

How to Easily Filter Data with Multiple Conditions in Google Sheets

Authored by
stats writer

December 1, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Easily Filter Data with Multiple Conditions in Google Sheets*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=102979>

Mastering Google Sheets requires proficiency in data manipulation, and few tools are as powerful for this purpose as its advanced filtering capabilities. While basic filtering helps isolate data based on a single column, complex analyses demand filtering with **multiple conditions** simultaneously. This robust approach allows users to meticulously define criteria that precisely dictate which rows of data are displayed.

The ability to specify multiple conditions--whether they involve comparing **text strings**, evaluating **numerical values**, checking **date ranges**, or even identifying **blank cells**--transforms raw data into actionable insights. Unlike standard filters, which often rely on simple dropdown menus, using functions like the dedicated FILTER function provides granular control over complex logical structures, allowing for dynamic result generation.

This guide delves into two primary, high-level methods available in Google Sheets for applying multi-criteria filters. These methods utilize array formulas to efficiently process large datasets, allowing you to dynamically return results--either within the current sheet or directed to a new location--without permanently altering the source data. Understanding the logic behind these functions is crucial for any serious spreadsheet user striving for efficient and precise data extraction.

Understanding the Core Concept: Logical Operators in Filtering

When filtering data based on more than one criterion, we inherently employ **logical operators**. The two fundamental operators we rely on are **AND** and **OR** logic. Determining whether a row must satisfy all specified conditions (AND) or merely one of them (OR) is the first critical step in constructing a multi-conditional filter formula. The method you choose drastically impacts the final result set.

In Google Sheets, the built-in FILTER function is the optimal tool for handling these complex requirements dynamically. It operates by evaluating a series of condition arrays against a defined data range, returning only those rows where the conditions evaluate to TRUE. The way we structure these condition arrays--specifically, the syntax used to separate or join them--determines whether we apply AND or OR logic.

We will explore how the syntax of the FILTER function is adapted to handle these two distinct logical requirements. By implementing either method, users gain flexibility when analyzing datasets that require nuanced segregation based on specific performance indicators or categorical traits.

Method 1: Implementing AND Logic Using the FILTER Function

When your analytical goal requires that **every specified condition must be met** for a row to be included in the result, you need to apply **AND logic**. This is the most common form of multi-criteria

filtering, ensuring strict adherence to all parameters simultaneously. In the FILTER function syntax, implementing AND logic is straightforward: you simply separate each condition with a comma.

The general structure of the FILTER function is `=FILTER(range, condition1, , , ...)`. Each condition you list acts as a separate, mandatory requirement. If a row satisfies `condition1` AND `condition2` AND any subsequent conditions, it is returned. If even one condition fails to evaluate as TRUE, the entire row is immediately discarded from the filtered output, demonstrating the strict nature of this conjunction.

For example, if you are analyzing inventory and want to see items where the "Status" is 'Active' **AND** the "Quantity" is less than 100, separating these two criteria with commas within the FILTER function automatically instructs Google Sheets to evaluate them conjunctively. This comma separation is the fundamental syntactic representation of **AND logic** when working with the array formulas in this environment.

Syntax Example: AND Logic

The following example formula demonstrates how to filter a dataset spanning cells A1 through C10 based on two specific attributes that must both be true for the row to be returned.

`=FILTER(A1:C10, A1:A10="A", C1:C10<20)`

In this formula, the filter processes the rows in the source range A1:C10. It includes a row only if the value in column A (range A1:A10) is exactly equal to the text string "A" **AND** the corresponding value in column C (range C1:C10) is numerically less than 20. This comma-separated syntax is the cleanest and most intuitive way to enforce **simultaneous criteria satisfaction**.

A crucial best practice when using the FILTER function is ensuring alignment. The condition ranges (e.g., A1:A10 and C1:C10) must be structured as single columns or rows and cover the exact same number of data points as the primary data range (A1:C10, excluding the header row if applicable). This alignment is critical to ensure that the logical test for row 5 of the source data correctly applies to row 5 of the condition array.

Method 2: Implementing OR Logic Using the FILTER Function

In contrast to strict AND requirements, **OR logic** is needed when you require a row to be returned if it meets **at least one** of several specified conditions. This results in a broader, more inclusive set of results. Because the FILTER function is designed primarily to accept multiple conditions separated by commas (implying AND), implementing OR logic requires a technique that leverages mathematical coercion.

To implement OR logic, we leverage the fact that Google Sheets treats the logical evaluation TRUE as the numerical value 1, and FALSE as 0. By wrapping each condition in parentheses and joining them using the **addition operator (+)**, we effectively sum the binary results (1s and 0s) of the logical tests for each row.

If the sum of the conditions for any given row is 1 or greater (e.g., $1+0=1$, or $1+1=2$), it means at least one condition evaluated to TRUE, and the row passes the filter. If the sum is 0 ($0+0=0$), it means all conditions evaluated to FALSE, and the row is excluded. This sophisticated use of the arithmetic addition operator as a stand-in for the logical OR operator is a crucial, high-level technique for complex spreadsheet users in [Google Sheets](#).

Syntax Example: OR Logic (Using Addition Operator)

The following formula illustrates how to filter the same dataset (A1:C10) to include any row that satisfies the first condition OR the second condition. Note the mandatory use of parentheses around each logical expression and the plus sign connecting them.

=FILTER(A1:C10, (A1:A10="A")+(C1:C10<20))

In this specific construction, the filter returns rows where the value in column A is equal to "A" **OR** the value in column C is less than 20. The key mechanism here is the use of **+** between the parenthetical conditions. The parentheses are essential because they ensure the logical comparisons are executed first, coercing the resulting TRUE/FALSE arrays into 1s and 0s, allowing the addition operator to combine them before the FILTER function processes the final condition array.

Visualizing the Dataset for Filtering

To effectively demonstrate both the AND and OR filtering methods, we will apply these formulas to a sample dataset. This dataset represents performance metrics across different teams, allowing us to see how each logical structure impacts the returned subset of data. Understanding the source data structure is necessary before applying any filtering technique.

The dataset includes three primary columns: Team (A), Member Name (B), and Points (C). The range A1:C10 will serve as our source data for all subsequent practical examples, allowing us to clearly visualize the results of filtering based on specific team assignments and point thresholds.

	A	B	C	D	E
1	Team	Position	Points		
2	A	Guard	15		
3	A	Guard	19		
4	A	Forward	22		
5	A	Forward	29		
6	A	Forward	25		
7	B	Guard	21		
8	B	Guard	30		
9	B	Forward	14		
10	B	Forward	12		
11					
12					
13					
14					
15					
16					
17					
18					

Practical Example 1: Strict Filtering with AND Logic

Using the dataset presented above, we will now apply the strict AND logic filter. Our objective is to isolate only those team members who belong to Team "A" **AND** who have scored fewer than 20 points. This dual requirement significantly narrows the returned dataset to only the most relevant entries.

The formula leverages the comma separation to enforce the two conditions simultaneously: the team designation in column A must match "A," and the points value in column C must satisfy the less-than inequality (<20). If a row satisfies one but not the other, it is omitted.

The specific formula used for this conjunctural filter is:

=FILTER(A1:C10, A1:A10="A", C1:C10<20)

The resulting output confirms that only records meeting **both** the team constraint and the point threshold are displayed. This efficiency in data retrieval highlights the power of the FILTER function for targeted analysis within large spreadsheets, eliminating noise and focusing only on the data points that satisfy the strictest requirements.

The following screenshot illustrates the result of applying this formula in a separate cell, showing the precise subset of data returned:

A12		=FILTER(A1:C10, A1:A10="A", C1:C10<20)			
	A	B	C	D	
1	Team	Position	Points		
2	A	Guard	15		
3	A	Guard	19		
4	A	Forward	22		
5	A	Forward	29		
6	A	Forward	25		
7	B	Guard	21		
8	B	Guard	30		
9	B	Forward	14		
10	B	Forward	12		
11					
12	A	Guard	15		
13	A	Guard	19		
14					
15					
16					
17					

As observed, the output contains only the rows where the team label is "A" **and** the points value is less than 20. Any rows belonging to Team B, or any rows from Team A that scored 20 points or higher, were successfully excluded by the strict nature of the AND logic implemented via comma separation.

Practical Example 2: Inclusive Filtering with OR Logic

In contrast to the strict AND requirement, we now implement the more inclusive **OR logic**. Here, our goal is to return any row where the team belongs to "A" **OR** the member scored less than 20 points. A row only needs to satisfy one of these conditions to be included in the results array, leading to a much larger, more comprehensive output.

As established previously, OR logic in the FILTER function is achieved by surrounding each condition array in parentheses and connecting them using the addition operator (+). This mathematical conversion transforms the TRUE/FALSE results into numerical arrays that the filter can process collectively as a single required argument.

The formula used to achieve this disjunctive filter is:

=FILTER(A1:C10, (A1:A10="A")+(C1:C10<20))

Notice how the use of the addition operator ensures that if a row is part of Team A (Condition 1 is TRUE/1), it is returned regardless of its points score. Similarly, if a row belongs to Team B but has 15 points (Condition 2 is TRUE/1), it is also returned. This results in a broader selection of data compared to the AND scenario, making it ideal for identifying all potential candidates that meet at least one benchmark.

The following visual representation confirms the outcome of applying the OR logic formula to the dataset:

A12		fx	=FILTER(A1:C10, (A1:A10="A")+(C1:C10<20))		
	A	B	C	D	
1	Team	Position	Points		
2	A	Guard	15		
3	A	Guard	19		
4	A	Forward	22		
5	A	Forward	29		
6	A	Forward	25		
7	B	Guard	21		
8	B	Guard	30		
9	B	Forward	14		
10	B	Forward	12		
11					
12	A	Guard	15		
13	A	Guard	19		
14	A	Forward	22		
15	A	Forward	29		
16	A	Forward	25		
17	B	Forward	14		
18	B	Forward	12		
19					
20					
21					

The resulting filtered dataset is notably larger than that derived using the AND operator. Every row returned either satisfies the criterion of being in Team "A" **or** satisfies the criterion of having points less than 20. This technique is invaluable when conducting analyses that require comprehensive

inclusion based on several acceptable characteristics.

Advanced Use: Combining AND and OR Logic

While the previous examples focused on purely AND or purely OR structures, the true power of multi-conditional filtering lies in its ability to handle combined logical requirements. By strategically using commas (AND) and the addition operator (OR), along with proper parentheses grouping, you can construct filters that answer highly complex and specific business questions.

For instance, suppose you want to filter for rows where the Team is "A" **AND** (Points are less than 20 **OR** Points are greater than 40). In this case, the OR conditions must be evaluated first. They are grouped using parentheses and the + operator, and the entire result of that OR grouping is then separated by a comma from the Team criterion (the AND criterion).

The syntax for such a combined requirement would look like this: `=FILTER(Range, Condition_AND_1, (Condition_OR_1 + Condition_OR_2))`. Mastering the grouping of conditions using parentheses is the key to executing complex, multi-level criteria filtering accurately in Google Sheets, ensuring that operations are executed in the correct logical order.