

How to Generate Unique Identifiers in Excel

Authored by
stats writer

November 18, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Generate Unique Identifiers in Excel*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=95847>

The Necessity of Unique Identifiers in Data Management

In complex datasets, the ability to assign a consistent and reliable label to each record is paramount for efficient data management and analysis. These labels, known as unique identifiers (UIDs), ensure that when data is manipulated, queried, or merged across different tables, we maintain accuracy and prevent ambiguity. Whether you are tracking inventory, managing customer records, or analyzing experimental results, generating UIDs is often a fundamental preliminary step. While specialized database systems automatically handle this process, users of Microsoft Excel frequently need robust, formula-driven methods to achieve the same result directly within their spreadsheets, especially when dealing with repeating entries that must be categorized under a single ID.

This guide explores two distinct methods for creating unique identifiers within **Excel**. The first method focuses on assigning a sequential ID only to **unique values** within a list (e.g., ensuring every instance of "Hawks" gets ID 3), which is essential for aggregation and pivot table analysis where grouping by category is necessary. The second method demonstrates how to create a **composite identifier** that is unique to every single row, often incorporating text elements and sequence numbers, which is vital for tracking individual transactions or specific occurrences of an event. Both techniques rely solely on powerful Excel functions, eliminating the need for complex VBA code or external tools.

Understanding which method to employ depends entirely on the analytical goal. If your objective is to categorize and group recurring entries, the sequential ID method is appropriate. If, however, you require absolute row-level uniqueness, even for identical entries, the composite identifier approach provides the necessary granularity and traceability. Mastering these formulas grants analysts powerful control over dataset normalization and organization directly within the spreadsheet environment.

Method 1: Assigning Sequential IDs to Distinct Values

A common challenge is generating a sequential numeric identifier for a list where certain values are repeated. For instance, if you have a list of basketball teams where "Lakers" appears multiple times, you want all instances of "Lakers" to share the same unique ID (e.g., 2), while the next unique team encountered (e.g., "Hawks") receives the next available ID (e.g., 3). This is achieved using a complex, yet highly efficient, combination of the **IF**, ISNA function, MATCH function, **MAX**, and VLOOKUP function.

To illustrate this process, let us consider an initial dataset containing a list of basketball team names in column A. Our goal is to populate column B with the unique identifiers.

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

	A	B	C	D	E
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					
18					

We begin by manually assigning the starting identifier value in cell B2. For simplicity and convention, we often start with the value **1** for the very first team listed in A2. This manually entered value serves as the anchor point for the subsequent formulas, allowing the system to reference existing IDs.

For example, we could use the value **1** for the first team:

	A	B	C	D	E
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					
17					

Step-by-Step Implementation and Formula Breakdown

The core mechanism of this method lies in checking if the current team name in column A has already been assigned an ID in the range above it. If it has been assigned an ID, we retrieve that existing ID using **VLOOKUP**. If it has not been assigned an ID, we locate the current highest ID used so far using the **MAX** function and add 1 to it, thereby generating a new, sequential unique identifier.

Next, we can type the following formula into cell **B3**, which is the first cell where the calculation will occur:

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

This sophisticated formula handles both conditions dynamically. Specifically:

The segment **MATCH(A3, A2:\$A\$2, 0)** attempts to find the value in A3 (the current team name) within the range of cells above it (A2 to A2, initially). Note the mixed referencing: A2 is absolute for the starting row of the ID range, while \$A\$2 remains fixed. As you drag the formula down, the range A2:\$A\$2 expands (e.g., to A2:\$A\$3, then A2:\$A\$4), allowing the formula to check against all preceding entries.

The **ISNA** function checks if the **MATCH** returned an error (i.e., if the team name is not found in the previously processed range).

If **ISNA** is **TRUE** (the team is new), the formula executes **MAX(B2:\$B\$2)+1**, assigning the next sequential ID.

If **ISNA** is **FALSE** (the team has been seen before), the formula executes the **VLOOKUP** function to retrieve the existing ID from the ID column (B), using the team name as the lookup value.

Analyzing the Results of Distinct ID Generation

Once the formula is entered into cell B3, the powerful utility of relative and mixed referencing comes into play. We can then click and drag this formula down to each remaining cell in column B, and **Excel** automatically adjusts the calculation ranges while maintaining the fixed anchor points.

	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						
16						
17						
18						

The result in Column B is a clean, sequential unique identifier assigned consistently across the dataset based on the distinct team name. This output is ideal for ensuring that all instances of a single entity are grouped under one numerical ID, which is often required before exporting data to a relational database or for statistical analysis.

Column B now contains a unique ID value for each team, demonstrating the success of the conditional sequencing. For example, we can observe the following pattern:

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**.

This method guarantees that regardless of how many times a team name appears or where it appears in the list, it is consistently assigned the same identifier, thereby normalizing the data based on categorical value rather than row position.

Method 2: Generating Row-Specific Composite Identifiers

While Method 1 is excellent for categorical grouping, sometimes the requirement is to generate a unique identifier for every single row, regardless of whether the cell entry repeats. This is particularly useful when creating transaction IDs, order numbers, or specific occurrence records where the ID must reflect both the entity and its sequence of appearance. This approach typically utilizes text manipulation functions combined with sequential counting.

This type of identifier is often a composite string, combining a fixed element (like an abbreviation of the team name) and a dynamic element (like the row count for that specific team). For instance, the first "Mavs" entry might be "MAV-1," and the second "Mavs" entry would be "MAV-2." This ensures that while both entries are "Mavs," they are uniquely traceable records.

To create this composite ID, we leverage the LEFT function to extract the abbreviation and the COUNTIF function to track the specific occurrence sequence.

Deconstructing the Composite ID Formula

The formula required for this method is significantly simpler than Method 1, relying primarily on string concatenation using the ampersand (&) operator. It is entered into cell B2 (assuming the data starts in A2) and then dragged down.

The composite formula structure involves three key components:

Extracting the text prefix: `LEFT(A2, 3)` takes the first three characters of the team name in cell A2.

The delimiter: `" - "&` inserts a hyphen to separate the text prefix from the sequential number.

Calculating the sequence number: `COUNTIF($A$2:A2, A2)` counts how many times the value in A2 has appeared starting from the beginning of the data set (\$A\$2) up to the current row (A2). The

mixed reference here is crucial: $\$A\2 is fixed, while A2 changes, creating an expanding range for counting.

Here is the exact implementation of the formula to achieve the row-specific unique identifier:

=LEFT(A2,3)&"-"&COUNTIF(\$A\$2:A2,A2)*1

The multiplication by *1 ensures that the output of the **COUNTIF** function is treated explicitly as a number for concatenation, although in most modern Excel versions, this is handled implicitly. The result is a highly readable and row-specific ID.

Applying the Composite Identifier Formula

Just as with the previous method, after inputting the formula into the first data row (B2), you can then click and drag this formula down to each remaining cell in column B. The expanding range in the COUNTIF function ensures that the sequence number correctly increments for duplicate entries.

	A	B	C	D	E	F	G
1	Team	Unique ID					
2	Mavs	Mav-1					
3	Lakers	Lak-1					
4	Mavs	Mav-2					
5	Hawks	Haw-1					
6	Hawks	Haw-2					
7	Warriors	War-1					
8	Mavs	Mav-3					
9	Lakers	Lak-2					
10	Nets	Net-1					
11	Spurs	Spu-1					
12	Lakers	Lak-3					
13							
14							
15							
16							

Column B now contains a unique ID for each row. Observe how even though the input value "Lakers" is repeated three times, the output identifiers are distinct: "Lak-1," "Lak-2," and "Lak-3."

This guarantees that every specific observation or record in the spreadsheet has its own traceable code, regardless of the text value contained within the cell. This method is particularly useful for auditing and tracking individual records in a dynamic list, providing a superior level of data integrity compared to relying solely on row numbers, which can change upon sorting or filtering.

Conclusion: Choosing the Right Identification Strategy

Generating unique identifiers in Microsoft Excel is a fundamental skill for transforming raw lists into structured datasets suitable for advanced analysis. The choice between Method 1 (Sequential IDs for Distinct Values) and Method 2 (Row-Specific Composite IDs) hinges entirely on the function the identifier is expected to perform.

Use the complex **IF/ISNA/MATCH/MAX/VLOOKUP** sequence when you need to group records consistently based on their categorical value. This normalized approach is essential for database preparation or aggregate reporting where "Mavs" must always be ID 1.

Conversely, utilize the simpler **LEFT/COUNTIF** string concatenation method when every single record, even if it is a duplicate category entry, requires its own absolute unique identifier for tracing specific occurrences. Both methods demonstrate the power of Excel's built-in formulas to handle complex data organization challenges without necessitating external programming.