How to Convert a Table to a List in Google Sheets

Authored by

stats writer

February 1, 2026

The need to convert two-dimensional table structures into a streamlined, single-column list format is a common requirement when working with data analysis in Google Sheets. This process, often referred to as 'unpivoting' or 'stacking' data, is critical for preparing datasets for subsequent operations like filtering, sorting, or running functions that require a vertical array input. While there are several techniques available, ranging from manual operations to advanced array formulas, selecting the most efficient method depends heavily on the size and dynamism of your source data. Understanding these methods is key to mastering data transformation workflows.

Data stored in wide formats--where related attributes are spread across multiple columns, such as quarterly sales across four columns--is often intuitive for human readers but highly inefficient for computational processing and analysis tools. By converting this tabular structure into a long, vertical list, every data point receives its own row. This adherence to the principles of 'tidy data' ensures that each variable forms a column and each observation forms a row, which dramatically simplifies tasks like creating pivot tables, generating charts, or performing statistical analysis across the entire dataset without complex referencing.

This comprehensive guide explores the various techniques available within Google Sheets to accomplish this conversion, highlighting the simplicity of manual methods, the structural utility of the TRANSPOSE function, and the powerful automation offered by the specialized FLATTEN function. Mastering these techniques will empower you to manage and analyze complex datasets with greater efficiency and accuracy.

Convert a Table to List in Google Sheets

Method 1: The Manual Approach Using Copy and Paste Special

The simplest and most straightforward method for transforming a static table into a vertical list involves using the standard copy and paste features, combined with the powerful "Paste special" options available in Google Sheets. This technique is ideal when you are dealing with a small dataset or when the source data will not change frequently, as it involves manual steps and creates a static output rather than a dynamic link to the original data range. It serves as a foundational skill for basic data manipulation.

To execute this manual transformation, you must first isolate the data elements you wish to convert. The crucial steps ensure that you are pasting only the raw values, preventing issues that might arise from transferring cell formatting, formulas, or conditional rules that conflict with the destination location. This isolation step ensures data integrity and a clean conversion. While effective, it is important to remember that any future updates to the source table will require repeating this entire process, making it less suitable for automated reporting or volatile datasets.

Selection: Select the entire range of the table you intend to convert into a list format. Ensure that you capture all necessary rows and columns containing the relevant data points.

Copying Data: Execute the copy command by right-clicking on the selected range and choosing "Copy" or by using the keyboard shortcut **Ctrl+C** (or Cmd+C on macOS) on your keyboard.

Destination Selection: Navigate to the cell where you want the resulting vertical list to begin. This should typically be an empty column to avoid overwriting existing information.

Paste Special Activation: Right-click on the destination cell and select "Paste special" or use the keyboard shortcut **Ctrl+Shift+V** (or Cmd+Shift+V on macOS).

Finalizing Conversion: In the subsequent pop-up or submenu, locate and select the option "Paste values only" (or sometimes "Paste values"). This step ensures that the resulting list is comprised purely of the data contents, effectively converting the tabular format into a stacked list. Click "OK" if a confirmation is required.

Upon completion of these steps, your original two-dimensional table will have been successfully converted into a streamlined list format, ready for further analysis or data aggregation. This method is highly accessible but lacks the dynamic updating capabilities provided by formula-based solutions.

Method 2: Leveraging the TRANSPOSE Function

For users seeking a semi-automated approach that still relies on built-in spreadsheet operations, the TRANSPOSE function offers a useful, albeit limited, alternative. The primary purpose of TRANSPOSE is to swap the rows and columns of an array or cell range. While it doesn't immediately "flatten" a multi-column table into a single list, it can be a preparatory step or utilized in combination with other functions (like **ARRAYFORMULA** or concatenation) to achieve the desired vertical orientation.

The core limitation of using TRANSPOSE alone for converting a table to a list is that it simply rotates the data. A table with three rows and four columns becomes a table with four rows and three columns. To achieve a single list, users would still need to manually copy and paste the transposed columns sequentially, or write complex formulas to stitch the resulting rows together. However, for specific data structures where the source table is already very narrow (e.g., two rows), transposing can sometimes suffice if only one row contains the actual values intended for the list.

The alternative approach mentioned in the original text typically refers to "Paste Transposed" within the "Paste special" menu. This manual action uses the function's logic without entering the formula itself, but still results in a rotated table, not a single list. The steps are straightforward: first,

select and copy the table; then, go to the destination cell, right-click, select "Paste special," and choose "Paste transposed." This provides static transposition.

While this method facilitates the reorganization of data axes, it falls short when the goal is a true single-column stacking of all cell values. This is where dedicated functions designed specifically for array manipulation provide a significantly cleaner and more dynamic solution, leading us to the modern approach offered by the **FLATTEN** function.

The Advanced Solution: Utilizing the FLATTEN Function

For modern data processing in Google Sheets, the most powerful and efficient method for converting a two-dimensional table into a single vertical list is the use of the **FLATTEN** function. This function was specifically introduced to simplify complex array operations, eliminating the need for convoluted combinations of functions like **QUERY**, **ARRAYFORMULA**, and concatenation operators that were previously necessary to achieve this stacking effect.

The **FLATTEN** function works by reading the input range, iterating through the cells row by row, and stacking all the values found into a single column array. This results in a clean, vertical list where the data maintains the original reading order (left-to-right, then top-to-bottom). This formula-based approach provides a crucial advantage over manual methods: **dynamism**. If the source data table changes, the resulting list automatically updates, making it essential for dashboards and reports built on volatile data.

The syntax for this function is remarkably simple, requiring only the array or range you wish to flatten as its argument. For example, if your data resides in the range `B2:E4`, the formula required is simply `=FLATTEN(B2:E4)`. This simplicity abstracts away the complexity of managing multi-dimensional arrays, making advanced data transformation accessible even to novice users.

Using **FLATTEN** is the professional standard when dealing with large volumes of data that need to be restructured for analysis. It drastically improves workflow efficiency compared to manual copying or attempting to construct complex array manipulation logic. The resulting output is not just a list of values, but an organized array ready for downstream processing, such as filtering out blanks or merging with other datasets.

Step-by-Step Guide to Implementing FLATTEN

To demonstrate the practical application of the **FLATTEN** function, we will use a common scenario: converting sales data presented quarterly across several years into a single chronological list. This kind of organizing data structure is often required before creating time-series charts or calculating overall averages.

Suppose we have a dataset that displays the total sales figures achieved by a company during each quarter (Q1, Q2, Q3, Q4) over three consecutive years. The data is structured with years in rows and quarters in columns, making it easy to read horizontally but difficult to analyze vertically.

The objective is to take this wide table and collapse all sales values into one long, vertical column. This allows us to treat the sales figures as a continuous stream of data points rather than segmented annual records. The crucial first step is identifying the exact range of the data, which in our example, is `B2:E4`, assuming the labels are in row 1 and column A.

The application of the formula is straightforward. You must select an empty cell--ideally cell A6 or any cell below or adjacent to the source data that has sufficient space--and input the function. It is important to ensure that the destination column is entirely empty below your starting cell, as the **FLATTEN** function will spill its results across numerous rows, potentially overwriting any existing content.

The required formula to achieve this transformation is as follows:

=FLATTEN(B2:E4)

This single formula references the entire array and instructs Google Sheets to process the data sequentially, starting from cell B2, moving across to E2, then dropping down to B3, and so forth, until the entire range E4 is included. The efficiency of this function cannot be overstated when compared to older, more convoluted methods.

Visualizing the FLATTEN Formula in Practice

To provide a clear context for how this function operates, consider the visual representation of the source data. A typical sales dataset might look like the structure illustrated in the image below, where the sales figures are clearly segmented by quarter and year.

Suppose we have the following dataset in Google Sheets that displays the total sales made by some company during each quarter of three consecutive years:

	A	B	C	D	E
1	**Year**	**Q1**	**Q2**	**Q3**	**Q4**
2	**2021**	8	7	4	13
3	**2022**	11	10	5	16
4	**2023**	14	15	8	14
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					

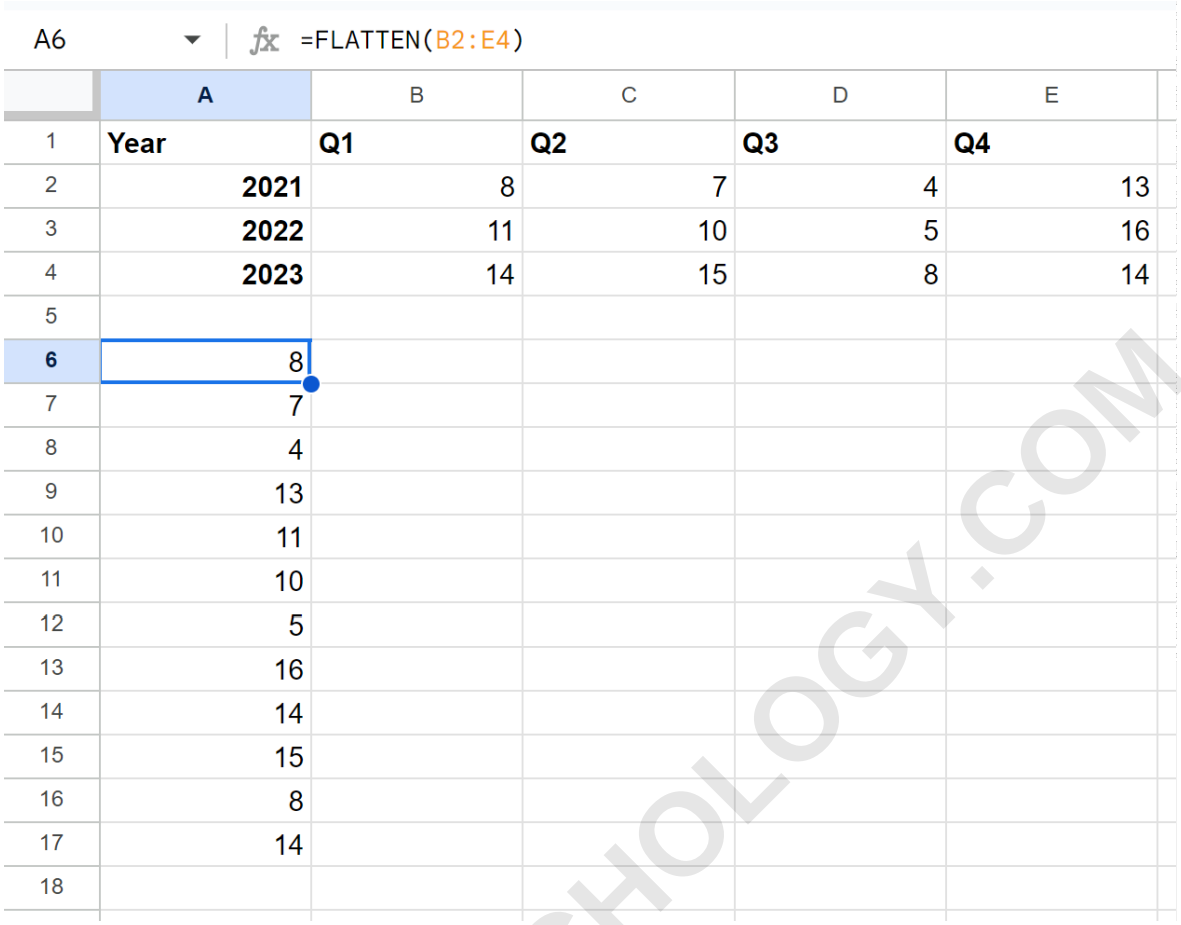
We are aiming to convert this visually organized table into a strict vertical list that preserves the sequence of the original data points. We choose an empty cell, for instance, cell **A6**, to initiate the transformation. By placing the FLATTEN formula here, the resulting array will populate downwards from A6.

We can type the following formula into cell **A6** to do so:

=FLATTEN(B2:E4)

The immediate result of applying this function demonstrates its power. The entire range `B2:E4` is successfully concatenated into a single column. The sequence of the output is crucial: it starts with the first value in the first row (B2), moves horizontally (C2, D2, E2), and then proceeds to the next row (B3) and repeats the horizontal pattern until the last cell (E4) is processed. This systematic processing ensures chronological integrity if the source table is ordered chronologically.

The following screenshot shows how to use this formula in practice:



	A	B	C	D	E
1	Year	Q1	Q2	Q3	Q4
2	2021	8	7	4	13
3	2022	11	10	5	16
4	2023	14	15	8	14
5					
6	8				
7	7				
8	4				
9	13				
10	11				
11	10				
12	5				
13	16				
14	14				
15	15				
16	8				
17	14				
18					

As clearly demonstrated, the single formula in cell A6 drives the entire resulting array. We can see that the formula was able to successfully convert the table of values into a single list that shows the sales values for each corresponding quarter and year.

Understanding the Output and Its Structure

The output generated by the FLATTEN function is a dynamic array. This means that the resulting list is not static values but is linked directly to the source range `B2:E4`. Any changes made to the figures within the original sales table will instantaneously be reflected in the resulting vertical list, maintaining data consistency without requiring manual intervention or recalculation.

The structure of the resulting list adheres precisely to the row-by-row, column-by-column traversal of the source data. This specific ordering is essential for correctly interpreting the data points, especially when contextual information (like the year and quarter) is implied by the order. Understanding this traversal pattern is critical for subsequent analysis where you might need to manually associate headers or labels with the flattened values.

For example, using our sales data illustration, the sequence of the flattened values corresponds

exactly to the temporal flow of the sales figures:

The first value in the list shows the sales for **Q1 2021** (originating from B2).

The second value in the list shows the sales for **Q2 2021** (originating from C2).

The third value in the list shows the sales for **Q3 2021** (originating from D2).

The fourth value in the list shows the sales for **Q4 2021** (originating from E2).

The pattern then seamlessly transitions to the next year's data (starting with B3) and continues this process until all sales figures from the original range `B2:E4` have been stacked vertically. This reliable ordering principle allows advanced users to combine FLATTEN with other array functions to generate corresponding columns for quarter and year labels, thereby fully normalizing the dataset.

Benefits of Vertical Organizing Data for Analysis

The conversion of wide data tables into long, vertical lists is not merely an aesthetic preference; it is a fundamental requirement for efficient data analysis, particularly within environments like Google Sheets or statistical software. When data is structured vertically, it aligns perfectly with the principles of relational databases and standard analytical tools, vastly improving readability and simplifying complex calculations.

One of the primary advantages of this vertical structure is its compatibility with powerful built-in tools, such as **Pivot Tables** and advanced filtering options. Pivot tables, for instance, are designed to aggregate data based on dimensions (or attributes) defined in separate columns. If all sales figures are in a single column, they can be easily summarized, averaged, or counted based on grouping variables (which would be generated alongside the flattened list), streamlining complex reporting tasks that would otherwise require multiple **SUMIFS** or **QUERY** criteria on the original wide table.

Furthermore, a vertical list simplifies the application of standard statistical functions. Functions like **AVERAGE**, **MEDIAN**, or **STDEV** can be applied directly to the entire list column, providing immediate insight into the central tendency or dispersion of the dataset as a whole. Trying to apply these functions across multiple, non-contiguous columns in the original table format would be cumbersome and error-prone. The structured, single-column approach promotes clarity and reduces the risk of calculation errors, which is crucial for maintaining data accuracy in business intelligence applications.

In essence, converting the data structure using **FLATTEN** transforms the data from a human-centric display into a machine-readable format optimized for computational efficiency. This data

transformation is a critical step in any robust analytical workflow, moving beyond simple spreadsheet management towards serious data science practices. (Note: You can find the complete documentation for the **FLATTEN** function in Google Sheets documentation.)

Conclusion: Choosing the Right Conversion Method

The choice between the manual Copy/Paste special method, the transitional TRANSPOSE function, and the automated **FLATTEN** function depends entirely on the specific requirements of the data task and the need for dynamism. For simple, one-off conversions of small, static datasets, the manual method is quick and requires no functional knowledge. It offers the fastest path to a raw, unlinked list.

However, when dealing with production-level data, frequently updated sheets, or large arrays, the **FLATTEN** function is unequivocally the superior choice. Its ability to dynamically link the list to the source table ensures that analysis is always performed on the most current data without requiring repetitive manual steps. This efficiency not only saves time but also significantly reduces the potential for human error inherent in repeated copying and pasting operations.

Mastering array manipulation functions like **FLATTEN** is essential for anyone serious about leveraging the full capabilities of Google Sheets for business intelligence or complex data modeling. By transforming your data into the optimal vertical format, you unlock powerful analytical capabilities, enabling easier integration with pivot tables, charts, and filtering operations, ultimately making your data clearer and significantly easier to read and analyze.