

How to Get Unique Values from Multiple Columns in Power BI

Authored by
stats writer

January 27, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Get Unique Values from Multiple Columns in Power BI*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=128073>

Power BI: Get Distinct Values from Multiple Columns

The Necessity of Distinct Combinations in Data Analysis

Power BI is recognized globally as a robust platform for data analytics and visualization, enabling organizations to transform complex raw data into actionable business intelligence. A core requirement in effective data manipulation is the ability to identify and extract unique values or distinct combinations across multiple dimensions within a large dataset. This capability is paramount when attempting to cleanse data, identify specific patterns, or establish the cardinality of relationships necessary for accurate modeling and reporting.

Often, when working with transactional or detailed datasets, redundant rows exist where the primary identifier (like a transaction ID) differs, but the descriptive attributes (such as Region, Product Category, or in our case, Team and Position) are identical. Identifying the unique combinations of these descriptive attributes is critical for creating dimension tables or ensuring that visualizations do not count redundant groupings. Without the ability to efficiently extract these unique pairings, data models can become inefficient, leading to slow performance and potentially misleading analytical results.

While Power BI offers various tools for data transformation, achieving a list of truly distinct row combinations often requires utilizing DAX (Data Analysis Expressions) to construct a new calculated table. Unlike filtering individual columns, finding distinct combinations across several columns simultaneously ensures that you capture every unique pairing or grouping present in your source data, which is essential for detailed segmentation and trend analysis. This method ensures data integrity and model efficiency by establishing a definitive list of unique keys.

Leveraging DAX and the SUMMARIZE Function

To efficiently obtain the distinct values across multiple columns, we move beyond the standard Power Query Editor functions and utilize a calculated table defined by DAX. DAX provides powerful functions for creating derived tables, columns, and measures, offering flexibility far beyond what standard graphical interfaces can provide. The key function for this specific task is SUMMARIZE.

The SUMMARIZE function, while commonly used for grouping data and performing aggregations (like sums or averages), has the powerful and essential side effect of providing a distinct list of combinations when no aggregation arguments are explicitly included. When you pass a source table and a list of grouping columns to SUMMARIZE, the resulting output is a new table containing every unique row combination defined by those specified columns. This is the most efficient and recommended way within DAX to generate a distinct list of composite keys from a larger dataset.

The fundamental syntax for using this function to create a new, derived table containing only the distinct values across multiple columns in a specific table in Power BI is both concise and powerful. It dictates the source of the data and the columns that define the unique grouping:

Distinct Table = SUMMARIZE(my_data, ,)

In this structure, `Distinct Table` represents the name of the new calculated table you are defining. The first argument, `my_data`, is the name of your source table residing in the data model. Subsequent arguments, such as `and` and `,` specify the columns from which you want to extract the unique combinations. The resulting table will include only those specified columns, ensuring a clean and concise output focused solely on the unique pairings, effectively eliminating any redundant rows based on this two-column key.

Step-by-Step Implementation: Reviewing the Dataset

To solidify the understanding of how the SUMMARIZE formula operates, we will walk through a clear, practical example using a sample dataset. We begin by assuming we have a detailed source table successfully imported into the Power BI environment. This table is named **my_data** and contains detailed information that includes player names, team affiliations, and specific positions.

It is crucial to note the redundancy in the descriptive columns. While the players themselves might be unique, the combination of their team and position repeats across many rows. Our goal is to derive a streamlined reference table that lists every possible unique pairing of Team and Position found within this raw data, ignoring the player-specific details.

Observe the structure and content of our sample source data, **my_data**:

The screenshot shows the Power BI Desktop interface with the 'Table tools' ribbon selected. The ribbon includes options like 'Name' (set to 'my_data'), 'Mark as date table', 'Manage relationships', and 'Calculations' (with sub-options: 'New measure', 'Quick measure', 'New column', 'New table'). Below the ribbon, a data table is displayed with the following columns: Team, Position, and Points.

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

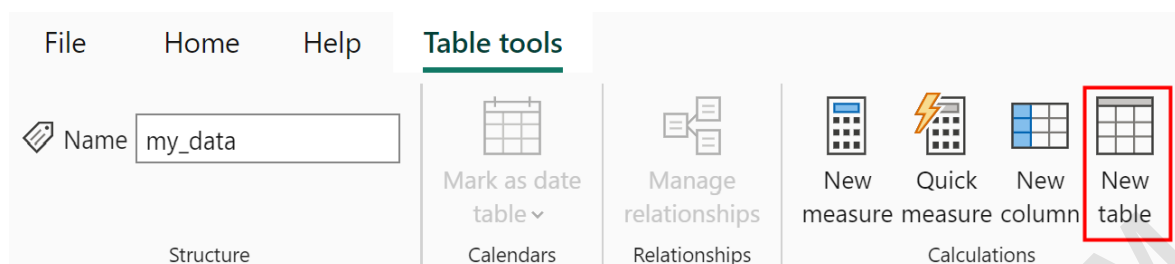
We can clearly observe that, for instance, Team "A" has several players listed as "Guard." If we were building a dimension table to filter future reports by unique Team-Position combinations, we would only need one instance of the "A / Guard" combination, regardless of how many players currently hold that assignment in the **my_data** fact table. This necessity drives the use of the distinct multi-column extraction technique.

Executing the DAX Formula for Calculated Tables

Having defined our objective--to extract the distinct values across the **Team** and **Position** columns--the next step involves initiating the creation of a new calculated table directly within the Power BI modeling environment. This process requires accessing the modeling ribbon and entering the precise DAX expression we identified earlier.

To begin this creation process, navigate your cursor to the **Table tools** tab, which is located prominently along the top ribbon of the Power BI Desktop interface. This section is dedicated to

data modeling operations, including the creation of derived tables and measures. Within this tab, locate and click the **New table** icon.



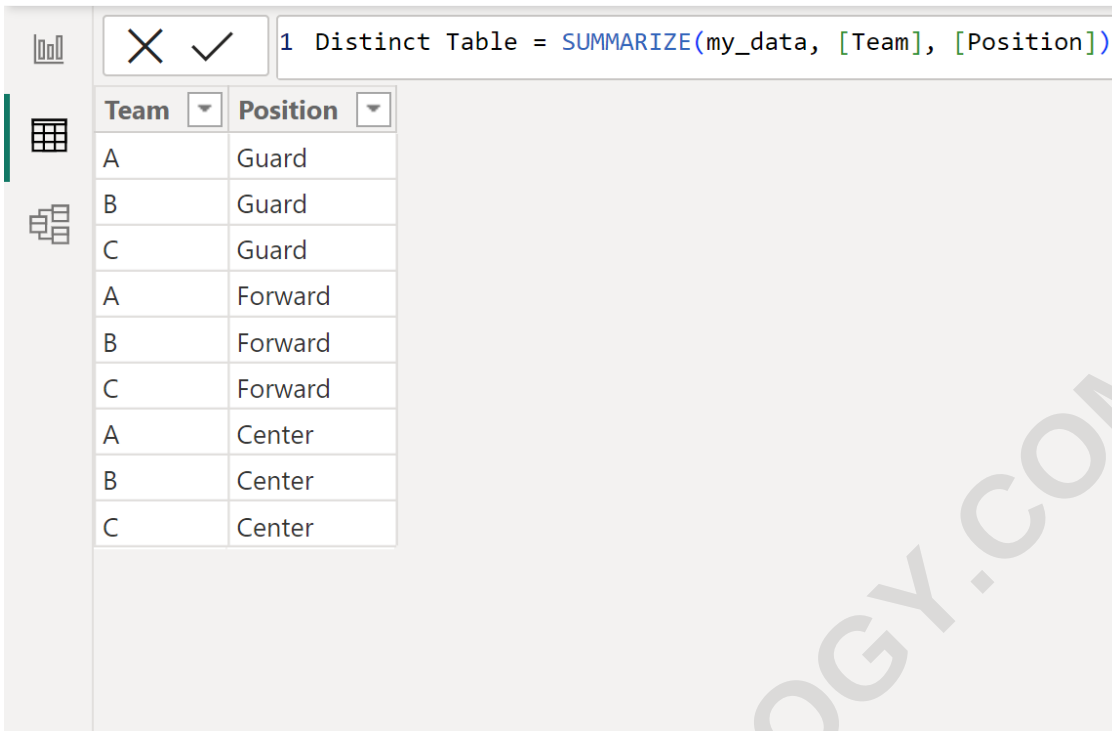
Clicking this icon immediately activates the DAX formula bar, prompting you to define the structure and content of your derived table. It is essential to enter the formula accurately, referencing the source table and the specific fields required for uniqueness:

Distinct Table = SUMMARIZE(my_data, ,)

Upon successful entry and confirmation of the formula (by pressing Enter), Power BI executes this DAX query against the internal data model. It meticulously calculates the result set, identifying all unique values and combinations of the **Team** and **Position** columns found within the large **my_data** table. The result is stored as a permanent, new table within the data model, ready for immediate use in establishing relationships or enhancing visualizations.

Analyzing the Calculated Output for Data Integrity

Following the execution of the DAX formula, a new object named **Distinct Table** is successfully generated and integrated into your data model. This table contains only the distinct combinations of values across the specified columns, **Team** and **Position**, from the original source table. This outcome verifies that the SUMMARIZE function performed its intended role: collapsing repeated multi-column entries into a single unique record.



The screenshot shows the Power BI interface. At the top, the DAX formula bar contains the formula: `1 Distinct Table = SUMMARIZE(my_data, [Team], [Position])`. Below the formula bar, a table is displayed with two columns: **Team** and **Position**. The table contains the following rows:

Team	Position
A	Guard
B	Guard
C	Guard
A	Forward
B	Forward
C	Forward
A	Center
B	Center
C	Center

A careful comparison between the original **my_data** table and the resulting **Distinct Table** immediately reveals the efficiency and cleanliness achieved by this method. For instance, there were multiple rows in the source data that shared a **Team** value of "A" and a **Position** value of "Guard." However, only one row in this new table contains this exact combination. All instances of duplicate pairings are successfully collapsed into a single, canonical record.

Similarly, the original dataset contained numerous occurrences where the **Team** value was "A" and the **Position** value was "Forward." The new calculated table ensures that this pairing, along with all other possible unique combinations found in the source data (such as Team B/Center, Team C/Guard, etc.), appears only once. This process guarantees a complete and accurate list of all unique organizational structures present in the raw data, which is crucial for building efficient dimension tables that link back to the fact data.

Alternative Data Transformation Methods

While the DAX SUMMARIZE method is highly effective for creating a calculated table within the modeling phase, it is important for expert Power BI users to be aware of alternative methods, particularly those used during the data loading and transformation phase via Power Query.

The Power Query Editor, which uses the M language, offers a highly visual and intuitive way to achieve the same result before the data even enters the data model. Inside the Query Editor, users can select the desired columns (e.g., Team and Position), right-click on the selection, and choose

the "Remove Duplicates" option. This operation modifies the query step, ensuring that only the unique values combinations across the selected fields are loaded into the data model.

The choice between using DAX (a calculated table) and Power Query (a transformation step) primarily depends on the context of the data lifecycle. Power Query handles ETL (Extract, Transform, Load) operations and is generally preferred for data cleansing, while DAX handles calculations and derivations that occur within the model structure itself. If your data source is static or requires minimal modification, Power Query is often the simpler approach. If the distinct combinations need to be dynamically derived based on pre-existing relationships or complex measures, DAX calculated tables are the superior choice.

Conclusion and Next Steps in Data Mastery

Mastering the creation of distinct combination tables is a foundational skill for advanced data analytics and modeling in Power BI. These unique combination tables are frequently utilized as dimension tables (lookup tables) in robust star schema designs, allowing complex measures from the original detailed fact table (like **my_data**) to be filtered accurately, consistently, and efficiently across all reports.

The ability to accurately define and extract these fundamental dimensions ensures that your reports are built on a robust and streamlined data structure, leading to improved dashboard performance and greater analytical precision. We highly recommend continuing the exploration of the full capabilities of DAX to unlock deeper insights and create more dynamic solutions from your datasets.

Note: You can find the complete documentation for the **SUMMARIZE** function in DAX, along with detailed examples and advanced usage scenarios, on the official Microsoft documentation pages. Understanding its optional arguments for aggregation is key for performing more complex grouping operations beyond simple distinct combination extraction.

The following tutorials explain how to perform other common tasks in Power BI, enabling you to continue building expertise in data manipulation and visualization:

How to calculate running totals in Power BI.

Methods for merging multiple tables using Power Query.

Techniques for dynamic row coloring in visual tables.