How to Remove Duplicate Rows in Google Sheets Using One Column

Authored by

mohammed looti

January 9, 2026

Understanding the Need for Data Normalization in Google Sheets

Data management often necessitates the identification and removal of redundant records to ensure accuracy and efficiency. When working within Google Sheets, encountering duplicate values is a common challenge, especially when compiling data from multiple sources or performing data entry tasks. While duplicates across an entire row are relatively easy to manage, the requirement often narrows down: removing entire rows based solely on a duplicate entry in one specific column. This is crucial for tasks like creating mailing lists where only one record per customer ID is needed, or generating summary statistics where each unique identifier should appear just once. Maintaining high quality, normalized data is paramount for reliable analysis and reporting, making the ability to quickly and accurately eliminate these redundancies an essential skill for any power user.

The native capabilities of Google Sheets provide powerful tools for this purpose, specifically designed to handle complex data transformation tasks without requiring extensive scripting or complex formulas. The goal is not just to identify the presence of duplicate entries, but to initiate a process that systematically deletes the entire corresponding row, ensuring that the remaining dataset retains only one representation of each unique value found in the designated column. This operation fundamentally alters the dataset, transforming raw, potentially messy input into clean, structured information ready for further processing or visualization. Understanding how this built-in functionality works is the most straightforward path to achieving swift and reliable data hygiene.

The Primary Method: Utilizing the "Remove Duplicates" Feature

The most efficient and user-friendly approach for tackling redundancy in Google Sheets involves accessing the dedicated **Remove Duplicates** feature. This tool is specifically housed within the **Data cleanup** submenu, providing a streamlined interface designed to prevent accidental data loss while giving the user precise control over the removal criteria. Instead of relying on manual sorting and filtering--a tedious and error-prone process--this feature automates the comparison across thousands of cells, identifying rows that share the same value in the chosen column and retaining only the first instance encountered. It is important to note that this action is permanent unless immediately undone using the standard undo command, emphasizing the need for careful execution and, ideally, working on a copy of the original data.

To initiate the process, you must first select the range of data you wish to analyze. Critically, even if you are only checking for duplicates in one column (e.g., Column A), you must select all columns (e.g., Columns A through Z) of the relevant rows if you intend for the corresponding entire row to be deleted. If you only select the single column, the removal process will only affect that column's cells, potentially misaligning the rest of your spreadsheet data and leading to catastrophic integrity issues. Once the entire range is highlighted, navigating to **Data**, then **Data cleanup**, and finally selecting **Remove duplicates** opens the configuration dialog box, which is the control center for

defining the parameters of the operation.

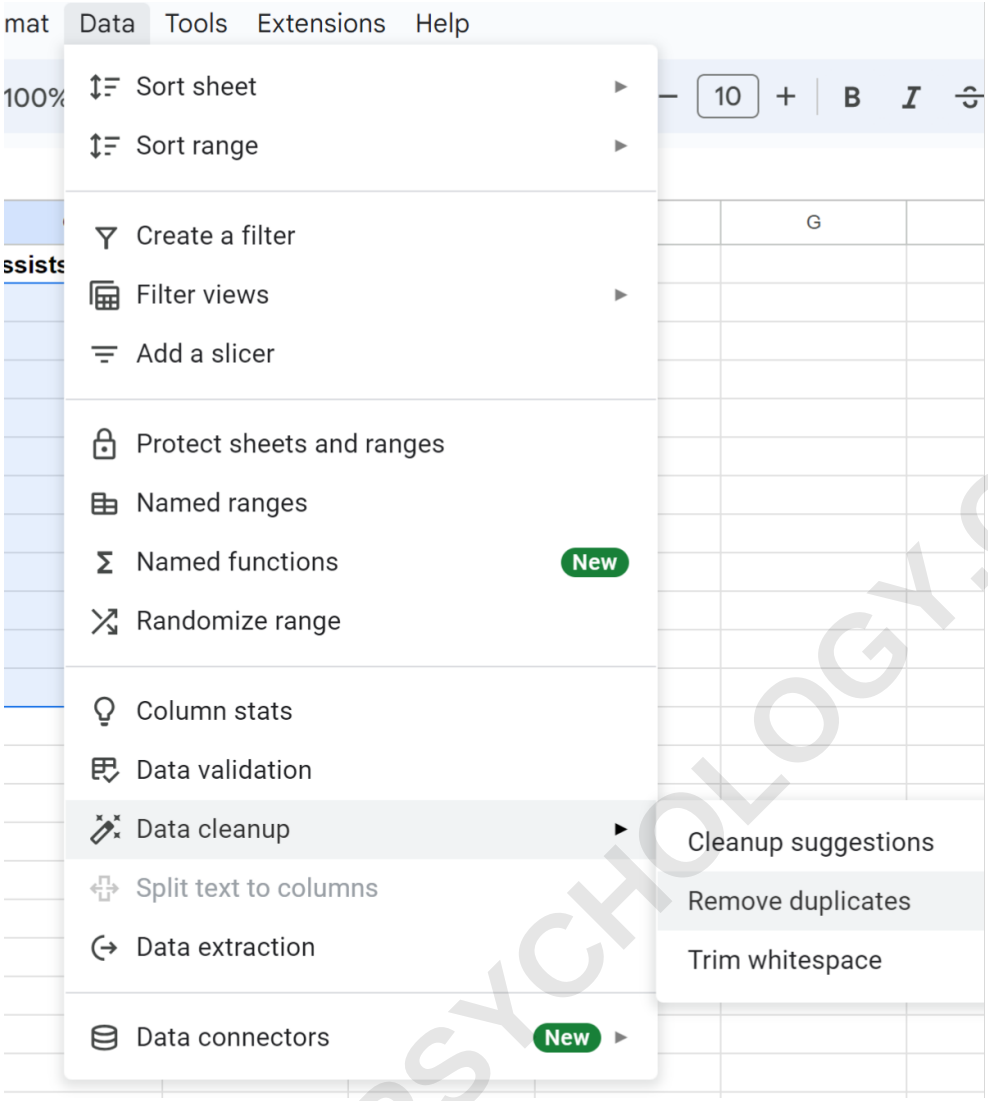
Step-by-Step Implementation: Executing the Duplication Check

Let us walk through a practical scenario involving a dataset of basketball players, where we want to ensure that each listed **Team** appears only once. Suppose we have the following initial dataset loaded into our spreadsheet:

	A	B	C	D
1	**Team**	**Points**	**Assists**	
2	Mavs	22	4	
3	Mavs	14	7	
4	Mavs	19	7	
5	Rockets	30	3	
6	Spurs	25	10	
7	Spurs	24	6	
8	Rockets	36	15	
9	Warriors	22	8	
10	Rockets	13	3	
11	Mavs	18	5	
12	Spurs	11	12	
13				
14				
15				
16				

As observed, there are clearly several duplicate values present in the **Team** column. Our objective is to perform a targeted removal operation that eliminates all rows where the team name is repeated, preserving only the first instance of each team entry. The process begins by meticulously selecting the entire relevant range of data. In this specific example, assuming the data spans from row 2 down to row 12 across columns A and B, we must highlight the cell range **A2:B12**. Selecting the header row (row 1) should generally be avoided unless the header row itself contains duplicates you wish to analyze, which is rare in standard data management tasks.

Once the data range is correctly highlighted, the next steps involve interacting with the menu bar features. Click the **Data** tab, then proceed to click **Data cleanup**. Within the dropdown menu that appears, select **Remove duplicates**. This action triggers the crucial configuration window:

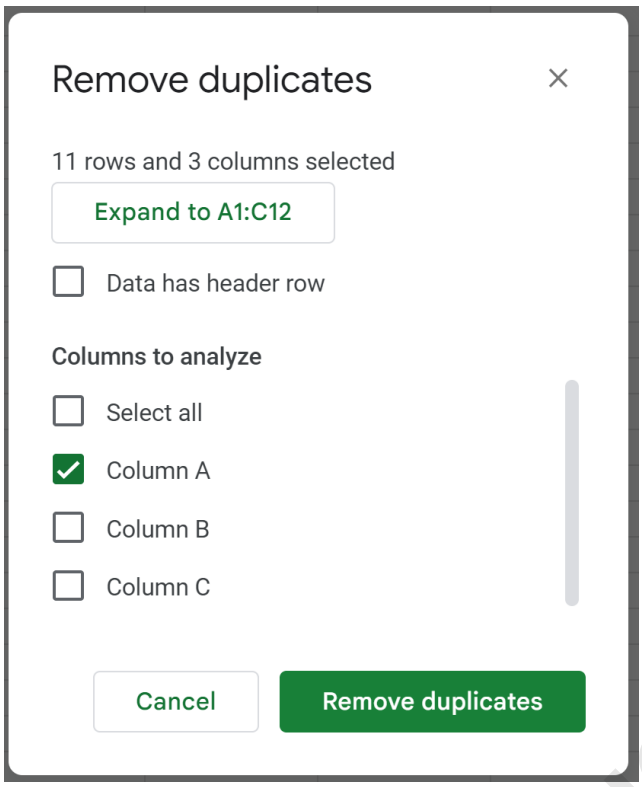


The configuration dialog box provides options for defining the scope of the duplication check. Since our goal is to identify duplicates based exclusively on the **Team** column (Column A), we must ensure that only the checkbox corresponding to **Column A** is selected under the **Columns to analyze** section. If multiple columns were checked, the tool would only remove a row if **all** checked columns had identical values across two or more rows. Because we only require uniqueness based on one column, we isolate the analysis to that specific field, ensuring maximum precision in the data reduction process.

Analyzing the Configuration and Confirmation

When the configuration window appears, pay close attention to the options presented. Although you selected the full range (A2:B12) in the previous step, the dialog box allows you to refine which column or columns must contain the matching data for a row to be considered a duplicate. In our basketball team example, we must deselect Column B (Player Name) and ensure that only Column

A (Team) remains active, as illustrated below:



It is standard practice to verify the option that states "Data has a header row." If you included your headers in the initial selection (A1:B12), checking this box prevents the tool from treating the header row as potentially removable data. However, in our current example, we selected the data range **A2:B12**, intentionally excluding the header, making this checkbox irrelevant but still a crucial setting to be aware of when performing data cleanup on larger datasets. After confirming that only the target column (Column A) is selected, clicking the **Remove duplicates** button executes the procedure immediately.

The system will then process the data, comparing every entry in the designated column against all others. For every instance where a value in Column A is repeated, the subsequent rows containing that value will be deleted entirely. The row that is retained is typically the first occurrence of that unique value within the selected range, preserving its associated data in Column B and any subsequent columns. This ensures that while the duplicate team names are eliminated, the context (the first listed player associated with that team) is kept intact.

Reviewing the Results of the Deletion Process

Upon completion, Google Sheets provides a summary message detailing how many duplicate rows were found and removed, and how many unique rows remain. This immediate feedback confirms

the successful execution of the task and allows for a quick assessment of the outcome. In our basketball dataset example, the resulting clean data set looks significantly smaller and is now guaranteed to have only unique team entries:

	A	B	C	D
1	**Team**	**Points**	**Assists**	
2	Mavs	22	4	
3	Rockets	30	3	
4	Spurs	25	10	
5	Warriors	22	8	
6				
7				
8				
9				
10				
11				
12				
13				
14				

A careful inspection confirms that all rows containing duplicate values in the **Team** column (Column A) have been successfully removed. According to the expected process outcome for this specific dataset, **7** rows with duplicate entries in the **Team** column were deleted, leaving exactly **4** rows that represent the unique set of teams. This result demonstrates the effective use of the **Remove Duplicates** feature for targeted data reduction based on a single key field, achieving high levels of data integrity and normalization swiftly. It is always recommended to visually verify a sample of the remaining data, especially if dealing with large, complex spreadsheets, to ensure no unintended removals occurred.

Alternative Method: Leveraging the UNIQUE Function

While the built-in **Remove Duplicates** feature is ideal for permanent data cleaning, there are scenarios where users might prefer a non-destructive approach--one that extracts the unique data into a new location without altering the original source sheet. This is where Google Sheets functions, specifically the **UNIQUE** function, prove invaluable. The **UNIQUE** function returns only the unique rows from a specified source range, allowing users to isolate unique values based on one column, or a combination of columns, and display them dynamically in a separate area of the sheet.

To use the UNIQUE function to isolate uniqueness based on a single column while returning the associated rows, the structure is surprisingly straightforward, although its application requires careful consideration of the array output. For instance, if our original data is in A2:B12, and we only want rows that have unique values in Column A (Team), we would typically use the formula `=UNIQUE(A2:B12)`. This function is designed to treat the entire row as a unit and return the first unique combination of values across the whole range. However, when we only care about uniqueness in Column A, the standard **UNIQUE** function will still compare Column B (Player Name). If two different players are listed for the same team, the function might return both rows if the entire row combination is unique.

For strictly enforcing uniqueness based on Column A while retaining the entire row's data, a more robust solution often involves combining the QUERY function with the **UNIQUE** function, or utilizing array formulas to create a primary key list and then filtering. A simpler, common workaround is to use the QUERY function with a specific grouping clause. For instance, if you wanted to select the unique team names and the first corresponding player, you could use a formula like `=QUERY(A2:B12, "SELECT A, B GROUP BY A, B LIMIT 1 LABEL B ''")` or, more simply, group solely by Column A and use a standard aggregation (like `MAX` or `MIN`) on Column B to ensure only one record is returned per unique team name. This method offers unparalleled flexibility for dynamic data extraction without data modification.

Limitations and Best Practices for Data Hygiene

While the **Remove Duplicates** tool is highly effective, it has certain limitations users must be aware of. Firstly, the tool is **case-sensitive**. This means that "Bulls" and "bulls" would be treated as two distinct, unique entries, and neither row would be removed. If your data source suffers from inconsistent capitalization, a preliminary data cleanup step--such as applying the `LOWER()` or `UPPER()` function to standardize the text before running the duplicate check--is absolutely necessary. Secondly, the tool treats leading or trailing spaces as unique characters. If a cell contains "Team A " (with a space) and another contains "Team A" (without a space), the tool will keep both rows. Using the `TRIM()` function across the target column is essential before initiating the removal process to avoid these common data entry errors skewing the results.

To maintain robust data hygiene, always adhere to a few fundamental best practices. Prioritize working on a duplicate copy of your primary sheet whenever performing destructive operations like row deletion. This provides an immediate safety net should the results be unexpected. Furthermore, before attempting to remove duplicates based on a single column, ensure that the data types within that column are consistent. Mixing numeric IDs stored as text with IDs stored as actual numbers can lead to incomplete removal of duplicate values. Finally, if data integrity is paramount and multiple columns relate to the uniqueness of the primary column, consider using a helper column to concatenate those related fields (e.g., combining Team Name and Year) to create

a composite key, and then running the duplicate removal process on this new helper column, guaranteeing that the uniqueness criteria are met across all necessary dimensions.