How to Retrieve the Next Cell Value Using VLOOKUP in Excel

Authored by

stats writer

January 3, 2026

The VLOOKUP function in Excel is a powerful tool for retrieving data from large tables based on a specific lookup value. However, traditional VLOOKUP is inherently rigid; it can only return a value from a column to the right of the search column, and it retrieves the corresponding value from the same row where the match was found. A common requirement, particularly in complex data analysis or financial modeling, is to find a specific entry but then retrieve a value offset vertically--for example, the data point immediately following the matched value in a separate column. Since VLOOKUP lacks the positional control necessary for this task, relying on it directly for vertical offsets is impossible. This guide details the robust and flexible methodology required to achieve this specific offset lookup, moving beyond the standard limitations of simple lookup functions by leveraging the synergistic power of the INDEX and MATCH functions.

Understanding the Limitations of VLOOKUP for Positional Offsets

While VLOOKUP remains a foundational function in Excel for quick data retrieval, its design inherently restricts advanced positional manipulation. The function requires four arguments: the **lookup_value**, the **table_array**, the **col_index_num** (column number to return data from), and the **range_lookup** (determining exact or approximate match). Crucially, the third argument, **col_index_num**, dictates which column within the specified table array contains the result, but it always returns the value from the **same row** where the initial match was found. There is no built-in mechanism to tell VLOOKUP to shift its result vertically, either up or down, to retrieve data from an adjacent record. This rigidity necessitates a more dynamic approach for scenarios requiring vertical offsets, such as retrieving the sales total from the month immediately following the month you searched for, or, as in our specific case, the value in the next cell down.

To overcome this limitation, expert Excel users universally prefer the combination of the INDEX and MATCH functions. This powerful pairing separates the task of finding the position (handled by MATCH) from the task of retrieving the value (handled by INDEX). By isolating these steps, we gain granular control over the row number, allowing us to easily add or subtract a numerical offset (like `+1` or `-1`) to target the specific adjacent cell we need. This technique is not just an alternative to VLOOKUP; it is often considered the superior, more flexible approach for complex lookups across many applications.

The Power Combination: INDEX and MATCH

The methodology for returning an offset value relies entirely on understanding how the INDEX and MATCH functions operate together. In essence, the MATCH function performs the heavy lifting of locating the position of the lookup value. It searches a specified range (a single column or row) and returns the relative numerical position of the match within that range. For instance, if you search for "Apple" in a list starting in A1 and "Apple" is in A5, MATCH returns the number 5. This output is a

critical component because it provides the exact row number needed for the value retrieval step.

Once the positional number is determined, the INDEX function takes over. The INDEX function is designed to return a value at the intersection of a specific row and column within a designated array (range). Its simplest form requires two main arguments: the **array** (the range containing the results) and the **row_num** (the position determined by MATCH). By nesting the MATCH function inside the INDEX function, we dynamically feed the row number to the retrieval function. This combination forms a robust and non-volatile lookup method that mirrors the functionality of VLOOKUP but with added dimensional control.

Implementing the Offset Logic

The key modification that allows us to return the value in the next cell--meaning the cell immediately below the standard match result--is achieved by manipulating the row number returned by the MATCH function. Since MATCH provides the exact position of the lookup value, adding **1** to this result forces the INDEX function to retrieve the value one row further down in the result array. Conversely, subtracting 1 would retrieve the value from the preceding row. This simple arithmetic operation transforms the basic lookup into a dynamic positional query, giving the user complete control over the vertical relationship between the lookup criterion and the desired result.

The resulting structure of the formula is elegantly simple, yet immensely powerful. It looks like this: `=INDEX(Result_Range, MATCH(Lookup_Value, Lookup_Range, 0) + Offset)`. The **Result_Range** is the column from which you want to return the value. The **Lookup_Value** is the item you are searching for. The **Lookup_Range** is the column containing the lookup value. The **Offset** is the integer (`+1` for the cell below, `-1` for the cell above). This particular formula looks up the value, determines its relative row number, adds 1 to shift the position downward, and then retrieves the corresponding value from the specified result range.

Here is the generalized structure that we will use in our example to return the corresponding value in the *next* cell in a different range:

=INDEX(B2:B11,MATCH("Lakers",A2:A11,0)+1)

In this implementation, the formula first finds the position of "Lakers" within the range **A2:A11**. It then increments that position by 1, and finally, INDEX uses this new, offset row number to pull the data from the corresponding position in the results range, **B2:B11**. This ensures that the returned value is always one row below the location of the lookup criterion.

Example: Using INDEX and MATCH for Vertical Offset Lookup in Excel

To illustrate this practical application, consider a scenario involving basketball team statistics, where we have a list of teams and the points scored by their respective players. Our objective is not just to find the points scored by a particular team, but specifically to retrieve the points score listed immediately after that team's entry. This often simulates scenarios where data points are related sequentially, and we need to compare a value against its subsequent entry in a time series or ordered list.

Suppose we have the following dataset in Excel showing points scored by basketball players, organized by team in Columns A and B:

	A	B	C	D	E
1	**Team**	**Points**			
2	Mavs	22			
3	Spurs	29			
4	Rockets	35			
5	Kings	13			
6	Warriors	18			
7	Nets	17			
8	Lakers	20			
9	Thunder	23			
10	Blazers	41			
11	Jazz	36			
12					
13					
14					
15					

Our specific task is to search for the team "Lakers" in the **Team** column (A2:A11) and then return the value from the **Points** column (B2:B11) that is one cell **below** the points value associated with the Lakers. This means if the Lakers are found in row 5, we want the points from row 6. If we were to attempt this using a complex combination of VLOOKUP and Array formula techniques, the syntax would become convoluted and highly inefficient, reinforcing why the INDEX/MATCH method is preferred for precision and clarity.

Step-by-Step Formula Application (Returning the Next Cell)

To execute the lookup and return the value from the subsequent cell, we will place the final formula

in a dedicated cell, such as **D2**. The construction of the formula starts with identifying the three core components necessary for the INDEX/MATCH structure: the return range, the lookup value, and the lookup range. For this specific goal--finding the points score for the entry immediately following the Lakers--we must ensure the crucial **+1** offset is applied directly to the row number generated by the MATCH function.

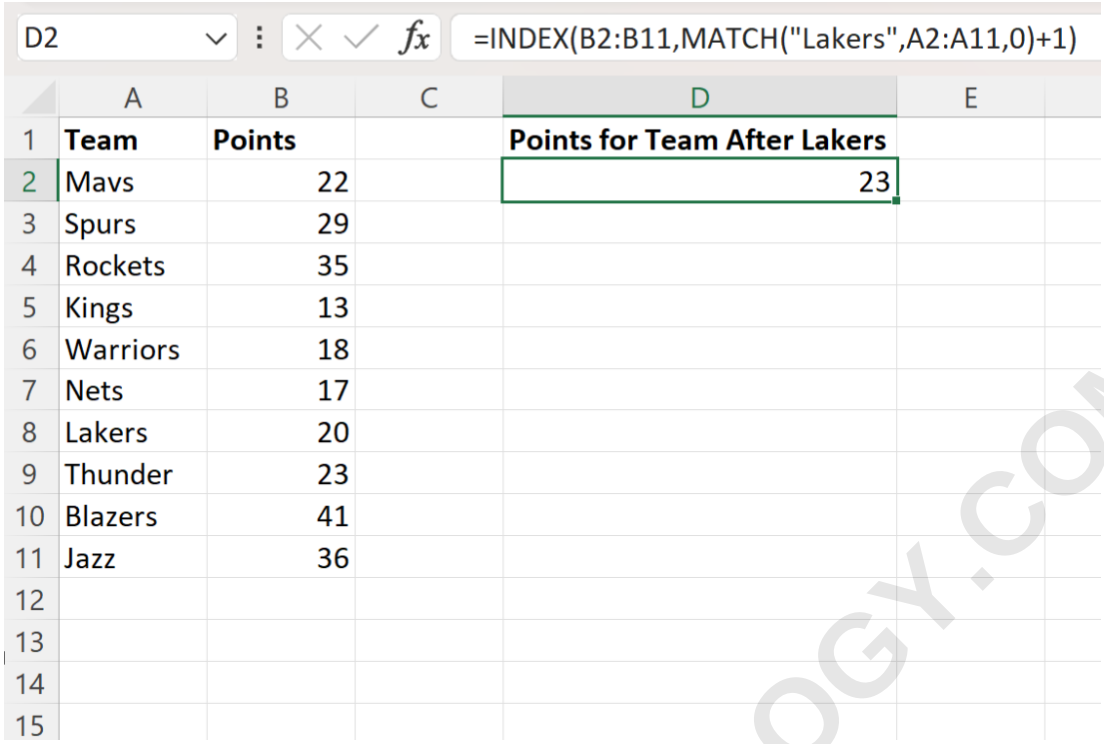
The detailed application involves these steps:

We define the **Return Range** (the array from which the answer will be pulled): `B2:B11` (The Points column).
We use MATCH to find the position of the **Lookup Value**: `MATCH("Lakers", A2:A11, 0)`. Assuming "Lakers" is found in row 5 of the entire sheet, and our range starts at row 2, MATCH will return the position 4 (since it is the fourth cell within the range A2:A11).
We apply the **Offset**: `+1`. This changes the position from 4 to 5, targeting the fifth cell in the return range B2:B11 (which corresponds to row 6 in the spreadsheet).
INDEX uses this offset position (5) to retrieve the value from the **Return Range** (B2:B11).

To execute this, we type the following formula into cell **D2**:

=INDEX(B2:B11,MATCH("Lakers",A2:A11,0)+1)

The subsequent screenshot clearly demonstrates the successful application of this formula within the Excel worksheet, resulting in the correct vertically offset value:



The successful execution of the formula returns a value of **23**. Analyzing the original table confirms that 23 is indeed the value in the **Points** column corresponding to the entry immediately following the "Lakers" entry. This confirms the efficacy of the `+1` offset in achieving the required positional shift relative to the initial lookup match.

Advanced Variation: Returning the Previous Cell (Offset -1)

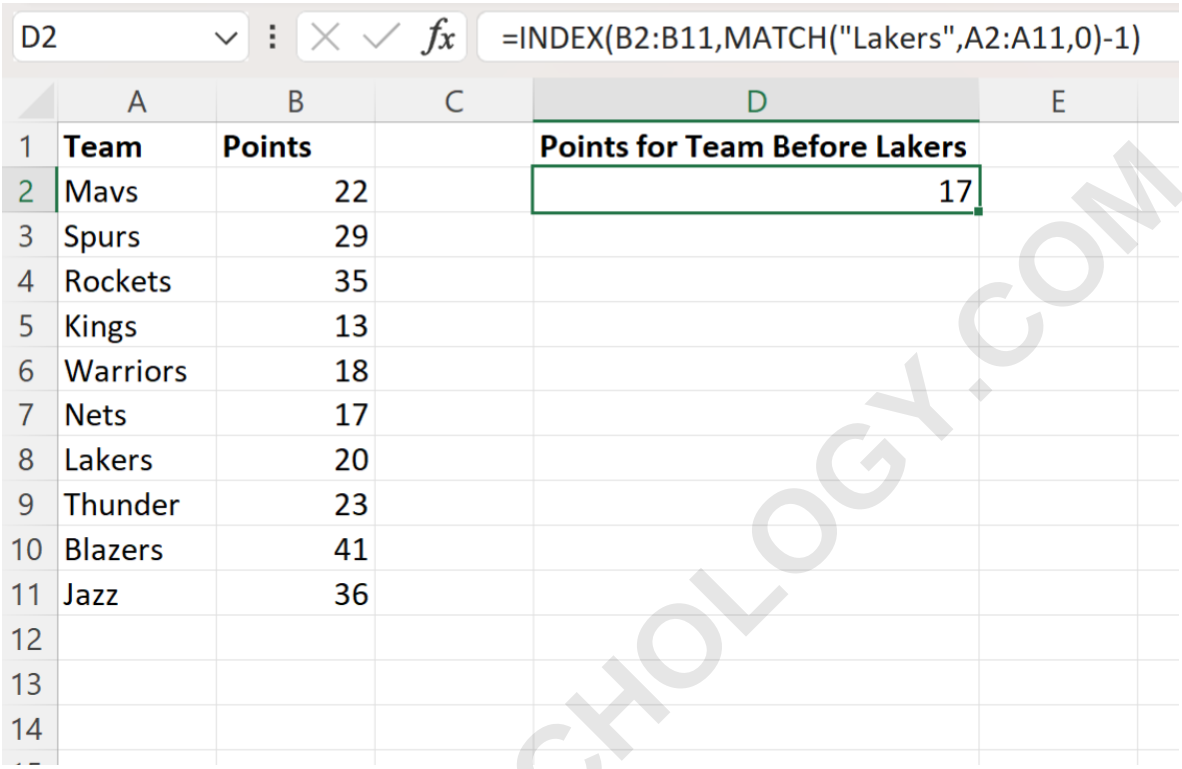
The flexibility of the INDEX and MATCH combination shines when we need to reverse the direction of the lookup. Instead of retrieving the value one cell **below** the match, we may occasionally need to retrieve the value one cell **above** the match. This is particularly useful in analyzing data where a subsequent entry might refer back to its predecessor, or when calculating differences between consecutive records. To accomplish this upward vertical shift, we simply change the positive offset to a negative offset in the formula structure.

If we wanted to return the value from the cell that is one *above* the points value for the Lakers, we would subtract one from the end of the formula instead of adding one. This instructs MATCH to find the position, and then subtracts 1 from that position before passing the final index number to the INDEX function. The structure remains identical, but the arithmetic operation dictates the direction of the retrieval.

The adjusted formula for retrieving the preceding value is as follows:

=INDEX(B2:B11,MATCH("Lakers",A2:A11,0)-1)

The following screenshot demonstrates the practical result of applying this negative offset formula:



In this case, the formula successfully returns a value of **17**. Reviewing the data confirms that 17 is the value in the **Points** column that immediately precedes the points value associated with the Lakers. This demonstrates the superior adaptability of the INDEX/MATCH technique compared to traditional single-function lookups like VLOOKUP, providing precise vertical navigation within data tables.

Considerations for Error Handling and Boundary Conditions

When implementing offset lookups, it is crucial to account for boundary conditions--situations where the offset pushes the row index outside the designated array range. For example, if we search for the first item in the list and apply a `-1` offset (seeking the value above it), the calculated row number will be zero, leading to a `#REF!` error because the INDEX function cannot retrieve a value from a position that is less than one. Similarly, if we search for the last item and apply a `+1` offset, the index will exceed the array size, also resulting in an error. Effective spreadsheet design requires incorporating robust error handling to manage these scenarios gracefully.

The most effective way to manage these potential errors is by wrapping the entire INDEX/MATCH

formula within an `IFERROR` function. This function allows the user to specify a default value or message to display if the lookup calculation encounters an error. For instance, the syntax `=IFERROR(INDEX(B2:B11, MATCH(A2, A2:A11, 0)-1), "N/A")` would return "N/A" if the search item is the first entry and the negative offset attempts to retrieve a non-existent cell above it. Implementing `IFERROR` significantly enhances the user experience and professionalism of the spreadsheet by preventing disruptive error codes from appearing in the output cells.

Summary of Lookup Techniques

The use of the INDEX and MATCH combination provides unparalleled flexibility in data retrieval within Excel, particularly when positional offsets are necessary. Unlike the rigid column indexing of VLOOKUP, this advanced technique allows analysts to precisely control the row number of the result, enabling lookups that retrieve values from adjacent records, regardless of whether those records appear before or after the initial match. Mastering this technique is essential for anyone progressing beyond basic Excel functionality and needing robust solutions for sequential data analysis.