

How to Generate Unique IDs in Google Sheets

Authored by
stats writer

January 31, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Generate Unique IDs in Google Sheets*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=128792>

Generating unique identifiers (UIDs) within Google Sheets is a fundamental requirement for effective data management and organization. While some spreadsheet environments offer dedicated functions, Sheets users typically rely on powerful combinations of standard formulas to achieve guaranteed uniqueness across their rows or groups. These identifiers are crucial for tasks such as creating primary keys, facilitating seamless data integration with external tools, and ensuring data integrity during sorting or filtering operations.

Effectively utilizing functions like ARRAYFORMULA, often used in conjunction with other logic functions, allows for the automatic generation of these identifiers across an entire range, streamlining the process of tracking and referencing individual records within a vast dataset. The ability to create robust, non-duplicative identifiers is especially useful when dealing with dynamic datasets or integrating information from multiple sources. This article explores highly effective, formula-driven methodologies for generating these critical identification values directly within your spreadsheet environment, ensuring clarity and consistency for all your organizational needs.

Generate Unique Identifiers in Google Sheets

The Necessity of Unique Identifiers in Data Management

In any substantial spreadsheet operation, ensuring that every record is distinctly identifiable is paramount. Without a system for unique identification, data manipulation tasks--such as merging datasets, performing advanced lookups, or applying database normalization principles--become complex and error-prone. Google Sheets, serving as a powerful data repository for many users, demands robust methods for assigning these identifiers to maintain integrity and facilitate sophisticated analysis. This requirement becomes especially acute when dealing with repetitive textual entries or when the inherent order of rows might change over time due to sorting or filtering.

The techniques we will explore bypass the limitations of relying solely on row numbers, which are volatile. Instead, we establish formula-based mechanisms that dynamically assign an ID based on content or position, ensuring that the identifier remains linked to its data regardless of how the spreadsheet is reorganized. We will delineate two primary methods: one for assigning persistent IDs to groups of identical items (Group IDs), and another for assigning strictly unique IDs to every individual row entry (Row IDs).

Method 1: Generating Persistent Group Identifiers

Often, data analysts need to assign a consistent, unique numerical code to all instances of a specific category or group name. For example, if you have multiple sales entries for "Product A," you might want "Product A" to always be assigned ID 1, "Product B" ID 2, and so forth, regardless of how many times they appear in the list. This is the concept of generating a persistent Group ID.

While seemingly simple, achieving this dynamically in Sheets requires a sophisticated combination of lookup and conditional logic functions that reference previously assigned values.

Consider a scenario where we track the performance of various sports teams, where team names frequently repeat across different rows. Our goal is to ensure that every instance of a team name, like "Lakers," receives the identical, pre-assigned unique numerical ID throughout the entire dataset. This method proves invaluable for pivot table analysis and grouping operations where aggregated summaries are required based on a canonical ID rather than relying solely on the textual group name.

Implementing the Group ID Formula: A Practical Example

Suppose we begin with a straightforward list of basketball team names in Column A, as shown below. We aim to populate Column B with the corresponding Group ID.

	A	B	C
1	Team		
2	Mavs		
3	Lakers		
4	Mavs		
5	Hawks		
6	Hawks		
7	Warriors		
8	Mavs		
9	Lakers		
10	Nets		
11	Spurs		
12	Lakers		
13			
14			
15			
16			
17			

To initiate the process, we must manually establish the first ID. This serves as the anchor point for the subsequent formulaic assignments. For simplicity and convention, we typically assign the value **1** to the first unique entry in our list, which sets the foundation for the auto-incrementing logic that follows.

	A	B	C	D
1	Team	Unique ID		
2	Mavs	1		
3	Lakers			
4	Mavs			
5	Hawks			
6	Hawks			
7	Warriors			
8	Mavs			
9	Lakers			
10	Nets			
11	Spurs			
12	Lakers			
13				
14				
15				
16				

Step-by-Step Breakdown of the Group ID Formula

The core mechanism for generating the Group ID relies on a complex formula entered into cell **B3**. This formula simultaneously checks two conditions: whether the current team name has been encountered previously, and if not, what the next sequential ID should be. If the team name is new, the formula increments the previous maximum ID by one; if the team name is a repeat, it performs a lookup to retrieve the previously assigned ID.

The following powerful formula is inserted into cell **B3** (and subsequently dragged down):

```
=IF(ISNA(MATCH(A3,A2:$A$2,0)),MAX(B2:$B$2)+1,VLOOKUP(A3,A2:$B$2,2,FALSE))
```

By clicking and dragging this formula down to apply it to all remaining cells in Column B, the system dynamically assigns the appropriate ID. It intelligently references the existing dataset above the current cell, ensuring that once a team receives an ID, that ID is consistently reused for all subsequent occurrences of that team name.

B3 $\text{=IF(ISNA(MATCH(A3, \$A\$2:A2, 0)), MAX(\$B\$2:B2)+1, VLOOKUP(A3, A\$2:\$B\$2, 2, FALSE))}$

	A	B	C	D	E	F
1	Team	Unique ID				
2	Mavs	1				
3	Lakers	2				
4	Mavs	1				
5	Hawks	3				
6	Hawks	3				
7	Warriors	4				
8	Mavs	1				
9	Lakers	2				
10	Nets	5				
11	Spurs	6				
12	Lakers	2				
13						
14						
15						

Analyzing the Logic of the Group ID Generation Formula

Understanding the components of this formula is essential for mastering dynamic ID assignment in Google Sheets. The function relies heavily on carefully constructed relative and absolute references, particularly the expanding range definitions (e.g., $A2:\$A\2) which are crucial for the incremental logic to work as the formula is dragged down the column.

MATCH(A3, A2:\$A\$2, 0): This segment attempts to find the value in cell A3 (the current team name) within the range of names already processed ($A2:\$A\2). The range $A2:\$A\2 expands dynamically as the formula is moved down, making it inclusive of all previous entries.

ISNA(...): If the **MATCH** function fails to find the team name in the processed range, it returns #N/A. The **ISNA** function captures this error, meaning the team name is encountered for the first time. In this case, the formula executes the TRUE condition.

TRUE Condition (New Team): $\text{MAX}(B2:\$B\$2)+1$: If the team is new, the formula determines the highest Group ID already assigned in Column B (within the processed range) and adds 1, generating the next sequential unique ID.

FALSE Condition (Existing Team): $\text{VLOOKUP}(A3, A2:\$B\$2, 2, \text{FALSE})$: If the team name is found (not ISNA), the formula executes a **VLOOKUP**. It searches for the current team name in the processed range ($A2:\$B\2) and returns the Group ID from the second column (Column B), ensuring the persistent ID is reused.

As a result of this meticulous construction, Column B accurately reflects the intended grouping:

Each team with the name "Mavs" has an ID value of **1**.

Each team with the name "Lakers" has an ID value of **2**.

Each team with the name "Hawks" has an ID value of **3**.

Method 2: Creating Truly Unique Row Identifiers

While Group IDs are excellent for categorization, they do not provide a unique identifier for every individual record, especially when the same group name repeats. In situations requiring a primary key where every single row must be distinguishable--such as log entries or transaction records--we need a mechanism that generates an ID that is unique across all rows, regardless of the data content. This is a crucial aspect of thorough data management.

This second method utilizes a combination of text concatenation and counting functions to generate a structured, truly unique row ID. The resulting ID typically incorporates the group name or a key feature from the row, appended with a sequential counter. This offers immediate readability while guaranteeing non-duplication.

Implementing the Unique Row ID Formula

The formula for generating a unique row ID uses the source value from Column A (the team name) and combines it with a sequential count based on how many times that name has appeared up to that specific row. This ensures that the ID for the first "Mavs" entry is different from the second "Mavs" entry.

The formula below is designed to be highly effective for this purpose. It concatenates the value in the current row's source cell (A2) with a hyphen and the count of that value in the preceding and current range:

=A2&"-"&COUNTIF(\$A\$2:A2,A2)*1

The critical element here is the mixed reference `A2:A2` within the `COUNTIF` function. The starting cell reference (`A2`) is absolute, while the ending cell reference (`A2`) is relative. As this formula is dragged down, the range expands (e.g., `A2:A3`, `A2:A4`, etc.), effectively counting the cumulative occurrences of the text in Column A up to the current row.

You can then click and drag this formula down to each remaining cell in column B:

	A	B	C	D
1	Team	Unique ID		
2	Mavs	Mavs-1		
3	Lakers	Lakers-1		
4	Mavs	Mavs-2		
5	Hawks	Hawks-1		
6	Hawks	Hawks-2		
7	Warriors	Warriors-1		
8	Mavs	Mavs-3		
9	Lakers	Lakers-2		
10	Nets	Nets-1		
11	Spurs	Spurs-1		
12	Lakers	Lakers-3		
13				
14				

Column B now contains a highly descriptive and unique ID for each row, such as "Mavs-1," "Lakers-1," and "Mavs-2." This style of identifier is extremely valuable for auditing, external database linking, and creating pivot keys where absolute row-level distinction is paramount.

Summary and Further Resources

Mastering the generation of calculated unique identifiers in Google Sheets provides a crucial boost to data integrity and organizational efficiency. Whether you require persistent Group IDs for consistent categorization or strictly unique Row IDs for primary key assignments, formulaic approaches offer reliable, scalable solutions without relying solely on external scripts or add-ons. The key lies in leveraging conditional logic functions like IF, MATCH, and VLOOKUP, combined with dynamic range referencing.

For scenarios demanding globally unique identifiers (GUIDs) that guarantee non-duplication even across different spreadsheets or users, integrating Google Apps Script remains the most robust alternative. Scripts allow access to functions like `Utilities.getUuid()`, which provides true 128-bit UUIDs, exceeding the scope of standard spreadsheet formulas but offering maximum security against collisions.

The following tutorials explain how to perform other common tasks and advanced data manipulation techniques in Google Sheets: