

# How to Use XLOOKUP to Retrieve Data from Another Sheet in Excel

Authored by  
**stats writer**

February 27, 2026

## RECOMMENDED CITATION

stats writer (2026). *How to Use XLOOKUP to Retrieve Data from Another Sheet in Excel*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=132947>

## An Introduction to XLOOKUP and Cross-Sheet Data Retrieval

In the modern landscape of **data analysis**, the ability to efficiently aggregate information from disparate sources is a fundamental skill. **Microsoft Excel** has long been the industry standard for such tasks, providing a suite of functions designed to manipulate and retrieve data. Among these, the **XLOOKUP** function represents a significant evolution over its predecessors, offering a more robust, flexible, and intuitive approach to searching for specific values within a **spreadsheet**. Unlike older functions such as VLOOKUP or HLOOKUP, **XLOOKUP** does not require the data to be sorted in a specific order, nor does it require the return column to be to the right of the lookup column, making it an essential tool for complex **workbook** management.

The true utility of **XLOOKUP** is realized when users need to bridge the gap between different worksheets. In a typical professional environment, data is rarely confined to a single sheet; instead, it is often distributed across multiple tabs to maintain organization and clarity. For instance, a financial **workbook** might have one sheet for monthly sales and another for employee commissions. By referencing a different sheet name within the **XLOOKUP** formula, users can create a dynamic link that fetches data from one location and displays it in another, thereby streamlining the reporting process and reducing the likelihood of manual entry errors. This cross-sheet capability is vital for maintaining a "single source of truth" within your **spreadsheet**.

Furthermore, utilizing **XLOOKUP** across sheets enhances the scalability of your **data analysis** models. As datasets grow in size and complexity, the traditional methods of copying and pasting data become unsustainable and prone to corruption. **XLOOKUP** allows for a modular design where raw data can be stored in "back-end" sheets while "front-end" dashboards display the calculated results. This separation of concerns not only improves the performance of **Microsoft Excel** but also makes the **workbook** easier to audit and update. In the following sections, we will delve into the technical mechanics of the **XLOOKUP** function and provide a comprehensive guide on how to implement it across different sheets.

## Deconstructing the XLOOKUP Syntax for Multiple Sheets

To master the **XLOOKUP** function, one must first understand its **syntax** and the specific **arguments** it requires. The basic structure of the formula is composed of three primary components: the lookup\_value, the lookup\_array, and the return\_array. When working within a single sheet, these components refer to local cells or ranges. However, when the data resides on a different sheet, the **cell reference** must be prepended with the name of the external sheet followed by an exclamation mark. This informs **Microsoft Excel** exactly where to look for the source data, effectively creating a bridge between the two locations.

Consider the following standard **XLOOKUP syntax** designed for cross-sheet retrieval:

**=XLOOKUP(A2, Sheet2!\$A\$2:\$A\$11, Sheet2!\$B\$2:\$B\$11)**

In this specific example, the first **argument**, **A2**, represents the value you are trying to find. This value is typically located on the active sheet where the formula is being written. The second **argument**, **Sheet2!\$A\$2:\$A\$11**, defines the "lookup\_array"--the vertical or horizontal range on the external sheet where **Microsoft Excel** will search for the match. Finally, the third **argument**, **Sheet2!\$B\$2:\$B\$11**, is the "return\_array," which specifies the range containing the data you wish to retrieve once the match is found.

Understanding the distinction between these **arguments** is crucial for troubleshooting errors. One of the most common mistakes is a mismatch in the dimensions of the lookup\_array and the return\_array. For **XLOOKUP** to function correctly, both arrays must have the same length. If the lookup range spans ten rows, the return range must also span ten rows. By adhering to this structural requirement, you ensure that the function can accurately map the index of the found item to the corresponding index in the return range, providing a seamless data transfer across your **workbook**.

## Setting Up the Workbook Structure for Cross-Sheet Lookups

Before implementing an **XLOOKUP** formula across sheets, it is essential to ensure that your **workbook** is structured logically. Data organization plays a pivotal role in the success of any **data analysis** project. Ideally, each sheet should have a clear purpose. For example, you might designate one sheet as a "Master Data" repository and another as a "Reporting" interface. This separation prevents accidental data deletion and makes it much easier to write and maintain formulas. When **XLOOKUP** is used in a well-organized **spreadsheet**, it acts as the connective tissue that brings these separate components together.

In our practical scenario, we begin with a sheet named **Sheet1**. This sheet serves as our primary workspace where we track basketball player performance, specifically focusing on the points scored by various teams. The data is organized into columns, providing a clear view of the team names and their respective point totals. However, this sheet is incomplete because it lacks information regarding assists, which is stored in a separate location. This is a common situation in business and sports analytics where different metrics are recorded by different departments or at different times, requiring a unified view for comprehensive **data analysis**.

	A	B	C	D	E
1	<b>Team</b>	<b>Points</b>			
2	Mavs	22			
3	Spurs	14			
4	Rockets	19			
5	Kings	20			
6	Warriors	30			
7	Nets	34			
8	Lakers	29			
9	Thunder	15			
10	Blazers	18			
11	Jazz	13			
12					
13					
14					
15					

Sheet1 Sheet2 +

On the other hand, we have **Sheet2**, which acts as our secondary data source. This sheet contains the same team names but focuses on a different metric: **Assists**. For an **XLOOKUP** to be successful, there must be a common identifier between the two sheets--in this case, the **Team** name. This identifier is often referred to as a "foreign key" in database terminology. By ensuring that the team names are spelled identically in both sheets, we create a reliable link that the **XLOOKUP** function can use to navigate between the datasets and retrieve the relevant information from **Sheet2**.

	A	B	C	D	E	F
1	<b>Team</b>	<b>Assists</b>				
2	Blazers	4				
3	Spurs	5				
4	Kings	2				
5	Jazz	4				
6	Warriors	9				
7	Nets	12				
8	Lakers	10				
9	Mavs	5				
10	Thunder	8				
11	Rockets	13				
12						
13						
14						
15						

## Step-by-Step Guide: Retrieving Basketball Stats Across Sheets

To perform a cross-sheet **XLOOKUP**, follow these precise steps to ensure accuracy and efficiency. First, navigate to the sheet where you want the results to appear--in our case, **Sheet1**. Click on the first empty cell where you want to populate the retrieved data, which is cell **C2**. This is where we will construct our formula to bridge the gap to **Sheet2**. By starting the formula with an equals sign, you signal to **Microsoft Excel** that you are initiating a calculation rather than entering static text.

Enter the **XLOOKUP** function and define its parameters. The first **argument** is the lookup value, **A2**, which contains the name of the team "Mavs." Next, you need to point the function to **Sheet2**. You can do this manually by typing **Sheet2!** or by clicking on the **Sheet2** tab while the formula is active and selecting the range **A2:A11**. This range is where **XLOOKUP** will search for the "Mavs" team name. Finally, select the return range on **Sheet2**, which is **B2:B11**, containing the assist values. The final formula should look like this:

**=XLOOKUP(A2, Sheet2!\$A\$2:\$A\$11, Sheet2!\$B\$2:\$B\$11)**

Once the formula is entered, press Enter. You will notice that **Microsoft Excel** instantly retrieves the value "5" for the Mavs, which corresponds to the data found in **Sheet2**. To apply this logic to the entire column, use the "fill handle"--the small square in the bottom-right corner of the cell. Click

and drag it down to cell **C11**. This action copies the formula while automatically adjusting the relative **cell reference** for the lookup value (changing **A2** to **A3**, **A4**, and so on), while keeping the source ranges on **Sheet2** locked in place due to the use of absolute references.

The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F	G	H
1	<b>Team</b>	<b>Points</b>	<b>Assists</b>					
2	Mavs	22	5					
3	Spurs	14	5					
4	Rockets	19	13					
5	Kings	20	2					
6	Warriors	30	9					
7	Nets	34	12					
8	Lakers	29	10					
9	Thunder	15	8					
10	Blazers	18	4					
11	Jazz	13	4					
12								
13								
14								
15								

The formula bar shows: `=XLOOKUP(A2, Sheet2!$A$2:$A$11, Sheet2!$B$2:$B$11)`

## The Role of Absolute References in Cross-Sheet Formulas

A critical component of the cross-sheet **XLOOKUP** formula is the use of **absolute references**, denoted by the dollar sign symbols (**\$**). In **Microsoft Excel**, references are relative by default. This means that if you copy a formula down one row, the references within that formula also shift down by one row. While this is helpful for the lookup value (as you want to look up the next team in the list), it is detrimental for the lookup and return arrays located on another sheet. If those ranges shift, the function may look at empty cells or miss data points entirely, leading to incorrect results or errors.

By applying **absolute references** to **Sheet2!\$A\$2:\$A\$11** and **Sheet2!\$B\$2:\$B\$11**, you are effectively "locking" these coordinates. No matter where you copy the formula in your **workbook**, **XLOOKUP** will always point to those exact cells on **Sheet2**. This is a best practice in **spreadsheet** design, as it ensures the integrity of your calculations. Without these locks, a simple drag-and-drop operation could silently break your **data analysis** model, resulting in difficult-to-detect inaccuracies that could compromise the quality of your findings.

To quickly apply **absolute references** while typing a formula, you can highlight the range and press the **F4** key on your keyboard. This cycles through the various types of referencing (absolute, mixed, and relative). Mastering this keyboard shortcut is a hallmark of an advanced **Microsoft Excel** user and is particularly useful when dealing with **XLOOKUP** operations involving large datasets spanning multiple worksheets. It provides the necessary stability for complex models to function reliably under varying conditions.

## Handling Missing Data and Errors in XLOOKUP

In real-world **data analysis**, datasets are rarely perfect. There may be instances where a team listed in **Sheet1** does not exist in the source data on **Sheet2**. In such cases, a standard **XLOOKUP** will return the **#N/A** error, which indicates that the lookup value was not found. While this is technically accurate, it can make a professional **spreadsheet** look messy and unpolished. Fortunately, **XLOOKUP** includes a built-in **argument** called **if\_not\_found** that allows you to specify a custom message or value when a match is missing.

To implement this, you would add a fourth **argument** to your formula. For example, `=XLOOKUP(A2, Sheet2!$A$2:$A$11, Sheet2!$B$2:$B$11, "Not Found")` would display the text "Not Found" instead of the **#N/A** error. This small addition significantly improves the readability of your reports and provides immediate feedback to the user that data is missing, rather than suggesting a breakdown in the formula's logic. This level of detail is essential when sharing **workbooks** with stakeholders who may not be familiar with **Microsoft Excel** error codes.

Beyond simple error handling, **XLOOKUP** also offers different match modes, such as exact matches (the default) or wildcard matches. Wildcard matching is particularly useful when there are slight variations in how names are recorded across sheets--for instance, "Mavericks" versus "Mavs." By utilizing wildcards, you can create more resilient formulas that are capable of handling minor discrepancies in data entry. This flexibility makes **XLOOKUP** a superior choice for **data analysis** compared to older, more rigid functions that require perfectly matching strings.

## Optimization and Workflow Integration for Large Datasets

When working with exceptionally large **spreadsheets** containing tens of thousands of rows, performance optimization becomes a primary concern. Every **XLOOKUP** formula consumes a small amount of processing power; when multiplied across an entire **workbook**, this can lead to slow calculation times and a laggy user experience. To mitigate this, consider using **Excel Tables** (formatted via **Ctrl+T**). Tables use structured references instead of standard cell addresses, which are not only easier to read but also more efficient for **Microsoft Excel** to process during calculations.

Another advanced technique for large-scale **data analysis** is the use of **Named Ranges**. Instead of referencing **Sheet2!\$A\$2:\$A\$11**, you could define that range as "TeamList" and the return range as "AssistData." Your formula would then become `=XLOOKUP(A2, TeamList, AssistData)`. This approach makes your formulas self-documenting and significantly easier to maintain. If the data on **Sheet2** expands, you only need to update the definition of the **Named Range** once, rather than searching through every sheet to update individual **cell references**.

Finally, always remember to audit your **XLOOKUP** results periodically. In a dynamic **workbook**, data is constantly being added, removed, or modified. By cross-referencing a few random samples manually, you can verify that your cross-sheet formulas are still pointing to the correct locations and returning accurate values. This disciplined approach to **spreadsheet** management ensures that your **data analysis** remains a reliable foundation for decision-making within your organization.

## Conclusion and Further Learning

Mastering the **XLOOKUP** function for cross-sheet operations is a transformative skill that unlocks the full potential of **Microsoft Excel**. By understanding the **syntax**, implementing **absolute references**, and planning your **workbook** structure, you can automate complex data retrieval tasks with ease. This not only saves time but also minimizes the risk of errors that are common in manual **data analysis** workflows. Whether you are managing sports statistics or corporate financial records, **XLOOKUP** provides the precision and flexibility needed to succeed.

To further expand your proficiency in **Microsoft Excel**, consider exploring related functions and features. The following tutorials explain how to perform other common operations in Excel, ranging from advanced conditional formatting to complex **data analysis** techniques that complement your lookup skills:

How to Use Nested XLOOKUP for Multi-Criteria Searches

Mastering INDEX and MATCH for 2D Data Retrieval

Using Power Query to Combine Data from Multiple Workbooks

Best Practices for Spreadsheet Documentation and Auditing

By continuously refining your technical abilities and staying updated with the latest **Microsoft Excel** features, you position yourself as a highly capable analyst in any data-driven environment. The journey of learning **XLOOKUP** is just the beginning of achieving total mastery over your digital workspace.