

Aggregate Functions

by Sophia Tutorial



WHAT'S COVERED

This tutorial explores the different types of aggregate functions that can be used for querying tables:

1. Introduction
2. Aggregate Functions

1. Introduction

Most of the queries that we have been working with so far have been focused on one row at a time and looking at individual records. However, a database works using sets of data rather than individual records. There are a lot of questions that can be answered with a database by aggregating the data together as if it were a single unit. This type of processing is done using aggregate functions. Although different databases may have some specialized aggregate functions, there are many common ones that are available in most databases. These aggregate functions take a collection of rows and reduce the resulting set to one row.

SQL can perform various mathematical summaries for us within the database to help answer questions. For example, how many invoices were there in January? Of course we could count, but sometimes counting is not a viable option. We also might want to know what was the smallest invoice total in the prior year, or the maximum invoice total of all time, or the average quantity ordered across all invoices.

2. Aggregate Functions

Aggregate functions can help us answer these questions. Here is a list of the most common aggregate functions. We will get into more specifics on each in later tutorials.

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

```
SELECT COUNT(customer_id)
FROM customer
WHERE country = 'USA';
```

```
SELECT MAX(total)
FROM invoice
```

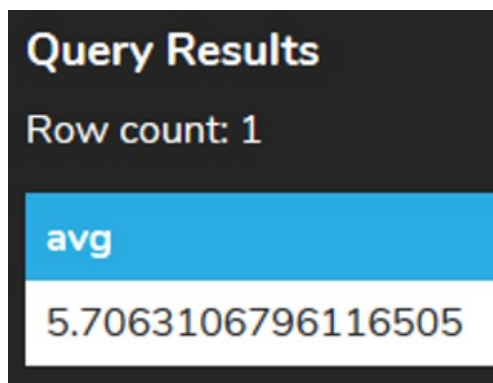
```
WHERE invoice_date between '2009-01-01' AND '2010-01-01';
```

```
SELECT MIN(total)
FROM invoice
WHERE invoice_date < '2011-01-01';
```

```
SELECT SUM(quantity)
FROM invoice_line;
```

AVG The AVG function calculates the average of non-null values.

As an example, if we were looking for the average of the total items purchased across all invoices, we could run:

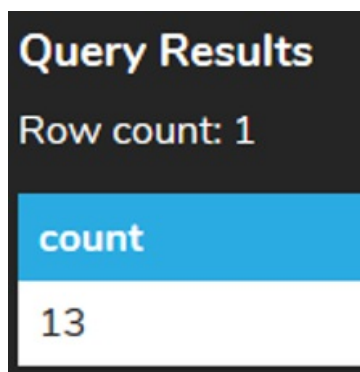


A screenshot of a database query result. It has a black header with 'Query Results' in yellow. Below it, 'Row count: 1' is shown in yellow. The result is a single row with a blue header 'avg' and a white value '5.7063106796116505'.

avg
5.7063106796116505

COUNT The COUNT function returns the number of rows in a group including those that have NULL values.

If we wanted to find out how many customers lived in the country USA, we would run:



A screenshot of a database query result. It has a black header with 'Query Results' in yellow. Below it, 'Row count: 1' is shown in yellow. The result is a single row with a blue header 'count' and a white value '13'.

count
13

MAX The MAX function returns the largest of the non-null values.

If we wanted to find the largest order made between January 1, 2009 and January 1, 2010, we would run:

Query Results	
Row count: 1	
max	
	14

MIN The MIN function returns the smallest of the non-null values.

If we wanted to find the smallest total prior to January 1, 2011, we would run:

Query Results	
Row count: 1	
min	
	1

SUM The SUM function returns the sum of all of the non-null values.

If we wanted to find out the number tracks ordered for all time, we would run:

Query Results	
Row count: 1	
sum	
	2240

In PostgreSQL, there are other unique aggregate functions that can be used, including:

BOOL_AND This returns true if all of the input values are true; otherwise, it returns false.

BOOL_OR This returns true if one of the input values is true; otherwise, it returns false.

STDDEV This function returns the standard deviation based on the non-null values.

VARIANCE This function returns the variance of the non-null values.

RANK This function returns the rank of the row based on the value. If a row has the same value as the prior row, it will return the same rank.

There are many others that can be used, but this will be a good starting point to build from.

Video Transcription

[MUSIC PLAYING] When it comes to aggregate functions, it allows us to be able to perform calculations based on multiple different rows, as a group, as a whole. So if we want to for example, add up all the different unit prices within the invoice table, we could certainly go through and then count and add up each one at a time, of all the 2,240 different rows. However, with the aggregate functions, we can be able to do that all in one go.

So in this case here, instead of adding them individually, we can do a Select. And then we'll call the function itself, which is going to be SUM. And then in parentheses, we have the column name, which is unit price in this case, and then the table name. Once you run that, it'll automatically add up all the different items for us, and then come up with the results.

There are multiple different aggregate functions that are available-- SUM, AVERAGE, COUNT, MAX, and MIN are the most common ones that you'll be able to utilize. Another example here is if we want to take a look at the average total for all the invoices, we can certainly run it like this, and it'll automatically calculate it for us.

With the COUNT, it allows us to be able to count the number of rows that fits a specific criteria. It can certainly do a row count as we see here. But if we're looking for specific things-- like for example, if we want to count the number of customers who live in the US, we can go ahead and enter it in this function here. Select COUNT as the function. And then in parentheses, we're going to select the customer ID.

And then we're still going to have the filter with the country, USA. And then it will return us the count. If we actually removed the aggregate function itself and ran that, we will still see that the real count that returns is going to be 13, with all the different customer IDs associated with them. Note that not all databases will actually display the row count as we want to return that value, in some cases. So that's why we would utilize that aggregate function itself.

[MUSIC PLAYING]



TRY IT

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

Aggregate functions allow us to take a collection of rows and reduce the results down to a single row.

Source: Authored by Vincent Tran

