

How to Easily Match Two Columns and Retrieve Data from a Third in Excel

Authored by
stats writer

December 3, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Easily Match Two Columns and Retrieve Data from a Third in Excel*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=104168>

In the realm of Excel mastery, one of the most fundamental and frequently performed tasks is the reconciliation of disparate datasets. Specifically, advanced users often need to match values residing in two distinct columns and subsequently retrieve associated information from a third column. This crucial procedure is efficiently executed using the established VLOOKUP function. This versatile formula is engineered to search for a specific target value within the leftmost column of a designated table and then return a value from a specified column in the same row.

The application of VLOOKUP extends beyond simple comparison; it is an indispensable tool for merging, validating, and enriching data. Whether you are comparing two extensive lists to identify common entries or utilizing a reference table to populate missing fields in your working sheet, Excel's lookup capabilities streamline complex database operations. Mastering this technique is a foundational step toward proficient data analysis and manipulation, ensuring accuracy and saving countless hours of manual data entry and cross-referencing.

Understanding the VLOOKUP Function Syntax

The core requirement for linking data across different columns is a structured approach to searching and retrieving. Whenever the goal is to match criteria between two columns and derive corresponding data into a third, the process hinges on the proper configuration of the VLOOKUP function. This function, whose name stands for "Vertical Lookup," requires four distinct arguments to execute successfully. Understanding each component is critical to avoiding common errors, such as returning #N/A or incorrect values.

The standard syntax for the function is formalized as follows:

VLOOKUP(lookup_value, table_array, col_index_num,)

To ensure clarity and precision in its deployment, we must delineate the role of each argument:

lookup_value: This is the specific item, identifier, or key that you are seeking. It must be present in both the column you are using for reference (Column D in our example) and the first column of the **table_array** (Column A). This value acts as the bridge connecting your two datasets. For instance, if you are looking up a product ID, that ID is your lookup_value.

table_array: This defines the entire range of cells containing the data you wish to search within and the data you wish to retrieve. It is imperative that the column containing the **lookup_value** is always the leftmost column of this array. When defining the table_array, it is standard practice to use **absolute references** (using dollar signs, e.g., \$A\$2:\$B\$16) to prevent the range from shifting when the formula is copied down.

col_index_num: This numerical index specifies which column within the defined table_array contains the result you want returned. If your table array spans three columns (A, B, C), and you

want the data from column C, your index number is 3. Note that this number is relative to the start of the table_array, not the entire sheet.

range_lookup: This optional boolean argument dictates the type of match Excel should perform. Using **TRUE** or omitting the argument results in an approximate match (useful for numerical ranges, but requires the lookup column to be sorted). Using **FALSE** forces an exact match, which is nearly always required when matching text strings or unique identifiers. For accurate matching of two columns, **FALSE** is the mandatory setting.

Example: Setting Up the Scenario for Data Matching

To solidify the understanding of VLOOKUP, let us walk through a practical scenario involving sports team statistics. Imagine we possess two separate datasets within the same Excel worksheet. The first dataset (our primary reference table) contains a comprehensive list of team names and their corresponding total points. The second dataset (our target list) contains a smaller, unorganized list of teams for which we need to retrieve the point values from the reference table.

This situation perfectly illustrates the power of combining data based on a common key--in this case, the team name. Our goal is to use the team names present in the target list to look up the associated point totals in the reference table and deposit those totals into a new column, thereby effectively matching column A and column D and returning the value from column B into column E.

The initial state of our worksheet is represented visually below. Notice the two distinct blocks of information that need to be correlated:

	A	B	C	D	E	F	G
1	Team	Points		Team	Points		
2	Mavericks	106		Suns			
3	Magic	89		Rockets			
4	Nets	93		Spurs			
5	Suns	96		Heat			
6	Celtics	95		Cavs			
7	Hornets	99					
8	Spurs	103					
9	Rockets	109					
10	Warriors	114					
11	Heat	86					
12	Knicks	89					
13	Cavs	88					
14	Thunder	93					
15	Lakers	96					
16	Clippers	99					
17							
18							
19							
20							
21							
22							

Step-by-Step Implementation of the VLOOKUP Formula

Our objective is to populate Column E ("Returned Points") by matching the team names listed in Column D ("Teams to Find") against the master list in Column A ("Team") and retrieving the respective scores from Column B ("Points"). We begin the process by focusing on the first entry in our target column, which is cell E2.

We need to instruct Excel to take the value in D2 (the team name 'Suns') and search for it within the range defined by our master data (A2 through B16). Once a match is found, Excel must return the data from the second column of that defined range (Column B).

The correct structure for the VLOOKUP formula, tailored specifically for the first cell (E2), is constructed as follows:

=VLOOKUP(D2, \$A\$2:\$B\$16, 2, FALSE)

This formula is entered directly into cell E2. Note the crucial use of **absolute cell references** (the dollar signs) for the table_array \$A\$2:\$B\$16. This fixation ensures that when we subsequently drag the formula down to apply it to the rest of Column D, the reference table itself does not shift, preserving the integrity of the lookup range for all subsequent calculations.

Analyzing the Formula Parameters and Output

Let us deconstruct the specific arguments used in the formula entered into E2 to understand precisely how the calculation is performed.

D2 (lookup_value): This is the team name 'Suns'. This relative reference ensures that when copied down, it changes to D3, D4, and so on, allowing us to dynamically look up each team in the target list.

\$A\$2:\$B\$16 (table_array): This fixed range defines our search domain, spanning the "Team" and "Points" columns. The search must begin in Column A.

2 (col_index_num): Since we want to return the "Points" value, and "Points" is the second column within the \$A\$2:\$B\$16 range, we use the index number 2.

FALSE (range_lookup): This critical setting demands an exact match for the lookup_value 'Suns'. If the exact name is not found, the formula will correctly return #N/A.

Upon execution, Excel locates 'Suns' in the master table (A2:B16) and retrieves the corresponding value from the second column (Column B), resulting in the desired output in cell E2. The screenshot below confirms the successful execution of the formula in the initial cell:

	A	B	C	D	E	F	G
1	Team	Points		Team	Points		
2	Mavericks	106		Suns	96		
3	Magic	89		Rockets			
4	Nets	93		Spurs			
5	Suns	96		Heat			
6	Celtics	95		Cavs			
7	Hornets	99					
8	Spurs	103					
9	Rockets	109					
10	Warriors	114					
11	Heat	86					
12	Knicks	89					
13	Cavs	88					
14	Thunder	93					
15	Lakers	96					
16	Clippers	99					
17							
18							
19							
20							
21							

As clearly illustrated, the 'Points' value associated with the 'Suns' entry in Column B is **96**, which is why this value is returned in column E. This demonstrates the seamless correlation between the key in the target list and the retrieved data in the reference table.

Expanding the Solution: Applying the Formula to the Entire Dataset

The true efficiency of this method is realized when the formula is applied consistently across the entire target list. Because we meticulously used **absolute references** for the table array (\$A\$2:\$B\$16) and **relative references** for the lookup value (D2), we can now simply drag the formula down Column E.

To propagate the formula, select cell E2, click and hold the small square (the fill handle) located at the bottom right corner of the cell, and drag it down to the last row corresponding to your target data (in this case, E16). Excel automatically adjusts the relative reference (D2 changes to D3, D4, etc.) while maintaining the fixed reference table, executing a precise lookup for every team listed in Column D.

This operation successfully completes the task of matching all values in the target column (D) against the reference column (A) and populating the third column (E) with the corresponding points from the reference data (B). The resulting filled column E provides a complete and verified dataset, ready for further data analysis.

	A	B	C	D	E	F	G
1	Team	Points		Team	Points		
2	Mavericks	106		Suns	96		
3	Magic	89		Rockets	109		
4	Nets	93		Spurs	103		
5	Suns	96		Heat	86		
6	Celtics	95		Cavs	88		
7	Hornets	99					
8	Spurs	103					
9	Rockets	109					
10	Warriors	114					
11	Heat	86					
12	Knicks	89					
13	Cavs	88					
14	Thunder	93					
15	Lakers	96					
16	Clippers	99					
17							
18							
19							
20							
21							
22							

Optimizing Performance: Handling Errors and Missing Data

While VLOOKUP is robust, it is essential to prepare for instances where a team name in Column D does not have a match in the reference table (Column A). In such cases, Excel returns the error value #N/A, indicating "Not Applicable" or "Not Found." While technically correct, this output can disrupt subsequent calculations or detract from the readability of the report.

To enhance the user experience and ensure cleaner output, the VLOOKUP function is frequently nested within the **IFERROR()** function. The IFERROR function allows the user to specify a custom value or message to display if the primary formula results in an error. For instance, modifying the formula to `=IFERROR(VLOOKUP(D2, A2:B16, 2, FALSE), "Missing Data")` ensures that

any team not found will display the text "Missing Data" instead of the disruptive #N/A error. This is a critical step in preparing data for professional presentation or automated processing.

Limitations of VLOOKUP and Modern Alternatives

While the VLOOKUP function remains a powerful tool, especially for backward compatibility, it does possess inherent structural limitations that modern Excel versions have addressed. The most significant constraint is the requirement that the lookup column (the column containing the lookup_value) must always be the leftmost column of the defined table_array. This constraint prevents looking "left" for data, forcing users to reorganize their reference tables simply to accommodate the function's architecture.

Furthermore, VLOOKUP often suffers from performance degradation when used across extremely large datasets, particularly due to its legacy implementation. For those utilizing newer versions of Excel, the use of more flexible and powerful alternatives is highly recommended. The two primary modern solutions are the **INDEX MATCH** combination and the newer **XLOOKUP** function.

The **INDEX MATCH** method overcomes the "lookup left" limitation by separating the search criteria (MATCH) from the retrieval mechanism (INDEX), offering greater flexibility and often better performance. However, for users with Microsoft 365 or Excel 2021, the **XLOOKUP** function is the preferred choice. XLOOKUP is designed as a streamlined, superior replacement for both VLOOKUP and INDEX MATCH, offering default exact matching, the ability to look both left and right, and built-in handling for missing data without needing a separate IFERROR wrapper.

Conclusion: Essential Skill for Data Analysis

The ability to conditionally match and merge data from distinct columns using formulas like VLOOKUP is a cornerstone skill for effective data management in Excel. Whether utilizing the traditional VLOOKUP method demonstrated here or transitioning to the advanced capabilities of XLOOKUP, the fundamental concept remains critical: establishing a common key to bridge information gaps between tables.

By implementing the correct syntax, particularly the precise definition of the table_array and the mandatory use of the FALSE argument for exact matching, users can reliably consolidate, clean, and prepare data for advanced quantitative reporting and data analysis. Mastery of these lookup functions transforms raw data into actionable insights.