

# SUM to Add Values

by Sophia Tutorial



## WHAT'S COVERED

This tutorial explores using the SUM function to add values within a column together in two parts:

1. Getting Started
2. Special Cases

## 1. Getting Started

SUM is an aggregate function that returns the sum of all of the values in a column given a filter. Similar to the COUNT function, the SUM function ignores NULL values in a column. The SUM function is used on numeric columns. For example, if we would like to find the sum of the total amount for all the invoices, then we run the following:

```
SELECT SUM(total)
FROM invoice;
```

This would SUM the values in the total column that are not NULL, which would give the result:

Query Results	
Row count: 1	
sum	
2351	

We may also want to SUM based on certain criteria by using the WHERE clause. For example, we could find the SUM of the total amount for all invoices in the USA:

```
SELECT SUM(total)
FROM invoice
WHERE billing_country = 'USA';
```

Query Results	
Row count: 1	
sum	
	528

Anything that we can filter with the WHERE clause, we can then aggregate to find the SUM for.

Similar to the COUNT function, we can also use the DISTINCT option to calculate only the SUM of the DISTINCT values. For example, if we had 1, 2, 2, and 4, and we used the regular SUM function, the function would return 9. However, if we used DISTINCT, the query would return 7, as 2 was repeated.

## 2. Special Cases

If we tried to use the SUM function on a non-numeric column, it would result in an error:

```
SELECT SUM(billing_country)
FROM invoice;
```

Query Results	
Query failed because of: error: function sum(character varying) does not exist	

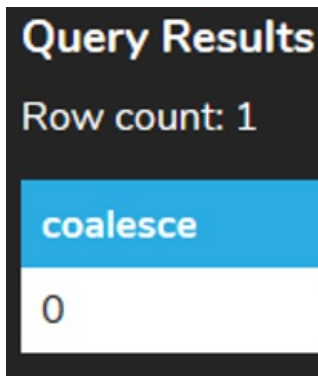
If the query returns only NULL values or if no rows are returned, the SUM returns NULL rather than 0. For example, if we're looking for the SUM of all of the invoices that have the invoice\_id < 1 (which does not exist):

```
SELECT SUM(total)
FROM invoice
WHERE invoice_id < 1;
The result will look like this:
```

Query Results	
Row count: 1	
sum	

If we wanted to return a 0 instead of NULL in this case, we would have to use the COALESCE function. This function returns the second argument if the first argument is NULL. For example:

```
SELECT COALESCE(SUM(total),0)
FROM invoice
WHERE invoice_id < 1;
```



The screenshot shows a 'Query Results' window. At the top, it says 'Row count: 1'. Below that, there is a table with one column named 'coalesce' and one row containing the value '0'.

coalesce
0

## Video Transcription

[MUSIC PLAYING] With the COUNT aggregate function you're able to count the number of rows that's going to be returned based on the data set. So in this case here, if we're selecting from the customer table, of course, by default it does show a row count. However, this is not shown in all cases and all databases, in which you'd actually have to utilize the COUNT function.

So one of the easy ways to be able to count all the different rows that's going to be returned is adding the COUNT with parentheses, and then the star in between. That'll count all the different rows associated with it. In this case here, it returns 59.

The other push that you can actually take is counting based on the primary key. In our case here, it's going to be customer\_id. And the reason you would actually do that in this case here is that every single customer\_id, because it's a primary key, would have a value.

One key, unique thing when it comes to the COUNT function is that it only counts rows that have a specific value associated with it if. There's a null value or an empty value, it will not be as part of the count.

So if you take a look back at the data itself, we'll notice that in the company column, not all the rows actually have a value for it. So if we utilize a COUNT on the company, it'll only count the number of rows that have a value for the company. So in this case here, only 10 customers actually have a company associated with them.

[MUSIC PLAYING]



Your turn! Open the SQL tool by clicking on the LAUNCH DATABASE button below. Then enter in one of the examples above and see how it works. Next, try your own choices for which columns you want the query to provide.



## SUMMARY

The SUM function returns the sum of the values in a column without NULL values.

Source: Authored by Vincent Tran