

# How to Combine Columns in Power BI for a Unified View

Authored by  
**stats writer**

January 28, 2026

## RECOMMENDED CITATION

stats writer (2026). *How to Combine Columns in Power BI for a Unified View*.  
PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=128118>

In the realm of business intelligence and data visualization, effective data manipulation is paramount. One of the most fundamental requirements when preparing raw data for visualization in Power BI is Concatenation, which refers to the process of combining two or more distinct text strings or columns into a single, cohesive entity. This operation is crucial for streamlining reporting and enhancing data clarity. For instance, analysts frequently need to combine separate fields like first name and last name into a unified full name column, or merge product codes and category identifiers into a single identifier field. This transformation not only simplifies the end-user experience when interacting with reports and slicers but also optimizes the underlying structure of the data model, minimizing complexity in subsequent calculations and filtering operations.

The central method for achieving column combination in Power BI involves creating a new calculated column utilizing the powerful Data Analysis Expressions (DAX) language. Unlike data transformations performed in Power Query (M language), creating a DAX calculated column ensures that the concatenated value exists directly within the table structure of your report, allowing it to react dynamically to any applied filters or context changes during runtime. This guide provides a comprehensive overview of mastering column concatenation using DAX, detailing both the basic operations using the ampersand (&) operator and the dedicated CONCATENATE function, offering robust methods for data organization and reporting efficiency.

By mastering the proper steps and syntax, you can easily concatenate columns and create a new, consolidated field in your Power BI report. The choice between the ampersand operator and the CONCATENATE function often comes down to the complexity of the merge; however, the ampersand (&) is generally preferred for its flexibility in incorporating separators and merging multiple strings simultaneously. Understanding these techniques is a foundational skill that leads to superior data governance and more reliable business insights derived from clean, well-structured data.

## Concatenating Columns in Power BI: Methods and Syntax

### Why Column Concatenation is Essential for Data Modeling

In data management, source data is frequently normalized, meaning information is meticulously separated across numerous columns to reduce redundancy and maintain data integrity. While this structure is ideal for database storage and transactional efficiency, it presents difficulties when the data must be presented visually or summarized in reports. For example, rather than displaying 'First Name' and 'Last Name' separately in a table visual or a slicer, combining them into a single 'Full Name' field drastically improves the clarity and navigability of the report for the end-user. Concatenation acts as the necessary translation layer between optimal data storage structure and optimal reporting presentation.

Furthermore, concatenation is often indispensable for generating **unique composite keys** or identifiers that may be missing in the original source data. When working with large data sets, ensuring data uniqueness, especially across different dimensions or tables, can be challenging. By combining two or three existing identifying fields (e.g., combining 'Store ID' and 'Product Code' to form a 'Unique Sales Item ID'), analysts can create a robust, single-column identifier. This new field can then be used effectively for defining relationships or performing accurate distinct counts within the Power BI [data model](#).

Ultimately, performing concatenation improves the readability and usability of reports. By condensing related information into a single field, analysts streamline the process of building visualizations and reduce the complexity required for filtering and sorting. This moves the burden of combining disparate data elements from the report consumer back to the data analyst during the preparation phase, ensuring a cleaner, more efficient reporting environment.

## Understanding the Primary Methods: Ampersand (&) vs. CONCATENATE Function

Within [DAX](#), data analysts have two principal tools for achieving string concatenation: the versatile **ampersand operator (&)** and the dedicated [CONCATENATE](#) function. Although both achieve the same fundamental result--combining text strings--they differ significantly in their application and flexibility. The ampersand operator is a binary arithmetic operator specifically repurposed for string merging. Its primary advantage lies in its simplicity for chaining multiple strings, allowing for complex joins involving many columns and separators in a single, easily readable expression (e.g., `& " - " & &` ).

The [CONCATENATE](#) function, conversely, is a traditional function that historically required exactly two arguments (text strings or column references). While it performs well for basic, two-column joins, merging three or more columns requires nesting the function, leading to formulas that are quickly convoluted and difficult to maintain (e.g., `CONCATENATE( , CONCATENATE( , ) )`). This structural limitation makes the [CONCATENATE](#) function less practical for typical real-world concatenation tasks which often require incorporating custom separators.

Given the restrictions and structural complexity imposed by nesting the [CONCATENATE](#) function, industry experts overwhelmingly favor the **ampersand operator (&)** for all forms of string concatenation in DAX calculated columns. The ampersand not only handles complex multi-part merges efficiently but also allows for the seamless inclusion of literal text separators (such as spaces, dashes, or pipes) directly into the formula chain, which is essential for creating well-formatted output strings.

## Deep Dive into the DAX Syntax for Concatenation

When defining a new calculated column in Power BI Desktop, you initiate the process via the 'Table tools' tab. The syntax used in the formula bar must precisely reference the columns you wish to join. For illustrative purposes, we will use a table named **my\_data** containing the columns we wish to merge.

The following formulas demonstrate the two ways to execute the concatenation operation in DAX. It is important to remember that DAX automatically handles the conversion of numerical or date types to text strings during concatenation, although explicit formatting is often recommended for precision.

### Formula 1: Concatenate Two Columns with No Separator

This approach uses the `CONCATENATE` function to directly merge the values from 'Column 1' and 'Column 2' without any intervening characters. This is suitable for combining identifiers that are meant to flow together seamlessly, like product codes:

**New Column = `CONCATENATE('my_data', 'my_data')`**

This particular formula concatenates the strings in Column 1 and Column 2 together with no separator between them.

### Formula 2: Concatenate Two Columns with Separator using Ampersand (&)

This formula leverages the preferred ampersand operator to combine 'Column 1', a space separator (" "), and 'Column 2'. This syntax is cleaner, more scalable, and highly recommended when readability is paramount:

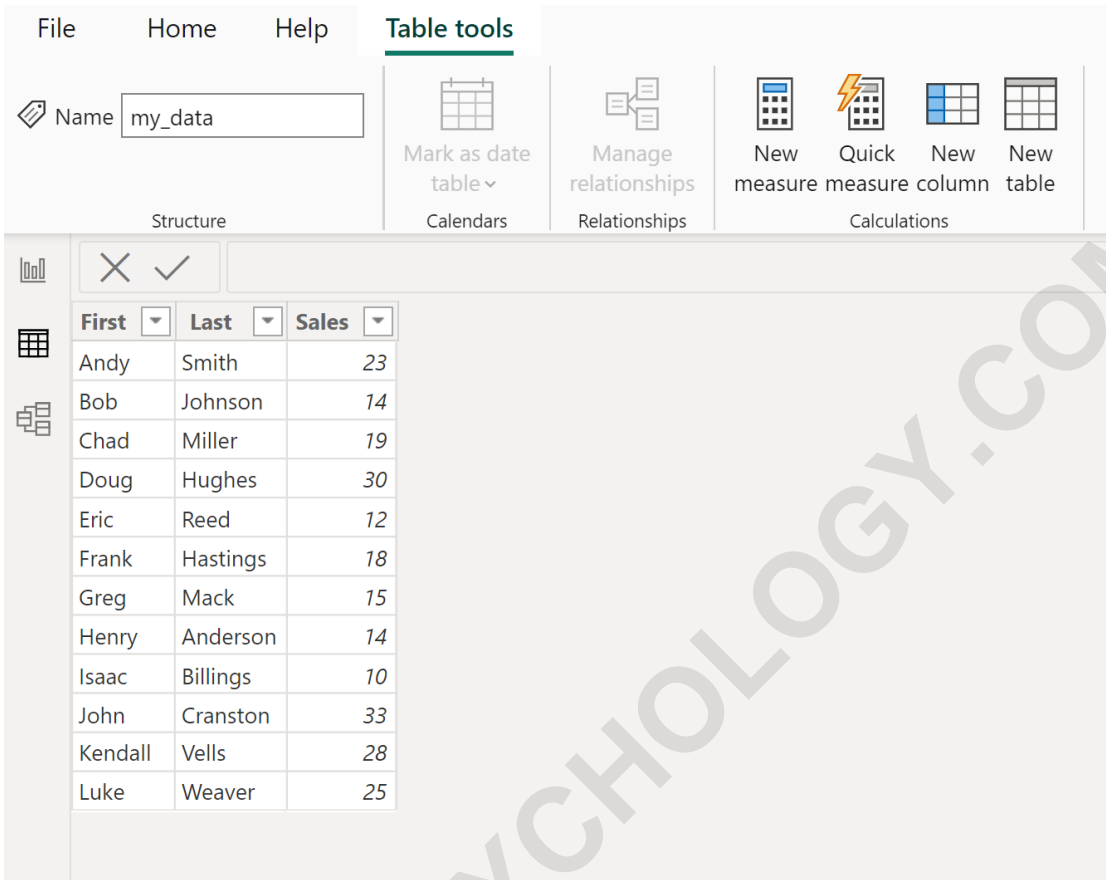
**New Column = `'my_data' & " " & 'my_data'`**

This particular formula uses the `&` symbol repeatedly: first to join the first column and the space literal, and then to join that result with the second column. This explicit inclusion of the separator string (" ") is what differentiates readable output from simply merged characters.

## Practical Example 1: Concatenating Columns Without a Separator

To demonstrate these formulas in action, we will use a sample table named **my\_data** that includes 'ID', 'First', and 'Last' columns. Our immediate objective is to create a new field by merging the 'First' and 'Last' columns directly, generating a unique identifier that lacks a separator.

The source table **my\_data**, which provides the input for our calculation, is structured as follows in the Power BI environment:



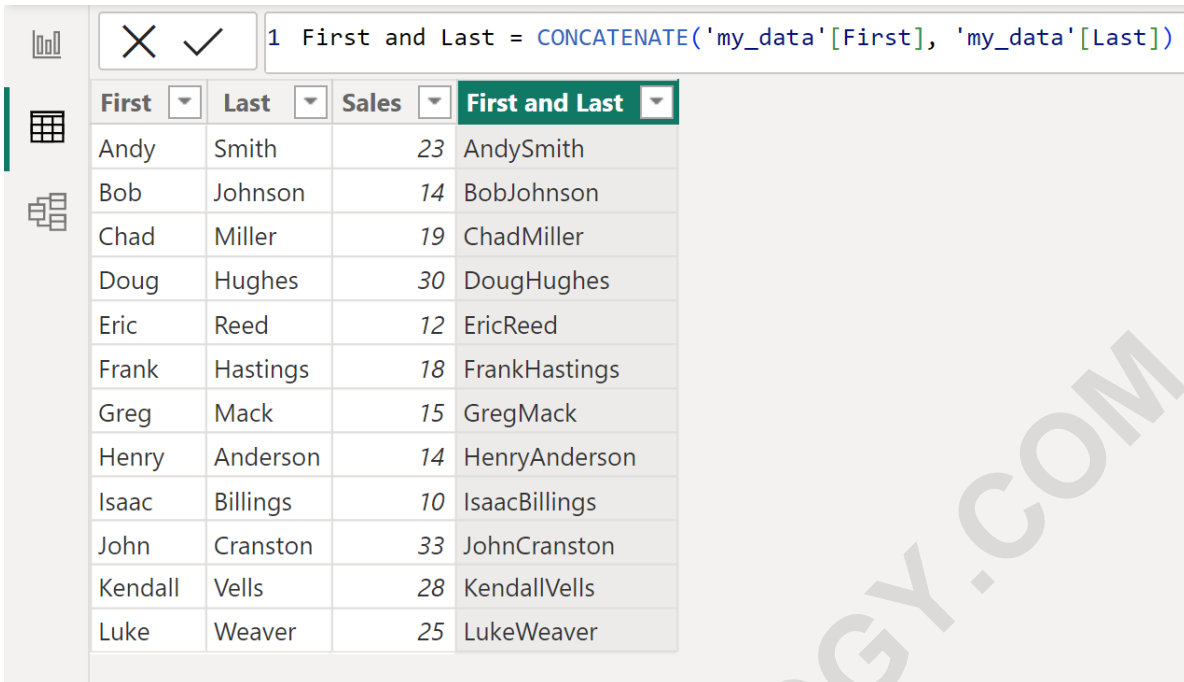
First	Last	Sales
Andy	Smith	23
Bob	Johnson	14
Chad	Miller	19
Doug	Hughes	30
Eric	Reed	12
Frank	Hastings	18
Greg	Mack	15
Henry	Anderson	14
Isaac	Billings	10
John	Cranston	33
Kendall	Vells	28
Luke	Weaver	25

To perform this operation, first navigate to the **Table tools** tab in the Power BI ribbon. Click the **New column** icon to open the DAX formula bar. We will use the **CONCATENATE** function here to illustrate its utility for basic, two-string mergers. We must ensure we fully qualify the column names by including the table reference ('my\_data').

The syntax for creating the unseparated combined column is:

**First and Last = CONCATENATE('my\_data', 'my\_data')**

Executing this formula creates a new calculated column named **First and Last** where the contents of the 'First' column are immediately followed by the contents of the 'Last' column. Notice how the resulting string is one continuous block of text, ideal for machine reading or internal IDs, but less so for human readability:



The screenshot shows the Power BI interface with a DAX formula bar at the top. The formula is: `1 First and Last = CONCATENATE('my_data'[First], 'my_data'[Last])`. Below the formula bar is a table with four columns: 'First', 'Last', 'Sales', and 'First and Last'. The 'First and Last' column contains the concatenated values of the 'First' and 'Last' columns without a separator.

First	Last	Sales	First and Last
Andy	Smith	23	AndySmith
Bob	Johnson	14	BobJohnson
Chad	Miller	19	ChadMiller
Doug	Hughes	30	DougHughes
Eric	Reed	12	EricReed
Frank	Hastings	18	FrankHastings
Greg	Mack	15	GregMack
Henry	Anderson	14	HenryAnderson
Isaac	Billings	10	IsaacBillings
John	Cranston	33	JohnCranston
Kendall	Vells	28	KendallVells
Luke	Weaver	25	LukeWeaver

## Practical Example 2: Implementing a Custom Separator

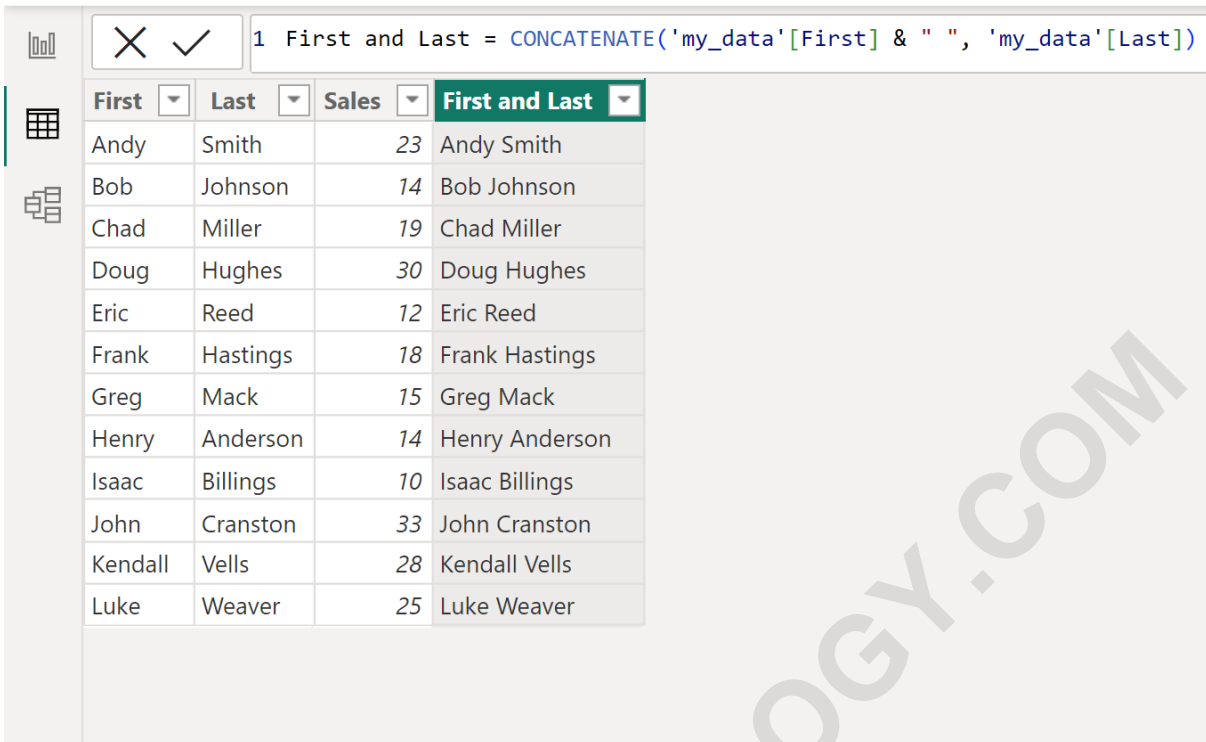
In most user-facing reports, concatenation requires a separator to ensure the combined data is legible. Using the same **my\_data** table, we will now combine the 'First' and 'Last' names using a standard space separator, which is necessary to create a professional "Full Name" field. This process requires using the powerful ampersand operator (&).

Start by clicking the **Table tools** tab, then clicking the **New column** icon, just as before. However, the **DAX** formula will now integrate the literal space string (" ") between the two column references. The ampersand operator is instrumental in binding these three distinct elements together efficiently:

The required formula using the space separator is:

**First and Last = 'my\_data' & " " & 'my\_data'**

This calculation results in a new column where the names are properly delimited by a space, significantly enhancing the readability and suitability of the field for use in visualizations and filters within your **Power BI** report. The image below confirms the successful implementation of the space separator:



The screenshot shows a Power BI interface with a DAX formula bar at the top and a table below. The formula bar contains the following formula: `1 First and Last = CONCATENATE('my_data'[First] & " ", 'my_data'[Last])`. The table has four columns: 'First', 'Last', 'Sales', and 'First and Last'. The 'First and Last' column contains the concatenated values of the 'First' and 'Last' columns, separated by a space.

First	Last	Sales	First and Last
Andy	Smith	23	Andy Smith
Bob	Johnson	14	Bob Johnson
Chad	Miller	19	Chad Miller
Doug	Hughes	30	Doug Hughes
Eric	Reed	12	Eric Reed
Frank	Hastings	18	Frank Hastings
Greg	Mack	15	Greg Mack
Henry	Anderson	14	Henry Anderson
Isaac	Billings	10	Isaac Billings
John	Cranston	33	John Cranston
Kendall	Vells	28	Kendall Vells
Luke	Weaver	25	Luke Weaver

It is crucial to understand that the separator string can be anything enclosed in double quotes. For example, if you wanted to format the output as "Last Name, First Name," the formula would be modified to `'my_data' & ", " & 'my_data'`. This demonstrates the immense flexibility offered by the ampersand operator in defining custom output formats for Concatenation.

### Advanced Techniques: Using Different Delimiters

While a space is common, many specialized applications require non-standard delimiters. For instance, creating internal system identifiers, file names, or URL slugs often necessitates using symbols like dashes or underscores to maintain data integrity across various platforms. The simplicity of the ampersand operator makes swapping delimiters an effortless task.

For example, if the requirement is to use a dash (hyphen) as the separator for a structured identifier, the literal string in the formula is simply changed from `" "` to `"-"`:

**First and Last = 'my\_data' & "-" & 'my\_data'**

This adjustment immediately changes the output format, resulting in values joined by a dash:



The screenshot shows the DAX editor interface. At the top, the formula bar contains the following DAX formula:

```
1 First and Last = CONCATENATE('my_data'[First] & "-", 'my_data'[Last])
```

Below the formula bar, a table is displayed with the following columns: First, Last, Sales, and First and Last. The data rows are as follows:

First	Last	Sales	First and Last
Andy	Smith	23	Andy-Smith
Bob	Johnson	14	Bob-Johnson
Chad	Miller	19	Chad-Miller
Doug	Hughes	30	Doug-Hughes
Eric	Reed	12	Eric-Reed
Frank	Hastings	18	Frank-Hastings
Greg	Mack	15	Greg-Mack
Henry	Anderson	14	Henry-Anderson
Isaac	Billings	10	Isaac-Billings
John	Cranston	33	John-Cranston
Kendall	Vells	28	Kendall-Vells
Luke	Weaver	25	Luke-Weaver

Beyond simple delimiter changes, advanced concatenation often involves handling missing data. If one of the source columns contains a BLANK or NULL value, the simple concatenation formula may produce unintended results, such as leading or trailing separators (e.g., if the middle name is blank, `& " " & & " " &` might result in a double space or a trailing space). To mitigate this, best practice involves using conditional logic, such as the `IF` function, to only include the separator if the subsequent column is not blank. This ensures that the final concatenated string is always clean and correctly formatted, regardless of the completeness of the source data.

## Summary and Next Steps in Power BI Data Transformation

Column Concatenation is a cornerstone of data preparation in Power BI, serving vital functions in enhancing report readability and optimizing the underlying data model. We have established that for most practical applications, the **ampersand (&)** operator offers superior flexibility and readability compared to the dedicated `CONCATENATE` function, especially when custom separators or multi-column joins are required.

By following the demonstrated procedures--creating a new calculated column and employing the appropriate `DAX` syntax--you can efficiently transform segmented data into consolidated, highly usable fields. Whether your goal is generating a simple unseparated code or a complex, properly spaced full name, mastering these techniques ensures data integrity and user satisfaction within your reports.

Mastering concatenation is merely the beginning of your journey in advanced data transformation within Power BI. To further enhance your data preparation skills, consider exploring related topics:

Using the **CONCATENATEX** function for iterative concatenation across rows or groups, typically used within measures.

Employing Power Query (M Language) transformations to perform concatenation before data loading, which can sometimes improve performance by reducing the number of calculated columns.

Techniques for splitting concatenated columns back into multiple fields, reversing the process when necessary.

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

ARABPSYCHOLOGY.COM