Learn to Identify Duplicate Data in Two Google Sheets Columns

Authored by

Mohammed looti

April 23, 2026

Introduction: Ensuring Data Integrity in Complex Datasets

In modern data management, maintaining **data integrity** is paramount, especially when handling large or merged datasets. The task of identifying and eliminating duplicate values is a fundamental requirement for accurate analysis, financial reporting, and efficient processing. Whether you are reconciling client lists, merging inventory records, or compiling statistical information, the presence of redundant entries can significantly skew results, complicate aggregation, and undermine the reliability of your data infrastructure. This challenge is particularly acute when comparing or consolidating information sourced from two separate lists or fields, often represented as distinct columns within a powerful spreadsheet application like Google Sheets.

Manually scanning thousands of rows across two columns for matching entries is time-consuming, prone to human error, and completely unsustainable in professional environments. Fortunately, Google Sheets offers highly effective, automated solutions to streamline this essential process. The most robust technique leverages the visual power of conditional formatting in conjunction with the analytical capabilities of the **COUNTIF() function**. This synergy allows users to instantly highlight cells that contain matching entries across two or more specified columns, providing immediate visual confirmation of redundancy.

This comprehensive guide is designed to provide you with a precise, step-by-step methodology for executing this task. We will demonstrate exactly how to configure the rules necessary to locate and highlight duplicate values that exist simultaneously in two distinct columns within a typical Google Sheets workbook. By mastering this technique, you will significantly enhance your ability to maintain cleaner, more reliable data, thereby improving the quality of all subsequent data-driven decisions. The visual outcome we aim to achieve, where common entries are clearly marked, is illustrated below:

	A	B	C	D
1	**Team 1**	**Team 2**		
2	Andy	Andy		
3	Bert	Ben		
4	Chad	Carl		
5	Derrick	John		
6	Eric	Mike		
7	Frank	Chad		
8	George	Frank		
9	Henry	Ethan		
10				
11				
12				
13				
14				
15				
16				
17				

Let us now proceed to set up our practical example and delve into the technical implementation of this powerful data cleansing strategy.

Preparing the Working Environment: Setting Up Our Example Dataset

To fully grasp the application of the duplicate detection method, we need a realistic, relatable scenario. Consider a situation where you are managing personnel data, perhaps comparing two separate rosters—Team Alpha and Team Beta—and need to identify which individuals are listed on both. This requires a direct, cell-by-cell comparison between two segregated columns. Our objective is not to find duplicates within Column A or Column B individually, but rather to identify entries from Column A that reappear in Column B, and vice versa. This cross-column verification is critical for reconciliation tasks.

For the purpose of this demonstration, we will use a dataset focused on **basketball players**. Column A will represent "Team 1," and Column B will represent "Team 2." We will structure our simple dataset in Google Sheets as shown in the image below, ensuring that we have a mix of unique names and names that overlap between the two teams. Note that the headers occupy row 1, and our data begins in row 2.

	A	B	C	D
1	**Team 1**	**Team 2**		
2	Andy	Andy		
3	Bert	Ben		
4	Chad	Carl		
5	Derrick	John		
6	Eric	Mike		
7	Frank	Chad		
8	George	Frank		
9	Henry	Ethan		
10				
11				
12				
13				
14				
15				
16				
17				
18				

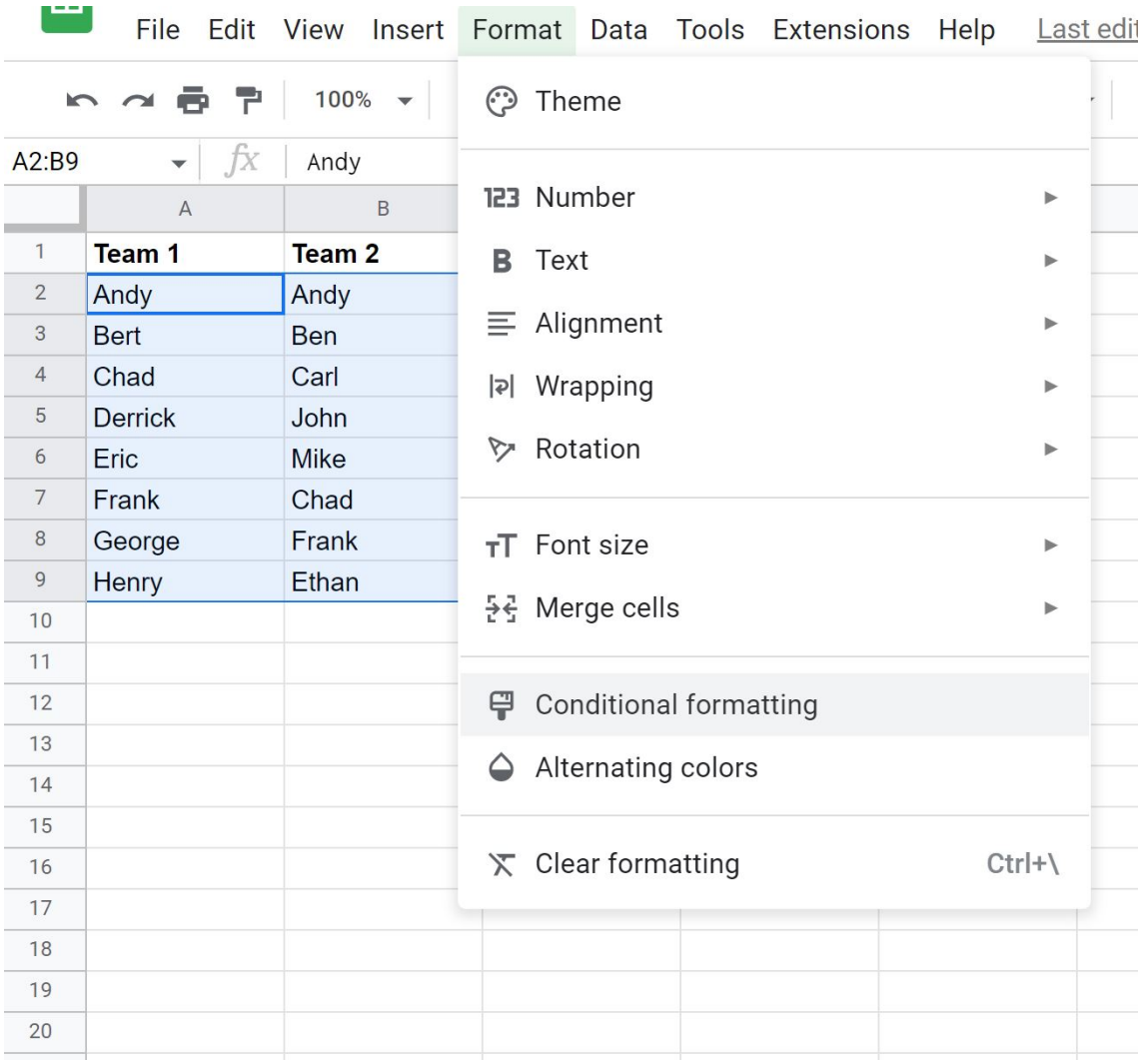
As illustrated, the dataset spans from cell A2 down to B9. Our subsequent steps will focus on applying a rule across the entire range A2:B9. This rule must efficiently scan every name in this combined list and flag any entry that appears more than once, regardless of whether the duplicate is located in the same column or the opposite column. This initial setup of a clearly defined, practical dataset is the necessary foundation before we implement the conditional logic.

Implementing Conditional Formatting Rules

The solution pivots on the intelligent use of conditional formatting, a powerful tool within Google Sheets that automatically applies aesthetic changes (such as highlighting) when specified logical criteria are met. Before defining the logic, we must carefully define the scope of our duplicate search. In our example, we need to select the entire target data range, which includes all the player names: **A2:B9**. Selecting the correct range is crucial, as the conditional rule will only be evaluated against the cells within this selection.

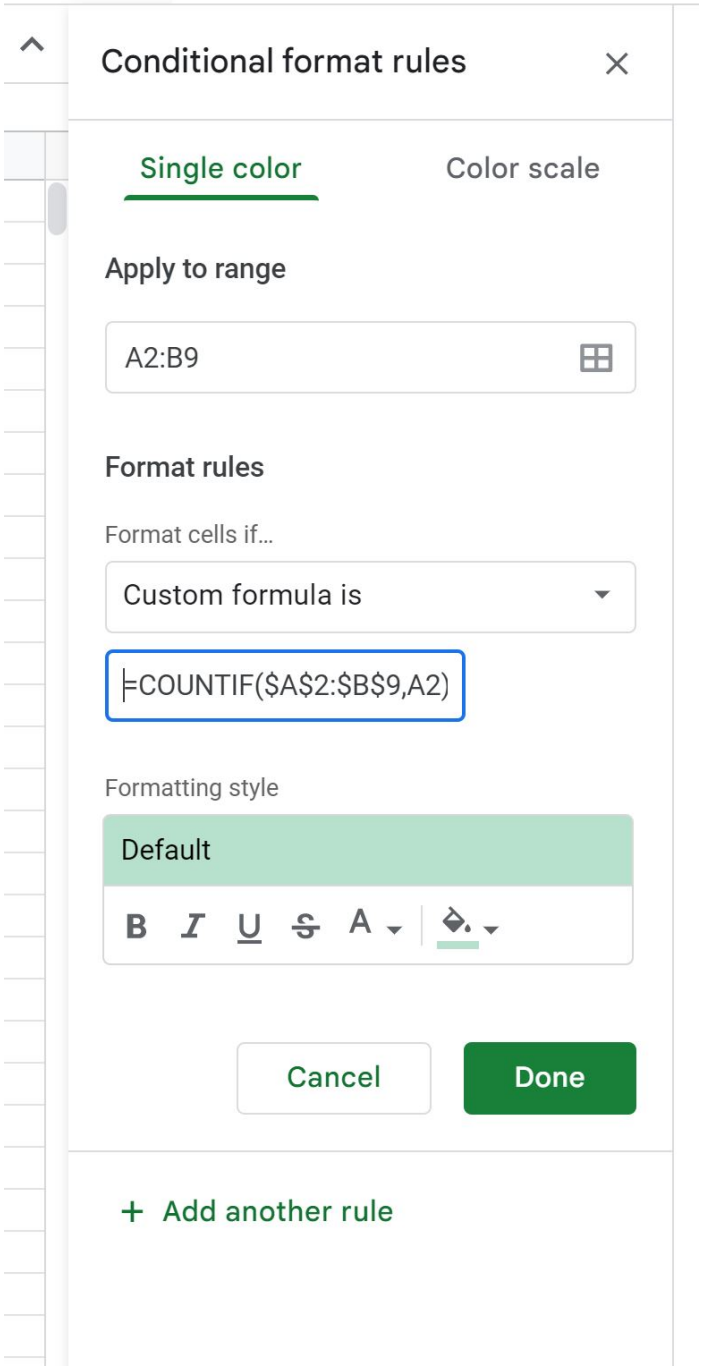
Once the range A2:B9 is selected, navigate to the main menu bar at the top of the Google Sheets interface. Click on the **Format** tab, and from the dropdown menu, select **Conditional formatting**. This action will trigger the appearance of the "Conditional format rules" sidebar on the right side of your screen. This panel is where we will construct the custom logic that determines which cells are

highlighted. Ensure that the "Apply to range" field correctly displays A2:B9.



Within the "Conditional format rules" panel, locate the dropdown menu labeled **Format cells if**. By default, this might be set to "Cell is not empty" or similar. We must change this default setting to the most flexible option: **Custom formula is**. Choosing this option enables us to input a specific spreadsheet formula that Google Sheets will test against every single cell within our selected range (A2:B9). In the input box provided for the custom formula, enter the following precise syntax:

=COUNTIF($A$2:$B$9,A2)>1



This single line of code is the core mechanism that intelligently compares values across the two columns. Understanding its structure, particularly the use of mixed references, is essential for successful implementation and troubleshooting.

Deconstructing the COUNTIF Formula for Cross-Column Comparison

The efficiency of this duplicate detection method relies entirely on the custom formula: `=COUNTIF($A$2:$B$9,A2)>1`. This sophisticated yet concise expression instructs the conditional

formatting engine on exactly how to evaluate the uniqueness of each entry. To truly master this technique, we must break down the purpose of each component:

The COUNTIF() function: This function is the primary counting mechanism. It is designed to count the number of cells within a specified range that satisfy a given criterion. Its standard syntax is `COUNTIF(range, criterion)`. In our application, the function determines how many times the current cell's value appears throughout the entire combined dataset.
`$A$2:$B$9` (The Range): This component defines the entire pool of data that the function must search. It encompasses all player names from both Column A and Column B. Crucially, the dollar signs (`$`) signify an **absolute reference**. When the conditional formatting rule is applied sequentially to every cell (A2, A3, B2, B3, etc.), this range remains fixed. The search boundary never shifts, ensuring that every comparison is made against the complete combined list.
`A2` (The Criterion): This element provides the value that the COUNTIF function is currently looking for. Because A2 lacks dollar signs, it functions as a **relative reference**. When the rule is evaluated for the cell A2, it counts the value in A2. When the rule shifts to evaluate cell A3, the criterion automatically shifts to A3. Similarly, when it moves to cell B2, the criterion dynamically becomes B2, and so on. This dynamic referencing is what allows the single formula to correctly test the content of every cell against the fixed, combined range.
`>1` (The Logical Condition): This is the trigger for the formatting. If the total count returned by COUNTIF is greater than 1, it means the current value appears more than once within the fixed range `$A$2:$B$9`. When this condition evaluates as TRUE, the cell is highlighted; otherwise, if the count is 1 (meaning the value is unique within the combined lists), the condition is FALSE, and no formatting is applied.

In summary, the formula effectively tests every cell against the entire two-column dataset. If the test reveals that the value is shared between the two lists, the cell is flagged, providing an efficient and automated method for cross-column duplicate identification.

Visual Confirmation and Analysis of Duplicate Values

Once you have accurately entered the custom formula and clicked the **Done** button in the Conditional format rules panel, Google Sheets processes the logic across the specified range A2:B9 instantaneously. The result is an immediate visual transformation of your data, with all cells containing duplicate values highlighted according to the default formatting (typically a light green background, unless otherwise specified).

	A	B	C	D
1	**Team 1**	**Team 2**		
2	Andy	Andy		
3	Bert	Ben		
4	Chad	Carl		
5	Derrick	John		
6	Eric	Mike		
7	Frank	Chad		
8	George	Frank		
9	Henry	Ethan		
10				
11				
12				
13				
14				
15				
16				
17				

Observing the finalized spreadsheet, the highlighted cells immediately draw attention to the names that are common to both the "Team 1" and "Team 2" rosters. This visual feedback is crucial for subsequent data cleansing or reconciliation efforts. Based on our example dataset, the following names were identified as duplicates, appearing exactly twice within the combined list:

Andy: Appears once in Team 1 and once in Team 2.
Chad: Appears once in Team 1 and once in Team 2.
Frank: Appears once in Team 1 and once in Team 2.

The remaining names, such as "Ben," "Elsa," and "Gina," are unique to their respective columns within the context of the entire range A2:B9, and therefore remain unformatted. This successful application confirms the effectiveness of the COUNTIF custom formula in identifying overlaps between disparate lists, providing a foundation for accurate data auditing.

Customization and Advanced Formatting Options

While the default light green highlighting successfully flags the duplicate values, conditional formatting offers extensive flexibility to customize the visual presentation. Tailoring the format rules ensures that the duplicates are easily recognizable and align with any internal reporting standards or aesthetic preferences you might have. Effective customization improves data comprehension, especially when dealing with complex spreadsheets that rely on various color codes.

Within the **Conditional format rules** panel, located directly beneath the custom formula input box, you will find a dedicated section for "Formatting style." Here, you can override the default settings and precisely control how the highlighted cells appear. Options include modifying the **background color**, selecting a contrasting **font color**, and applying **font styles** such as bolding, italicizing, or using strikethrough.

For high-visibility data auditing, for example, you might choose a vibrant red background color to ensure immediate attention is drawn to the duplicated entries. Alternatively, if the goal is data archival or soft auditing, a more subtle change, such as italicizing the text, might be preferred. Experimentation with these controls—color palettes, border styles, and font weights—allows you to create a visual schema that maximizes the clarity and utility of your spreadsheet, making the duplicate identification process not just functional, but highly intuitive.

Conclusion and Further Data Management Techniques

The method detailed in this guide—combining conditional formatting with the powerful `COUNTIF` formula—provides an efficient, non-destructive way to identify overlapping entries between two separate columns in Google Sheets. By understanding the critical role of absolute references in defining the fixed search range and relative references for dynamic criterion evaluation, you can adapt this technique to datasets of any size and complexity.

While identifying duplicates is a crucial step toward achieving **data integrity**, it is only one facet of advanced spreadsheet management. Google Sheets is equipped with a vast ecosystem of functions and features designed to enhance every stage of the data lifecycle, from input and cleaning to complex analysis and visualization. Continual learning and exploration of these tools are essential for maximizing productivity.

To further expand your proficiency in data manipulation, we strongly encourage exploring related functions and techniques. Consider mastering advanced filtering options to isolate duplicate rows, utilizing pivot tables for aggregated analysis, or employing other logical functions such as `VLOOKUP` or `INDEX/MATCH` for cross-sheet reconciliation. By building on the foundation of conditional logic demonstrated here, you can automate increasingly complex workflows and transform how you manage and interpret your spreadsheet data.