

Learn How to Use XLOOKUP with Multiple Criteria in Excel

Authored by
Mohammed looti

June 4, 2026

RECOMMENDED CITATION

Mohammed looti (2026). *Learn How to Use XLOOKUP with Multiple Criteria in Excel*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=3697>

The [XLOOKUP function](#) in [Excel](#) revolutionized how users perform lookups compared to its predecessors, VLOOKUP and HLOOKUP. While XLOOKUP is powerful on its own, tackling scenarios that require matching against multiple conditions simultaneously often necessitates a creative approach. When you need to retrieve a value based on two, three, or even more criteria being met across different columns, standard usage is insufficient. This guide details the advanced technique utilizing array [concatenation](#) to force XLOOKUP to treat multiple criteria fields as a single lookup value, enabling highly flexible and precise data retrieval within complex spreadsheets. This method is crucial for analysts and data professionals who manage large datasets where unique identification often relies on a combination of attributes.

The core concept involves combining the individual lookup values (the criteria) into one string, and then similarly combining the corresponding lookup columns into one lookup array. This transforms the multi-criteria problem into a standard single-criterion lookup, allowing the function to execute seamlessly. Understanding this process of merging data ranges is fundamental to unlocking the full potential of [XLOOKUP](#) when dealing with specific records that lack a single unique identifier.

Introduction to Advanced Lookup Capabilities in Excel

For many years, achieving multi-criteria lookups in Excel required complex array formulas, often involving the INDEX and MATCH functions combined with [Boolean logic](#), which could be cumbersome and difficult to debug. The introduction of [XLOOKUP](#) provides a cleaner, more intuitive path to solve this common analytical challenge. By leveraging the power of implicit intersection and the ampersand (&) operator, we can effectively merge the criteria and lookup arrays into a single, cohesive unit that the function can process efficiently. This approach simplifies the syntax significantly while maintaining high performance, even when querying extensive [datasets](#).

The following formula structure represents the most robust way to execute a multi-criteria lookup using XLOOKUP. Note how the lookup value argument and the lookup array argument are constructed using the ampersand (&) operator to join multiple cells and ranges, respectively, creating a composite value that must match exactly across all criteria fields.

=XLOOKUP(F2&G2&H2,A2:A13&B2:B13&C2:C13,D2:D13)

This powerful formula is designed to locate the exact value within the result range (here, the cells specified in **D2:D13**) only if all specified conditions are simultaneously satisfied across the corresponding row. If even one criterion fails to match, the entire concatenated string fails the lookup, resulting in an error or a default value, depending on how the optional arguments of [XLOOKUP](#) are configured. It is important to remember that this technique relies heavily on the positional integrity of the data; the criteria cells must correspond exactly to the order of the lookup

columns.

Deconstructing the XLOOKUP Formula for Multiple Criteria

To fully grasp this technique, we must break down the three primary arguments of the XLOOKUP function as they relate to multi-criteria searching. The first argument, the `lookup_value`, is constructed by joining the individual criteria cells (e.g., `F2&G2&H2`). This creates a single, unified search string. If F2 holds "Red," G2 holds "Large," and H2 holds "Cotton," the effective lookup value becomes "RedLargeCotton." This step is the crucial foundation of the method, converting three separate criteria into one actionable search target.

Next, the second argument, the `lookup_array`, must mirror this construction. We use [concatenation](#) to join the corresponding lookup ranges (e.g., `A2:A13&B2:B13&C2:C13`). When [Excel](#) processes this, it effectively creates a temporary, virtual column where each row contains the concatenated values from columns A, B, and C. For instance, Row 5 in this virtual array might contain "BlueSmallWool." The function then searches this temporary, concatenated array for the exact match of the combined lookup value, "RedLargeCotton."

Finally, the third argument, the `return_array` (`D2:D13` in the example), remains standard. This range specifies the values that should be returned upon finding the first match in the virtual lookup array. This particular formula will look for the cell in the range `D2:D13` where the following specific criteria are all met within a single row:

The value in cell range `A2:A13` is equal to the value in cell `F2`

The value in cell range `B2:B13` is equal to the value in cell `G2`

The value in cell range `C2:C13` is equal to the value in cell `H2`

This technique is immensely powerful because it relies on simple string manipulation rather than complex array processing, making it faster and less resource-intensive than some legacy methods. By aligning the criteria cells (F2, G2, H2) with the column ranges (A2:A13, B2:B13, C2:C13) in the exact same order, we ensure that the function is comparing identical structural data, guaranteeing accurate retrieval based on the intersection of all desired conditions.

The Role of Concatenation in Multi-Criteria Searches

The ampersand symbol (&) is the key operator that enables this multi-criteria functionality within [Excel](#) formulas. In this context, it performs [string concatenation](#), merging multiple text or numerical values into a single text string. When applied to ranges, Excel's calculation engine handles the merging element-by-element, creating the temporary lookup array described previously. This process is essential because the [XLOOKUP function](#) is fundamentally designed to search for a single value in a single column or array. By concatenating the input criteria and the lookup

columns, we trick XLOOKUP into treating the combined criteria as that single search value.

While effective, one consideration when using concatenation is the potential for false positives if the data contains overlapping or ambiguous substrings. For example, if one row contains criteria "A" and "B," resulting in the string "AB," and another row contains criteria "A B" (with a space), these would produce different lookup strings. However, if one column contains "12" and the next "34" (resulting in "1234"), and another row contains "123" and "4" (also resulting in "1234"), this would lead to an incorrect match. To mitigate this risk, it is a best practice to insert a unique delimiter, such as a pipe symbol (|) or a hyphen (-), between the concatenated fields both in the lookup value and the lookup array. For instance, using `F2&"|"&G2&"|"&H2` for the lookup value and `A2:A13&"|"&B2:B13&"|"&C2:C13` for the lookup array ensures that the boundaries between criteria are explicitly defined, eliminating ambiguity.

This technique is a perfect illustration of how modern [Excel](#) functionality allows users to work around the inherent limitations of functions by manipulating the input data structure. By relying on the ampersand operator for robust string joining, we can ensure that the multi-criteria search remains precise and efficient, providing accurate results even in complex scenarios where traditional VLOOKUP would fail without significant helper column construction. The following example demonstrates how to apply this formula in a real-world scenario involving a [sports dataset](#).

Practical Example: Analyzing Basketball Player Data

To illustrate the application of XLOOKUP with multiple criteria, let us consider a common scenario involving a [dataset](#) that tracks basketball player statistics. Suppose we have detailed information organized by team, position, starter status, and points scored. Since multiple players might share the same team or position, we need a way to combine these factors to isolate a single, specific record.

Suppose we have the following dataset that contains information about various basketball players:

	A	B	C	D	E	F
1	Team	Position	Starter	Points		
2	Mavs	Guard	Yes	29		
3	Mavs	Guard	No	13		
4	Mavs	Forward	Yes	19		
5	Mavs	Center	Yes	23		
6	Cavs	Guard	No	12		
7	Cavs	Guard	Yes	30		
8	Cavs	Forward	Yes	31		
9	Cavs	Center	No	8		
10	Lakers	Guard	No	12		
11	Lakers	Guard	Yes	19		
12	Lakers	Forward	No	22		
13	Lakers	Center	Yes	24		
14						
15						
16						
17						
18						
19						
20						
21						

The objective is to retrieve the 'Points' value (Column D) for a player who satisfies three distinct conditions simultaneously. This is where the power of the concatenated XLOOKUP formula becomes evident. We are not just looking for a player on the 'Cavs'; we are looking for the player who is on the 'Cavs' **AND** is a 'Guard' **AND** is designated as a 'Yes' starter. If we only used a single criterion lookup, we would likely retrieve the first matching record, which may not be the correct player.

Now suppose we would like to look up the points value for the player who meets all of the following criteria, defined in cells F2, G2, and H2:

Team = **Cavs**

Position = **Guard**

Starter = **Yes**

The criteria cells (F2, G2, H2) contain the desired values. We must now construct the lookup string by joining these three cells and comparing it against the combined lookup array composed of the corresponding data columns (A2:A13, B2:B13, C2:C13). By precisely matching the sequence of

criteria to the sequence of lookup columns, we ensure that the function accurately identifies the unique row that meets all requirements. We can use the following formula to do so:

```
=XLOOKUP(F2&G2&H2,A2:A13&B2:B13&C2:C13,D2:D13)
```

Implementing the Formula and Verifying Results

Implementation of this formula is straightforward. The user simply needs to enter the formula into the desired output cell, which in this demonstration is cell **I2**. Upon pressing **Enter**, [Excel](#) immediately calculates the virtual arrays and returns the corresponding value from the return array. This calculation happens dynamically and does not require the user to explicitly create or manage the temporary concatenated columns, maintaining a clean and efficient workbook structure. This is a significant advantage over methods that rely on helper columns, which can clutter a worksheet.

We can type this formula into cell **I2** and then press **Enter** to execute the lookup. The result is the retrieval of the 'Points' statistic for the player uniquely identified by the combination of Team, Position, and Starter status. The dynamic nature of the [XLOOKUP function](#) means that if the criteria in cells F2, G2, or H2 are changed, the result in I2 will update instantly, providing powerful interactivity for data analysis.

	A	B	C	D	E	F	G	H	I
1	Team	Position	Starter	Points		Team	Position	Starter	Points
2	Mavs	Guard	Yes	29		Cavs	Guard	Yes	30
3	Mavs	Guard	No	13					
4	Mavs	Forward	Yes	19					
5	Mavs	Center	Yes	23					
6	Cavs	Guard	No	12					
7	Cavs	Guard	Yes	30					
8	Cavs	Forward	Yes	31					
9	Cavs	Center	No	8					
10	Lakers	Guard	No	12					
11	Lakers	Guard	Yes	19					
12	Lakers	Forward	No	22					
13	Lakers	Center	Yes	24					
14									
15									
16									
17									
18									
19									
20									

As demonstrated in the resulting image, this [XLOOKUP function](#) is able to successfully look up "Cavs" in the Team column, "Guard" in the Position column, and "Yes" in the Starter column, and subsequently return the associated points value of **30**. This confirms that the concatenated lookup value ("CavsGuardYes") was successfully matched against the concatenated lookup array in the **A2:C13** range, locating the corresponding value in cell **D6**, which holds the value 30. The ability to perform this complex matching in a single, elegant formula underscores the efficiency gain provided by modern [Excel](#) functionality.

We can check the original [dataset](#) and confirm that this is indeed the correct points value for the player that meets all of these criteria. Verification is a critical step in complex data retrieval to ensure that the logic of the formula has executed as intended. By manually cross-referencing the three criteria columns--Team (Cavs), Position (Guard), and Starter (Yes)--we find that only one row satisfies this combination, and the corresponding points value is 30, validating the successful execution of the multi-criteria lookup.

	A	B	C	D	E	F	G	H	I
1	Team	Position	Starter	Points		Team	Position	Starter	Points
2	Mavs	Guard	Yes	29		Cavs	Guard	Yes	30
3	Mavs	Guard	No	13					
4	Mavs	Forward	Yes	19					
5	Mavs	Center	Yes	23					
6	Cavs	Guard	No	12					
7	Cavs	Guard	Yes	30					
8	Cavs	Forward	Yes	31					
9	Cavs	Center	No	8					
10	Lakers	Guard	No	12					
11	Lakers	Guard	Yes	19					
12	Lakers	Forward	No	22					
13	Lakers	Center	Yes	24					
14									
15									
16									
17									
18									
19									
20									
21									

Essential Considerations and Troubleshooting

While the concatenated XLOOKUP technique is highly effective, users must be aware of several essential considerations to ensure stability and accuracy. Firstly, performance can become a factor when dealing with extremely large [datasets](#) (tens of thousands of rows or more) because [Excel](#) is generating a large temporary array in memory. For maximum efficiency, especially in volatile workbooks, it is always recommended to use structured table references instead of static cell ranges (e.g., `Table1` instead of `A2:A1000`). Structured references are generally more readable and dynamic, adjusting automatically as data is added or removed.

Secondly, troubleshooting often centers on data type and formatting issues. If one of the criteria columns contains numerical data and another contains text, ensure that the formatting is consistent, or use the `TEXT()` function to explicitly convert numbers to text within the concatenation process if necessary. Mismatched data types between the lookup value and the lookup array can cause the formula to return an error, typically the `#N/A` error, indicating no match found. Reviewing the explicit concatenated strings using a separate helper cell can often reveal subtle inconsistencies, such as leading or trailing spaces, that prevent an exact match.

Finally, remember the importance of defining what happens when no match is found. The standard

syntax for [XLOOKUP](#) includes an optional fourth argument, `if_not_found`. It is highly recommended to utilize this argument to provide a meaningful response instead of the default `#N/A` error. For instance, modifying the formula to `=XLOOKUP(..., "No Match Found")` provides clearer feedback to the end-user, enhancing the professional quality and usability of the spreadsheet model. Mastering these nuances ensures that the multi-criteria lookup remains robust and reliable under varying data conditions.

Additional Resources for Excel Mastery

To further enhance your proficiency in advanced [Excel](#) data manipulation and lookup techniques, exploring related tutorials is highly recommended. Understanding how to integrate conditional logic and array functions will solidify your ability to handle complex data challenges that extend beyond simple single-criteria lookups. These resources often delve into specialized topics such as dynamic array formulas, advanced filtering, and pivot table construction, all of which complement the power of the [XLOOKUP function](#).

The following tutorials explain how to perform other common operations in [Excel](#), which are often necessary steps in preparing data for successful multi-criteria analysis: