

How to Easily Perform a Reverse VLOOKUP in Google Sheets

Authored by
stats writer

November 30, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Easily Perform a Reverse VLOOKUP in Google Sheets*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=102229>

A VLOOKUP function is typically designed to search for a value in the leftmost column of a specified data range and then return a corresponding value from a column to the right. However, situations often arise where you need to search for a criterion located in a rightward column and extract data from a column positioned to its left. This is precisely when a **reverse VLOOKUP** becomes necessary in applications like Google Sheets. While there is no dedicated `REVERSEVLOOKUP` function, we can efficiently achieve this functionality by leveraging the powerful combination of standard `VLOOKUP` and the **Array literal** syntax `{}`.

To successfully execute a reverse lookup in Google Sheets, the key is dynamically restructuring your data range within the formula itself. This technique effectively tricks the `VLOOKUP` function into treating a different column (the one containing your search key) as the new leftmost column. This tutorial will meticulously guide you through the process, ensuring you can perform sophisticated data extraction regardless of the column order.

The VLOOKUP Limitation and the Need for Reversal

The core limitation of the standard VLOOKUP function stems from its design parameter: it must search the first column of the provided range. If your search criterion (the lookup value) is located in the third column, and the desired return value is in the first column, `VLOOKUP` will fail because it only evaluates the first column for matches. This rigidity makes simple `VLOOKUP` unsuitable for finding data positioned to the left of the search column.

In data management, particularly when dealing with large or fixed **datasets**, restructuring the source data merely to satisfy `VLOOKUP`'s requirements is often impractical or inefficient. For instance, if you have a product ID in column B and a descriptive name in column A, a standard `VLOOKUP` cannot use the ID (B) to retrieve the name (A). Therefore, mastering the reverse `VLOOKUP` technique is critical for advanced users who need flexibility in data retrieval without altering the underlying spreadsheet layout.

Example: Reverse VLOOKUP in Google Sheets

Suppose we are working with the following **dataset** which tracks the performance of various basketball teams, specifically noting their respective scores. Our goal is to illustrate both the standard `VLOOKUP` behavior and the necessary reversal technique.

	A	B	C	D
1	Team	Points		
2	Mavericks	90		
3	Rockets	92		
4	Spurs	94		
5	Hornets	96		
6	Nets	97		
7	Knicks	99		
8	Suns	104		
9	Bucks	107		
10				
11				
12				
13				
14				
15				
16				
17				

First, let's confirm how a traditional lookup works. We can use the following **VLOOKUP** formula to find the number of points associated with the team named "Knicks." In this standard scenario, the team name is in column A (the first column of the range), and the score is in column B (the second column).

=VLOOKUP("Knicks", A1:B9, 2)

This formula is structured according to the typical **VLOOKUP** syntax: `VLOOKUP(search_key, range, index,)`. Here's a detailed breakdown of its parameters:

It identifies **"Knicks"** as the value that the function needs to find in the first column.

It specifies **A1:B9** as the complete range of the data to be analyzed.

It designates that we wish to return the corresponding value located in the **2nd** column within the specified range (Column B).

When applied in practice, the spreadsheet successfully executes the lookup:

D2 fx =VLOOKUP("Knicks", A1:B9, 2)

	A	B	C	D
1	Team	Points		Points for Knicks
2	Mavericks	90		99
3	Rockets	92		
4	Spurs	94		
5	Hornets	96		
6	Nets	97		
7	Knicks	99		
8	Suns	104		
9	Bucks	107		
10				
11				
12				
13				
14				
15				
16				

As anticipated, this standard formula correctly returns the numerical value of **99**, which corresponds to the "Knicks" team. This demonstrates the required left-to-right search flow.

The Breakthrough: Using Array Literals for Data Reordering

The true challenge arises when we flip the requirement. Suppose we instead want to know which team is associated with a specific points value, such as **99**. In this case, our search key (99, in column B) is to the right of the value we want to return (the Team Name, in column A). This is where the standard VLOOKUP fails, but the **Array literal** technique provides an elegant workaround.

The array literal syntax { } in Google Sheets allows us to virtually reconstruct the data table within the formula itself. By using semicolons (;) to stack rows and commas (,) to concatenate columns, we can select our original columns and present them to VLOOKUP in a new order. For a reverse lookup, we simply place the column containing the lookup value first, followed by the column containing the return value.

We can use the following **reverse VLOOKUP** formula to find the team name associated with a score of 99:

=VLOOKUP(99, {B2:B9, A2:A9}, 2)

Detailed Analysis of the Reverse VLOOKUP Formula Components

Understanding the structure of the array literal is paramount to mastering this technique. The formula `{B2:B9, A2:A9}` effectively creates a temporary, two-column array where Column B (Points) is the first column, and Column A (Team) is the second column. This temporary structure satisfies VLOOKUP's requirement that the search key must be in the first column of the range.

Here is a precise breakdown of the specific arguments used in the reverse VLOOKUP:

The **Search Key (99)**: It identifies the score **99** as the value we need to find. Since we reordered the columns, VLOOKUP will search for this value in the temporary Column 1 (which corresponds to B2:B9).

The **Range ({B2:B9, A2:A9})**: This is the crucial part. It specifies the ranges to analyze, combining them into a single virtual range where the Points column (B) comes first, and the Team column (A) comes second. Note that we typically exclude the header row (starting at B2 and A2) when using this array method to prevent errors, depending on the exact implementation.

The **Index (2)**: This specifies that we want to return the value in the **2nd** column of the newly created array. Since the Team column (A) is now the second column in our virtual range, this returns the team name.

The **Is Sorted (Omitted/FALSE Default)**: Although not explicitly included, leaving the last parameter out defaults to `FALSE`, enforcing an exact match search, which is essential for accurate lookups.

Implementing the Reverse Lookup and Verifying Results

Once the formula is correctly entered, Google Sheets processes the array creation and executes the VLOOKUP against the virtual table. The output confirms the efficacy of this method, demonstrating how effortlessly we can bypass the standard left-to-right search constraint.

Here's how to use this formula in practice:

D2 fx =VLOOKUP(99, {B2:B9, A2:A9}, 2)

	A	B	C	D	E
1	Team	Points		Team Who Scored 99	
2	Mavericks	90		Knicks	
3	Rockets	92			
4	Spurs	94			
5	Hornets	96			
6	Nets	97			
7	Knicks	99			
8	Suns	104			
9	Bucks	107			
10					
11					
12					
13					
14					
15					
16					
17					

The formula correctly identifies the **Knicks** as the team associated with a points value of **99**. This successful operation validates the use of the **Array literal** syntax as the definitive method for performing a reverse VLOOKUP within Google Sheets environments.

Alternative Methods: INDEX MATCH and QUERY Function

While the VLOOKUP array method is highly effective, it is important to recognize that other, often more flexible, methods exist for performing bidirectional lookups. The **INDEX MATCH** combination is widely regarded as the most robust alternative, offering superior performance and flexibility, particularly across non-contiguous ranges or when dealing with extremely large datasets.

The INDEX function returns the value of a cell at a specific intersection (row, column), while the MATCH function returns the relative position (row number) of a search key within a specified range. By nesting MATCH inside INDEX, we can locate the row number corresponding to the search key in one column and then extract the data from that same row in another column, irrespective of their relative positions.

For our example, the equivalent INDEX MATCH formula to find the team name for score 99 would be:

=INDEX(A2:A9, MATCH(99, B2:B9, 0))

Another powerful option in Google Sheets is the `QUERY` function, which utilizes SQL-like commands. This method is exceptionally useful when you need to perform additional filtering or aggregation alongside your lookup. A reverse lookup using `QUERY` would look like this:

```
=QUERY(A1:B9, "SELECT A WHERE B = 99", 0)
```

Both `INDEX MATCH` and `QUERY` offer sophisticated solutions that eliminate the need for the array literal trick, providing valuable alternatives for complex data manipulation.

Best Practices and Troubleshooting Common Reverse VLOOKUP Errors

When employing the reverse VLOOKUP technique using array literals, several factors must be carefully managed to ensure accuracy and prevent errors. The most common pitfall is incorrectly defining the virtual array range. Ensure that the search column is always listed first within the curly braces {}, followed by the column containing the result you want to retrieve.

Another critical consideration is matching the size of the ranges. Both ranges within the array literal must cover the exact same number of rows. For instance, if you use {B2:B9, A2:A10}, the resulting array will be mismatched, leading to errors or inaccurate results. Always verify that your column references start and end on the same rows.

Finally, remember that VLOOKUP inherently stops at the first match it finds. If multiple teams achieved a score of 99, the formula will only return the name of the first team listed in the data that meets the criterion. If you need to retrieve multiple matching results, you should explore more advanced functions such as `FILTER` or the aforementioned `QUERY` function.

Summary of Reverse VLOOKUP Methodology

To summarize, performing a reverse VLOOKUP in Google Sheets requires an intelligent circumvention of the function's inherent limitation. By using the Array literal syntax {}, we can dynamically restructure the relationship between the lookup column and the return column. This technique is invaluable for analysts who require quick, targeted data retrieval without manipulating the core structure of their spreadsheet. While alternatives like `INDEX MATCH` offer greater flexibility, the VLOOKUP array method remains a concise and powerful tool in the Google Sheets arsenal.

Mastering this specific application of VLOOKUP enhances your ability to manage and query data effectively, transforming a rigid function into a versatile tool capable of handling searches in any direction across your data table.