Google Sheets Query: Remove Header from Results

Authored by

Mohammed looti

October 28, 2025

Introduction: Mastering Header Control in Google Sheets Queries

The QUERY function in Google Sheets is arguably the most powerful tool available for advanced data handling, enabling users to perform complex selections and transformations akin to professional SQL operations. However, when generating reports or preparing data for integration into other systems, the default inclusion of header rows in the query output often proves cumbersome. Controlling and removing these headers is a critical skill for any serious Google Sheets user aiming for clean, finalized data presentation.

Whether you are aggregating metrics, extracting specific subsets of data, or simply requiring a purely numerical output, the need to suppress column labels arises frequently. These headers, while useful for identifying the source columns, can interfere with subsequent calculations or automated processes that expect raw data. This comprehensive guide details three expert methods, offering solutions for targeted header removal in aggregated columns as well as complete, system-wide header eradication using advanced formulaic techniques.

By mastering the clauses and structural techniques discussed here, you will significantly enhance your data manipulation capabilities within the spreadsheet environment. We will transition from simple, single-column adjustments to complex, nested query structures, providing you with precise control over every aspect of your data's appearance. Let us begin exploring the specialized syntax that transforms raw query results into polished, header-free datasets.

Method 1: Suppressing Headers with the LABEL Clause (Targeting Aggregated Columns)

The first and most straightforward method focuses on removing headers generated by aggregated fields, such as sums, averages, or counts. In a standard QUERY function, when you use an aggregate function like `SUM(B)`, the system automatically assigns a default header (e.g., "sum of B"). To eliminate this default label, we utilize the **`LABEL`** clause.

The primary function of the **`LABEL`** clause is to allow users to rename column headers to something more descriptive than the default output. Crucially, by instructing the clause to assign an empty string (`""`) as the new label, we effectively hide the header entirely without altering the underlying data or the query logic. This technique is highly precise, allowing you to selectively remove headers from calculated columns while retaining the labels of standard, non-aggregated columns.

The following syntax demonstrates how to apply this technique. Note how the **`LABEL`** clause is appended to the main query string, targeting the specific aggregated column and replacing its header with nothing:

=QUERY(A1:C7,"select A, sum(B) group by A label sum(B) "")

In this example, the formula selects column `A` and calculates the sum of column `B`, grouping the results by `A`. The critical instruction, **`label sum(B) ''`**, signals to the QUERY function that the header for the aggregated results of column B should be displayed as a blank entry, providing a clean, header-suppressed column in the final output.

Method 2: Applying LABEL to Multiple Columns for Enhanced Control

Building upon the previous method, when your data involves multiple complex calculations, you often need to suppress several headers simultaneously. Fortunately, the **`LABEL`** clause is designed to handle multiple assignments efficiently within a single query string, making it an excellent choice for complex reports requiring multi-column data aggregation.

To remove headers from several aggregated columns, you simply extend the **`LABEL`** clause list, separating each column assignment with a comma. This approach ensures that your formula remains compact and readable, avoiding the need for overly complex or nested structures when dealing solely with aggregated fields. It provides granular control, which is essential for reporting where clarity and conciseness are paramount.

Consider a scenario where you are calculating the sum of two different metrics (B and C) grouped by a single identifier (A). Both resulting aggregate columns would generate default headers that need removal. The enhanced formula integrates both suppression commands seamlessly:

=QUERY(A1:C7,"select A, sum(B), sum(C) group by A label sum(B) '', sum(C) ''")

In this structure, the statement **`label sum(B) '', sum(C) ''`** ensures that both the header for the aggregated values of column B and the header for column C are replaced by empty strings. This method is highly efficient for generating streamlined summaries where the column context is derived implicitly from the row data or will be added manually elsewhere in the sheet.

Method 3: Comprehensive Header Removal via Nested Queries and OFFSET

There are many instances where the goal is to produce a completely header-free dataset, irrespective of whether the columns are aggregated or raw data. For this universal suppression, neither the **`LABEL`** clause nor standard query syntax is sufficient. Instead, we must employ a powerful technique involving a nested query combined with the **`OFFSET`** clause.

A nested query utilizes two QUERY functions: an inner query that performs the necessary data selection, transformation, and aggregation, and an outer query that processes the results of the inner query. The key to header removal lies in the outer query, which uses **`offset 1`**. This clause instructs the outer query to simply skip the first row of its input data--which, by design, is the header row generated by the inner query.

Furthermore, to ensure that the outer query itself does not introduce a new header row, we must carefully manage the header argument. The inner query often includes a header argument of `1` (to correctly process the input range), but the outer query must specify a header argument of `0`, telling Google Sheets that the data it is processing (the output of the inner query minus the skipped row) has no headers. This two-part approach guarantees a completely clean data output.

=QUERY(QUERY(A1:C7, "select A, sum(B) group by A", 1), "select * offset 1", 0)

This sophisticated formula ensures that the initial selection and grouping occur correctly, but the final display mechanism (the outer query) simply ignores the resultant header row before presenting the data. This technique is invaluable when preparing data for machine consumption or when generating reports where headers must be explicitly managed by external formatting tools.

Practical Demonstration: The Illustrative Dataset

To effectively illustrate these three methods, we will utilize a cohesive and easily understandable dataset. This context is crucial for demonstrating how each specific QUERY clause alters the final output structure. Our sample data, representing fictional team performance metrics, includes team names, points scored, and assists made across several entries.

We will assume this sample data is located in the range `A1:C7` of your Google Sheet. Column A contains the team name, Column B contains points, and Column C contains assists. The consistent application of the formulas to this exact dataset will allow you to replicate the results and observe the header removal techniques in action.

Understanding this structure--especially the presence of a single header row in the input data--is essential for correctly interpreting the role of the header argument (e.g., the `1` in the inner query of Method 3). Below is the visual representation of the foundational data we will manipulate throughout the following examples:

	A	B	C	D
1	**Team**	**Points**	**Assists**	
2	Mavs	22	4	
3	Mavs	15	9	
4	Mavs	28	3	
5	Heat	22	13	
6	Heat	25	12	
7	Heat	12	6	
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				

With this dataset established, we can proceed to apply each of the three specialized QUERY function methods to see the precise output generated by targeted and comprehensive header suppression.

Example 1: Isolating and Suppressing a Single Aggregated Header

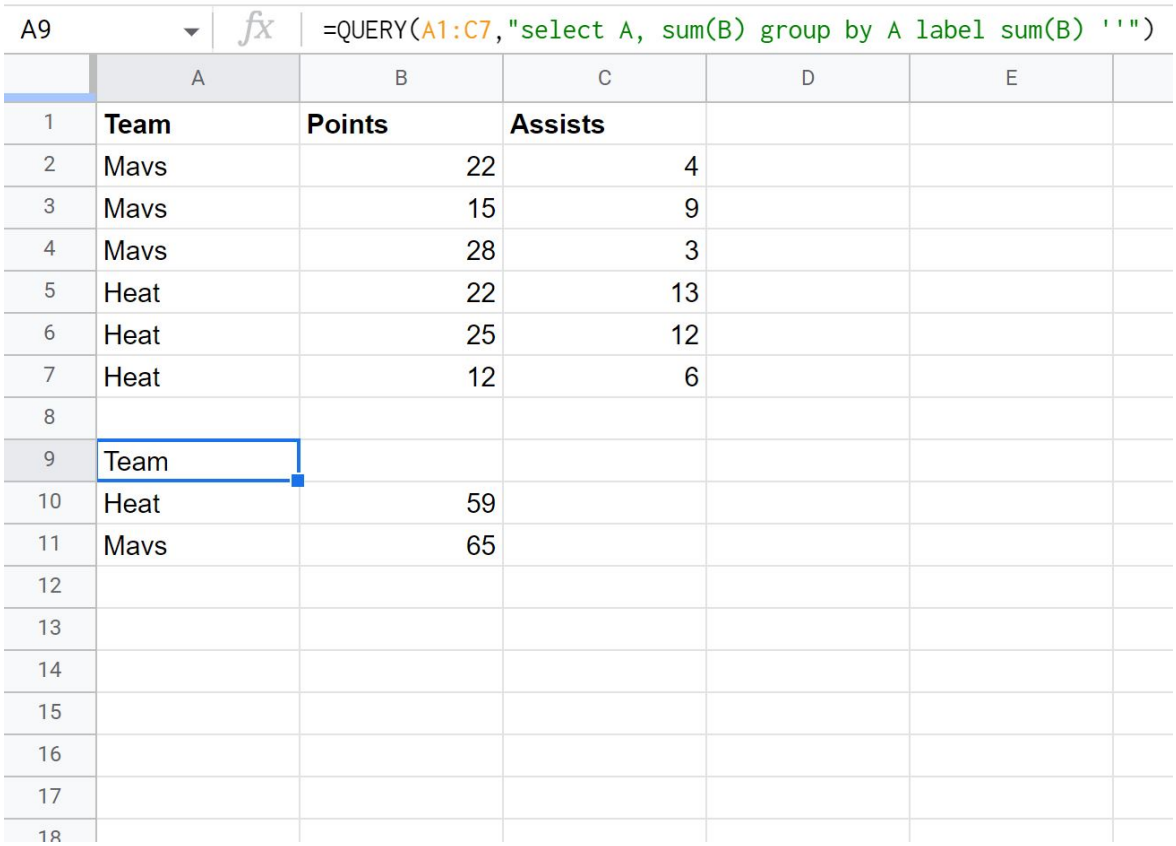
Our first practical example applies Method 1 to remove the header from a single calculated column. The objective is to aggregate the total points scored by each team (grouping by Column A and summing Column B) and ensure that the resulting "sum of B" label is completely suppressed, leaving the team names header intact for context.

We achieve this focused header removal using the `LABEL` clause set to an empty string. This approach maintains the integrity of the non-aggregated column headers while cleaning up the derived metrics:

=QUERY(A1:C7,"select A, sum(B) group by A label sum(B) """)

The output clearly demonstrates the effectiveness of the targeted `LABEL` instruction. The header for the team name (Column A) is preserved, offering necessary context, while the calculated point

totals appear without the generic "sum of B" header, resulting in a cleaner, more professional summary of the aggregated data.



	A	B	C	D	E
1	**Team**	**Points**	**Assists**		
2	Mavs	22	4		
3	Mavs	15	9		
4	Mavs	28	3		
5	Heat	22	13		
6	Heat	25	12		
7	Heat	12	6		
8					
9	Team				
10	Heat	59			
11	Mavs	65			
12					
13					
14					
15					
16					
17					
18					

Example 2: Streamlining Output with Multiple Header Removals

Expanding on the concept of targeted removal, Example 2 applies Method 2 to suppress headers from two distinct aggregated columns. Here, we calculate both the total points (sum of B) and total assists (sum of C) for each team. The goal is to eliminate the headers for both calculated metrics while retaining the original header for the grouping column (Team Name).

This requires concatenating the suppression commands within a single **LABEL** clause, ensuring that the formula remains efficient and clear:

=QUERY(A1:C7,"select A, sum(B), sum(C) group by A label sum(B) '', sum(C) ''")

The resulting table showcases the simultaneous application of the empty string labels to both aggregated columns. The output is highly streamlined, perfect for dashboards or reports where only the essential column identifier (Team) needs a label, and the metrics themselves are self-explanatory or defined elsewhere in the report structure.

A9 =QUERY(A1:C7,"select A, sum(B), sum(C) group by A label sum(B) '', sum(C) ''")

	A	B	C	D	E	F	G
1	**Team**	**Points**	**Assists**				
2	Mavs	22	4				
3	Mavs	15	9				
4	Mavs	28	3				
5	Heat	22	13				
6	Heat	25	12				
7	Heat	12	6				
8							
9	Team						
10	Heat	59	31				
11	Mavs	65	16				
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							

Example 3: Achieving Total Header Suppression Using Nested OFFSET

The final and most powerful example utilizes Method 3 to achieve complete header suppression across all resulting columns, including the grouping column (Team Name). This technique is essential when the output must be raw data, devoid of any header row, which is often the case when integrating data across different systems or software environments.

We employ the nested query structure. The inner query performs the aggregation (Team and Sum of Points), generating a temporary result set with a header row. The outer query then strictly skips this header row using `offset 1` and ensures no new header is added by setting its final argument to `0`.

=QUERY(QUERY(A1:C7, "select A, sum(B) group by A", 1), "select * offset 1", 0)

As expected, the final output table is a pure data matrix. Every column, including the initial grouping column, is displayed without any accompanying header label. This method provides the ultimate solution for scenarios demanding a truly header-free data stream, confirming the mastery of advanced QUERY techniques.

A9 =QUERY(QUERY(A1:C7, "select A, sum(B) group by A", 1), "select * offset 1", 0)

	A	B	C	D	E	F	G
1	**Team**	**Points**	**Assists**				
2	Mavs	22	4				
3	Mavs	15	9				
4	Mavs	28	3				
5	Heat	22	13				
6	Heat	25	12				
7	Heat	12	6				
8							
9	Heat	59					
10	Mavs	65					
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							

Conclusion and Advanced Resources for Google Sheets Mastery

Achieving precise control over the output of the QUERY function is a hallmark of advanced spreadsheet proficiency. By understanding and applying the `LABEL` clause for targeted suppression and the nested query `OFFSET` technique for comprehensive removal, you can ensure that your data is always presented in the most suitable format, optimizing it for reporting, integration, or further analytical processing.

Effective header management is a key component of robust data manipulation workflows. These methods empower you to transform complex query results into clean, production-ready outputs, significantly reducing the manual cleanup often associated with aggregating data in spreadsheets. This capability ensures that your Google Sheets environment functions less like a basic ledger and more like a sophisticated data engine.

To further solidify your understanding of the QUERY function and explore other advanced functionalities, we recommend consulting the following authoritative resources and related tutorials. Continuous learning in these areas will unlock the full potential of your spreadsheet analysis:

Google Sheets Query: How to Use Group By
Official Google Sheets QUERY function documentation
Google Sheets: Basic Spreadsheet Functions

By integrating these advanced techniques into your routine, data management and sophisticated reporting tasks in Google Sheets become effortless and highly automated, paving the way for more impactful data analysis.