

How to Extract Unique Values from Multiple Columns in Excel: A Simple Guide

Authored by
stats writer

November 30, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Extract Unique Values from Multiple Columns in Excel: A Simple Guide*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=102721>

Analyzing datasets often requires identifying distinct entries, especially when data is spread across several columns. Finding unique values from multiple columns in **Microsoft Excel** presents a common challenge for data professionals. While Excel offers several built-in features, combining results from non-contiguous ranges requires either intricate formula construction or the use of specialized tools. Historically, two primary techniques have been employed: utilizing the **COUNTIFS function** for quick counting or leveraging the Advanced Filter for extraction.

The **COUNTIFS function** excels at determining the frequency of specific criteria across different ranges, allowing users to quickly ascertain how many unique items exist by setting up complex criteria checks. However, COUNTIFS primarily returns a count, not the list of unique items itself, necessitating further steps to actually extract the list. Conversely, the Advanced Filter is a powerful data management tool designed specifically to extract unique values from a specified range and paste them into a new location or sheet. While highly effective, the Advanced Filter typically operates on a single consolidated column or range, making its application across disparate columns less straightforward without first merging the data, which can be computationally inefficient for large datasets. This article will focus on a robust **array formula** approach that dynamically lists these unique entries without requiring data reorganization, offering a highly efficient and reusable solution for pre-2021 versions of Excel.

The Challenge of Consolidation and Extraction

When dealing with data distributed across multiple columns, the difficulty lies in treating these separate ranges as a single, virtual array for uniqueness testing. Simple methods like using the standard **UNIQUE function** (available in Excel 365/2021) do not natively handle multiple non-contiguous column ranges simultaneously. For users relying on older versions of Excel (2019 and earlier), a dynamic, self-adjusting solution is required--one that can iterate through all cells in the specified columns, check for prior appearances, and sequentially return the novel entries. This extraction process demands a formula capable of handling array operations, where the formula processes multiple values simultaneously rather than just a single cell reference, culminating in the necessity of entering the formula using the special keystroke combination, **Ctrl+Shift+Enter**, to confirm it as an array formula.

The core philosophy behind this complex array method is generating a list of "addresses" (row and column numbers) for every cell containing unique data within the specified multi-column range. Once these addresses are mapped, the formula identifies the smallest numerical address that corresponds to a unique entry and uses redirection functions to retrieve the value stored at that location. This approach ensures that the output list is comprehensive, returning every unique item found across all designated columns, irrespective of their column location. Furthermore, because this solution is formula-based, the output list automatically updates whenever the source data changes, providing significant advantages over static extraction methods like the Advanced Filter,

which require manual re-execution after data modification.

Deconstructing the Complex Array Formula for Uniqueness

To successfully extract the unique values dynamically, we utilize a powerful combination of functions, primarily **INDIRECT**, **TEXT**, **MIN**, **IF**, and **COUNTIF**, alongside **ROW** and **COLUMN** functions for positional mapping. Understanding the role of each component is essential for both implementing and adapting the solution. This intricate formula structure effectively creates an index of all unique occurrences within the defined multi-column dataset, treating the entire source range as one unified entity. The formula essentially searches for non-blank cells that have not yet appeared in the list of results generated so far, marking their position within the source data for extraction.

The formula provided below is specifically designed to perform this matrix operation. It finds the unique values in a specified cell range and returns a sequential list of these unique values. You must commit this formula using the **Ctrl+Shift+Enter** keystroke combination to ensure it functions as an array formula, surrounding the formula with curly braces { } in the formula bar, although these braces are inserted by Excel and should not be typed manually.

You can use the following formula to find unique values from multiple columns in Excel:

```
=INDIRECT(TEXT(MIN(IF(($A$2:$C$11<>"")*(COUNTIF($E$1:E1,$A$2:$C$11)=0),ROW($2:$11)*100+COLUMN($A:$C),7^8),"R0C00"),)&"")
```

This particular formula finds the unique values in the specified cell range **\$A\$2:\$C\$11** and, when dragged down, dynamically returns a list of the unique values in the output column, typically starting at cell **E2** in this example. Note the use of absolute references (\$) to lock the source range, ensuring consistency as the formula is copied down the output column. This structure leverages a combination of cell referencing and array logic to perform what would otherwise require VBA or specialized tools.

Step-by-Step Breakdown: The Core Logic

Analyzing the formula from the inside out reveals its clever mechanism for identifying and locating unique entries across the specified columns. The central component relies on the **IF function**, which utilizes two logical conditions checked simultaneously using array multiplication (the * operator). The first condition, `((A2:C11<>"")`, ensures that the formula only processes non-blank cells within the source range. This prevents empty cells from interfering with the counting process and ensures that only actual data points are considered for inclusion in the unique list.

The second, and arguably most crucial, condition is `(COUNTIF(E1:E1,A2:C11)=0)`. This

segment uses the semi-relative range `E1:E1` (which expands as the formula is copied down, e.g., to `E1:E2`, `E1:E3`, etc.) to check if the value currently being examined in the source range `A2:C11` has already appeared in the output list above the current cell. If the count of that value in the previously generated output is zero, it confirms the value is unique at this stage of the list generation. When both conditions (non-blank AND unique) are true, the IF function proceeds to calculate a unique numerical code for the cell.

The positional encoding `ROW($2:$11)*100 + COLUMN($A:$C)` is the value returned by the IF function when a unique, non-blank match is found. The **ROW function** returns the row number (e.g., 2, 3, 4, etc.), and the **COLUMN function** returns the column number (1 for A, 2 for B, 3 for C, etc.). By multiplying the row number by a large factor (like 100) and adding the column number, we create a unique numerical address (e.g., cell A2 becomes 201, B2 becomes 202, C3 becomes 303). If the conditions are not met (the cell is blank or already listed), the IF function returns a very large number (here, 7^8 , which is 5,764,801), effectively pushing these non-unique addresses to the bottom of the list when we search for the minimum value.

Extracting the Value: MIN, TEXT, and INDIRECT

Once the IF function generates a massive array of numerical addresses (where unique addresses are small and repeated/blank addresses are very large), the **MIN function** is employed to find the smallest valid address code in the entire array. This smallest code corresponds to the next unique entry that has not yet been extracted. The MIN function is the mechanism that ensures the list is populated sequentially, always moving to the next unlisted unique item across the multi-column range.

This numerical address must then be converted back into a cell reference that Excel can understand. This conversion is handled by the **TEXT function** using the "R0C00" format, which interprets the number (e.g., 201) as an R1C1 style reference (Row 2, Column 1). For example, if MIN returns 302, TEXT converts this into the R1C1 string equivalent of cell B3. Finally, the **INDIRECT function** takes this text string (the cell address) and resolves it, retrieving the actual text or numerical value contained within that cell. The trailing "&" ensures that when the formula reaches the point where all unique values have been extracted and only the large placeholder numbers remain (which MIN returns), the formula outputs a blank string instead of an error value, resulting in a clean, self-terminating list.

Practical Application: Setting Up the Data

To illustrate the practical implementation of this powerful array formula, we will use a common scenario involving multiple lists of related, yet potentially overlapping, data. Suppose we are tracking the names of basketball teams across three separate scouting lists or divisions, and we

need a master list of every distinct team name mentioned, regardless of which column it appeared in. This necessitates treating columns A, B, and C as a single source for unique identification.

The source data must be structured clearly, typically starting with headers in Row 1, and the data itself beginning in Row 2. Consider the following dataset, which features basketball team names organized across three columns (A, B, and C) spanning rows 2 through 11:

Example: Find Unique Values from Multiple Columns in Excel

Suppose we have the following three lists of basketball team names in Excel:

	A	B	C	D	E	F
1	Team List 1	Team List 2	Team List 3			
2	Mavs	Magic	Spurs			
3	Spurs	Hornets	Heat			
4	Rockets	Spurs	Rockets			
5	Rockets	Warriors	Rockets			
6	Warriors	Lakers	Nets			
7	Nets	Spurs	Mavs			
8	Heat	Heat	Suns			
9	Lakers	Nets	Heat			
10	Rockets	Hornets	Nets			
11	Kings	Magic	Warriors			
12						
13						
14						
15						
16						
17						

Our objective is to generate the resultant unique list in column E, starting at cell E2. This location is chosen because the COUNTIF component of the formula needs a cell immediately above the starting point (E1) for the initial uniqueness check. Since E1 is directly referenced as the upper bound of the semi-relative range $\$E\$1:E1$, it must be empty or contain text that will not conflict with the actual data being extracted. The process relies on this relative referencing mechanism to ensure that each subsequent cell only checks against the values already generated in the list above it.

Implementing and Confirming the Array Formula

To find the unique values from each of the three columns (A2:C11), we must input the formula into

the first destination cell, **E2**. It is critical that the range references within the formula accurately reflect the source data (**\$A\$2:\$C\$11**) and the correct row range (**\$2:\$11**) and column range (**\$A:\$C**) for positional mapping. Any deviation in these ranges will lead to incorrect or incomplete results. The initial source range defines the scope of the search, while the row and column ranges define the boundaries for positional calculation.

The formula to be entered into cell **E2** is repeated here for clarity:

```
=INDIRECT(TEXT(MIN(IF(($A$2:$C$11<>"")*(COUNTIF($E$1:E1,$A$2:$C$11)=0),ROW($2:$11)*100+COLUMN($A:$C),7^8)),&"R0C00"),)&"")
```

Crucially, after typing or pasting this formula into cell **E2**, we must use the special command combination: pressing **Ctrl+Shift+Enter** simultaneously. This action signals to Excel that the formula should be treated as an array operation, enabling it to process the large multi-cell ranges defined within. Excel will automatically enclose the formula in curly braces, confirming successful array entry, as shown in the subsequent image where the formula is applied.

We can type this complex array formula into cell **E2** and press **Ctrl+Shift+Enter**:

	A	B	C	D	E	F
1	Team List 1	Team List 2	Team List 3			
2	Mavs	Magic	Spurs		Mavs	
3	Spurs	Hornets	Heat			
4	Rockets	Spurs	Rockets			
5	Rockets	Warriors	Rockets			
6	Warriors	Lakers	Nets			
7	Nets	Spurs	Mavs			
8	Heat	Heat	Suns			
9	Lakers	Nets	Heat			
10	Rockets	Hornets	Nets			
11	Kings	Magic	Warriors			
12						
13						
14						
15						

Once the formula is correctly entered in E2, the final step involves replicating the formula down the output column. This is achieved by clicking on the fill handle (the small square at the bottom right

corner of cell E2) and dragging it downwards. As the formula is copied down, the semi-relative range $\$E\$1:E1$ expands (e.g., to $\$E\$1:E2$ in cell E3), allowing the **COUNTIF** component to correctly check against the growing list of already extracted unique items. The formula must be dragged down far enough to account for the maximum possible number of unique entries plus a few extra rows to ensure all unique values are captured and the list terminates with blank cells.

We can then click and drag this formula down to other cells in column E until blank values start appearing, signaling the end of the unique list:

	A	B	C	D	E	F	G
1	Team List 1	Team List 2	Team List 3				
2	Mavs	Magic	Spurs		Mavs		
3	Spurs	Hornets	Heat		Magic		
4	Rockets	Spurs	Rockets		Spurs		
5	Rockets	Warriors	Rockets		Hornets		
6	Warriors	Lakers	Nets		Heat		
7	Nets	Spurs	Mavs		Rockets		
8	Heat	Heat	Suns		Warriors		
9	Lakers	Nets	Heat		Lakers		
10	Rockets	Hornets	Nets		Nets		
11	Kings	Magic	Warriors		Suns		
12					Kings		
13							
14							
15							
16							
17							
18							
19							
20							
21							

The resulting output clearly shows the master list of distinct team names. From the final extracted output, we can observe that there are precisely 11 unique team names across the three source columns (A2:C11). This dynamic list serves as a robust solution for consolidating data from multiple disparate sources within a single worksheet.

Adapting the Formula for Different Datasets

A significant advantage of understanding the component parts of this array formula is the ability to easily adapt it to datasets involving more or fewer columns and rows. The key to modification lies

in consistently adjusting the three critical range components within the formula structure. These must always be updated together to maintain positional integrity for the extraction logic. If, for instance, a fourth column (D) were added to the dataset, every instance of the column range C would need to be extended to D.

The three ranges requiring modification are:

The main source range used for the IF and COUNTIF uniqueness check (e.g., **\$A\$2:\$C\$11**).

The ROW range defining the potential row numbers (e.g., **\$2:\$11**).

The COLUMN range defining the potential column numbers (e.g., **\$A:\$C**).

For example, if the data spanned from Column A to Column F and Rows 5 through 50, the formula ranges would need to be updated to **\$A\$5:\$F\$50**, **\$5:\$50**, and **\$A:\$F**, respectively. Failure to update all three corresponding ranges correctly will lead to errors, such as returning incorrect values or misidentifying unique entries due to faulty numerical address generation. It is also important to ensure that the initial output cell (e.g., E2) and the cell immediately above it (E1) are correctly set up relative to the semi-relative range **\$E\$1:E1**, as this controls the sequential uniqueness check.

Note: To use this formula with more columns or rows, simply modify the **\$A\$2:\$C\$11** cell range in all its appearances, along with the corresponding **ROW** and **COLUMN** ranges, to accurately include the new boundaries of your source data. This flexibility makes the array formula highly versatile for users who cannot access newer Excel features.

Modern Alternatives: Leveraging the UNIQUE Function (Excel 365/2021)

For users who have access to modern versions of **Microsoft Excel**, specifically those with the dynamic array capabilities introduced in Microsoft 365 or Excel 2021, the entire process of finding unique values across multiple columns is dramatically simplified. These modern versions introduce the powerful **UNIQUE function**, which is part of the dynamic array functionality, enabling complex array operations without the need for **Ctrl+Shift+Enter**. Dynamic array functions "spill" their results automatically, eliminating the need to drag the formula down manually.

While the standard UNIQUE function handles single ranges effortlessly, combining non-contiguous columns requires a preliminary step: consolidating the data into a single array using functions like **VSTACK** (for vertical stacking) or **CHOOSE**. If the data is truly sparse or non-contiguous, the recommended approach is to use the **CHOOSE function** to virtually stack the column arrays before feeding them into UNIQUE. For instance, to stack columns A, B, and C (rows 2 to 11) using CHOOSE, the formula would look something like `=UNIQUE(CHOOSE({1;2;3}, A2:A11, B2:B11, C2:C11))`. This single, non-array formula performs the entire task instantly, demonstrating the significant efficiency gains provided by the newer Excel versions over the complex historical array

formulas, effectively replacing the need for **INDIRECT** and the complex positional calculations.

Conclusion: Selecting the Right Technique

Finding unique values across multiple columns in **Excel** can be approached using several techniques, each suited to different versions of the software and specific requirements. For those utilizing pre-2021 versions of Excel, the complex **Ctrl+Shift+Enter array formula** involving **INDIRECT** and **COUNTIF** remains the most robust and dynamic solution for generating a self-updating list. This method requires a deep understanding of array logic and positional mapping but delivers unparalleled flexibility for legacy systems.

Alternatively, the **Advanced Filter** provides a simple, wizard-based graphical user interface for one-time extraction tasks, suitable if the source data is static or if merging columns temporarily is acceptable. Finally, users on modern Excel platforms should prioritize the streamlined approach provided by the **UNIQUE function** combined with stacking functions like **CHOOSE** or **VSTACK**. Mastery of these techniques ensures efficient data management, regardless of the complexity or dispersion of the source data across the worksheet.