

How to Concatenate Multiple Columns in Power BI

Authored by
mohammed looti

January 11, 2026

RECOMMENDED CITATION

mohammed looti (2026). *How to Concatenate Multiple Columns in Power BI*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=125578>

Concatenating columns in Power BI is a foundational data transformation technique that enables users to merge data from multiple sources or fields into a single, cohesive column. This process is essential when preparing data for analysis, particularly when creating composite keys, forming complete addresses, or generating user-friendly identifiers like a full name derived from separate first, middle, and last name fields. While the basic concatenation function in some tools limits combination to just two strings, Power BI offers robust solutions through DAX (Data Analysis Expressions) that allow you to combine far more than two columns effortlessly. Understanding how to handle multiple column concatenations efficiently ensures cleaner data models and more effective visualization results.

The need to combine three or more text fields is extremely common in real-world datasets. Imagine having employee data stored with granular separation--a best practice in database design--but requiring a consolidated view for reporting dashboards or filtering slicers. Attempting to manage multiple separate columns for a single logical entity (like a name) adds unnecessary complexity to report creation. By leveraging specific DAX operators, we can perform this multi-column merge directly within the Power BI data model, creating calculated columns that dynamically update as the underlying data changes. This approach maintains the integrity of the original granular columns while providing the required composite field for enhanced reporting capabilities.

While there are specialized functions like CONCATENATEX designed for iterative string merging, the most straightforward and performance-efficient method for combining a fixed, small number of columns (such as three, four, or five) is through the simple, yet powerful, ampersand operator (&). This article focuses specifically on how to harness the versatility of the & operator within DAX to handle the common scenario of concatenating more than two columns effectively, ensuring that proper separators are included for readability and consistency.

The Foundational Role of the Ampersand (&) Operator in DAX

The easiest and most common methodology used by Power BI professionals to concatenate more than two columns involves utilizing the **&** symbol within a DAX calculated column formula. Unlike languages that rely on functions like `CONCAT()` or `JOIN()`, DAX treats the ampersand as the primary operator for merging two text strings sequentially. The elegance of this method lies in its simplicity: you string together column references and necessary delimiters using repeated ampersands. Each ampersand acts as a concatenation point, instructing Power BI to link the text immediately preceding it with the text immediately following it.

When constructing a formula using the **&** operator, it is crucial to understand how to handle the separators, or delimiters. Since DAX only combines the elements you explicitly provide, you must manually insert spaces, hyphens, commas, or any other separator as a literal string within double quotes. For example, to join three columns, the formula requires four elements (Column A,

Delimiter 1, Column B, Delimiter 2, Column C) linked by four ampersands. This explicit inclusion of separators is vital; without them, the resulting concatenated column would merge all values directly into a single, unreadable block (e.g., "JohnMichaelSmith" instead of "John Michael Smith").

This method is executed entirely within a calculated column, meaning the operation is performed row-by-row across the entire table--a concept known in DAX as the Row Context. When Power BI evaluates the formula, it retrieves the specific value from the first column for the current row, joins it with the defined delimiter string, then joins that result with the value from the second column for the same row, and so on. This ensures that the final concatenated string correctly represents the merged data associated with that specific record. The resulting output is a new column in your data model ready for visualization and reporting purposes.

Syntax and Structure for Multi-Column Concatenation

The easiest way to concatenate more than 2 columns together in Power BI is to use the **&** symbol in DAX. This approach provides maximum control over the placement and type of separators used between the concatenated elements.

For instance, if you have three separate columns containing components of a name and you wish to combine them into a single **Full Name** column, you can use the following concise and highly readable syntax:

```
Full Name = 'my_data' & " " & 'my_data' & " " & 'my_data'
```

This particular example creates a new column named **Full Name** that concatenates the strings from the **First**, **Middle** and **Last** columns. Crucially, it uses the literal string " " (a single space enclosed in double quotes) as the separator between the individual string components. This explicit definition of the separator ensures that the resulting merged data is clean, formatted, and easily readable for end-users.

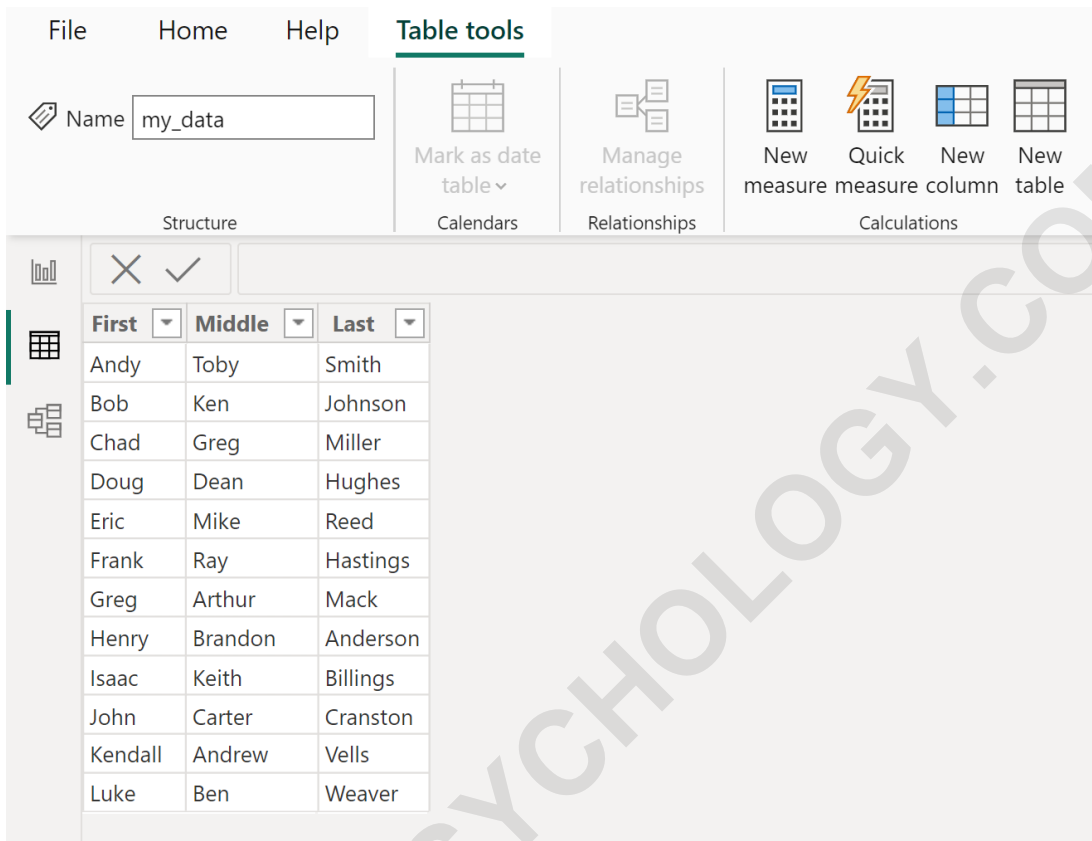
The following section provides a detailed, step-by-step example demonstrating how to implement this syntax within the Power BI desktop environment, transforming raw, separated data into a unified field.

Step-by-Step Example: Implementing the Calculated Column

To illustrate the application of the **&** operator for multi-column concatenation, let us consider a practical scenario. Suppose we are working with a table named **my_data** that contains granular details for personnel records, including the distinct columns for the first, middle, and last names. Our goal is to derive a single column that presents the complete name, properly formatted with

spaces, without altering the source columns.

Suppose we have the following table named **my_data** in Power BI, which represents a typical structure encountered during data import and preparation:

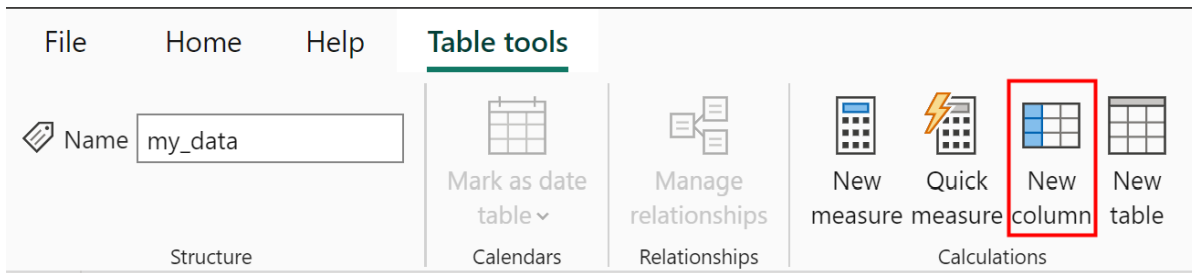


The screenshot displays the Power BI interface with the 'Table tools' ribbon selected. The ribbon includes options for 'Mark as date table', 'Manage relationships', and 'Calculations' (New measure, Quick measure, New column, New table). Below the ribbon, a table is visible with the following data:

First	Middle	Last
Andy	Toby	Smith
Bob	Ken	Johnson
Chad	Greg	Miller
Doug	Dean	Hughes
Eric	Mike	Reed
Frank	Ray	Hastings
Greg	Arthur	Mack
Henry	Brandon	Anderson
Isaac	Keith	Billings
John	Carter	Cranston
Kendall	Andrew	Vells
Luke	Ben	Weaver

Our immediate objective is to create a new derived column that successfully concatenates the strings from the **First**, **Middle**, and **Last** columns. This transformation requires navigating the Power BI interface to initiate the creation of a new calculated column and then entering the required DAX formula. This process ensures that the new column is an integral part of the data model and responds dynamically to changes in the source data.

To begin the process of creating this calculated field, you must first ensure you are focused on the correct table in the Data view or Report view. Then, click the **Table tools** tab located in the ribbon menu at the top of the Power BI Desktop application. Within this tab, locate and click the **New column** icon. This action immediately generates a formula bar, awaiting your DAX expression, and sets up the necessary Row Context for the calculation.



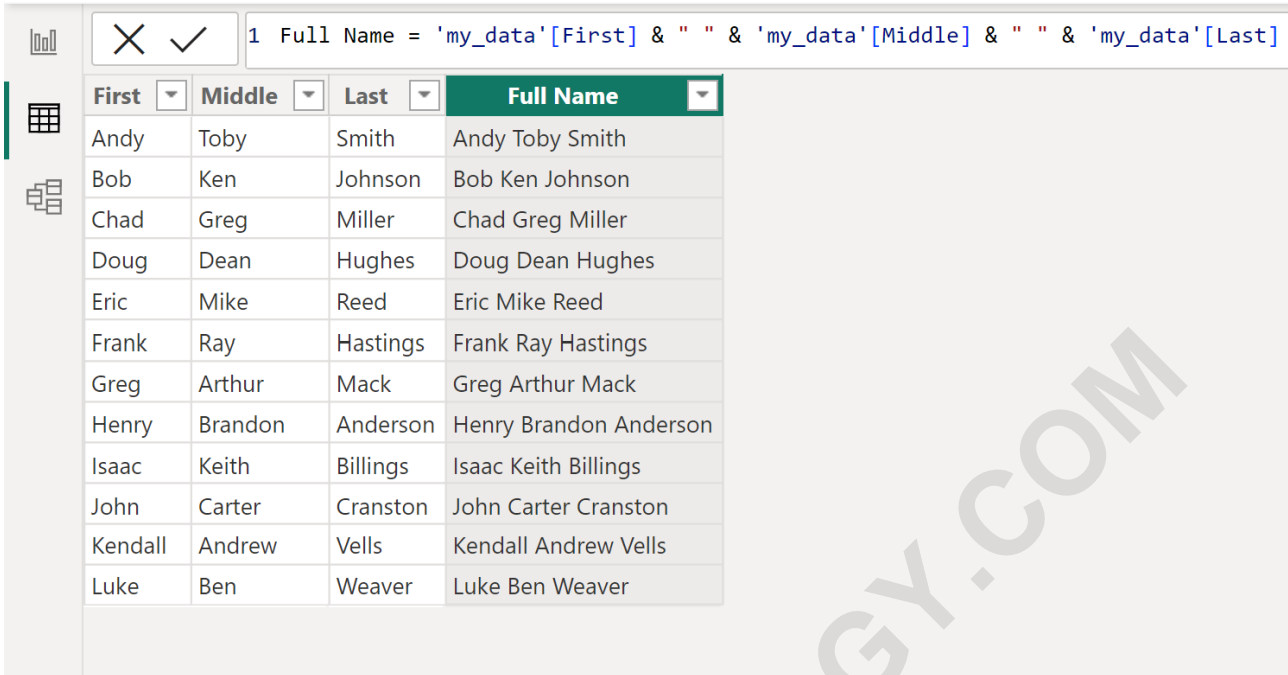
Practical Application: Creating the 'Full Name' Column

Once the formula bar is visible after clicking the **New column** button, we can input the specific DAX expression designed for multi-column concatenation. The structure of the formula must meticulously reference the table name, the column names (enclosed in square brackets), and the desired separators (enclosed in double quotes), all linked by the ampersand operator.

Then type the following formula into the formula bar. Note the strict sequence: column reference, ampersand, separator string, ampersand, next column reference, and so on. This formula precisely defines how the three separate text fields will be merged:

Full Name = 'my_data' & " " & 'my_data' & " " & 'my_data'

Upon confirming the formula (by pressing Enter or clicking the checkmark), Power BI calculates the resulting string for every row in the **my_data** table. This creates a new column named **Full Name** that successfully concatenates the strings from the **First**, **Middle** and **Last** columns, using spaces as the designated separator between the strings. The immediate visual result confirms the success of the operation, providing a clean, single field ready for use in reports.



The screenshot shows the DAX formula bar with the following formula: `1 Full Name = 'my_data'[First] & " " & 'my_data'[Middle] & " " & 'my_data'[Last]`. Below the formula bar is a table with four columns: First, Middle, Last, and Full Name. The Full Name column contains the concatenated values of the other three columns, separated by a space.

First	Middle	Last	Full Name
Andy	Toby	Smith	Andy Toby Smith
Bob	Ken	Johnson	Bob Ken Johnson
Chad	Greg	Miller	Chad Greg Miller
Doug	Dean	Hughes	Doug Dean Hughes
Eric	Mike	Reed	Eric Mike Reed
Frank	Ray	Hastings	Frank Ray Hastings
Greg	Arthur	Mack	Greg Arthur Mack
Henry	Brandon	Anderson	Henry Brandon Anderson
Isaac	Keith	Billings	Isaac Keith Billings
John	Carter	Cranston	John Carter Cranston
Kendall	Andrew	Vells	Kendall Andrew Vells
Luke	Ben	Weaver	Luke Ben Weaver

Customizing Delimiters and Handling Separators

A key advantage of using the **&** operator in DAX for concatenation is the ability to fully customize the delimiter used to separate the fields. While a space is often appropriate for names, many data requirements necessitate different separators, such as a comma and space (e.g., for address formatting), a hyphen, or a pipe symbol (e.g., for creating unique key identifiers).

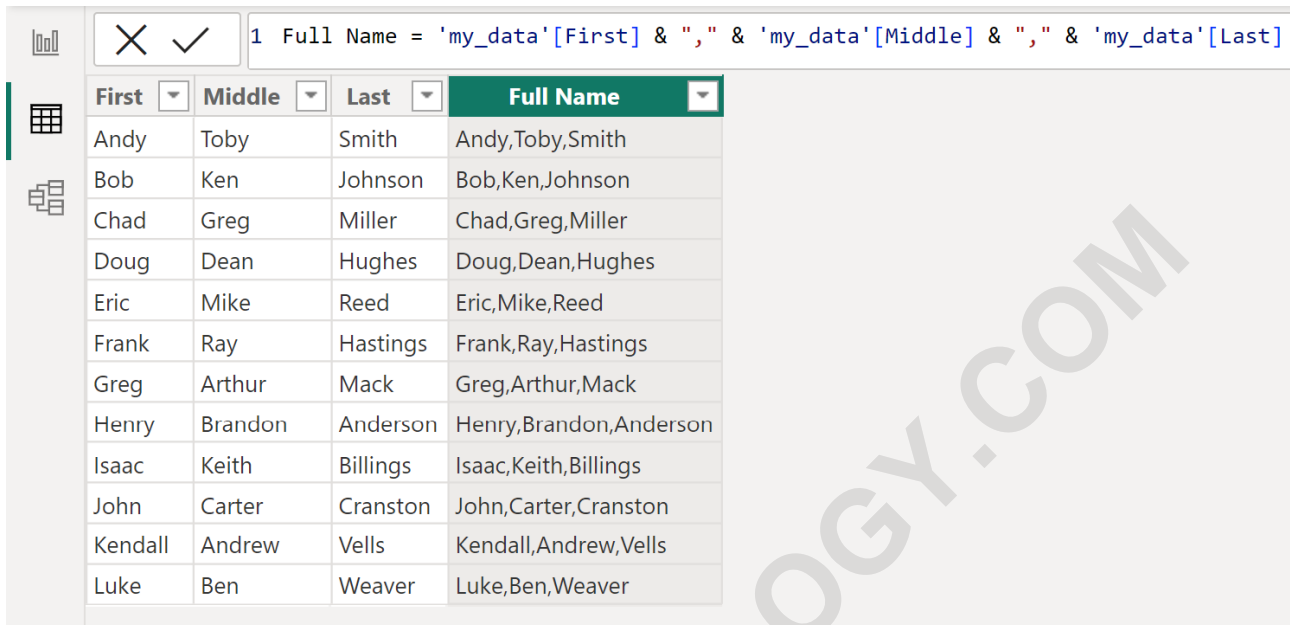
It is important to note that we chose to use a simple space as a separator in the previous example, but you could use any separator you'd like by explicitly defining the text string after the **&** symbols. The choice of separator should be governed by the intended use of the concatenated column. If the column is meant purely for display, readability is paramount. If, however, the column is intended to be used as a key or needs to be parsed by another system later, a non-standard character like a pipe (|) might be preferable, as it is less likely to appear within the actual data itself.

For example, suppose the requirement shifts, and the middle name should be separated from the first and last names by a comma, perhaps to align with specific academic or administrative formatting standards. You could easily modify the formula to replace the space delimiter " " with a comma and space ", ", or just a comma, as shown below:

Full Name = 'my_data' & ", " & 'my_data' & ", " & 'my_data'

Executing this adjusted formula results in a change in how the values are presented in the new column, demonstrating the flexibility of the DAX concatenation operator. This will concatenate the

values from the three columns with a comma placed between each string, providing a different textual presentation of the data:



The screenshot shows the Power BI interface. At the top, the DAX formula bar contains the following formula: `1 Full Name = 'my_data'[First] & "," & 'my_data'[Middle] & "," & 'my_data'[Last]`. Below the formula bar is a table with four columns: 'First', 'Middle', 'Last', and 'Full Name'. The 'Full Name' column contains the concatenated values of the first three columns, separated by commas.

First	Middle	Last	Full Name
Andy	Toby	Smith	Andy,Toby,Smith
Bob	Ken	Johnson	Bob,Ken,Johnson
Chad	Greg	Miller	Chad,Greg,Miller
Doug	Dean	Hughes	Doug,Dean,Hughes
Eric	Mike	Reed	Eric,Mike,Reed
Frank	Ray	Hastings	Frank,Ray,Hastings
Greg	Arthur	Mack	Greg,Arthur,Mack
Henry	Brandon	Anderson	Henry,Brandon,Anderson
Isaac	Keith	Billings	Isaac,Keith,Billings
John	Carter	Cranston	John,Carter,Cranston
Kendall	Andrew	Vells	Kendall,Andrew,Vells
Luke	Ben	Weaver	Luke,Ben,Weaver

Scaling Concatenation: Beyond Three Columns

The principles demonstrated for three columns extend seamlessly to any arbitrary number of columns, making the `&` operator highly scalable for fixed concatenation tasks. If you needed to combine six different columns--perhaps a complex product code consisting of category, size, color, year, region, and ID--you would simply continue the pattern: column reference followed by ampersand, followed by the delimiter string, and so on, until all required fields are included.

Note: In this example, we concatenated three columns together, but you can use the `&` symbol as many times as you'd like to concatenate even more columns together. The primary constraint, in this case, is not the DAX functionality, but rather the complexity and manageability of the formula itself. For very long concatenation chains (e.g., more than eight or ten components), the formula can become lengthy and prone to manual error, which might prompt consideration of alternative iterative functions.

A crucial consideration when dealing with multi-column concatenation is handling potential blank or null values in the source columns. If the middle name column, for instance, is blank for certain records, the formula `'my_data' & " " & 'my_data' & " " & 'my_data'` would result in an extra space appearing where the middle name should be (e.g., "John Smith"). To prevent this, advanced DAX logic involving functions like `IF` and `ISBLANK` or `COALESCE` is often integrated into the calculation to conditionally include the delimiter only if the intermediate column has a value. This attention to detail ensures the resulting concatenated string is perfectly formatted regardless of

data completeness.

Alternative Approach: Leveraging CONCATENATEX for Iterative Merging

While the **&** operator is ideal for fixed, row-by-row merging of individual columns, situations may arise where you need to concatenate multiple strings across different rows within a group (e.g., listing all cities visited by a customer). For these scenarios, the CONCATENATEX function provides a more powerful, iterative solution.

DAX's CONCATENATEX is an iterator function that evaluates a string expression for every row of a specified table and then combines the resulting strings into a single text value, using a defined delimiter. This function is typically used in measures rather than calculated columns and excels when summarizing text data based on relationships in your data model. For example, you could use CONCATENATEX to create a single field on a Customer table that lists all product categories that customer has purchased, separated by commas.

The key difference is context. The **&** operator works purely within the Row Context to merge fixed columns for that single record. CONCATENATEX works within a Filter Context, allowing it to aggregate strings from related rows into a single cell, thereby offering a solution for complex grouping and summary tasks that simple concatenation cannot address. Although overkill for merging three simple name columns, understanding CONCATENATEX provides a comprehensive toolset for all types of text merging challenges in Power BI.

Summary of Best Practices for Column Concatenation

Successfully implementing multi-column concatenation enhances data clarity and simplifies reporting in Power BI. To ensure accuracy and maintainability, adhere to the following best practices:

Always Use Explicit Delimiters: Never rely on natural spacing; always insert a literal string (e.g., " ", ", ", " ", or "| ") using the **&** operator to maintain readability and structure in the resulting concatenated column. The choice of delimiter should be consistent across similar data types.

Proactively Handle Missing Data: Implement DAX logic (such as nested **IF** statements checking **ISBLANK** or utilizing specialized functions) to suppress delimiters or entire segments of the concatenation formula if the source column value is null or empty. This prevents unsightly double separators (e.g., "Name , , ID") in your final output.

Use Descriptive Column Names: Name the new calculated column clearly (e.g., "Full_Address," "Composite_Key," or "Formatted_Name") to differentiate it from the original granular source columns. This practice is crucial for data governance and clarity when other report developers

interact with the data model.

By mastering the use of the **&** operator for fixed concatenation and understanding the iterative power of CONCATENATEX, Power BI users can efficiently transform raw data into optimized structures, significantly improving the analytical and visualization capabilities of their reports.

The following tutorials explain how to perform other common tasks in Power BI:

ARABPSYCHOLOGY.COM