

Learn How to Combine Data from Multiple Columns into One in Excel

Authored by
Mohammed looti

November 10, 2025

RECOMMENDED CITATION

Mohammed looti (2025). *Learn How to Combine Data from Multiple Columns into One in Excel*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=15468>

Microsoft [Excel](#) is an indispensable tool for data management, but often raw data needs consolidation before it can be effectively analyzed or presented. Combining multiple columns--such as first name, middle initial, and last name, or product code and location identifier--into a single, cohesive string is a fundamental requirement for creating unique identifiers or generating cleaner reports. Fortunately, Excel offers two robust and highly efficient methods for achieving this column consolidation: the [CONCAT Function](#) and the [TEXTJOIN Function](#). Understanding the subtle yet crucial differences between these two functions will allow you to select the optimal approach based on whether your final output requires separators, such as spaces or dashes, between the combined values.

Overview of Column Combination Methods

When faced with the task of merging data from three separate columns into one, users typically have to decide whether they need a seamless string (e.g., "JohnDoe123") or a readable, delimited string (e.g., "John Doe 123"). The choice between the available Excel functions hinges entirely on this requirement. While older versions of Excel relied heavily on the ampersand (&) operator or the now-deprecated [CONCATENATE](#) function, modern Excel provides streamlined solutions that handle ranges of [cell references](#) far more efficiently, making the process faster and less error-prone, especially when dealing with large datasets.

The first primary method, utilizing the [CONCAT Function](#), is the simplest choice for direct merging. This function takes a range of cells and joins them together sequentially without inserting any characters between the combined strings. This method is ideal when generating condensed data, such as product keys, database primary keys, or internal tracking identifiers where spaces or separators are forbidden or unnecessary.

The second, and often more flexible, method involves the [TEXTJOIN Function](#). This powerful formula not only merges the content of the specified cells but also allows the user to define a specific [delimiter](#) (separator) to be placed between each cell's value. Furthermore, it includes an essential argument that dictates whether blank cells should be ignored during the concatenation process, a feature highly valued in cleaning messy, incomplete datasets. Both methods leverage the ability to reference a range of cells, such as **A2:C2**, which significantly simplifies the formula structure compared to older techniques.

Method 1: Using the CONCAT Function for Seamless Merging

The [CONCAT function](#) is designed for straightforward, rapid combination of text strings or ranges. It requires only the range of cells you wish to merge. When combining columns **A**, **B**, and **C** starting in row 2, the syntax is remarkably concise, focusing purely on the source data. This simplicity makes it the fastest formula to write when delimiters are not required, directly addressing

the core need to combine values from multiple locations into a single output cell without any intermediate characters.

To merge the values residing in cells **A2**, **B2**, and **C2** into a single cell, typically **D2**, you would implement the following formula. Note that the output will be a single continuous string, as the function does not insert spaces or any other form of separation.

=CONCAT(A2:C2)

This particular formula will combine the values extracted from the specified range, **A2** through **C2**, into one single cell with absolutely no spaces or intervening characters between the concatenated strings. If **A2** contains "Smith", **B2** contains "J", and **C2** contains "101", the resulting output in **D2** would be "SmithJ101". This function is particularly useful for technical data aggregation where a non-spaced identifier is mandatory, ensuring the combined data remains compact and structurally uniform across the entire dataset.

Example 1: Combine Three Columns into One Using CONCAT

Let us consider a practical scenario where we have a dataset detailing employee sales, separated by Employee ID (Column A), Product Code (Column B), and Sales Region (Column C). Our goal is to create a unique, non-spaced tracking key in Column D by combining these three pieces of information using the **CONCAT** function. This example clearly demonstrates the utility of **CONCAT** for generating compact identifiers, which are often necessary for database lookups or internal reporting systems where space characters can cause errors.

We begin by setting up our initial dataset. The following visualization illustrates the data structure before the combination process begins. The columns are distinct, and we require them to be merged into a new column, **D**, which will hold the concatenated string for each record.

	A	B	C	D	E	F
1	Store	Employee	Sales			
2	North	Andy	24			
3	North	Bob	19			
4	South	Chad	15			
5	South	Doug	30			
6	East	Eric	12			
7	East	Frank	17			
8	West	Greg	18			
9	West	Henry	30			
10						
11						
12						
13						
14						
15						
16						
17						

To execute the concatenation for the first row, we type the following formula directly into cell **D2**. This formula instructs Excel to take the contents of the range **A2:C2** and merge them sequentially. This single formula efficiently replaces the need to manually reference each cell individually (e.g., A2 & B2 & C2), which was common in older versions of the software.

=CONCAT(A2:C2)

Once the formula is entered into **D2**, we can apply this logic to the rest of the column. By clicking and dragging the fill handle (the small square at the bottom-right corner of cell **D2**) down to the last row of our data, Excel automatically adjusts the [cell references](#) (e.g., D3 uses A3:C3, D4 uses A4:C4, and so on). The resulting Column D clearly demonstrates that the values from columns A, B, and C have been merged into a single cell with no spaces or intervening characters, providing a clean, compact tracking identifier.

D2				=CONCAT(A2:C2)	
	A	B	C	D	E
1	Store	Employee	Sales	Combined Values	
2	North	Andy	24	NorthAndy24	
3	North	Bob	19	NorthBob19	
4	South	Chad	15	SouthChad15	
5	South	Doug	30	SouthDoug30	
6	East	Eric	12	EastEric12	
7	East	Frank	17	EastFrank17	
8	West	Greg	18	WestGreg18	
9	West	Henry	30	WestHenry30	
10					
11					
12					
13					
14					
15					

As visible in the output, Column D successfully combines the values from columns **A**, **B**, and **C** into a single cell string. This method ensures data integrity by preventing the introduction of any extraneous characters, fulfilling the requirement for a unified, non-delimited data field. For users who strictly require minimal formula complexity and do not need visual separation within the combined string, the [CONCAT function](#) remains the ideal choice.

Method 2: Using the TEXTJOIN Function with Delimiters

While **CONCAT** is excellent for generating seamless strings, most data presentation requires some form of separation to maintain readability. This is where the [TEXTJOIN Function](#) excels. TEXTJOIN allows the user to specify three critical elements: the [delimiter](#) (the character or text to be inserted between values), a logical argument specifying whether to ignore empty cells, and the range of cells to be joined. This makes TEXTJOIN the preferred function for preparing data for human consumption or for creating standardized outputs like addresses or full names.

The syntax for TEXTJOIN is more expansive than CONCAT but offers vastly superior control over the final output. The general structure is: **=TEXTJOIN(delimiter, ignore_empty, text1, , ...)**. In our case, if we wish to use a standard space character to separate the data from columns A, B, and C, the formula is constructed as shown below. The first argument, " ", specifies the space as the delimiter, and the second argument, **TRUE**, tells Excel to ignore any blank cells within the specified range, which is a powerful feature for data cleaning.

To combine the values from cells **A2**, **B2**, and **C2**, inserting a space between each value for enhanced readability, we would use the following formula in cell **D2**. This configuration is widely applicable when merging name components, date segments, or geographical identifiers where visual separation is mandatory.

=TEXTJOIN(" ", TRUE, A2:C2)

This formula dictates that a space must be inserted after the contents of **A2** and **B2** before the next cell's content is appended. Crucially, by setting the **ignore_empty** argument to **TRUE**, if cell **B2** were empty (e.g., no middle initial), TEXTJOIN would intelligently join A2 and C2 with only a single space between them, rather than inserting two spaces (one for the missing B2 content and one after it), preventing unsightly gaps in the final output string.

Example 2: Joining Data with Spaces Using TEXTJOIN

Building upon our previous employee sales dataset, let's now use **TEXTJOIN** to create a combined field that is easily readable, separating the Employee ID, Product Code, and Sales Region with a space. This is essential for reports or dashboards where the combined field serves as a descriptive label rather than a technical key. The implementation process mirrors that of **CONCAT**, starting with the formula insertion in the first data row.

We insert the [TEXTJOIN function](#) into cell **D2**. The use of the space character (" ") as the delimiter ensures that the output is segmented, enhancing the clarity of the combined information. This immediate visual separation makes the resulting data much more accessible for general users who do not require the compact nature of the **CONCAT** output.

=TEXTJOIN(" ", TRUE, A2:C2)

After entering the formula in **D2**, we utilize the fill handle once more to propagate the formula down column D for the remaining rows. As the formula is dragged down, it dynamically updates the [cell references](#), ensuring that each row correctly processes its corresponding A, B, and C values. The resulting column D now showcases the combined data with a clear space between each component, transforming the technical identifiers into a more human-readable format.

D2 ✕ ✓ <i>fx</i> =TEXTJOIN(" ", TRUE, A2:C2)					
	A	B	C	D	E
1	Store	Employee	Sales	Combined Values	
2	North	Andy	24	North Andy 24	
3	North	Bob	19	North Bob 19	
4	South	Chad	15	South Chad 15	
5	South	Doug	30	South Doug 30	
6	East	Eric	12	East Eric 12	
7	East	Frank	17	East Frank 17	
8	West	Greg	18	West Greg 18	
9	West	Henry	30	West Henry 30	
10					
11					
12					
13					
14					

The screenshot above confirms that Column D successfully combines the values from columns A, B, and C into a single cell, with a space inserted in between the values, achieving the desired separation. This outcome highlights the primary advantage of TEXTJOIN over CONCAT: the ability to structure the combined output for clarity and presentation, making it suitable for a broader range of reporting needs.

Advanced Delimiters with TEXTJOIN

One of the most powerful features of the [TEXTJOIN function](#) is the flexibility it provides in defining the [delimiter](#). The first argument of the TEXTJOIN function is not restricted to simple spaces; any character, string, or combination of characters enclosed in quotation marks can serve as the separator. This allows analysts to create highly structured data outputs, such as hyphenated codes, comma-separated lists, or even outputs separated by complex strings like " - AND - ". This level of customization is invaluable when data must conform to external system requirements or specific organizational formatting standards.

For instance, we might need the combined data to look like a hyphenated product code (e.g., "EID-PC-REGION"). To achieve this, we simply replace the space delimiter (" ") in the first argument with a dash ("-"). The resulting formula effectively instructs Excel to insert a dash after the content of the first and second columns before appending the subsequent content, providing a standardized, structured output that resembles a traditional part number or SKU.

To demonstrate this flexibility, we use the following formula. This structure maintains the efficiency of range referencing (**A2:C2**) while changing the visual output dramatically by introducing a hyphenated structure.

=TEXTJOIN("-", TRUE, A2:C2)

The following screenshot illustrates the practical application of this modified formula. Notice how the resulting combined strings in Column D are now separated by dashes, providing a highly organized structure that might be preferred for inventory management or system integration purposes. This capability underscores why TEXTJOIN is often considered the superior, modern function for complex concatenation tasks in Excel, offering control that CONCAT simply cannot match.

	A	B	C	D	E
1	Store	Employee	Sales	Combined Values	
2	North	Andy	24	North-Andy-24	
3	North	Bob	19	North-Bob-19	
4	South	Chad	15	South-Chad-15	
5	South	Doug	30	South-Doug-30	
6	East	Eric	12	East-Eric-12	
7	East	Frank	17	East-Frank-17	
8	West	Greg	18	West-Greg-18	
9	West	Henry	30	West-Henry-30	
10					
11					
12					
13					
14					
15					

As clearly shown, Column D now successfully combines the values from columns A, B, and C into a single cell string, with a distinct dash inserted in between the values. This customization confirms the power and versatility of the **TEXTJOIN** function in handling diverse data formatting requirements, moving beyond simple space separation to complex, standardized delimiters.

Summary of Best Practices and Additional Resources

Choosing between **CONCAT** and **TEXTJOIN** depends entirely on the intended use of the

combined data. If the goal is to create a compact, non-spaced technical identifier (e.g., database keys), **CONCAT** is the more streamlined and appropriate choice. However, if the combined string is intended for reporting, user presentation, or requires handling potential missing data gracefully, **TEXTJOIN** provides the necessary control via its delimiter and **ignore_empty** arguments. The ability of both functions to handle cell ranges (e.g., A2:C2) efficiently marks a significant improvement over legacy concatenation methods, ensuring faster formula creation and improved worksheet performance across large datasets in [Excel](#).

When implementing these functions, always ensure that your cell references are absolute if you intend to copy the formula across columns, or relative if you intend to drag the formula down rows, as demonstrated in the preceding examples. For complex concatenation tasks involving conditional formatting or combining data based on external criteria, nested functions might be required, but for simple column merging, these two functions provide the most direct and reliable solution. Mastering both **CONCAT** and **TEXTJOIN** is fundamental to effective data manipulation within Excel.

For further learning and to explore related data manipulation techniques in Excel, the following tutorials explain how to perform other common operations, building upon the foundational knowledge of merging and preparing data:

How to Extract Substrings from Text in Excel

Using the TRIM Function for Data Cleaning

Techniques for Combining Dates and Times in Excel