How to Count Rows in Google Sheets Using the QUERY Function: A Step-by-Step Guide

Authored by

stats writer

December 3, 2025

The **QUERY function** in **Google Sheets** is arguably the most powerful tool for data manipulation, going far beyond standard filtering and sorting. While many users rely on simple formulas, QUERY provides a SQL-like interface that allows for complex data extraction, aggregation, and transformation. Fundamentally, QUERY allows you to perform sophisticated operations, such as calculating sums, finding averages, grouping results, and, crucially for this tutorial, accurately counting the number of rows that meet specific conditions across a defined range of cells.

Understanding how to leverage the **QUERY function** for counting requires familiarity with its core syntax. The function generally requires three arguments: the data range to analyze, a query string written in the Google Visualization API Query Language, and an optional number specifying whether the header row is included. When focusing on counting, the query string must employ the **COUNT()** aggregation function alongside optional filtering clauses (like `WHERE`) to define which records should be included in the final tally.

This tutorial focuses specifically on utilizing the `COUNT()` clause within the QUERY framework. We will explore how to implement various constraints, including simple row counts, counts based on single criteria, and advanced counts utilizing complex conditional logic. The ability to use **logical operators** such as `AND`, `OR`, and `NOT` within the criteria text string is essential for refining the results and ensuring the query accurately reflects the specific business or analytical question you are trying to answer.

To accurately determine the number of rows that satisfy specific requirements within your dataset, you must integrate the **COUNT()** aggregation function directly into your **Google Sheets** query string. This technique offers far greater flexibility than standard `COUNTIF` or `COUNTIFS` formulas when dealing with large datasets or when complex combinations of filters are necessary.

Method 1: Counting the Total Number of Rows

The simplest application of the **QUERY function** for counting is determining the total number of non-empty rows within a specified data range. This is often necessary for verifying data integrity or establishing the size of the population you are analyzing. When counting total rows, the `WHERE` clause is omitted, as the goal is to count every record present in the range.

In the query string, we use `select count(A)`. The column chosen within the parentheses (e.g., A, B, or C) should ideally be a column that is guaranteed to contain data in every row you wish to count, although often the first column (A) is sufficient. The function then tallies all non-null values in that column across the defined range.

Method 1: Count Total Rows Syntax

=QUERY(A1:C13, "select count(A)")

Method 2: Counting Rows Based on a Single Criterion

When analytical needs become more specific, we often need to filter the data before counting. This is where the `WHERE` clause is introduced. Using a single criterion allows you to target rows where a specific column meets a certain numerical comparison or text match. For example, we might want to count how many records have a value greater than 100 in Column B, or how many records contain the text 'Active' in Column C.

The syntax requires that the criterion be placed immediately after the `WHERE` keyword. In the example below, we are instructing the **QUERY function** to first filter the range `A1:C13`, retaining only rows where the value in Column B is strictly greater than 100, and then apply the **COUNT()** aggregation to the remaining subset of data. This two-step process (filter then count) is fundamental to conditional counting using QUERY.

Method 2: Count Total Rows that Meet One Criteria Syntax

=QUERY(A1:C13, "select count(A) where B>100")

Method 3: Counting Rows Based on Multiple, Complex Criteria

For advanced data analysis, it is frequently necessary to count rows that satisfy several conditions simultaneously. The **QUERY function** excels here because it supports robust **logical operators** within the `WHERE` clause, allowing us to combine multiple conditions using `AND` and `OR` logic. This enables complex filtering that would be cumbersome using nested native formulas.

When using the `OR` operator, as demonstrated in the first formula below, the function counts a row if it meets either Condition X or Condition Y (or both). This is useful for capturing broad categories or finding records that fit any one of several permissive criteria. Conversely, the `AND` operator, shown in the second formula, demands that a row satisfy both Condition X and Condition Y simultaneously. This is highly restrictive and precise, often used to pinpoint specific intersections in the data, such as finding records that belong to a certain group and exceed a specific performance threshold.

Method 3: Count Total Rows that Meet Multiple Criteria Syntax

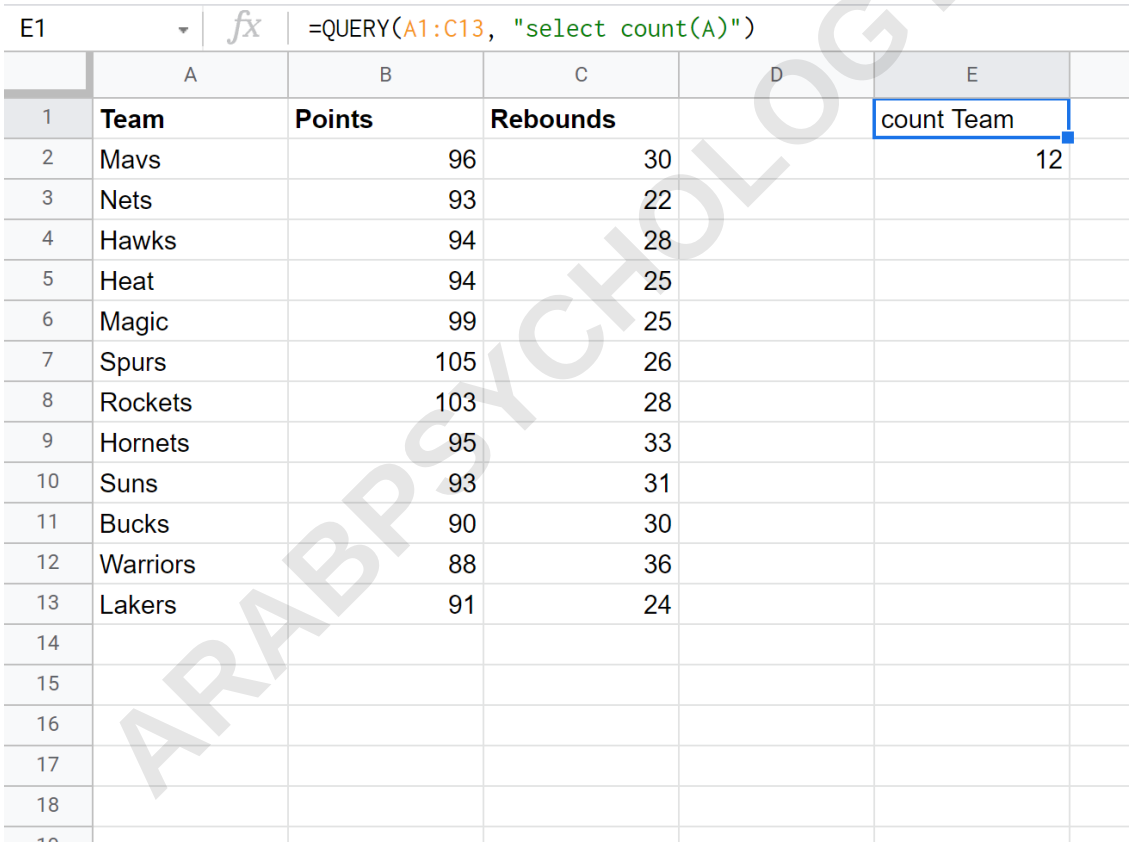
=QUERY(A1:C13, "select count(A) where A='Mavs' OR B>100")

=QUERY(A1:C13, "select count(A) where A='Mavs' AND B>100")

The following detailed examples illustrate the practical application of each of these three methods using sample sports team data, demonstrating exactly how the formulas execute within **Google Sheets**.

Example 1: Counting the Entire Dataset

To begin, let's determine the total population size of our dataset, which in this case represents the total number of teams listed in the range A1:C13. This simple count confirms the scope of our data before we apply any restrictive filters. We utilize the formula `=QUERY(A1:C13, "select count(A)")`. Note that the range A1:C13 includes one header row, which QUERY intelligently handles when counting records, focusing on the data entries themselves.

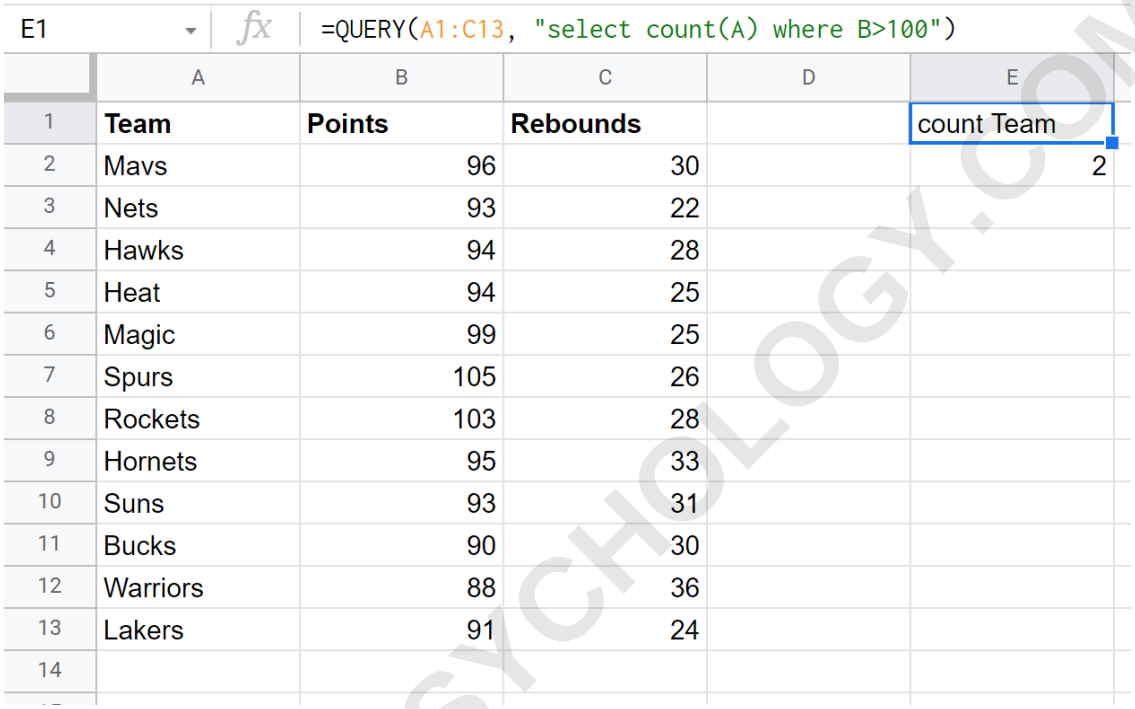


	A	B	C	D	E
1	**Team**	**Points**	**Rebounds**		count Team
2	Mavs	96	30		12
3	Nets	93	22		
4	Hawks	94	28		
5	Heat	94	25		
6	Magic	99	25		
7	Spurs	105	26		
8	Rockets	103	28		
9	Hornets	95	33		
10	Suns	93	31		
11	Bucks	90	30		
12	Warriors	88	36		
13	Lakers	91	24		
14					
15					
16					
17					
18					

As demonstrated in the visualization above, executing this formula yields a result of **12**. This confirms that there are 12 total teams entered into the spreadsheet, providing a baseline count for subsequent conditional analysis.

Example 2: Applying a Single Conditional Filter

This example moves beyond a simple total count by introducing the `WHERE` clause to filter the results. We are interested in identifying only those teams whose points total exceeds 100. This requires the use of the greater-than operator (`>`) within the query string. The query structure becomes `select count(A) where B>100`, targeting Column B (Points) for the comparison.



E1 =QUERY(A1:C13, "select count(A) where B>100")

	A	B	C	D	E
1	**Team**	**Points**	**Rebounds**		count Team
2	Mavs	96	30		2
3	Nets	93	22		
4	Hawks	94	28		
5	Heat	94	25		
6	Magic	99	25		
7	Spurs	105	26		
8	Rockets	103	28		
9	Hornets	95	33		
10	Suns	93	31		
11	Bucks	90	30		
12	Warriors	88	36		
13	Lakers	91	24		
14					

Upon execution, the **QUERY function** processes the dataset, isolating teams with more than 100 points. The result clearly shows that there are only **2** teams that satisfy this specific performance criterion.

Conversely, if our goal is to count teams with lower performance metrics, we can easily adjust the **logical operator**. The following formula utilizes the less-than operator (`<`) to count the total number of teams whose points are strictly less than 100. This flexibility in conditional operators is a key benefit of using the QUERY syntax.

E1 =QUERY(A1:C13, "select count(A) where B<100")

	A	B	C	D	E
1	**Team**	**Points**	**Rebounds**		count Team
2	Mavs	96	30		10
3	Nets	93	22		
4	Hawks	94	28		
5	Heat	94	25		
6	Magic	99	25		
7	Spurs	105	26		
8	Rockets	103	28		
9	Hornets	95	33		
10	Suns	93	31		
11	Bucks	90	30		
12	Warriors	88	36		
13	Lakers	91	24		
14					
15					
16					
17					
18					

By applying the criteria `where B<100`, the count reveals that exactly **10** teams fall into the category of having fewer than 100 points, confirming the results are complementary to the previous count (10 + 2 = 12 total teams).

Example 3: Counting Rows with Multiple Criteria using OR and AND

Applying the OR Logical Operator

The `OR` operator is essential when counting rows that could potentially belong to one of several distinct categories. In this specific scenario, we want to count any team that meets at least one of the following two conditions: either the team name in Column A is exactly equal to "Mavs", OR the points total in Column B is greater than 100. The use of **logical operators** allows us to aggregate these disjoint sets of data.

E1 =QUERY(A1:C13, "select count(A) where A='Mavs' or B>100")

	A	B	C	D	E
1	**Team**	**Points**	**Rebounds**		count Team
2	Mavs	96	30		3
3	Nets	93	22		
4	Hawks	94	28		
5	Heat	94	25		
6	Magic	99	25		
7	Spurs	105	26		
8	Rockets	103	28		
9	Hornets	95	33		
10	Suns	93	31		
11	Bucks	90	30		
12	Warriors	88	36		
13	Lakers	91	24		
14					
15					
16					
17					
18					

The query `select count(A) where A='Mavs' OR B>100` successfully identifies and counts all rows satisfying either criterion. The resulting count is **3**, indicating three teams meet one or both of these criteria.

Applying the AND Logical Operator

In contrast to `OR`, the **AND logical operator** provides a much stricter filter, requiring that all specified conditions must be true for a row to be included in the count. Here, we aim to find high-performing teams by counting only those rows where the points are greater than 90 **AND** the rebounds (in Column C) are greater than 30. This ensures we are only counting teams excelling in both metrics.

E1 =QUERY(A1:C13, "select count(A) where B>90 and C>30")

	A	B	C	D	E
1	**Team**	**Points**	**Rebounds**		count Team
2	Mavs	96	30		2
3	Nets	93	22		
4	Hawks	94	28		
5	Heat	94	25		
6	Magic	99	25		
7	Spurs	105	26		
8	Rockets	103	28		
9	Hornets	95	33		
10	Suns	93	31		
11	Bucks	90	30		
12	Warriors	88	36		
13	Lakers	91	24		
14					
15					
16					
17					
18					
19					

By executing the restrictive query, `select count(A) where B>90 AND C>30`, the output confirms that only **2** teams simultaneously meet both the required points and rebounds thresholds. This demonstrates the precision and power of combining multiple filters using the `AND` operator within the **Google Sheets QUERY** function.

Further Resources on Google Sheets Query Operations

Mastering the **QUERY function** involves understanding its full range of capabilities beyond simple counting. To further enhance your data analysis skills in **Google Sheets**, we recommend exploring other complex operations that rely on the Google Visualization API Query Language, such as grouping and aggregation.

The following tutorial provides detailed instructions on how to group and summarize data effectively:

Google Sheets Query: How to Use Group By