

Learning to Create a Unique List from Multiple Columns in Google Sheets

Authored by
Mohammed looti

October 30, 2025

RECOMMENDED CITATION

Mohammed looti (2025). *Learning to Create a Unique List from Multiple Columns in Google Sheets*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=6333>

Introduction to Efficient Data Management in Google Sheets

In the contemporary, data-driven environment, the ability to effectively manage and refine information is crucial for accurate decision-making. A frequent and significant challenge encountered by users of powerful spreadsheet applications like [Google Sheets](#) is the presence of [duplicate data](#). Such redundant entries can severely compromise analytical results, diminish operational efficiency, and ultimately lead to flawed insights. Therefore, mastering the extraction of unique values is a foundational skill in modern data handling. Ensuring high [data integrity](#)--whether compiling extensive lists, aggregating survey responses, or overseeing complex inventories--begins with the precise identification and isolation of distinct entries.

[Google Sheets](#) is equipped with robust, built-in [formulas](#) designed to streamline the deduplication process, especially when unique values are distributed across several columns. This comprehensive guide will detail two distinct, highly effective methodologies for generating unique lists from multi-column ranges. The selection between these methods hinges entirely upon your primary objective: do you need to extract unique rows, thereby preserving the original structure of your data, or do you require all unique individual values compiled into a single, comprehensive list?

A deep understanding of these specific [formulas](#) will significantly elevate your data manipulation capabilities within the platform. We will systematically explore each formula, providing detailed practical examples and transparent explanations to ensure you can confidently apply these advanced techniques to manage and refine your own [datasets](#), transforming raw data into reliable, structured information.

Core Formulas for Multi-Column Uniqueness

To efficiently create a unique list of values sourced from data spanning multiple columns in [Google Sheets](#), we rely upon the power of two primary approaches, each addressing a specific need in data cleaning and consolidation:

Formula 1: Extracting Unique Rows (Preserving Columnar Structure)

=UNIQUE(A2:C20)

This formula is engineered specifically to extract entire rows where the combination of values across those columns is distinct. It is the ideal solution when you must preserve the original columnar structure of your [dataset](#) while simultaneously eliminating identical records. The `UNIQUE` [function](#) evaluates every row as a single, complete entity. It returns only the first occurrence of any row whose composite values are not duplicated elsewhere within the designated range. For instance, the range `A2:C20` represents the specific block of cells you intend to analyze for these unique, full-row combinations.

Formula 2: Consolidating All Unique Values into a Single Column

=UNIQUE(FLATTEN(A2:C20))

This alternative approach employs the powerful `FLATTEN` [function](#) in conjunction with `UNIQUE` to achieve a dramatically different output. The `FLATTEN` function takes any multi-column range and efficiently transforms it into a single column, essentially stacking all data entries vertically to form a [dynamic array](#). Once the data is flattened, the `UNIQUE` [function](#) then processes the resulting single column. It processes the entire list, returning every individual unique value, regardless of its initial column or row position. This method is indispensable when your goal is to generate a comprehensive master list of all distinct items, tags, or categories present across your entire [dataset](#).

The following practical scenarios will clearly illustrate how to implement each of these robust [formulas](#) effectively in real-world data management challenges, providing clear insights into their operational differences and the resulting outputs.

Example 1: Creating a Unique List of Rows Across Multiple Columns

Consider a common data management task: maintaining a roster of basketball players, where the [dataset](#) includes the player's name and their corresponding team. Due to factors such as merged reports or repeated data entry, it is highly likely you will encounter duplicate entries where the exact player-team combination is repeated. Our objective in this example is strictly to obtain a clean, consolidated roster where every unique player-team pairing appears exactly once.

Below is the initial [dataset](#) containing information for various basketball players:

	A	B	C	D	
1	Team	Position			
2	Mavs	Guard			
3	Mavs	Guard			
4	Mavs	Forward			
5	Mavs	Forward			
6	Mavs	Center			
7	Spurs	Guard			
8	Spurs	Guard			
9	Spurs	Forward			
10	Spurs	Forward			
11	Spurs	Center			
12					
13					
14					
15					
16					
17					
18					
19					
20					

As clearly visible in the source data, certain rows, such as the record for "Michael Jordan" on "Bulls" and "LeBron James" on "Lakers," are repeated. To effectively consolidate this list and retain only the distinct player-team combinations, we must employ the `UNIQUE` [function](#). This [formula](#) meticulously assesses the combined data within each row, returning only the first occurrence of any identical composite row entry.

We implement the following [formula](#) to filter the range and return only the truly unique rows across both columns:

=UNIQUE(A2:B11)

By inputting this formula into an available empty cell (e.g., D2) in your spreadsheet, [Google Sheets](#) automatically generates the results as a [dynamic array](#). Crucially, the output maintains the exact columnar structure of the original data, ensuring that the Player and Team columns remain aligned and visible.

The resulting clean output is demonstrated in the screenshot below:

	A	B	C	D	E
D1				=UNIQUE(A2:B11)	
1	Team	Position		Mavs	Guard
2	Mavs	Guard		Mavs	Forward
3	Mavs	Guard		Mavs	Center
4	Mavs	Forward		Spurs	Guard
5	Mavs	Forward		Spurs	Forward
6	Mavs	Center		Spurs	Center
7	Spurs	Guard			
8	Spurs	Guard			
9	Spurs	Forward			
10	Spurs	Forward			
11	Spurs	Center			
12					
13					
14					
15					
16					
17					
18					
19					

This illustration confirms that the formula successfully returns a highly condensed list containing only the truly unique rows from the source data. Every player-team pairing now appears precisely once, providing an accurate and reliable record. This method proves invaluable for creating unique attendance registers, managing inventory records based on distinct product codes, or compiling clean customer interaction logs.

Example 2: Creating a Unique List of Values Consolidated into One Column

Now, let us consider a scenario where your data details player positions across various basketball teams, with positions potentially listed across multiple columns (e.g., Primary Position, Secondary Position, Tertiary Position). Unlike the previous example, your objective is not to find unique rows, but rather to aggregate every single unique position mentioned across the entire range, regardless of its originating column, and present it as a single, consolidated list. This is the perfect application for combining the `UNIQUE` and `FLATTEN` [functions](#).

Observe the following source data, which contains player positions distributed across three columns:

	A	B	C	D	E
1	Team 1	Team 2	Team 3		
2	Guard	Guard	Guard		
3	Guard	Forward	Guard		
4	Forward	Forward	Guard		
5	Forward	Forward	Center		
6	Center	Forward	Center		
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					

Positions like "Guard," "Forward," and "Center" are repeated numerous times across different columns. If you were to apply the `UNIQUE` function alone to this multi-column range, it would only identify unique rows (e.g., a row containing "Guard," "Forward," "Center" is unique from a row containing "Guard," "Guard," "Forward"). Our specific requirement, however, is to extract every unique **individual** position, independent of its context within the row or column.

To successfully achieve this consolidation, we first must convert the two-dimensional range `A2:C6` into a one-dimensional list. This critical step is accomplished using the `FLATTEN` function, which stacks all values from columns A, B, and C into a single, continuous column. Following this transformation, the `UNIQUE` function processes the resulting flattened list, effectively eliminating all duplicate individual text strings and presenting only the distinct positions.

=UNIQUE(FLATTEN(A2:C6))

Placing this formula into an empty cell (e.g., E2) instantly generates a clean, vertical list of every unique position found throughout the defined range. This technique is extremely efficient for tasks such as consolidating category tags, keywords, or descriptive values that are frequently dispersed across various spreadsheet fields.

The screenshot below illustrates the result of applying this powerful combined formula:

	A	B	C	D	E
1	Team 1	Team 2	Team 3		Guard
2	Guard	Guard	Guard		Forward
3	Guard	Forward	Guard		Center
4	Forward	Forward	Guard		
5	Forward	Forward	Center		
6	Center	Forward	Center		
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					

The output clearly demonstrates the formula's success in isolating and displaying only the unique positions across all three original columns. All instances of "Guard," "Forward," and "Center" have been successfully reduced to single, distinct entries. This consolidated view is often essential for building filter menus, generating summary reports, or populating dropdown validation lists where a definitive inventory of all possible values is required.

Advanced Considerations and Optimization Techniques

While the `UNIQUE` and `FLATTEN` functions provide extraordinary power for generating unique lists, optimizing their application and understanding potential pitfalls is key to advanced data management. Addressing these nuances ensures that your solutions are robust, precise, and perform efficiently even when dealing with complex or evolving data structures.

One common issue is the handling of blank cells. By default, both `UNIQUE` and `FLATTEN` will often treat empty cells within your specified range as a unique value, which may clutter your final list. To effectively exclude these blank entries, you should nest your `UNIQUE` function within a

`FILTER` [function](#) to systematically remove empty strings. For instance, the expanded formula `=FILTER(UNIQUE(FLATTEN(A2:C6)), UNIQUE(FLATTEN(A2:C6)))` would first flatten and find unique values, and then exclude all resulting blanks from the final output.

Another crucial consideration is case sensitivity. [Google Sheets'](#) `UNIQUE` function is case-sensitive by default, meaning that "apple" and "Apple" are registered as two distinct unique values. If your goal requires a case-insensitive unique list, you must first normalize all text entries to a consistent case (either upper or lower) before applying the deduplication logic. A robust solution uses `ARRAYFORMULA` alongside `LOWER`, such as in the formula `=UNIQUE(ARRAYFORMULA(LOWER(FLATTEN(A2:C6))))`. This structure converts all values in the range to lowercase, then flattens them, and finally extracts the unique entries, successfully treating variations like "APPLE," "Apple," and "apple" as a single unified entry.

Finally, maintaining awareness of performance is essential when working with very large [datasets](#). While Google Sheets is engineered to be highly robust, extremely large ranges or overly complex nested formulas can occasionally introduce calculation latency. If you anticipate performance issues, it is advisable to test your [formulas](#) on smaller data subsets or consider decomposing very complex operations into several sequential, simpler steps to maintain optimal processing speed.

Conclusion: Gaining Control Over Your Data

Mastering the techniques for creating unique lists is the cornerstone of efficient data analysis and comprehensive data management within the Google Sheets environment. By confidently applying the `UNIQUE` function--whether used independently to define unique rows or combined with the `FLATTEN` [function](#) to create a consolidated list of individual unique values--you gain precise, granular control over your data outputs. These highly effective formulas empower users to transform disorganized, redundant source data into clean, readily actionable information.

Whether your immediate task involves cleaning raw inputs, preparing specialized reports, or architecting interactive dashboards, the solutions detailed in this guide provide flexible and powerful answers to pervasive data challenges. We highly recommend experimenting with these formulas using your own data samples, adapting them to meet specific requirements, and unlocking the full potential of Google Sheets for superior, professional-grade data management.

Additional Resources for Google Sheets Proficiency

To further enhance your skills and explore more sophisticated operations within Google Sheets, we encourage you to delve into the following related tutorials. These resources cover a spectrum of common tasks and advanced functionalities that can further streamline your workflow and dramatically improve your data analysis capabilities: