

Congratulations! You passed!

Grade received 100% To pass 100% or higher

Go to next item

1.

1 / 1 point



Activity overview

In previous lessons, you were introduced to SQL. In this activity, you will practice it by learning how to write basic calculations in an SQL query.

By the time you complete this activity, you will be able to use operators in SQL queries. This will enable you to make calculations using SQL and combine multiple arithmetic operations in a single query. This will help you to work faster and more efficiently in SQL as you quickly discover significant patterns in your data—a key skill for data analysts.

Operations in SQL

In SQL, the symbols for the four basic arithmetic operations are:

- + addition
- subtraction
- \* multiplication
- / division

For this activity, imagine you've been asked to analyze subway ridership data to help improve the quality of the city's public transportation. You can use basic calculations in SQL to help you make sense of this data.

Access the dataset

Log in to [BigQuery Sandbox](#). If you have a free trial version of BigQuery, you can use that instead. On the BigQuery page, click the **Go to BigQuery** button.

- Note:** BigQuery Sandbox frequently updates its user interface. The latest changes may not be reflected in the screenshots presented in this activity, but the principles remain the same. Adapting to changes in software updates is an essential skill for data analysts, and it's helpful for you to practice troubleshooting. You can also reach out to your community of learners on the discussion forum for help.

In this activity, you'll use data that describes the average weekly subway ridership in New York City from 2013-2018. The columns include: subway stations, subway routes, weekly ridership for each year (2013-2018), the change in ridership from 2017 to 2018 (both in raw numbers and percent), and the ridership rank of each station in 2018. Each row corresponds to a different station in the subway system.

- In order to find the **new\_york\_subway** dataset, you will want to make sure you have the **bigquery-public-data** pinned in your explorer menu. Follow these steps to pin the dataset.

- Navigate to the **Explorer** menu in BigQuery.
- Type the word **public** in the search box and enter.
- Click "Broaden search to all projects".
- Find the **bigquery-public-data** and pin it.

To examine the dataset, follow these steps:

- Make sure that **bigquery-public-data** is pinned to the **Explorer** menu of your SQL Workspace.
- If it is not pinned, click **+ ADD DATA** and add any BigQuery public dataset (such as **noaa\_lightning**).
- Open the bigquery-public-data dropdown in the **Explorer** menu and scroll until you find **new\_york\_subway**.
- Open the dropdown and click **subway\_ridership\_2013\_present** to open and examine the dataset.

Use a calculation with a single operator

The **change\_2018\_raw** column describes the change in weekly ridership from 2017 to 2018 in raw numbers. Suppose you want to find data on the change in weekly ridership from 2013 to 2014. You can use SQL to subtract the number of riders in 2013 from the number of riders in 2014.

To do that, follow these steps:

- In the Query editor, type **SELECT**. Add the names of the columns you want to use in your calculations. You're selecting several columns, so press **Enter** (Windows) or **Return** (Mac) after **SELECT**. Add a **comma** after each column name. Include the following columns: **station\_name**, **ridership\_2013**, and **ridership\_2014**.
- Add the calculation to the query. Enter the names of the two columns with a minus sign between them: **ridership\_2014 - ridership\_2013**.
- List the result in a new column. To do this, type **AS** followed by the new column's name. Name it **change\_2014\_raw** since it represents the change in ridership from 2013 to 2014 in raw numbers.
- End your query with the **FROM** command and the name of the dataset and subset you're pulling data from. After FROM, press **Enter** or **Return** and type **bigquery-public-data.new\_york\_subway.subway\_ridership\_2013\_present**.

Query editor

```
1 SELECT
2 station_name,
3 ridership_2013,
4 ridership_2014,
5 ridership_2014-ridership_2013 AS change_2014_raw
6 FROM
7 bigquery-public-data.new_york_subway.subway_ridership_2013_present
```

- Now, click **Run** and get the results.

The results show the change in ridership from 2013 to 2014. For example, the *Atlantic Av - Barclays Ctr* station gained an average of 1,774 riders per week. The *4 Av* station lost 321 riders.

By including a basic calculation in your query, you can get an idea of the change in ridership for each subway station in any given year. Basic calculations help you quickly gain important knowledge about your data.

Query results				
<div><div>SAVE RESULTS</div><div>EXPLORE DATA</div></div>				
Query complete (0.0 sec elapsed, cached)				
<div>Job informationResultsJSONExecution details</div>				
Row	station_name	ridership_2013	ridership_2014	change_2014_raw
1	Atlantic Av - Barclays Ctr	39871	41645	1774
2	4 Av	13156	12835	-321
3	14 St / 6 Av	49316	49990	674
4	Jamaica - Van Wyck	4992	5121	129
5	Crown Hts - Utica Av	27780	28287	507

Use a calculation with multiple operators

The original dataset describes average weekly ridership for each individual year. Suppose you want to find average weekly ridership for a longer period of time, such as the multi-year period from 2013-2016.

To do this, you can combine multiple arithmetic operations in a query. The average of a set of numbers is the sum of the numbers divided by the total number of values in the set. There are four values in your new set (ridership data for 2013, 2014, 2015, 2016). You can use SQL to sum the numbers for each year and divide that sum by 4.

- First, click the **COMPOSE NEW QUERY** button to refresh the query editor.
- Type **SELECT** to select the columns you want to pull from the table. You're selecting several columns, so press **Enter** or **Return** after SELECT and add a **comma** after each column name. For this query, enter **station\_name**, **ridership\_2013**, **ridership\_2014**, **ridership\_2015**, and **ridership\_2016**.
- Add the calculation to the query. If you use more than one arithmetic operator in a calculation, you need to use parentheses to control the order of the operations. In this case, you want to sum the years, and then divide the sum by 4. So, put parentheses around the sum of the four column names. Type **(ridership\_2013 + ridership\_2014 + ridership\_2015 + ridership\_2016)**. Then type a division operator **/** and the number **4**.
- Next, list the result in a new column by typing **AS** followed by the new column's name. Call the new column **average** since it represents average weekly ridership for the period 2013-2016.
- End your query with the **FROM** command and the name of the dataset and subset that you're pulling data from. After FROM, press **Enter** and type **bigquery-public-data.new\_york\_subway.subway\_ridership\_2013\_present**.

At this point, your screen should display like this:

Query results				
<div><div>RUN</div><div>MORE</div><div>SCHEDULE</div><div>CANCEL</div><div>This query will process 19.1 KiB when run.</div></div>				
1	SELECT			
2	station_name,			
3	ridership_2013,			
4	ridership_2014,			
5	ridership_2015,			
6	ridership_2016,			
7	(ridership_2013 + ridership_2014 + ridership_2015 + ridership_2016) / 4 AS average			
8	FROM			
9	bigquery-public-data.new_york_subway.subway_ridership_2013_present			

- Click the **Run** button to get the results. The results clearly show the trend in ridership at each station from 2013 to 2016. For example, weekly ridership at the *Atlantic Av - Barclays Ctr* station increased every year since 2013. Further, for the years 2014, 2015, and 2016, weekly ridership at *Atlantic Av - Barclays Ctr* exceeded the overall average for the period 2013-2016 (listed in the **average** column).

Query results						
<div><div>SAVE RESULTS</div><div>EXPLORE DATA</div></div>						
Query complete (0.3 sec elapsed, 19.1 KB processed)						
<div>Job informationResultsJSONExecution details</div>						
Row	station_name	ridership_2013	ridership_2014	ridership_2015	ridership_2016	average
1	Atlantic Av - Barclays Ctr	39871	41645	42231	42711	41614.5
2	4 Av	13156	12835	13126	13116	13058.25
3	14 St / 6 Av	49316	49990	50335	50692	50083.25
4	Jamaica - Van Wyck	4992	5121	5092	5022	5056.75
5	Crown Hts - Utica Av	27780	28287	28146	29069	28320.5

This kind of data is useful for managing public transportation. It can help you determine which stations or routes to expand due to increased ridership. Using basic calculations in your query allows you to quickly discover significant patterns in your data.

Confirmation and reflection

Using the same average formula as the last query you wrote, write a query to find the average weekly ridership from the years 2016-2018. What is the average weekly ridership for the Atlantic Av - Barclays Ctr station in this timeframe?

- ☐ 49255
- ☒ 42672.33
- ☐ 13212.67
- ☐ 4903.67

Correct

The average weekly ridership for the Atlantic Av - Barclays Ctr from the years 2016-2018 is 42672.33. To find the station's total average weekly ridership, you would have to use a SQL calculation to add the three columns ridership\_2016, ridership\_2017, and ridership\_2018, then divide the sum by 3. Going forward, you can use SQL to perform mathematical calculations with data and analyze patterns in real-world situations.

- In this activity, you wrote queries with single and multiple calculations to find patterns in subway ridership data. In the text box below, write 2-3 sentences (40-60 words) in response to each of the following questions:

- What other patterns can you discover in the subway ridership dataset using calculations with multiple operators?
- How can basic calculations in SQL help you make sense of a large amount of data?

What other patterns can you discover in the subway ridership dataset using calculations with multiple operators?

How can basic calculations in SQL help you make sense of a large amount of data?

Correct

Congratulations on completing this hands-on activity! A good response would include a pattern you found in the ridership dataset, as well as some of the following insights about SQL calculations:

Making basic calculations is a key part of any data analysis. Using basic calculations in SQL can help you quickly discover significant patterns, relationships, and trends in a large dataset. When you include a calculation in a query with your other commands, you can work faster and more efficiently.

1 / 1 point