

# How to Sum Multiple Columns in Power BI: A Step-by-Step Guide

Authored by  
**mohammed loot**

January 11, 2026

## RECOMMENDED CITATION

mohammed loot (2026). *How to Sum Multiple Columns in Power BI: A Step-by-Step Guide*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=125628>

One of the most common requirements in data analysis is the aggregation of numerical data. In Power BI, summing values across multiple columns within a single row requires a specialized approach, moving beyond simple aggregation functions used for single columns. To achieve this, we leverage the power of DAX (Data Analysis Expressions) to create a new calculated column that iterates through the desired fields.

The standard `SUM` function in DAX is designed to aggregate an entire column, operating within the filter context of a visualization. However, when the goal is to calculate a total based on values across several columns for each specific row--such as summing `Sales`, `Expenses`, and `Profit` to get `Total Revenue` per transaction--we must shift our focus to row-level operations. This introduction explains the critical functions and syntax required to define this row-by-row summation successfully, providing valuable insights for granular analysis.

## Understanding Row Context vs. Filter Context

A fundamental concept in DAX is the distinction between filter context and row context. Standard aggregation functions, like `SUM()`, inherently operate within the filter context defined by the visuals, slicers, or measures applied. They calculate a result across a filtered set of rows.

In contrast, when defining a calculated column, DAX operates under a naturally occurring row context. This means the calculation is performed sequentially for every single row in the table, allowing us to reference the individual values of columns on that specific row. While we could simply add the columns directly (e.g., `+`), the most robust and flexible method for row-level operations, especially when needing to ensure accurate context transitions, involves using the `SUMX` function.

Understanding this context is vital because simple arithmetic addition (`+`) works perfectly fine within the row context of a calculated column. However, integrating iteration functions like SUMX offers greater control and prepares the user for more complex scenarios, such as incorporating virtual tables or applying custom filters during the aggregation process.

## Introducing the SUMX and CALCULATE Functions

To perform a summation across multiple columns effectively, we often rely on a combination of two powerful DAX functions: `SUMX` and, less frequently but importantly for context modification, `CALCULATE`.

**SUMX Function:** This function is an iterator. It takes two primary arguments: a table, and an expression. It iterates through every row of the specified table (the first argument) and evaluates the expression (the second argument) for that row, finally summing up the results of the expression evaluation. Because it iterates row by row, `SUMX` is ideal for performing row-level arithmetic

before aggregation.

**CALCULATE Function:** While `SUMX` handles the row-by-row aggregation, `CALCULATE` is the primary function for context transition and modification in Power BI. When used in conjunction with `SUMX`, it can ensure that the calculation operates within a controlled filter context, although for simple calculated columns, its inclusion is often redundant or used as a standard wrapper for best practice, ensuring the expression is evaluated correctly regardless of surrounding context changes.

The combination of these functions allows us to define precisely which values across multiple columns should be totaled for each record in the underlying data table, providing maximum flexibility and performance in large datasets.

You can use the following standard DAX syntax to sum the values across multiple columns of a table in Power BI:

**Sum Points = CALCULATE(SUMX('my\_data', + + ))**

This particular formula defines a new calculated column named **Sum Points**. This column will contain the resulting sum of the values found in the **Game 1**, **Game 2**, and **Game 3** columns. The calculation is executed row by row across the entire table named **my\_data**. The expression `+ +` is evaluated sequentially for each player, and the results are aggregated by SUMX (though since this is a calculated column, the aggregation happens effectively on a single row, and CALCULATE ensures it works correctly in the context of the model).

The subsequent section provides a detailed, practical example demonstrating how to implement and utilize this formula within the Power BI Desktop environment.

### Step-by-Step Example: Calculating Total Player Points

To illustrate the process of summing multiple columns, we will use a common scenario involving sports data. Imagine we have tracking information for basketball players, detailing their performance across several games. Our objective is to determine the cumulative points scored by each player across all recorded games.

This approach is essential when the source data maintains scores or metrics in separate fields rather than in a single normalized column. By creating a total points column, we simplify subsequent analysis, enabling easier ranking, filtering, and visualization of overall player performance directly within our Power BI reports. The steps outlined below will guide you through creating the required calculated column using the DAX syntax introduced previously.

## Defining the Sample Dataset

For this example, let us assume we have imported the following table into Power BI Desktop. We have named this table **my\_data**. The table contains the `Team`, `Game 1` points, `Game 2` points, and `Game 3` points scored by various basketball players:

The screenshot shows the Power BI Desktop interface with the 'Table tools' ribbon selected. The ribbon includes options like 'Name' (set to 'my\_data'), 'Mark as date table', 'Manage relationships', and 'Calculations' (with sub-options: 'New measure', 'Quick measure', 'New column', 'New table'). Below the ribbon, a data table is displayed with the following data:

Team	Game 1	Game 2	Game 3
Mavs	14	5	10
Spurs	10	17	12
Rockets	22	18	12
Kings	20	9	14
Warriors	30	40	34
Nets	14	24	23
Lakers	18	28	28
Thunder	15	13	25
Blazers	12	17	12
Jazz	21	10	19

Our specific goal is to enrich this dataset by introducing a new column that consolidates the individual game scores. We seek to create a new column that explicitly shows the total sum of points scored by each corresponding player across all three games (`Game 1`, `Game 2`, and `Game 3`). This transformation will allow analysts to instantly grasp the overall contribution of each team's player across the measured period.

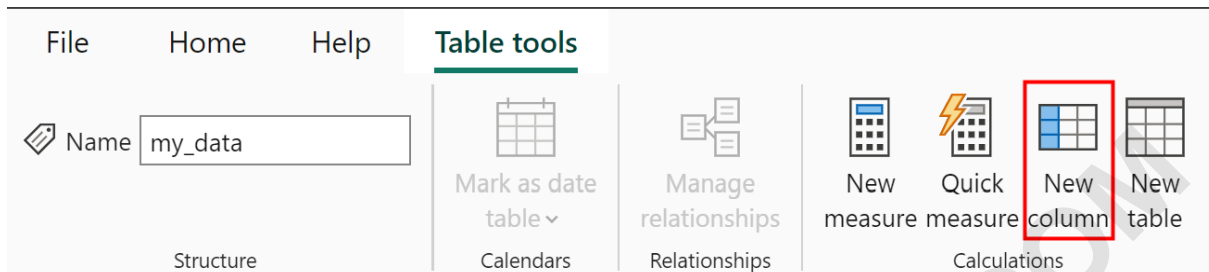
## Implementation Steps in Power BI Desktop

The implementation process begins in the Data View or Report View within Power BI Desktop, ensuring you have the necessary permissions to modify the data model by adding a calculated element.

**Navigate to Table Tools:** First, ensure you are in the Data View, or select the table in the Fields pane while in Report View. Click on the **Table tools** tab located on the ribbon interface at the top

of the application.

**Create a New Column:** Within the **Table tools** tab, locate and click the **New column** icon. This action opens the formula bar, allowing you to input your DAX expression for the new field.



**Input the DAX Formula:** Once the formula bar is active, type or paste the precise DAX formula required to sum the desired columns. Note that since we are defining a calculated column, the system automatically initiates the row context necessary for the calculation.

Type the following formula into the formula bar, ensuring the table name (`my\_data`) and column names (``, etc.) exactly match your data model:

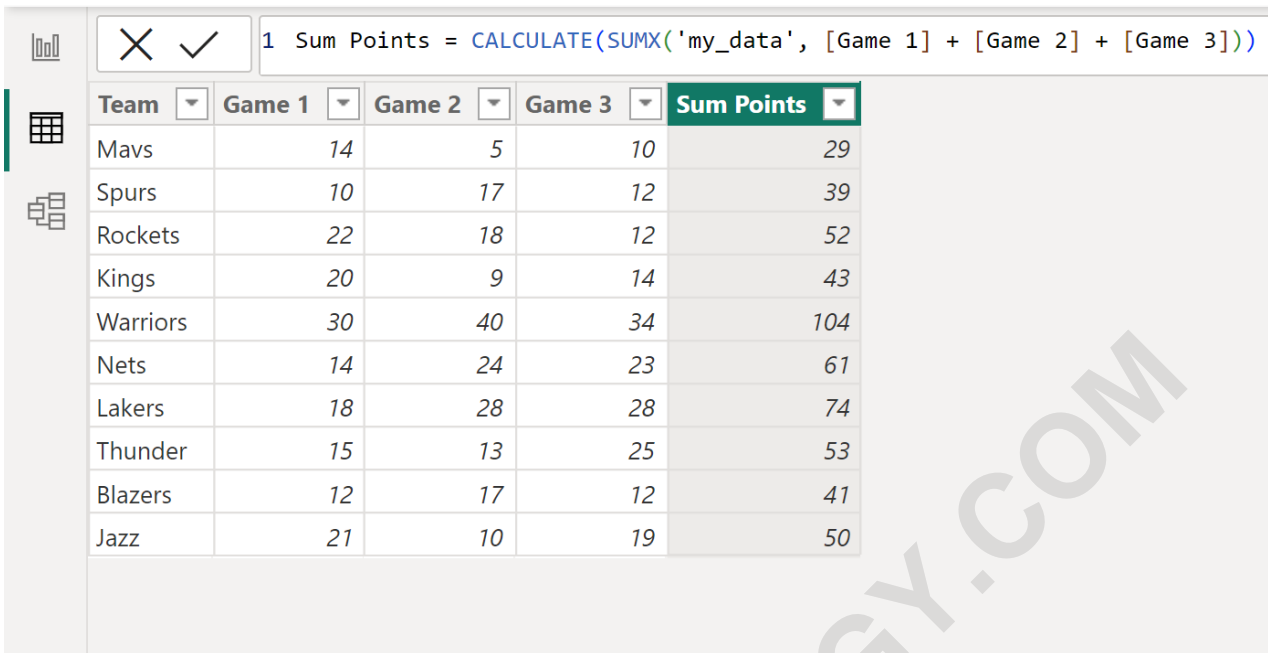
**Sum Points = CALCULATE(SUMX('my\_data', + + ))**

**Execution and Verification:** Press Enter to commit the formula. Power BI will instantly process the calculation across all rows of the `my\_data` table, creating the new column.

It is important to understand that while a simple addition like `Sum Points = + +` often suffices for this specific row-context calculated column scenario, utilizing the CALCULATE and SUMX structure is highly recommended. This practice provides a template that is easily adaptable for complex scenarios, such as adding conditional logic or filter modifications later on.

## Analyzing the Results

Upon successful execution of the DAX formula, a new column labeled **Sum Points** is added to your table. This column immediately displays the aggregated score for each player, combining the values from `Game 1`, `Game 2`, and `Game 3` as intended:



The screenshot shows the Power BI interface with a calculated column formula bar at the top. The formula is: `1 Sum Points = CALCULATE(SUMX('my_data', [Game 1] + [Game 2] + [Game 3]))`. Below the formula bar is a table with the following data:

Team	Game 1	Game 2	Game 3	Sum Points
Mavs	14	5	10	29
Spurs	10	17	12	39
Rockets	22	18	12	52
Kings	20	9	14	43
Warriors	30	40	34	104
Nets	14	24	23	61
Lakers	18	28	28	74
Thunder	15	13	25	53
Blazers	12	17	12	41
Jazz	21	10	19	50

The inclusion of this calculated column enhances the table's utility by providing a readily available metric for total performance. We can verify the accuracy of this new column by examining a few records:

The player representing the **Mavs** has scores of 14, 5, and 10. The summation confirms:  $14 + 5 + 10 = 29$  points.

The player representing the **Spurs** has scores of 10, 17, and 12. The summation confirms:  $10 + 17 + 12 = 39$  points.

The player representing the **Rockets** has scores of 22, 18, and 12. The summation confirms:  $22 + 18 + 12 = 52$  points.

This result validates the use of the `SUMX` function combined with column addition within the expression. This new column can now be used efficiently in visuals, measures, and further calculations without needing to repeatedly reference the three individual game score columns.

## Summary and Further Resources

Generating a column that sums multiple existing columns in Power BI is a foundational technique in data preparation. By mastering the usage of calculated columns and the iterative function SUMX, analysts can efficiently transform wide datasets into formats more suitable for insightful reporting and analysis.

While the direct addition of columns works in a simple row context, adopting the structured DAX

syntax involving `CALCULATE` and `SUMX` establishes a robust standard for your data modeling practice. This ensures your calculations are resilient and scalable as your data model grows in complexity.

To further enhance your skills in data manipulation using DAX within Power BI, consider exploring the following advanced tutorials which delve into related techniques:

The following tutorials explain how to perform other common tasks in Power BI:

ARABPSYCHOLOGY.COM