

Learning to Extract Distinct Values from Multiple Columns in Power BI Using DAX

Authored by
Mohammed loot

November 12, 2025

RECOMMENDED CITATION

Mohammed loot (2025). *Learning to Extract Distinct Values from Multiple Columns in Power BI Using DAX*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=17631>

Introduction to Data Aggregation and Distinct Combinations in Power BI

When working with complex datasets in [Power BI](#), analysts frequently encounter scenarios where they need to derive a list of unique combinations of attributes across multiple columns. While identifying distinct values within a single column is straightforward, extracting unique rows based on the combination of several fields requires a specialized approach utilizing the powerful capabilities of [DAX](#) (Data Analysis Expressions). This process is crucial for effective [data modeling](#), especially when preparing dimension tables or simply reducing the cardinality of a view to focus solely on unique categorical pairings. A common requirement, for instance, is generating a master list of all unique Team and Position pairings present in a large transactional or fact table, discarding all duplicate rows that share those exact attribute values.

Achieving this goal requires creating a calculated table using a specific DAX function designed for summarization and grouping. This method ensures that the resulting structure is stable, reusable, and optimized for subsequent reporting or relationship creation within the Power BI environment. Unlike simple filtering, which only affects visualization scope, creating a new calculated table permanently defines a new data structure in the model, based on the unique combinations derived from the source table. This approach is superior for analytical purposes as it isolates the necessary unique identifiers into a clean, referenceable entity, simplifying complex measure calculations later on.

The syntax used to define this new distinct table must be precise, leveraging the core functionalities of DAX to perform the necessary grouping operation efficiently. Understanding how this function interprets the column arguments is essential; it processes the source table row by row, building a unique list of pairings defined by the combination of the specified columns. Any row combination that has already appeared is skipped, ensuring that the final output contains only one instance of each unique combination, thereby addressing the common analytical challenge of isolating distinct categorical intersections.

Understanding the DAX Solution: The Role of SUMMARIZE

The most reliable and performance-optimized function in [DAX](#) for deriving a table of distinct values across multiple columns is the [SUMMARIZE](#) function. While its name suggests aggregation, when used without explicit aggregation arguments (like SUM or COUNT), [SUMMARIZE](#) effectively acts as a grouping tool. It returns a table containing the unique combinations of the columns listed as its grouping parameters. This function is fundamental to advanced [data modeling](#) in Power BI, allowing for the creation of new dimensional structures directly within the data model.

The fundamental syntax for creating a new table containing only the distinct values across multiple columns in a specified source table is structured as follows. We begin by defining the name of the new calculated table, followed by the call to the [SUMMARIZE](#) function, supplying the source table

name, and then listing all columns whose unique combinations we wish to retain. This structure clearly communicates the intent: group the source data by these specific attributes and return the resulting unique groups as a new table.

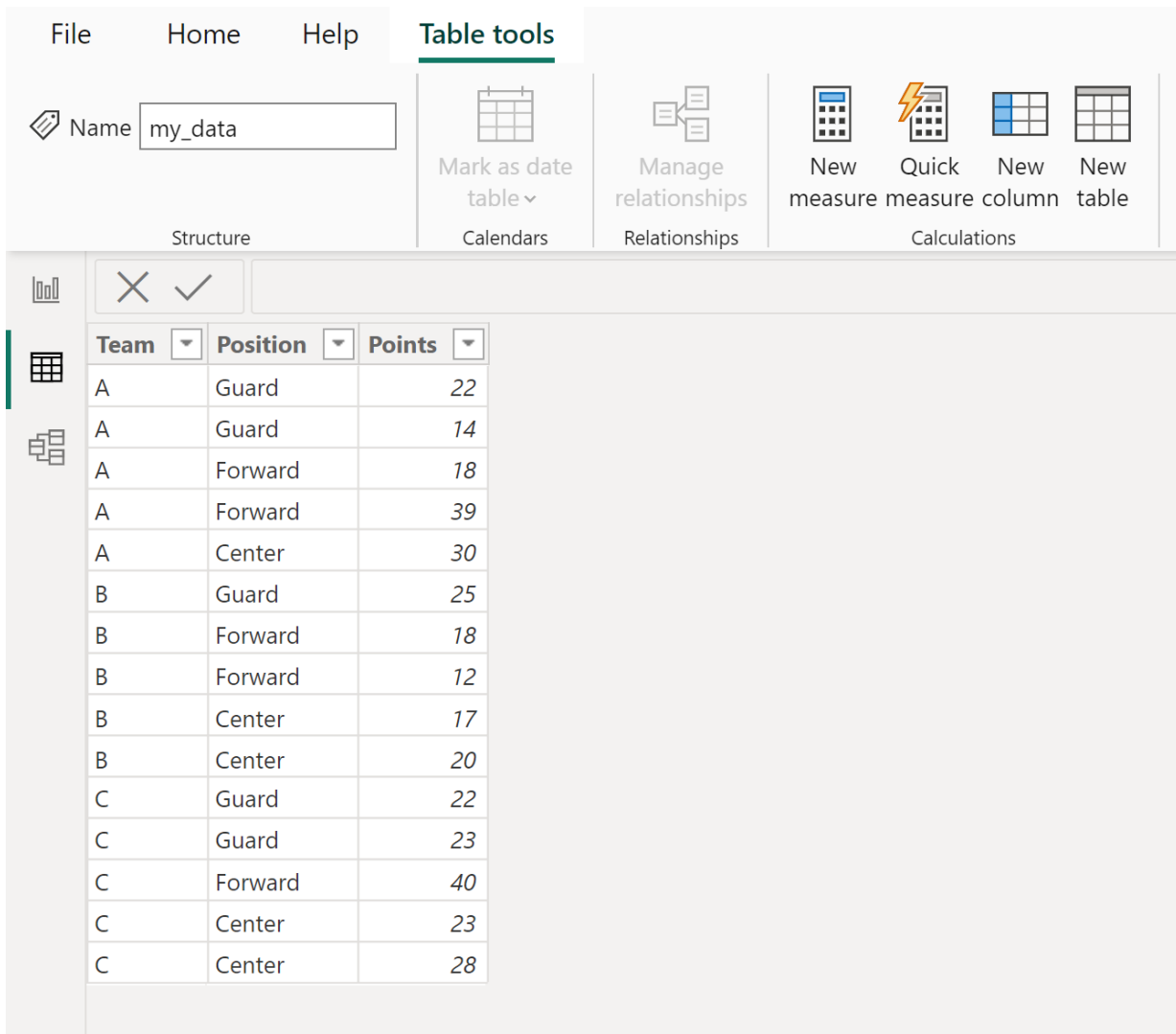
Distinct Table = SUMMARIZE(my_data, ,)

In the above example, a new table named **Distinct Table** is generated. This table will exclusively contain the distinct combination of values found across the **Team** and **Position** columns of the **my_data** table. It is crucial to note that if the source table had a row where Team='A' and Position='Guard', and another row exactly matching that combination, only one row representing that unique pairing will be present in the resulting **Distinct Table**. This method is highly efficient because it inherently performs the necessary de-duplication based on the combined keys specified within the function arguments, ensuring data integrity and reduced redundancy for subsequent analysis within [Power BI](#).

Practical Example: Extracting Unique Player Attributes

To illustrate the application of the [SUMMARIZE](#) function, consider a typical scenario involving a sports data table in [Power BI](#). Suppose we have a table named **my_data** that logs detailed information about basketball players, including multiple entries for players who have identical team and position assignments, or perhaps multiple records over different time periods that share the same core attributes. Our goal is to derive a clean, definitive list of every unique combination of **Team** and **Position** that exists within this raw dataset, effectively creating a dimensional attribute table based on these two categorical fields.

The source table, **my_data**, might look like the representation below. Notice the repetitive nature of the entries: Team 'A' has multiple players listed as 'Guard', and Team 'B' similarly has multiple instances of 'Center'. If we were to analyze this table directly for unique combinations, we would quickly realize the redundancy. For effective reporting and filtering, we need to collapse these repeated combinations into single, definitive rows. This step is foundational for robust [data modeling](#), where dimensional tables must represent unique entities.



The screenshot shows the 'Table tools' ribbon in Power BI Desktop. The ribbon includes a 'Name' field set to 'my_data', a 'Structure' section, and several tool groups: 'Mark as date table' (with a 'Calendars' dropdown), 'Manage relationships' (with a 'Relationships' dropdown), and 'Calculations' (with 'New measure', 'Quick measure', 'New column', and 'New table' options). Below the ribbon, a data table is displayed with columns 'Team', 'Position', and 'Points'. The table contains 15 rows of data.

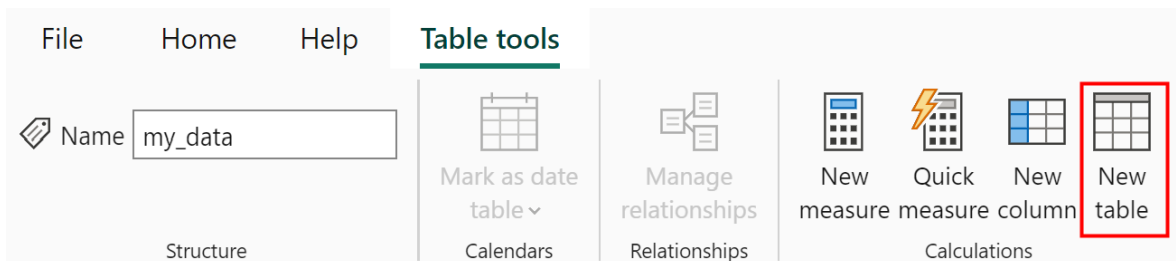
Team	Position	Points
A	Guard	22
A	Guard	14
A	Forward	18
A	Forward	39
A	Center	30
B	Guard	25
B	Forward	18
B	Forward	12
B	Center	17
B	Center	20
C	Guard	22
C	Guard	23
C	Forward	40
C	Center	23
C	Center	28

The objective is clear: we seek to create a new table that extracts the distinct values based on the intersection of the **Team** and **Position** columns. This new table will serve as a lookup table, ensuring that when we build slicers or filters based on these attributes, the user is presented with only the actual, unique pairings that exist in the underlying data, preventing potential confusion or misleading analysis caused by unnecessary repetition in the data model structure.

Step-by-Step Implementation in Power BI Desktop

Implementing the solution requires navigating the [Power BI](#) Desktop interface to create a new calculated table. This process is initiated from the modeling tools section, which grants access to the DAX calculation engine outside of standard measure or column creation. First, ensure you are in the Data view or Model view within Power BI Desktop, as the relevant tools are context-dependent and typically available when focusing on the overall data structure rather than individual report visuals.

To begin, locate and click the **Table tools** tab positioned along the top ribbon menu. Within this section, locate and click the **New table** icon. This action opens the DAX formula bar, prompting the user to define the expression that will calculate the contents of this new data structure. This is where the powerful grouping capability of [DAX](#) is leveraged to perform the necessary de-duplication across multiple columns simultaneously, transforming the raw data into a refined dimensional structure.



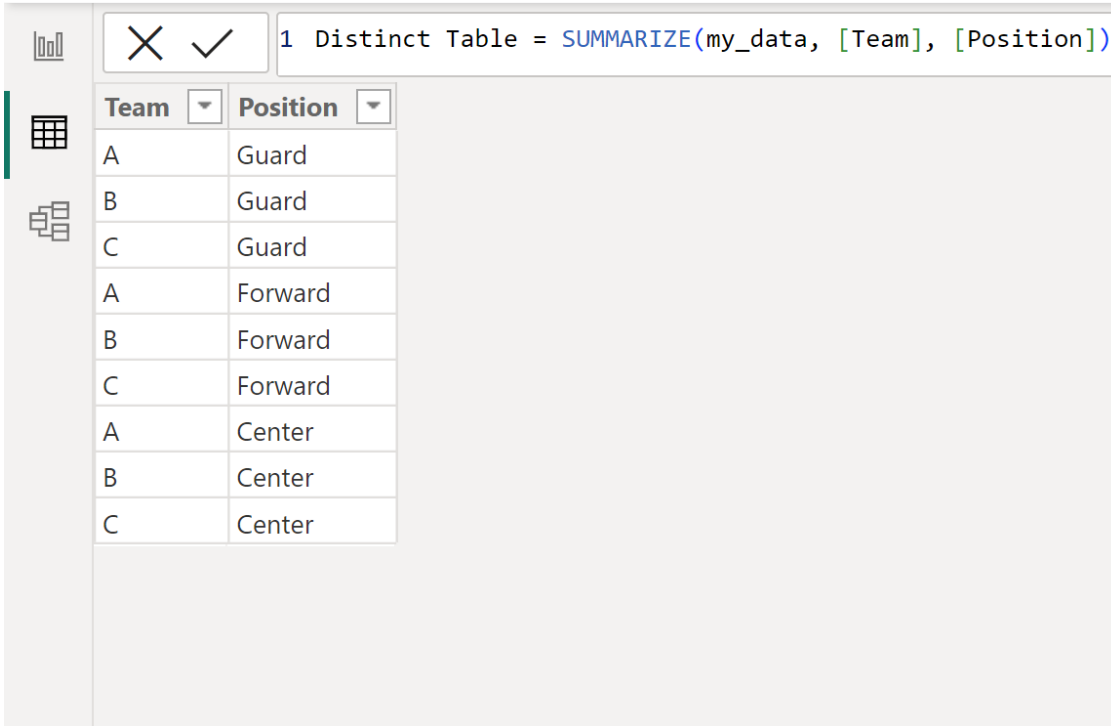
After activating the formula bar, the analyst must input the precise [DAX](#) expression utilizing the [SUMMARIZE](#) function, referencing the source table (**my_data**) and the columns intended for grouping (**Team** and **Position**). The structure should adhere strictly to the previously defined syntax, assigning the result to the desired table name, in this case, **Distinct Table**. This execution immediately calculates the result set and adds the new table to the data model alongside the original source data, ready for use in relationships or visualizations.

Distinct Table = SUMMARIZE(my_data, ,)

The execution of this formula results in the immediate creation of the **Distinct Table**. This new table contains only the distinct, non-redundant pairings of values across the **Team** and **Position** columns derived from the original **my_data** table. The system performs an internal grouping operation, ensuring that the cardinality of the new table is significantly reduced compared to the source, providing a highly efficient reference list for these combined attributes.

Analyzing the Output and Alternative Approaches

Upon successful execution of the [SUMMARIZE](#) statement, the resulting **Distinct Table** will present a clean, concise view of all unique Team and Position combinations that existed in the source data. This output is a powerful example of how DAX can be used to engineer dimensional tables from transactional data. For instance, while the original **my_data** table contained multiple rows where **Team** was 'A' and **Position** was 'Guard', the new table contains only a single entry for that specific combination, confirming the success of the de-duplication process based on the combined key.



```
1 Distinct Table = SUMMARIZE(my_data, [Team], [Position])
```

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

Similarly, any other combination that appeared multiple times in the source data--such as Team 'A' and Position 'Forward'--is similarly condensed into a single row in the new table. This reduction in data granularity is vital for creating effective relationships in the data model and ensuring that measures and aggregates are calculated correctly against a clean set of dimensions. The unique pairings derived here can be used as the primary key for a new dimension table, linking back to the original fact table (**my_data**) if necessary, thereby improving the overall structure and performance of the [data modeling](#) effort within [Power BI](#).

While [SUMMARIZE](#) is the preferred method for its directness and traditional grouping capability, it is worth noting that alternative [DAX](#) functions, such as `DISTINCT` combined with `SELECTCOLUMNS`, or even `GROUPBY` in conjunction with `ADDCOLUMNS`, could potentially achieve similar results. However, [SUMMARIZE](#) provides the most concise and readable syntax for simple multi-column distinct extraction, making it the industry standard for this specific requirement. Analysts should choose the function that best balances performance, maintainability, and clarity for their specific [data modeling](#) needs.

Conclusion and Further Resources

Mastering the creation of calculated tables containing distinct combinations of values is a foundational skill for advanced data analysis in [Power BI](#). By leveraging the power of the [SUMMARIZE](#) function within [DAX](#), analysts can efficiently transform raw, highly granular data into clean, dimensional structures that enhance the performance and accuracy of reports. The ability to

isolate unique categorical pairings is essential for creating robust data models that accurately reflect business realities without redundant information.

The method demonstrated, using the precise grouping capabilities of [SUMMARIZE](#), ensures that the resulting table is optimized for use as a lookup or dimension table, streamlining subsequent data relationships and analytical queries. This technique is applicable across numerous industries and data types whenever the requirement is to identify and isolate the unique intersections of attributes.

For those seeking to deepen their understanding of DAX capabilities, particularly regarding table manipulation and aggregation functions, the official documentation provides comprehensive details.

Note: You can find the complete documentation for the **SUMMARIZE** function in DAX [here](#).

Additional Resources

The following tutorials explain how to perform other common tasks in Power BI, complementing the skills learned in handling distinct values:

How to create complex measures in Power BI.

Techniques for optimizing data refresh schedules in Power BI Service.

Advanced filtering and slicer synchronization across multiple report pages.