

How to Sum Multiple Rows with VLOOKUP in Google Sheets

Authored by
stats writer

January 17, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Sum Multiple Rows with VLOOKUP in Google Sheets*.
PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=126499>

The VLOOKUP function is one of the most foundational and widely used tools within Google Sheets, designed primarily for vertical data retrieval. Its basic purpose is to search for a specified key value within the first column of a data range and return a corresponding value from a designated column in the same row. While typically engineered to retrieve a single data point, advanced spreadsheet users often seek to leverage VLOOKUP to perform more complex operations, such as summing multiple numeric columns related to that matched key, or, more ambitiously, aggregating values across multiple rows that contain the same lookup criterion. This guide details how to move beyond standard lookups to achieve sophisticated multi-row and multi-column aggregation in your spreadsheets.

Achieving the summation of multiple values requires combining VLOOKUP with other powerful functions like ARRAYFORMULA or SUMPRODUCT. This integration allows the formula to handle arrays of results rather than just single-cell outputs. When dealing with business intelligence tasks--such as summing total sales for a specific product across different regions, or calculating a player's total points across various games--these methods become indispensable. We will explore two distinct methodologies that cater to different aggregation needs: summing values across columns in the **first matched row**, and summing values across **all matched rows** within the dataset.

To effectively manage dynamic data aggregation in Google Sheets, understanding the limitations and extensions of standard lookup functions is key. Below, we introduce the two primary formulas used to perform a VLOOKUP operation that culminates in summing multiple numerical results.

Method 1: Summing Across Columns in the First Matched Row

The first method addresses the need to aggregate data points horizontally--that is, across multiple columns--from the **very first instance** where the lookup key appears. This technique is ideal when your data is structured such that all relevant metrics (e.g., Game 1, Game 2, Game 3 scores) for a unique identifier (e.g., Player ID) are contained within a single row. Standard VLOOKUP only allows returning a single column index; however, by combining it with the ARRAYFORMULA and SUM functions, we can instruct VLOOKUP to return an array of values from several specified columns simultaneously, which are then immediately summed.

The core innovation here lies in using curly braces `{ }` within the column index argument of the VLOOKUP. When VLOOKUP sees an array of column numbers (e.g., `{2, 3, 4}`) instead of a single number, it generates a horizontal array containing the values from columns 2, 3, and 4 corresponding to the matched row. Because VLOOKUP is inherently designed for single-cell output, we must wrap this entire calculation in ARRAYFORMULA to force the evaluation of the array output. Finally, the SUM function aggregates these returned numerical values into a single total.

The formula syntax is as follows:

```
=ARRAYFORMULA(SUM(VLOOKUP(A14, $A$2:$D$11, {2,3,4}, FALSE)))
```

In this specific example, this particular formula sums the values in columns 2, 3, and 4 in the **first row** of the range **A2:D11** where the value in column A is equal to the lookup criterion specified in cell **A14**. The ARRAYFORMULA wrapper is essential for handling the array of column indexes {2, 3, 4} returned by the VLOOKUP, which are then processed by the outer SUM.

Method 2: Aggregating Values Across All Matched Rows Using SUMPRODUCT

The second, and arguably more powerful, method is required when the data you wish to aggregate is distributed across **multiple rows** associated with the same lookup key. Standard VLOOKUP is incapable of handling such a requirement, as it always stops after finding the first match. To overcome this limitation, we turn to the SUMPRODUCT function, which is expertly suited for performing conditional summation and multiplication across entire arrays without requiring the use of ARRAYFORMULA.

The SUMPRODUCT function works by creating a conditional mask. The structure $(A2:A11=A14)$ evaluates every cell in the lookup column (A2:A11) against the criteria (A14). This comparison returns an array of Boolean Logic values (TRUE or FALSE). When this array is multiplied by the numerical data range (B2:D11), Google Sheets coerces TRUE values into 1 and FALSE values into 0. Thus, only the rows where the condition is TRUE (i.e., where the name matches the criteria) are multiplied by 1, allowing their corresponding numerical values to be included in the final summation; non-matching rows are effectively multiplied by 0 and eliminated.

This streamlined approach aggregates the target values for **every row** that satisfies the specified condition, delivering a true conditional sum across multiple columns. The formula is significantly cleaner than the VLOOKUP array method and is generally preferred for calculating grand totals based on a specific criterion that appears multiple times in the source data.

The corresponding formula structure is shown below:

```
=SUMPRODUCT((A2:A11=A14)*B2:D11)
```

Specifically, this particular formula sums the values in columns B, C, and D for each row where the value in column A is equal to the lookup value provided in cell **A14**. This method successfully calculates the aggregate total across all matching rows, achieving a multi-row conditional sum.

Prerequisites: Setting Up the Data for Demonstration

To illustrate these two distinct methods in a practical context, we will utilize a sample dataset that tracks points scored by various basketball players across three separate games. This dataset inherently contains duplicate player names, which is essential for highlighting the difference between Method 1 (First Match) and Method 2 (All Matches).

The data range spans A2:D11 and includes the player's name and their score for Game 1, Game 2, and Game 3. Our goal is to look up a specific player and calculate their total scores based on the requirements of each method. The following examples show how to use each method in practice with the following dataset in Google Sheets:

	A	B	C	D	E
1	Player	Game 1	Game 2	Game 3	
2	Andy	10	8	31	
3	Bob	12	17	9	
4	Chad	14	41	19	
5	Derrick	22	20	24	
6	Erin	29	24	23	
7	Frank	14	24	29	
8	Greg	19	28	34	
9	Henry	10	14	20	
10	Chad	12	15	22	
11	John	33	15	11	
12					
13					
14					
15					
16					

Practical Application 1: Calculating Totals for the Initial Match (Method 1 Walkthrough)

In our first demonstration, we will apply Method 1, which utilizes `VLOOKUP`, `ARRAYFORMULA`, and `SUM`, to find the aggregated score across Game 1, Game 2, and Game 3 for the player listed in the first matching row. This is particularly useful if the dataset is sorted chronologically or by transaction ID, and you only need the cumulative information from the earliest entry.

We can type the following formula into cell **B14**, assuming the lookup value (in this case, "Chad") is located in cell **A14**, to sum the points values scored by Chad in all three games for the first row

found:

=ARRAYFORMULA(SUM(VLOOKUP(A14, \$A\$2:\$D\$11, {2,3,4}, FALSE)))

Once we press **Enter**, the results will be shown, reflecting the sum of the scores only from the first row that matches the criterion:

B14 **fx** =ARRAYFORMULA(SUM(VLOOKUP(A14, \$A\$2:\$D\$11, {2,3,4}, FALSE)))

	A	B	C	D	E
1	Player	Game 1	Game 2	Game 3	
2	Andy	10	8	31	
3	Bob	12	17	9	
4	Chad	14	41	19	
5	Derrick	22	20	24	
6	Erin	29	24	23	
7	Frank	14	24	29	
8	Greg	19	28	34	
9	Henry	10	14	20	
10	Chad	12	15	22	
11	John	33	15	11	
12					
13	Player				
14	Chad	74			
15					
16					

This formula uses VLOOKUP to find "Chad" in the Player column and then returns the sum of the points values for each game found in the **first row** that matches "Chad."

We can see that the formula returns a value of **74**, which is the sum of the points scored by Chad in the first row where "Chad" appears (24 + 30 + 20). This confirms that this methodology aggregates data horizontally but is vertically limited to the initial match.

Practical Application 2: Calculating Grand Totals Across All Matches (Method 2 Walkthrough)

For scenarios requiring a comprehensive aggregate score--where all appearances of the lookup key must contribute to the total--Method 2, leveraging the power of SUMPRODUCT, is the

appropriate solution. Unlike VLOOKUP, SUMPRODUCT scans the entire array, applying conditional multiplication row by row to ensure all matching data is included in the final calculation.

We apply the SUMPRODUCT formula, targeting the player "Chad" specified in cell A14, to retrieve the grand total of scores:

=SUMPRODUCT((A2:A11=A14)*B2:D11)

Once we press **Enter**, the comprehensive results will be shown, encompassing all matching entries:

B14 \sum =SUMPRODUCT((A2:A11=A14)*B2:D11)

	A	B	C	D
1	Player	Game 1	Game 2	Game 3
2	Andy	10	8	31
3	Bob	12	17	9
4	Chad	14	41	19
5	Derrick	22	20	24
6	Erin	29	24	23
7	Frank	14	24	29
8	Greg	19	28	34
9	Henry	10	14	20
10	Chad	12	15	22
11	John	33	15	11
12				
13	Player			
14	Chad	123		
15				

This formula looks up "Chad" in the Player column and then returns the sum of the points values for each game in **each row** that matches "Chad." This process correctly accounts for all instances where Chad is listed.

We can see that the formula calculates that Chad scored a total of **123** points across the two rows he appeared in (74 points from the first row + 49 points from the second row). This demonstrates the efficacy of SUMPRODUCT for multi-row aggregation based on a non-unique lookup criterion.

Conclusion: Choosing the Right Aggregation Method

Selecting the correct formula depends critically on whether your data requirement is based on the first match only (Method 1) or a full aggregation across all matches (Method 2). For most analytical tasks involving non-unique identifiers and the need for a comprehensive total, Method 2 (using **SUMPRODUCT**) is the superior and more straightforward choice.

The following tutorials explain how to perform other common tasks in Google Sheets, building upon these principles of conditional lookups and array manipulation:

ARABPSYCHOLOGY.COM