

Excel: Use VLOOKUP to Return Multiple Values Vertically

Authored by
stats writer

November 17, 2025

RECOMMENDED CITATION

stats writer (2025). *Excel: Use VLOOKUP to Return Multiple Values Vertically*.
PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=94454>

The Challenge of Multiple Matches in Excel

Microsoft Excel remains the industry standard for data manipulation and analysis. One of its most powerful and frequently utilized tools is the **VLOOKUP** function, designed to quickly search for a value in the first column of a table and return a corresponding value from a specified column in the same row. While indispensable for simple lookups, users frequently encounter a major limitation when dealing with real-world datasets: the need to retrieve multiple matching entries for a single lookup criterion.

The core issue stems from the fundamental design of **VLOOKUP**. It is engineered to stop processing and return the result immediately upon locating the very first instance of the lookup value within the designated range. This behavior means that if your data contains duplicates--which is common when tracking metrics over time or grouping details by category--you cannot use **VLOOKUP** alone to extract all related records. If you are tracking sales data for a region that appears multiple times, or, as we will demonstrate, basketball scores for a team that played several games, a single lookup will only capture the first recorded event, omitting crucial information necessary for comprehensive reporting in **Excel**.

Addressing this constraint traditionally required complex, resource-intensive techniques, often involving elaborate combinations of functions like **INDEX**, **MATCH**, **ROW**, and sometimes necessitated the creation of helper columns using concatenation. These legacy methods were prone to errors, difficult to audit, and required the user to enter them as Dynamic Array Formulas (CSE formulas) in older versions of Excel. Fortunately, modern versions of Excel (Microsoft 365) have introduced far more elegant and powerful solutions that simplify this process dramatically.

Understanding the Limitations of VLOOKUP

To fully appreciate the necessity of modern alternatives, it is vital to understand precisely why **VLOOKUP** fails in a multiple-match scenario. The function operates sequentially down the table's first column. Once it finds an exact match (or the closest match, depending on the range lookup argument), it locks onto that row and returns the corresponding value from the specified index column. It does not possess the inherent capability to continue scanning the remaining rows for additional matches, making it inherently unsuitable for returning multiple values vertically.

Consider a scenario where a lookup value, such as a Product ID, appears five times in the source data, each with a different associated transaction value. If you run **VLOOKUP**, you will only retrieve the transaction value linked to the first appearance of that Product ID. The other four essential data points are completely ignored by the function, potentially leading to incomplete analysis or skewed reporting. This inherent limitation dictates that for retrieving comprehensive subsets of data based on a single criterion, we must look beyond the standard vertical lookup methodology.

The requirement here is not simply to find one corresponding cell value, but to dynamically extract and spill an entire vertical array of results. This shift in requirement--from retrieving a single scalar value to retrieving a dynamic array--is precisely what necessitated the development of Excel's powerful new generation of functions, including the one we will use to solve this problem: **FILTER**.

Introducing the Modern Solution: The FILTER Function

The introduction of the **FILTER function** marks a significant milestone in Excel's evolution, especially for users dealing with complex data extraction requirements. Unlike traditional lookup functions that return a single cell value, **FILTER** is designed as a Dynamic Array Formulas function. This means it can automatically return an array of results that "spill" into adjacent cells, either horizontally or, as required for this vertical lookup solution, downwards.

The **FILTER** function's primary role is to filter a range or array based on a boolean (TRUE/FALSE) criteria defined by the user. It allows for highly customizable filtering directly within a cell formula, eliminating the need for manual data filtering or complex legacy formulas. When utilized effectively, it completely replaces the need for cumbersome workarounds previously required to handle multiple returns in lookups, providing a clean and intuitive solution.

By leveraging **FILTER**, we can instruct Excel to examine the entire lookup column (the criteria range), identify every row that meets our specified condition, and then return the corresponding values from the desired results column, stacking them vertically below the cell where the formula is entered. This capability transforms how complex lookups are executed in Excel, providing both efficiency and clarity.

Syntax Breakdown: Utilizing FILTER for Vertical Lookups

To perform a vertical lookup that returns multiple matching values, we utilize **FILTER** with a specific structure. The general syntax for the **FILTER** function is: `=FILTER(array, include, )`.

In our specific application, which replicates the vertical lookup capability but with multiple returns, the formula is highly concise. This syntax allows us to look up some value in a range and return multiple values vertically:

=FILTER(B2:B12, D2=A2:A12)

Let's analyze the components of this powerful formula:

The "Array" Argument (B2:B12): This is the range containing the values you wish to return. In a traditional **VLOOKUP**, this would be the 'result' column. Here, we are telling Excel, "I want to retrieve data from cells B2 through B12." These are the potential results (e.g., the Points scored).

The "Include" Argument (D2=A2:A12): This is the crucial conditional expression that defines the lookup criteria. It tells Excel which rows of the array to include. We are comparing the lookup value (in cell **D2**, which might contain "Mavs") against every value in the lookup range (A2:A12, the Team column). This comparison generates an array of TRUE and FALSE values (a boolean array). A TRUE value indicates a match, triggering the return of the corresponding value from the B2:B12 range.

The Optional "if_empty" Argument: Although not used in this specific example, this third argument allows the user to specify a custom message (e.g., "No Match Found") to display if the filter criteria returns no results, enhancing user experience and clarity.

In essence, this compact formula returns all values from the range **B2:B12** where the corresponding value in the range **A2:A12** is exactly equal to the criterion specified in cell **D2**. The result automatically spills vertically down the spreadsheet, fulfilling the requirement to return multiple values.

Practical Example Setup: Preparing the Dataset

To illustrate the application of this method, let us consider a practical example involving a dataset tracking basketball performance. Suppose we have compiled data detailing points scored by various players across different teams. Our goal is to query this data based on the Team name and retrieve all corresponding Point totals associated with that team.

Our sample dataset in Excel contains three columns: Player, Team, and Points. We will organize the data within cells A1 to B12. Column A contains the Team name, and Column B contains the Points scored. Column D will be reserved for our lookup value, and Column E will be where the dynamic results spill.

Suppose we have the following dataset in Excel that contains information about points scored by basketball players on various teams:

	A	B	C	D	E
1	Team	Points			
2	Mavs	22			
3	Rockets	19			
4	Nets	34			
5	Spurs	13			
6	Spurs	18			
7	Mavs	15			
8	Kings	17			
9	Mavs	30			
10	Celtics	32			
11	Warriors	27			
12	Nets	15			
13					
14					
15					

Our objective is clear: we intend to look up the team identifier "Mavs" (which we will place in cell **D2**) within the Team column (A2:A12) and subsequently retrieve every single points value (from B2:B12) associated with that team, displayed vertically in column E. This requires a solution that is dynamic and handles all matching entries, precisely where the **FILTER** function excels.

Step-by-Step Implementation of the Formula

The following example shows how to use this formula in practice. The implementation of the **FILTER function** is straightforward, provided the user has access to a version of Excel that supports Dynamic Array Formulas.

First, identify the cell where the lookup criterion will reside. For this example, we place our target team, "Mavs," into cell **D2**.

Next, select the starting cell for the results. Since we want the returned points values to appear vertically, we will start by typing the formula into cell **E2**.

Finally, input the **FILTER** formula, specifying the array to return (B2:B12) and the condition (D2 = A2:A12):

We can type the following formula into cell **E2** to execute the multi-value vertical lookup:

=FILTER(B2:B12, D2=A2:A12)

Upon pressing Enter, the formula immediately performs the calculation. Because this is a dynamic array function, the results will "spill" automatically into the cells below **E2** (i.e., E3, E4, E5, etc.) until all matching values have been returned. No manual copying or array entry (Ctrl+Shift+Enter) is required, highlighting the efficiency of this modern approach.

Interpreting the Results and Application Scenarios

Once the formula is entered, the output clearly demonstrates its effectiveness in resolving the limitations of the traditional **VLOOKUP** function. The following screenshot shows how to use this formula in practice:

	A	B	C	D	E	F
1	Team	Points		Team	Points	
2	Mavs	22		Mavs	22	
3	Rockets	19			15	
4	Nets	34			30	
5	Spurs	13				
6	Spurs	18				
7	Mavs	15				
8	Kings	17				
9	Mavs	30				
10	Celtics	32				
11	Warriors	27				
12	Nets	15				
13						
14						

As shown in the output, the formula correctly returns the values **22**, **15**, and **30**. These are the three point totals associated with the team "Mavs" within the original dataset. The function successfully scanned the entire range (A2:A12) and returned the corresponding values from the points column (B2:B12) for every match encountered.

Notice that each of these values represents a value in the points column that corresponds precisely to the "Mavs" entry in the team column:

	A	B	C	D	E	
1	Team	Points		Team	Points	
2	Mavs	22		Mavs	22	
3	Rockets	19			15	
4	Nets	34			30	
5	Spurs	13				
6	Spurs	18				
7	Mavs	15				
8	Kings	17				
9	Mavs	30				
10	Celtics	32				
11	Warriors	27				
12	Nets	15				
13						
14						
15						

This approach is particularly valuable in various professional application scenarios, such as financial modeling (retrieving all transactions for a specific account), inventory management (listing all components belonging to a specific product line), or human resources (extracting all payroll entries for an employee across different cost centers).

Why FILTER is Superior to Legacy Array Formulas

Before the introduction of the **FILTER** function, achieving this type of multi-value vertical lookup was significantly more challenging. Users relied on complex traditional array formulas, typically combining **INDEX**, **SMALL**, **IF**, and **ROW** functions. These formulas were notoriously difficult to write, debug, and maintain, especially for intermediate users of **Excel**.

The primary disadvantages of legacy array methods included: the requirement for specific array entry (Ctrl+Shift+Enter); the necessity of dragging the formula down a sufficient number of rows (often resulting in #NUM or #N/A errors if dragged too far or not far enough); and the reliance on helper components like **COUNTIF** to track occurrences.

The **FILTER** function, conversely, eliminates all these hurdles. It is simple to write, does not require array entry shortcuts, and, critically, it handles the "spilling" of the results automatically, dynamically adjusting the output size based on the number of matches found. This transition represents a major improvement in data handling capabilities within modern **Excel**, allowing users to perform sophisticated lookups with unprecedented ease and reliability. This powerful, yet

succinct, method using **FILTER** should be the preferred technique for anyone operating on current versions of **Excel** who needs to return multiple corresponding values based on a single lookup criterion, moving far beyond the inherent limitations of the legacy **VLOOKUP** function.

ARABPSYCHOLOGY.COM