

Your grade: 100%

Next item →



Activity Overview

Previously, you learned about basic calculations with SQL. Just like spreadsheets, you can use the four basic arithmetic operations to perform addition, subtraction, multiplication, and division. Now, you'll practice writing basic calculations using single and multiple operators in BigQuery. This will enable you to combine multiple arithmetic operations in a single query, allowing you to quickly and efficiently discover significant patterns in your data.

Step-By-Step Instructions

Follow the instructions to complete each step of the activity. Then answer the question at the end of the activity before going to the next course item.

Step 1: Access the dataset

For this activity, you'll analyze subway ridership data to help improve the quality of the city's public transportation. To do this, you'll use data that describes the average weekly subway ridership in New York City from 2013-2018.

1. Navigate to your [BigQuery console](#).

- **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.

Step 2: Examine the dataset

1. Locate the public data by **typing** the word **public** in the **search bar**. Select **SEARCH ALL PROJECTS** and **SHOW MORE** to access all of the public datasets.

Explorer

Type to search X ?

Found 39 results.

SEARCH ALL PROJECTS

- bigquery-public-data
- covid19_public_forecasts
- patents-public-data
- google_patents_research

SHOW MORE

2. Select the **star icon** next to **bigquery-public-data**. This will pin it to the Explorer menu.

Explorer

Type to search X ?

Found 39 results.

SEARCH ALL PROJECTS

- bigquery-public-data
- covid19_public_forecasts
- patents-public-data
- google_patents_research
- publications

3. Select the **dropdown arrow** next to the **bigquery-public-data** to expand all the available datasets.

4. Navigate down the list until you find the **new_york_subway** dataset.

Note: You may have to select the **SHOW MORE** button twice to reach the correct dataset. Now, select the **dropdown arrow** next to **new_york_subway** and select **subway_ridership_2013_present**.

Explorer

Type to search X ?

Found 3 results.

SEARCH ALL PROJECTS

- ncaa_basketball
- nces_ipeds
- new_york
- new_york_311
- new_york_citibike
- new_york_mv_collisions
- new_york_subway

Note: This dataset may not be accessible through other search methods. Follow the previous steps to locate the correct data for this activity.

Note also that any of the public databases on BigQuery are living records and, as such, are periodically updated with new data. Throughout this course (and, if your results differ from those you encounter in videos or screenshots, there's a good chance it is due to a data refresh).

Step 3: Use a calculation with a single operator

First, identify data exploring the change in weekly ridership from 2013 to 2014. Use SQL to subtract the number of riders in 2013 from the number of riders in 2014.

- Select the **dropdown arrow next to QUERY** and select **In split tab**. You will then be able to see the Query editor and the datasets. If you prefer to only see the Query editor, you could select **In new tab**.

Field name	Type	Mode
station_name	STRING	NULLABLE
routes	STRING	NULLABLE
ridership_2013	INTEGER	NULLABLE
ridership_2014	INTEGER	NULLABLE
ridership_2015	INTEGER	NULLABLE
ridership_2016	INTEGER	NULLABLE

- In the Query editor, add the names of the columns you want to use in your calculations in your **SELECT** statement. In this case, select the following columns:

- [station_name](#)
- [ridership_2013](#)
- [ridership_2014](#)

Be sure to add a comma after each column name. 2. Add the calculation to the query by adding the names of the two columns with a minus sign between them.

- [ridership_2014 - ridership_2013](#)

3. List the result in a new column. To do this, enter **AS** followed by the new column's name. Name it **change_2014_raw** because 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**, select **Enter** or **Return** and enter **bqquery-public-data.new_york_subway.subway_ridership_2013_present**.

```

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

- Select **RUN** to get the results.

The results show the change in ridership from 2013 to 2014 at different stations. 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 quickly gain important knowledge about your data. In this example, you now have insights into the change in ridership for each subway station in any given year.

Query results			
Job complete (0.0 sec elapsed, cached)			
	station_name	ridership_2013	ridership_2014
1	Atlantic Av - Barclays Ctr	31971	41645
2	4 Av	13156	12835
3	14 St / 6 Av	49316	49990
4	Jamaica Ave - Van Wyck	4992	5121
5	Crown Hts - Utica Av	27780	28287

Step 4: Use a calculation with multiple operators

Next, find the average weekly ridership for a longer period of time, such as the multi-year period from 2013-2016.

To do this, 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). Use SQL to sum the numbers for each year and divide that sum by 4.

- Select the **COMPOSE NEW QUERY** button to refresh the query editor.
- Enter **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. The columns selected for this query will be:
 - [station_name](#)
 - [ridership_2013](#)
 - [ridership_2014](#)
 - [ridership_2015](#)
 - [ridership_2016](#)
- Add the calculation to the query. Use parentheses to control the order of the operations if you use more than one arithmetic operator in a calculation. In this case, sum the years, then divide the sum by 4. Put parentheses around the sum of the four column names. Enter **(ridership_2013 + ridership_2014 + ridership_2015 + ridership_2016) / 4**. Then enter a division operator / and the number 4.
- List the result in a new column by entering **AS** followed by the new column's name. Name the new **average** because 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**, select

Enter and type `bigrquery-public-data.new_york_subway.subway_ridership_2013_present`.

RUN **MORE** **SCHEDULE** **CANCEL** This query will process 19.1 kB when run.

```
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   bigrquery-public-data.new_york_subway.subway_ridership_2013_present
```

6. Select **RUN** 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

SAVE RESULTS

EXPLORE DATA

Query complete (0.3 sec elapsed, 19.1 KB processed)

Job information **Results** JSON Execution details

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.

1. Reflection

1 / 1 point

Now you've spent some time performing calculations using SQL. In the text box below, write 2-3 sentences (40-60 words) in response to each of the following questions:

- When working with datasets, when might you use basic SQL calculations?
- How can those basic SQL calculations help make sense of large quantities of data?

any time you need to quickly aggregate, summarize, or transform the raw data within a database. This is essential when creating initial reports, preparing data for more complex analysis, or simply getting a snapshot of performance metrics. simple calculations are crucial for making large quantities of data manageable because they allow you to reduce thousands of records into a few meaningful numbers. By providing instant summaries (e.g., total sales, average customer age), they help you quickly identify key trends and patterns in the data.

- An effective reflection would note that you can use basic calculations in your query to help you quickly discover significant patterns in your data. While this example used public transportation, queries can be used with all sorts of datasets to provide you with useful information and patterns.

 Like  Dislike  Report an issue