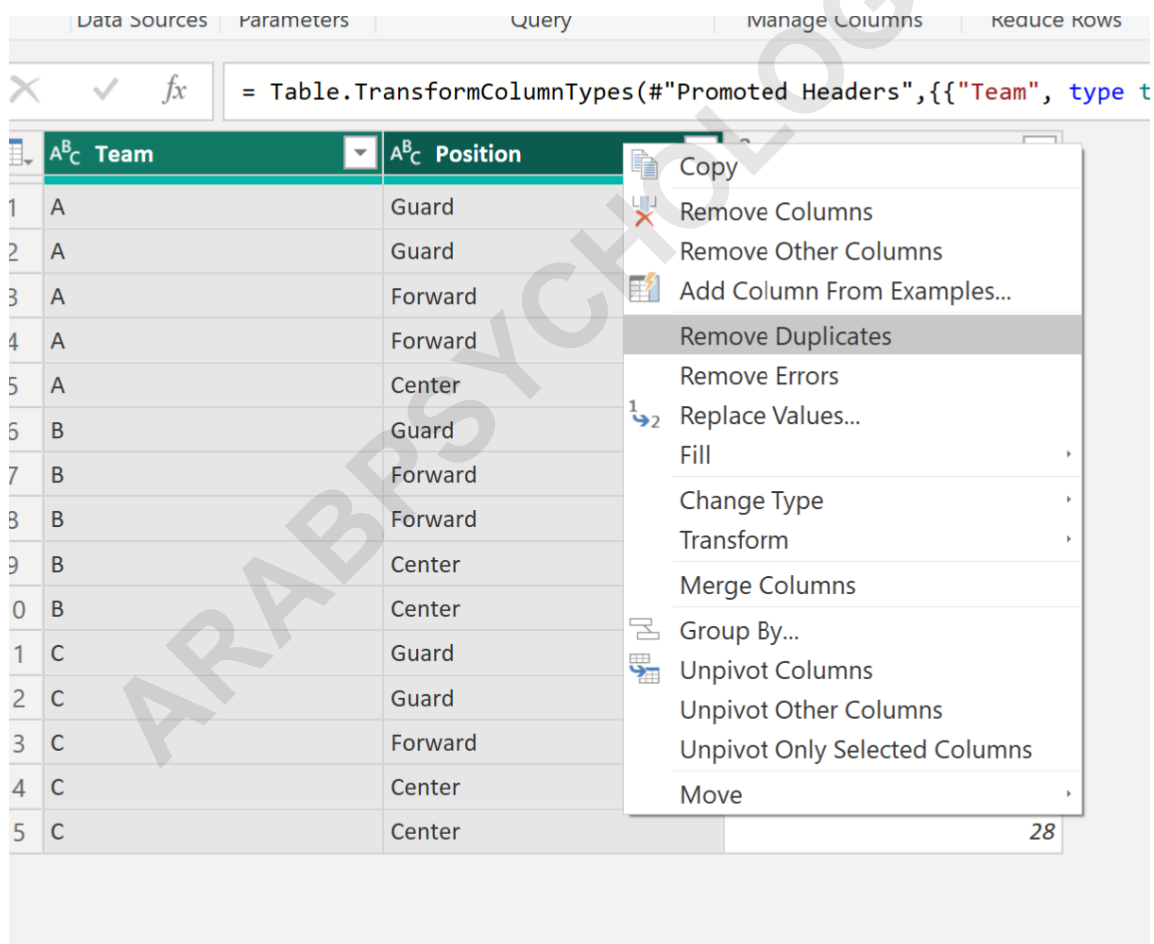
Once inside the Power Query Editor, the process of removing duplicates requires precise identification of the columns that define the uniqueness criteria. It is vital to understand that the "Remove Duplicates" function operates based on the selected columns. If no columns are selected, Power Query attempts to find and remove rows that are entirely identical across all columns, a rare occurrence in real-world data. However, in our specific example, we are interested in removing duplicates based only on the combination of **Team** and **Position**, preserving the differences in Player Name or other associated statistics, which is a much more common requirement in relational data analysis.

To perform this targeted cleansing, the user must first select the relevant columns. While holding down the **Ctrl** key on the keyboard, click on the column header named **Team**, and then click on the column header named **Position**. This multiple selection action highlights these two columns, signaling to Power Query that the subsequent action should be scoped specifically to the values

within these fields. If a row shares identical values in both selected columns with another row, one of them will be flagged for removal, retaining only the first instance encountered in the current table order.

With the required columns highlighted, proceed to right-click anywhere within the header area of the selected columns (or access the "Remove Rows" section under the Home tab). A context menu will appear, presenting various transformation options tailored to column manipulation. Locate and click the option explicitly labeled **Remove Duplicates**. This single action triggers the Power Query engine to analyze the selected data, identify all instances where the combined key (Team + Position) is repeated, and retain only the first instance it encounters, effectively deleting all subsequent duplicate rows associated with that key set.

The visual confirmation of invoking this critical function is demonstrated below, illustrating the context menu selection that initiates the transformation:



The screenshot shows the Power BI interface with a table containing two columns: 'Team' and 'Position'. The table has 15 rows of data. A context menu is open over the table, and the 'Remove Duplicates' option is highlighted. The formula bar at the top shows the DAX formula: `= Table.TransformColumnTypes("#Promoted Headers",{{"Team", type t`. The table data is as follows:

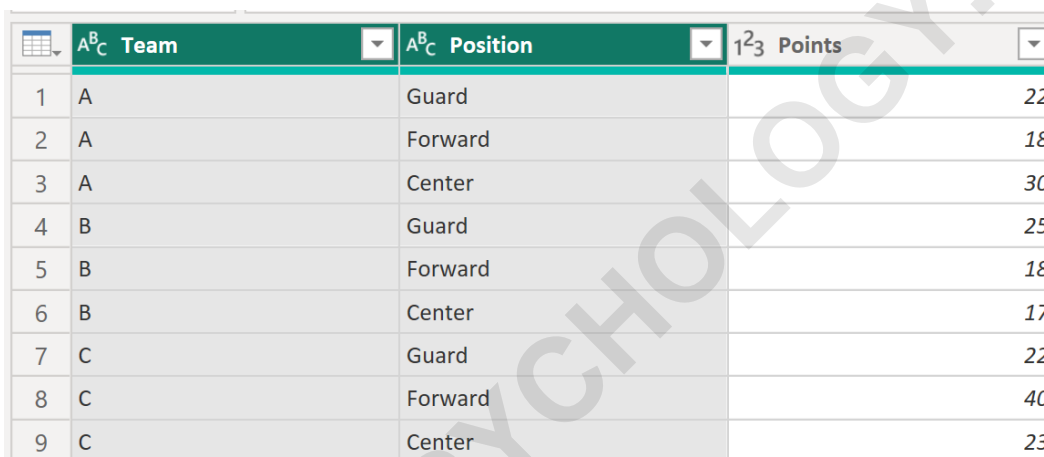
	Team	Position
1	A	Guard
2	A	Guard
3	A	Forward
4	A	Forward
5	A	Center
6	B	Guard
7	B	Forward
8	B	Forward
9	B	Center
0	B	Center
1	C	Guard
2	C	Guard
3	C	Forward
4	C	Center
5	C	Center

Analyzing the Results and Data Integrity

Immediately following the execution of the **Remove Duplicates** command, the table within the

Power Query Editor will refresh, displaying the newly cleansed dataset. You will observe that the total number of rows has decreased, reflecting the removal of the redundant entries. Crucially, if you examine the columns **Team** and **Position**, you will find that every combination now appears only once, fulfilling the requirement for unique team-position pairings. This confirms that the targeted transformation based on the two selected key columns was successful, ensuring data integrity based on our defined uniqueness criteria and preparing the data for reliable aggregation.

The resulting table, post-transformation, should look significantly leaner, demonstrating that all rows sharing duplicate values across both the **Team** and **Position** columns have been successfully eliminated. For example, if there were four entries for "Celtics" and "Center," only one of those original four rows will remain, typically the one that appeared highest in the original table list. The output illustrates the refined dataset, ready to be loaded into the data model:



	Team	Position	Points
1	A	Guard	22
2	A	Forward	18
3	A	Center	30
4	B	Guard	25
5	B	Forward	18
6	B	Center	17
7	C	Guard	22
8	C	Forward	40
9	C	Center	23

This stage is vital for verification. Before moving on, it is good practice to review the "Applied Steps" pane on the right-hand side of the Power Query Editor. A new step, likely labeled "Removed Duplicates," will have been added. This feature ensures traceability, allowing the user to click back through the steps to see the data before and after the duplication removal, providing absolute confidence in the data preparation workflow. If the result is unsatisfactory or an error was made in column selection, the step can easily be deleted or modified here without affecting the original data source, maintaining the flexibility of the transformation process.

Applying Changes and Concluding the Transformation

The transformations performed within the Power Query Editor are staging changes; they do not

automatically apply to the main Power BI data model until explicitly instructed. Once you are satisfied with the results of the duplicate removal and any other subsequent data cleaning steps, you must formally conclude the transformation session and load the refined data back into Power BI Desktop for analysis and reporting. This critical final step bridges the data preparation environment with the visualization environment, making the cleansed data available for report creation.

To finalize the process, locate the **Home** tab within the Power Query Editor ribbon. Click the **Close & Apply** button. This action performs two vital functions: it closes the Power Query Editor window, and more importantly, it executes all the recorded steps (including the duplicate removal) against the underlying data source, loading the resulting, cleansed table into the data model of Power BI Desktop. The system will then process the data, and you might see status messages indicating the data load and structure application progress, confirming the successful transformation.

If you attempt to close the editor without explicitly clicking "Close & Apply," a prompt will appear asking if you would like to apply the changes you made to the original table. Click **Yes** to ensure the transformation steps, including the removal of duplicate rows, are permanently incorporated into the data model that Power BI will use for all subsequent reports, measures, and visualizations. Failing to apply the changes means the original, uncleaned data will remain in the model, rendering the transformation effort ineffective. Always confirm the application of changes to complete the data preparation pipeline successfully and maintain data consistency.

Considerations for Advanced Duplicate Management

While the standard **Remove Duplicates** feature is highly effective for exact matches or matches based on selected key columns, advanced scenarios sometimes require more nuanced handling. For instance, what if two records are almost identical but have minor differences in non-key fields (e.g., slight spelling variations in a secondary column, or different timestamps)? In such cases, the standard tool might fail to identify them as duplicates if the entire row is not identical, or if the user requires a conditional removal based on specific criteria not related to the primary key, such as retaining the record with the most complete set of supplementary information.

For scenarios demanding conditional removal--such as keeping the latest record based on a date column, or the record with the highest value in a sales column--users must employ more sophisticated Power Query techniques. One common approach involves using the **Group By** transformation. By grouping the data based on the key columns (Team and Position, in our example), the user can then specify an aggregation function to determine which row is retained. For example, applying a "Max" function to a Date/Time column during grouping ensures that the row corresponding to the most recent entry for that duplicate key combination is preserved, providing maximum flexibility and control over data lineage and integrity when dealing with time-

series data.

Furthermore, it is worth considering the performance implications when dealing with exceptionally large datasets. Removing duplicates is generally a resource-intensive operation, as it requires Power Query to scan and compare every row based on the selected key columns. For very large tables, ensuring that the necessary data source optimizations are in place (e.g., proper indexing or using dataflow partitioning in Power BI Service) can help expedite the transformation process and reduce load times. Ultimately, mastering the **Remove Duplicates** function is a foundational skill, but understanding when to pivot to advanced techniques like Group By ensures that Power BI analysts can handle data quality issues across the entire spectrum of complexity, optimizing both accuracy and efficiency.

ARABPSYCHOLOGY.COM