

How to Return the Last Matching Value with VLOOKUP in Google Sheets

Authored by
mohammed looti

January 9, 2026

RECOMMENDED CITATION

mohammed looti (2026). *How to Return the Last Matching Value with VLOOKUP in Google Sheets*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=125165>

The VLOOKUP function is a cornerstone of Google Sheets data manipulation, enabling powerful database queries within a spreadsheet environment. Its primary role is to search for a specific value in the leftmost column of a range and return a corresponding value from a designated column. While straightforward, using standard VLOOKUP presents a limitation: it inherently retrieves the **first matching value** it encounters. However, in dynamic datasets, retrieving the **last matching value**—often representing the most recent entry or the final status of a record—is frequently required. To overcome this limitation and harness VLOOKUP for this specialized purpose, we must integrate it with auxiliary functions like the SORT function and the ROW function. This combination allows us to invert the dataset order, forcing VLOOKUP to recognize the chronologically last entry as the first match in the sorted range.

Understanding the Default VLOOKUP Behavior

By default, the **VLOOKUP** function in Google Sheets is designed for efficiency and speed when performing a vertical search. When you provide it with a search key, a data range, and a column index, it scans the first column of that range from top to bottom. As soon as it finds the first instance that matches the criteria, it stops the search and returns the corresponding value. This behavior is ideal for ensuring data integrity when records are unique or when the oldest record is the desired outcome.

However, many datasets are transactional, meaning they include multiple entries for the same identifier (e.g., product IDs, customer names, or team scores) recorded over time. In these scenarios, the standard VLOOKUP result, which returns the **first match**, might be outdated or irrelevant. For instance, if you are tracking sequential updates or player scores, you often need the most current, or **last match**, entry to reflect the true, most recent state of the data.

The Challenge of Retrieving the Last Match

The inherent limitation of the VLOOKUP function necessitates a workaround when searching for the most recent data point. Simply changing the lookup type argument to `FALSE` (for exact match) does not resolve the issue of multiple matches; it only ensures accuracy for the first match found. We must strategically manipulate the underlying data structure before VLOOKUP processes it. This manipulation involves rearranging the entire lookup range so that the last row in the original dataset effectively becomes the first row in the temporary, virtual lookup range, overriding VLOOKUP's natural search order.

Implementing the Advanced VLOOKUP Solution

To successfully instruct Google Sheets to return the value associated with the final occurrence of a search key, we must nest the data range within the SORT function. The crucial component of this

technique is using the [ROW function](#) as the sorting criterion, ensuring that the entire dataset is temporarily sorted in descending order based on the row numbers. This ensures that the rows are processed from bottom to top.

The following syntax is used to combine these functions, leveraging the internal capabilities of the SORT function to create an inverted table array for VLOOKUP to process:

```
=VLOOKUP(E2, SORT(A2:C11, ROW(A2:A11), FALSE), 3, FALSE)
```

Breaking Down the Last Match Formula Syntax

This formula is highly effective because it treats the dynamically sorted range as the lookup table array. The formula is designed to search for the value contained in cell **E2** (the lookup value) within the temporary array generated by the inner functions. It then targets column **3** of that array, retrieving the value based on an exact match (`FALSE`).

The inner [SORT function](#) is responsible for the rearrangement. It takes the original data range **A2:C11** and uses the array generated by the `ROW(A2:A11)` function as the basis for sorting. By setting the final argument of SORT to `FALSE`, we enforce a descending sort, effectively reversing the chronological order of the entries. This reversal ensures that the last record of any matching entry appears at the top of the virtual table, where VLOOKUP will find it first.

In summary, this construction looks up the value in cell **E2** within the range **A2:C11** (after it has been sorted in reverse row order) and returns the corresponding value from column **3**, which represents the data associated with the **last matching value** in the original dataset.

Practical Example: Tracking Latest Player Scores

To solidify this concept, let us work through a concrete example using a dataset that tracks basketball player statistics. This scenario perfectly illustrates why finding the last matching score is vital—it provides the most up-to-date performance metric when multiple entries exist for the same team.

Setup: Sample Basketball Dataset

Imagine we have the following dataset where teams are listed multiple times, reflecting different games or reporting periods. We aim to find the most recent reported score for a specific team.

	A	B	C	D
1	Team	Assists	Points	
2	Mavs	8	22	
3	Nets	4	14	
4	Warriors	4	29	
5	Warriors	9	30	
6	Mavs	3	36	
7	Nets	10	18	
8	Hawks	13	12	
9	Nets	5	17	
10	Hawks	8	25	
11	Mavs	2	24	
12				
13				
14				
15				

In this table (spanning A2:C11), we have columns for Player, Team, and Points. Notice that the team "Nets" appears three times, each with a potentially different score (8, 12, and 17). Our goal is to query the Team column for "Nets" and return the corresponding value from the Points column associated with the row furthest down the sheet (row 11), as this represents the last recorded entry.

Step-by-Step Implementation

Suppose we define our search key in cell **E2** as "Nets". We would like the result to appear in cell **F2**. Since the Points column is the third column (index 3) in our range (A:C), we construct the formula to target that index after the data has been inverted by the SORT function.

We can type the following comprehensive formula into cell **F2** to execute this dynamic lookup:

```
=VLOOKUP(E2, SORT(A2:C11, ROW(A2:A11), FALSE), 3, FALSE)
```

Verifying the Result

Upon executing the formula, [Google Sheets](#) successfully processes the request. The outcome demonstrates the efficacy of the VLOOKUP/SORT combination in retrieving the latest information from a repeated entry within the dataset.

F2 **fx** =VLOOKUP(E2, SORT(A2:C11, ROW(A2:A11), FALSE), 3, FALSE)

	A	B	C	D	E	F
1	Team	Assists	Points		Team	Points
2	Mavs	8	22		Nets	17
3	Nets	4	14			
4	Warriors	4	29			
5	Warriors	9	30			
6	Mavs	3	36			
7	Nets	10	18			
8	Hawks	13	12			
9	Nets	5	17			
10	Hawks	8	25			
11	Mavs	2	24			
12						
13						
14						

The formula correctly returns the value **17**. If you manually scan the original data, the last time "Nets" appears in the **Team** column is in row 11, where the corresponding **Points** value is indeed 17. This confirms that the reverse sorting mechanism successfully presented the final occurrence as the first match to the VLOOKUP function.

	A	B	C	D	E	F
1	Team	Assists	Points		Team	Points
2	Mavs	8	22		Nets	17
3	Nets	4	14			
4	Warriors	4	29			
5	Warriors	9	30			
6	Mavs	3	36			
7	Nets	10	18			
8	Hawks	13	12			
9	Nets	5	17			
10	Hawks	8	25			
11	Mavs	2	24			
12						
13						
14						

Detailed Explanation of Function Synergy

Understanding how the auxiliary functions interact is crucial to mastering this technique. The power lies in the calculated array created by the inner functions, which VLOOKUP then treats as its primary source data.

```
=VLOOKUP(E2, SORT(A2:C11, ROW(A2:A11), FALSE), 3, FALSE)
```

The Role of the ROW Function

The ROW function, when used with a range argument like `ROW(A2:A11)`, is vital. It does not return a single row number, but rather an array of numbers corresponding to the row index of every cell specified in the range. If we consider the rows in the example range (A2:A11), this returns the array {2; 3; 4; 5; 6; 7; 8; 9; 10; 11}. This generated array acts as an implicit, temporary sorting key attached to our data range, providing a numerical basis for rearrangement that corresponds exactly to the original order of entry.

The Role of the SORT Function

The SORT function then takes the original data range (A2:C11) and sorts it based on the array generated by the ROW function. By setting the third argument (`FALSE`) to indicate **descending order**, the SORT function reverses the chronological sequence of the data. Row 11 (the largest row number) moves to the top, followed by Row 10, and so on, creating a completely inverted virtual table array which is passed to VLOOKUP.

Because VLOOKUP is inherently coded to find the **first match** as it scans from top to bottom, when it searches this inverted table, the entry that was originally the last match in the spreadsheet (e.g., the data from row 11) is now the first entry it finds for that search key. This strategic use of data manipulation allows us to bypass the standard lookup limitations and successfully perform a search for the most recent data point.

Alternative Methods and Considerations

While the VLOOKUP/SORT/ROW combination is powerful and effective for obtaining the last match, it is important to note its computational cost. Since the entire range must be sorted internally upon every calculation, performance can slow down significantly when dealing with extremely large datasets (tens of thousands of rows). The SORT function is volatile in this context and recalculates frequently.

For those seeking alternatives or more robust solutions in complex scenarios, experienced

spreadsheet users often prefer alternative approaches. These include using the combination of **INDEX** and **MATCH** functions nested with other array formulas, or utilizing the **QUERY** function, which is Google Sheets' powerful native database language capability. However, for most common operational tasks, this modified VLOOKUP approach offers a clear, understandable, and highly effective method for retrieving the last matching value.

The following tutorials explain how to perform other common tasks in Google Sheets:

ARABPSYCHOLOGY.COM