

How to Perform Wildcard Searches with XLOOKUP in Excel

Authored by
stats writer

February 3, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Perform Wildcard Searches with XLOOKUP in Excel*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=129263>

The **XLOOKUP function** in **Excel** represents a significant evolution over older lookup tools like VLOOKUP and HLOOKUP. Its primary purpose is to simplify the process of searching for specific values within a designated range of data and efficiently retrieving corresponding data from a return array. This modern function offers greater flexibility, including the ability to perform lookups in any direction and return entire rows or columns, making it an indispensable tool for advanced spreadsheet management.

While standard lookups require an exact match, the true power of **XLOOKUP** is unlocked when combined with **wildcard** characters. By incorporating symbols such as the asterisk (*) or the question mark (?), users can dramatically enhance the function's capabilities, enabling **fuzzy searches** or partial text matching. This is particularly useful when dealing with messy datasets where entries might contain slight variations, prefixes, or suffixes, providing a level of robustness that streamlines **data analysis** workflows.

Mastering the application of **XLOOKUP** with a **wildcard** allows for highly efficient and accurate data retrieval, especially from large and complex datasets in **Excel**. This technique moves beyond simple exact matches, providing the necessary flexibility to tackle real-world data challenges where absolute precision in search terms is often unattainable. The following sections will provide a detailed, step-by-step guide on how to structure and implement this advanced lookup technique effectively.

Excel: Use XLOOKUP with Wildcard

Understanding Wildcard Characters in Excel

In Microsoft **Excel**, **wildcard** characters are non-literal symbols used to represent one or more characters in a search string. When performing complex lookups, these characters are essential for enabling partial matches. The two most commonly utilized wildcards are the asterisk (*) and the question mark (?). The asterisk represents any sequence of zero or more characters, while the question mark represents any single character.

When implementing a partial text search using the **XLOOKUP function**, the asterisk is the most relevant wildcard, as it allows us to identify entries that merely contain a specific substring, regardless of what precedes or follows it. For instance, searching for "East*" would match "Eastern," "Eastwood," and "East." Conversely, searching for "*ern" would match "Western" and "Eastern." Combining these, as in "*land*", would find any entry containing "land" anywhere within the text string.

To successfully integrate these symbols into your lookup formula, you must concatenate the **wildcard** with the lookup value using the ampersand (&) operator. This construction informs **Excel**

that the search is not for a literal string including the asterisk, but rather for a pattern defined by the combination of the partial text and the wildcard boundaries. This technique is fundamental for conducting effective **fuzzy searches** across large datasets.

Setting Up the XLOOKUP Wildcard Syntax

To leverage wildcard functionality within **XLOOKUP**, a specific syntax for the `lookup_value` argument must be utilized, and importantly, the `match_mode` argument must be configured to enable wildcard recognition. The essential structure uses concatenation to wrap the target text with asterisks. If the partial text is stored in cell `E1`, the modified `lookup_value` appears as `"*"&E1&"*"`.

The standard formula structure for a wildcard **XLOOKUP** search is shown below. This configuration is designed to find the first occurrence of a value that contains the text specified in the lookup cell (`E1`) within the search array (`A2:A11`), returning the corresponding data from the return array (`B2:B11`). Note the inclusion of the final two arguments:

```
=XLOOKUP("*"&E1&"*", A2:A11, B2:B11,,2)
```

This formula finds the first value in the range **A2:A11** that contains the partial text specified in cell **E1**, and returns the corresponding value located in the range **B2:B11**. The crucial element here is the number `2` as the fifth argument, which explicitly sets the `match_mode` parameter to enable **wildcard** matching. Without this setting, the formula would search for the literal string `"*ock*"` and likely fail to find a match.

Parameters of the XLOOKUP Function for Partial Matching

Understanding the standard arguments of **XLOOKUP** is vital before applying wildcards. The function accepts six total arguments, but only the first three are mandatory. For a wildcard search, the fifth argument (`match_mode`) becomes mandatory, while the fourth (`if_not_found`) is often left optional, hence the double comma (`,,`) placeholder:

`lookup_value`: This is the item you are searching for. For partial matching, this must be a concatenated string incorporating the lookup text and the asterisk wildcard (e.g., `"*"&E1&"*"`).

`lookup_array`: The range where the lookup value is expected to be found (e.g., `A2:A11`).

`return_array`: The range containing the data you wish to retrieve upon finding a match (e.g., `B2:B11`).

`if_not_found` (Optional): The value to return if no match is found. If omitted, XLOOKUP returns the standard `#N/A` error.

`match_mode`: This is the critical setting for wildcard searches. Setting this argument to **2** activates

the **wildcard** character match capability. Other modes include 0 (exact match), -1 (exact match or next smaller item), and 1 (exact match or next larger item).

`search_mode` (Optional): This determines the search direction. Default is 1 (search from first to last).

By defining `match_mode` as 2, we explicitly instruct **Excel** to interpret the asterisk and question mark symbols as wildcards rather than literal text. This technical configuration is what differentiates a standard exact-match lookup from a powerful partial-match **fuzzy search**, greatly expanding the scope of solvable **data analysis** problems.

Practical Example: Using XLOOKUP for Partial Text Match

The following example shows how to use this formula in practice. To illustrate the application of wildcards in **XLOOKUP**, let us consider a common scenario involving a dataset where we need to retrieve corresponding numerical data based on a partial text identifier. The goal is to efficiently locate information without requiring the full, precise spelling of the lookup value.

Suppose we have the following dataset in Excel that contains information about points scored by various basketball players:

	A	B	C	D	E
1	Team	Points			
2	Mavs	12			
3	Spurs	31			
4	Rockets	14			
5	Kings	29			
6	Warriors	34			
7	Nets	10			
8	Lakers	26			
9	Thunder	18			
10	Blazers	22			
11	Jazz	15			
12					
13					
14					
15					
16					

Assume we are specifically interested in finding the points scored by the first team whose name

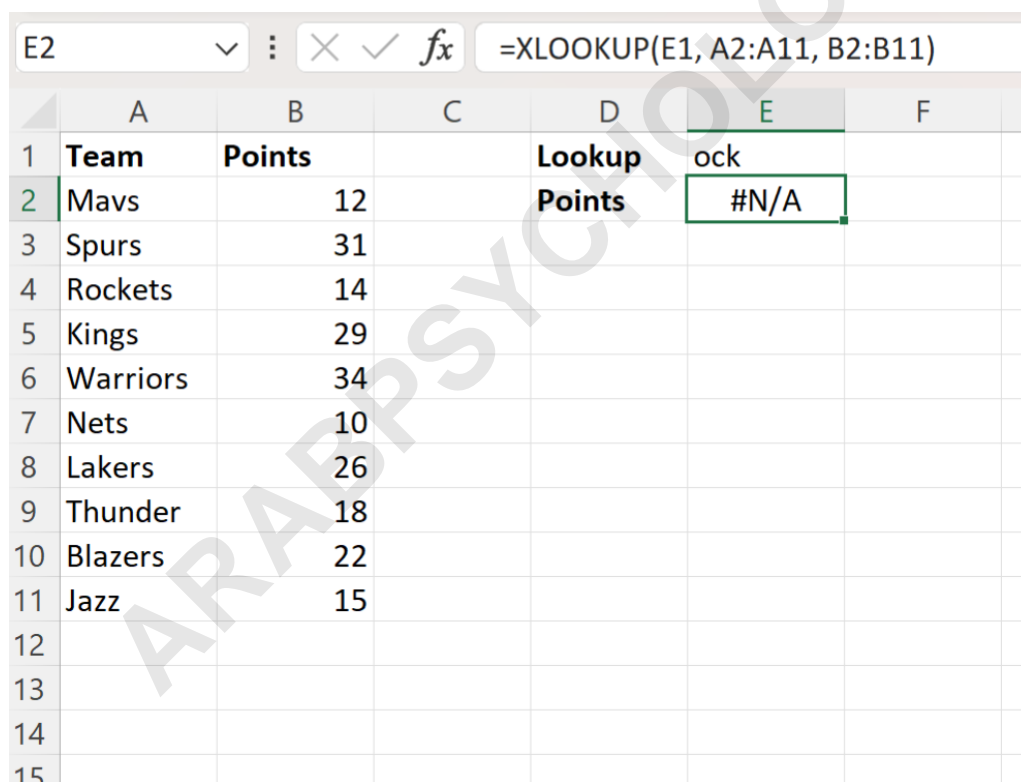
contains the substring "ock." If we place the search term "ock" in cell E1, our objective is to return the corresponding value from the **Points** column. This requires a partial match search, which necessitates the use of the asterisk **wildcard**.

Demonstrating the Failure of Exact Match Lookup

Before implementing the successful wildcard solution, it is instructive to demonstrate why a standard, exact-match **XLOOKUP** fails in this context. If we attempt to use the formula below, searching for the literal term contained in cell E1 ("ock"), the lookup will not succeed:

=XLOOKUP(E1, A2:A11, B2:B11)

The screenshot below illustrates the result of this attempted exact match lookup. Since the value "ock" does not exist as a standalone entry within the **Team** column (range A2:A11), **Excel** correctly returns the standard error message indicating that no exact match was found:



The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F
1	Team	Points		Lookup	ock	
2	Mavs	12		Points	#N/A	
3	Spurs	31				
4	Rockets	14				
5	Kings	29				
6	Warriors	34				
7	Nets	10				
8	Lakers	26				
9	Thunder	18				
10	Blazers	22				
11	Jazz	15				
12						
13						
14						
15						

The formula bar shows the formula: **=XLOOKUP(E1, A2:A11, B2:B11)**. The cell E2 contains the result: **#N/A**.

The formula returns **#N/A** since there is no value in the **Team** column that is equal to "ock" as an exact match. This failure highlights the necessity of enabling the wildcard functionality to conduct the required partial match or **fuzzy search** based on the substring "ock."

Step-by-Step Implementation of the Wildcard Formula

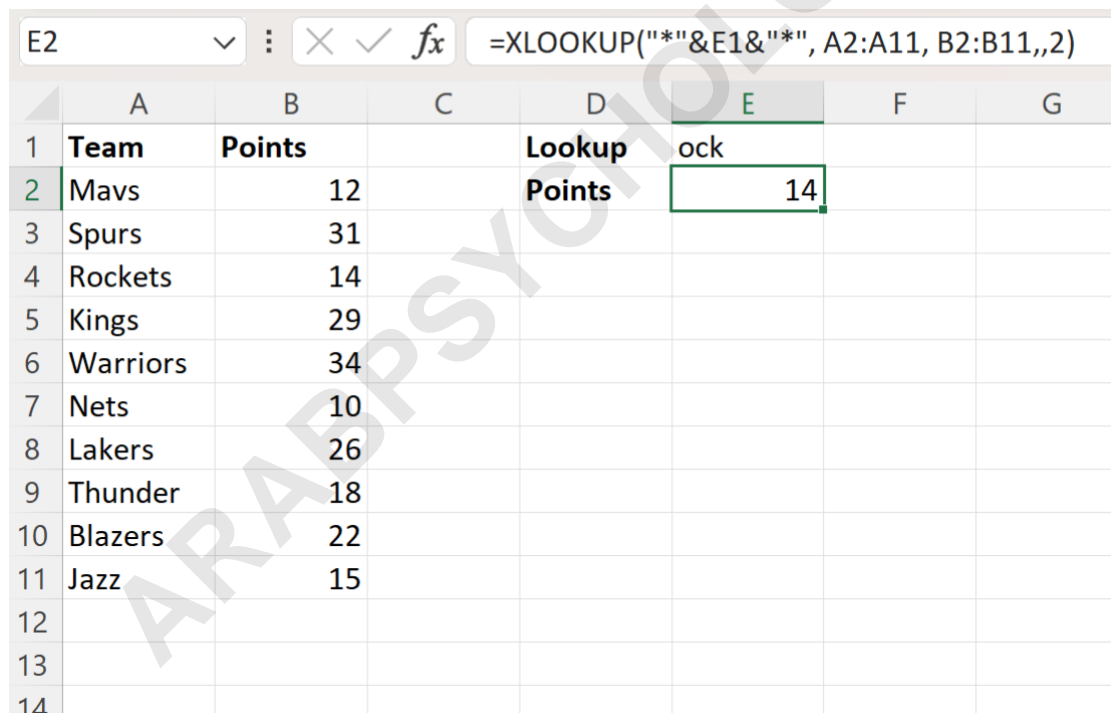
To achieve the desired partial match, we must modify the formula by incorporating the asterisk wildcards around cell E1 and explicitly setting the `match_mode` argument to 2. This modification transforms the search criteria from an exact literal string to a flexible pattern.

We use the following refined syntax:

```
=XLOOKUP("*"&E1&"*", A2:A11, B2:B11,,2)
```

This formula instructs **Excel** to concatenate the asterisk, the content of E1 ("ock"), and another asterisk, resulting in the lookup pattern `"*ock*"`. The `match_mode` of 2 ensures that this pattern is interpreted as a request for any cell in A2:A11 that contains "ock" anywhere within its text. The double comma placeholder skips the optional `if_not_found` argument.

The following screenshot shows how to use this formula in practice:



	A	B	C	D	E	F	G
1	Team	Points		Lookup	ock		
2	Mavs	12		Points	14		
3	Spurs	31					
4	Rockets	14					
5	Kings	29					
6	Warriors	34					
7	Nets	10					
8	Lakers	26					
9	Thunder	18					
10	Blazers	22					
11	Jazz	15					
12							
13							
14							

The formula returns a value of **14**, which represents the value in the **Points** column for the first row that contains "ock" in the **Team** column ("Rockets"). Since `search_mode` defaults to 1 (Search first-to-last), the function retrieves the data associated with the first team name found containing the "ock" substring.

Summary and Further XLOOKUP Resources

Utilizing the **XLOOKUP function** with **wildcard** characters provides a powerful mechanism for overcoming the limitations of exact matching in **data analysis** within **Excel**. By setting the lookup value dynamically using concatenation (e.g., `"*"&value&"*`) and crucially enabling `match_mode=2`, users can efficiently retrieve data based on partial string matches.

This technique is essential for tasks such as cleaning large lists, consolidating data from various sources with inconsistent naming conventions, or performing investigative lookups where only a fragment of the desired text is known. The ability to perform these **fuzzy searches** dramatically improves the speed and reliability of data manipulation compared to relying solely on older, less flexible functions.

Note: You can find the complete documentation for the **XLOOKUP** function in Excel here.

Related Excel Tutorials

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