Learn How to Concatenate Multiple Columns in Power BI Using DAX

Authored by

Mohammed looti

November 12, 2025

One of the most frequent requirements in data preparation and modeling is the ability to combine textual information from multiple fields into a single, cohesive string. In Power BI, this process, known as concatenation, is essential for tasks such as creating full names, standardized addresses, or unique identifiers. While the standard `CONCATENATE` function in DAX is limited to combining only two strings, the most straightforward and versatile method for combining more than two columns is leveraging the powerful ampersand (**&**) operator within a Calculated Column. This approach offers superior flexibility and readability when dealing with complex string combinations.

The **&** symbol acts as a dedicated string concatenation operator in Data Analysis Expressions (DAX), allowing developers to link an unlimited number of column references and literal strings together seamlessly. This capability is paramount when modeling dimensional data where multiple attributes must be unified for reporting purposes.

Utilizing the Ampersand Operator (&) for Multi-Column Concatenation in DAX

The **&** operator provides an intuitive and scalable solution for joining strings in Power BI. Unlike the restrictive `CONCATENATE` function, the ampersand allows for chaining operations, meaning you can link Column A, a separator, Column B, another separator, and Column C, all within a single, concise formula. This method is preferred by experienced Power BI developers due to its clarity and directness when defining new attributes in the data model.

For example, if the goal is to create a complete name from three distinct fields--First, Middle, and Last--while ensuring a space separates each component, the syntax is clear and expressive. This flexibility extends beyond simple spaces; developers can insert any required literal text, special characters, or symbols (such as hyphens, commas, or parentheses) directly into the concatenation sequence by enclosing them in double quotes.

You can use the following syntax structure to combine three columns and their necessary separators effectively:

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

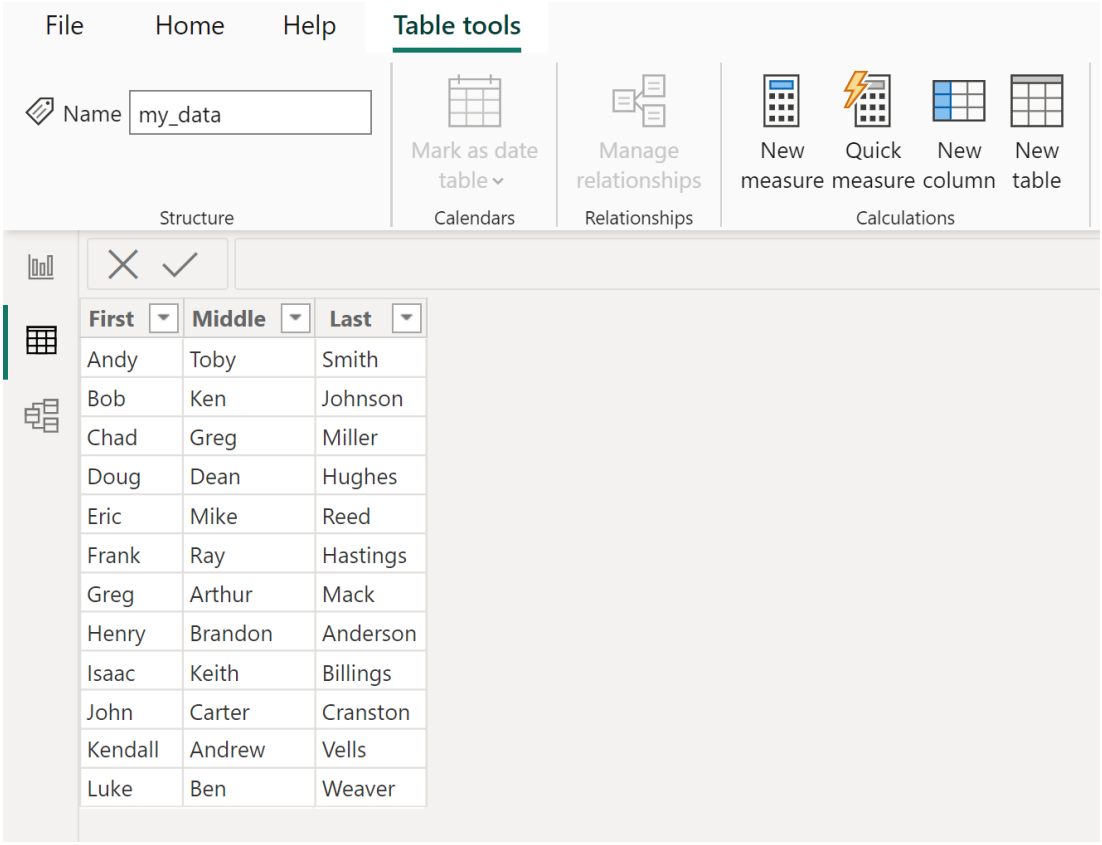
This specific formula generates a new attribute named **Full Name**. It methodically combines the text values retrieved from the **First**, **Middle**, and **Last** columns of the table named **my_data**, using a single space (" ") as the designated separator placed between the individual string components. Understanding this basic structure is the foundation for performing any multi-column concatenation operation within the DAX environment.

Step-by-Step Implementation: Creating a Calculated Column

To demonstrate the practical application of the **&** operator, let us walk through the process using a

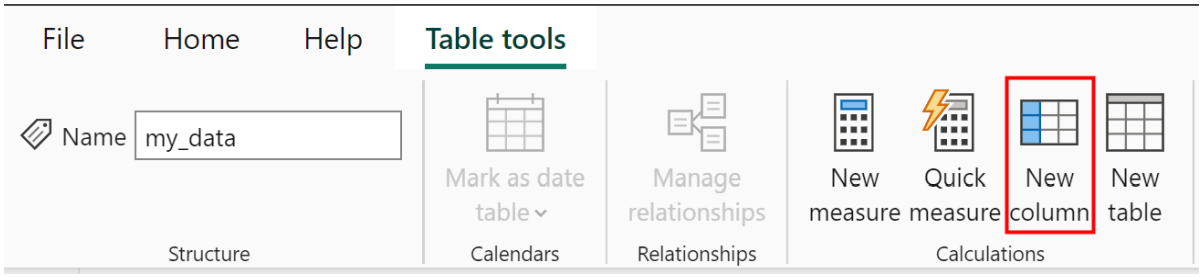
sample dataset. We will assume we are working with a table named **my_data** which contains segregated name components. The objective is to merge these components into a single, user-friendly field that can be used effectively in visualizations and reports. This process requires creating a Calculated Column, as string manipulation resulting in a new physical column is best handled at the data model layer rather than in a visual measure.

Suppose we begin with the following raw data table structure defined within Power BI:



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

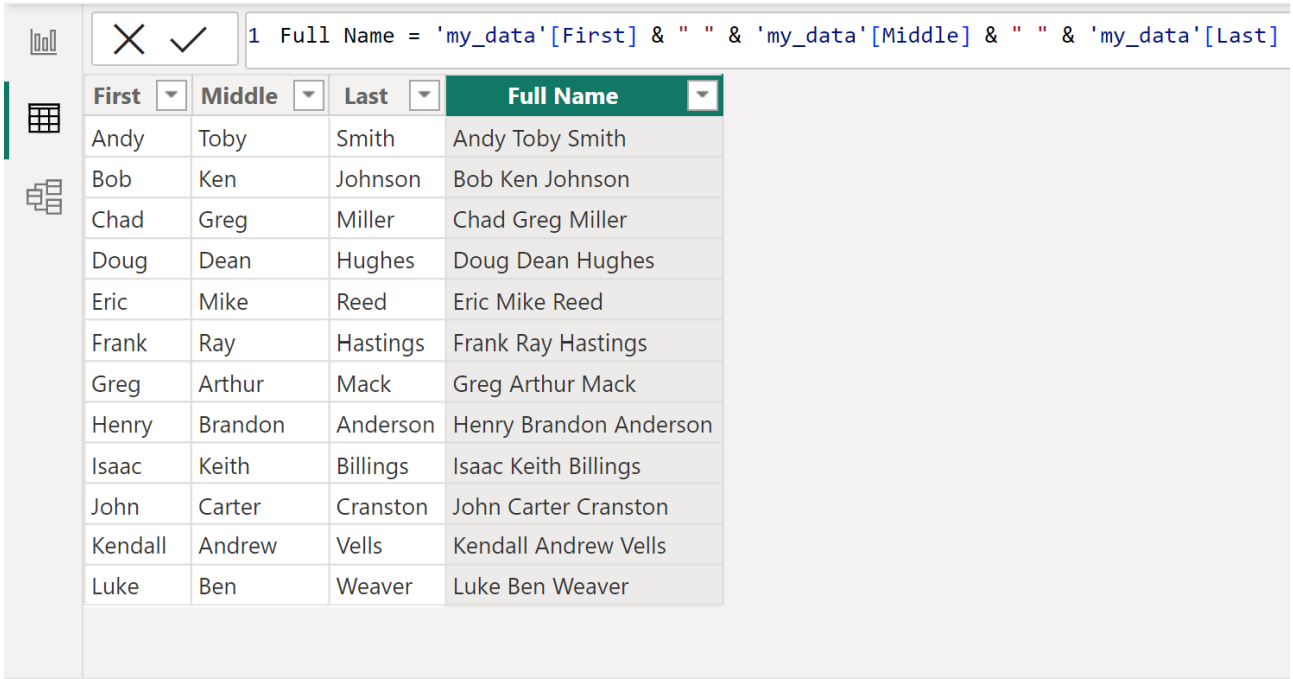
Our goal is to create a comprehensive column that joins the strings from the **First**, **Middle**, and **Last** columns. To initiate the creation of this new column, navigate to the **Table tools** tab within the Power BI interface while viewing the data model. Once there, locate and click the **New column** icon. This action opens the DAX formula bar, prompting us to input the desired expression for the new column.



In the newly opened formula bar, input the DAX formula defined earlier. Remember that column references must be enclosed in square brackets `` and preceded by the table name, enclosed in single quotes `' '`, ensuring that the formula correctly references the source data fields for each row calculation. The complete formula should be entered precisely as follows:

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

Upon confirming the formula, Power BI calculates the new column row-by-row. This process yields the desired attribute named **Full Name**, which successfully combines the strings from the **First**, **Middle**, and **Last** columns, utilizing spaces as separators to ensure the resulting text is formatted correctly and is easily readable, demonstrating the power and simplicity of the **&** operator for concatenation.



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 Separators and Handling Data Nuances

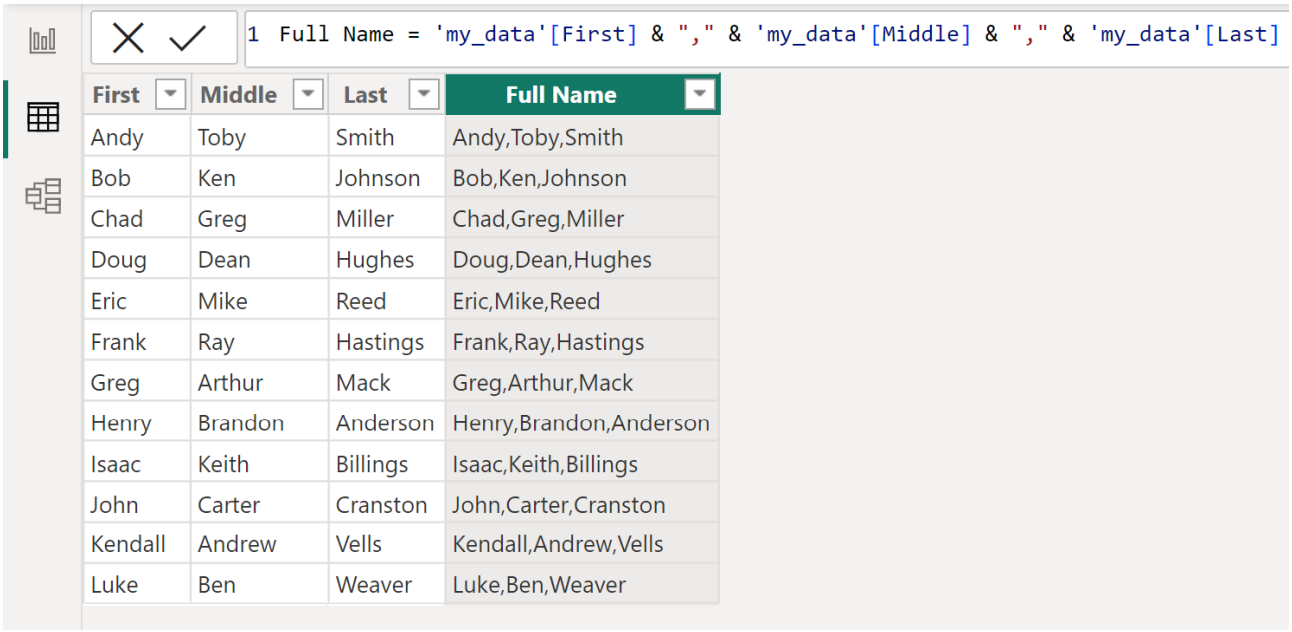
A key advantage of using the **&** operator is the absolute control it grants over the separators used between concatenated strings. While the example above utilized a simple space, any string literal can be inserted, allowing for highly specific formatting requirements. This is particularly useful when creating standardized keys or when outputting data intended for external systems that require specific delimiters.

For instance, if the business requirement dictates that the name components must be separated by a comma and a space--a common format for listing names (e.g., "Doe, John, M.")--the formula can be easily modified. The separator string enclosed in double quotes is simply replaced with the desired delimiter, which in this case would be `", "`. This small adjustment significantly alters the presentation of the final aggregated string without requiring complex logical functions.

To illustrate using a comma and space as a separator, the corresponding DAX syntax would look like this:

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

Executing this revised formula will concatenate the values from the three source columns, inserting the specified comma and space delimiter between each string segment. This results in an output that adheres strictly to the required comma-separated format, demonstrating the granular control over string assembly provided by the **&** operator in the DAX language.



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

Scalability and Best Practices for Concatenation

A major strength of the **&** operator is its inherent scalability. Although the examples focused on concatenating three columns, this operator can be chained indefinitely to combine any number of columns necessary for the data model. If a table had five or ten relevant text fields, the developer would simply continue adding the column reference followed by the **&** operator and the required separator string until all components are included. This makes it the ideal tool for simple, direct string combinations in Power BI.

However, it is crucial to consider data quality when using direct concatenation. If any of the source columns--such as `` in our example--contain a **BLANK** or **NULL** value, the direct use of the **&** operator will typically treat the blank as an empty string (""). While this is often acceptable, it can lead to redundant separators if not managed. For instance, if `` is null, the result might look like "John Doe" (two spaces) or "John, , Doe" (two commas). For scenarios where robust null handling is required, particularly with large numbers of columns where nulls are common, developers may need to embed conditional logic (e.g., using `IF` or `ISBLANK`) around the separators to ensure they only appear if the adjacent column contains a value.

For significantly more complex concatenation tasks, particularly those involving iterative assembly or handling long lists of attributes with conditional formatting, alternatives like the `CONCATENATEX` function might be explored. However, for the vast majority of standard data modeling needs--such as combining First/Middle/Last names, or address components--the use of the simple, clean, and highly performant **&** operator remains the gold standard within DAX for creating well-formed Calculated Column values.

Additional Resources for Advanced DAX String Operations

Mastering string manipulation is fundamental to effective data preparation in Power BI. The following tutorials provide further explanation on how to perform other common and advanced tasks within the DAX environment, building upon the foundational knowledge of simple concatenation:

How to handle conditional logic when concatenating fields with potential nulls.
Exploring the `CONCATENATEX` function for iterating over related tables.
Methods for extracting substrings (LEFT, RIGHT, MID) for complex parsing needs.