Congratulations! You passed!

Grade received 100% To pass 100% or higher

Go to next item

1. 1 / 1 point

Activity overview

So far, you've learned about SQL and used SQL queries to interact with databases. In this activity, you'll practice sorting data by using SQL queries with ORDER BY and WHERE clauses.

By the time you complete this activity, you will be able to write queries that sort data depending on your needs. This will enable you to organize and use data more effectively in your career as a data analyst.

Sorting with SQL

To practice sorting data with SQL, you'll query the CDC Births Data Summary public dataset. The queries you write will help you obtain some answers about which counties in the United States have the most and least births in the years 2016-2018.

Load the dataset

1. 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.

2. If you have never created a BigQuery project before, click **CREATE PROJECT** in the right side of the screen. If you have created a project before, you can use an existing one or create a new one by clicking the project dropdown in the blue header bar and selecting **NEW PROJECT**.

3. Name your project something that will help you identify it later. You can give it a unique project ID or use an auto-generated one. Don't worry about selecting an organization if you don't know what it is.

4. Now, you'll see the **Editor** interface. In the middle of the screen is a window where you can type code, and to the left is the **Explorer** menu where you can search for datasets.

5. Click **+ ADD DATA** at the top of the Explorer menu, then **Explore public datasets** from the resulting dropdown.

6. In the **Search Marketplace** bar, type **sdoh_cdc_wonder_natality**.

7. Click the **CDC Births Data Summary**.

8. Click **View Dataset**. This will bring you back to the BigQuery Sandbox interface in a new tab.

   - **Note:** This may pin the **bigquery-public-data** dropdown to the **Explorer** menu. You can use this to browse datasets and tables.

9. Click back to the **Editor** tab. This is where you'll write your queries.

10. Copy, paste, and run the following query to display the first 1,000 rows of the **county_natality** table:

```
SELECT
  *
FROM
  `bigquery-public-data.sdoh_cdc_wonder_natality.county_natality`
LIMIT
  1000
```

After the query has run, your results should appear like this:

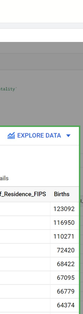
Row	Year	County_of_Residence	County_of_Residence_FIPS	Births
1	2018-01-01	Calhoun County, AL	01015	1265
2	2018-01-01	Tulsa County, OK	40143	8933
3	2018-01-01	Carroll County, GA	13045	1540
4	2018-01-01	Saginaw County, MI	26145	2182
5	2018-01-01	Hillsborough County, FL	12057	17126
6	2018-01-01	Lake County, IN	18089	5785
7	2018-01-01	St. Tammany Parish, LA	22103	2932
8	2018-01-01	Osceola County, FL	12097	4437
9	2018-01-01	Sarpy County, NE	31153	2386
10	2018-01-01	Kane County, IL	17089	6337
11	2018-01-01	San Juan County, NM	35045	1444

Use the ORDER BY clause

Examine the dataset you just loaded. Take a moment to familiarize yourself with the columns and get a feel for what each can tell you.

Now, imagine you were asked by your manager to figure out which 10 counties had the lowest birth count for 2016-2018. You could accomplish this by modifying your query to use the ORDER BY clause:

```
SELECT
  *
FROM
  `bigquery-public-data.sdoh_cdc_wonder_natality.county_natality`
ORDER BY
  Births
LIMIT
  10
```



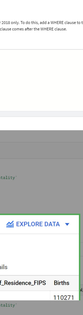
Row	Year	County_of_Residence	County_of_Residence_FIPS	Births
1	2018-01-01	Tompkins County, NY	36109	735
2	2018-01-01	Unidentified Counties, HI	15999	749
3	2017-01-01	Tompkins County, NY	36109	767
4	2016-01-01	Tompkins County, NY	36109	787
5	2018-01-01	Unidentified Counties, MA	25999	802
6	2016-01-01	Unidentified Counties, HI	15999	845
7	2017-01-01	Washington County, RI	44009	852
8	2017-01-01	Unidentified Counties, HI	15999	857
9	2017-01-01	Unidentified Counties, MA	25999	866
10	2018-01-01	Fayette County, GA	13113	867

You may have noticed that the query did not specify whether it should be sorted **ASC** (ascending) or **DESC** (descending). When this is not specified, SQL defaults to sorting by ascending order. You can run another query to confirm this.

Copy, paste, and run the following query that includes ASC:

```
SELECT
  *
FROM
  `bigquery-public-data.sdoh_cdc_wonder_natality.county_natality`
ORDER BY
  Births
ASC
LIMIT
  10
```

You'll find that the results did not change. Notice that Tompkins County, NY, had just 735 births in 2018—the lowest birth count of any county in the US between 2016-2018.



Row	Year	County_of_Residence	County_of_Residence_FIPS	Births
1	2018-01-01	Tompkins County, NY	36109	735
2	2018-01-01	Unidentified Counties, HI	15999	749
3	2017-01-01	Tompkins County, NY	36109	767
4	2016-01-01	Tompkins County, NY	36109	787
5	2018-01-01	Unidentified Counties, MA	25999	802
6	2016-01-01	Unidentified Counties, HI	15999	845
7	2017-01-01	Washington County, RI	44009	852
8	2017-01-01	Unidentified Counties, HI	15999	857
9	2017-01-01	Unidentified Counties, MA	25999	866
10	2018-01-01	Fayette County, GA	13113	867

Use DESC to reverse sorting order

Now, modify the query to sort in the other direction, returning the top 10 counties with the highest yearly birth counts between 2016-2018.

Copy, paste, and run the following query:

```
SELECT
  *
FROM
  `bigquery-public-data.sdoh_cdc_wonder_natality.county_natality`
ORDER BY
  Births
DESC
LIMIT
  10
```

Your results should appear like this:

Row	Year	County_of_Residence	County_of_Residence_FIPS	Births
1	2016-01-01	Los Angeles County, CA	06037	123092
2	2017-01-01	Los Angeles County, CA	06037	116950
3	2018-01-01	Los Angeles County, CA	06037	110271
4	2016-01-01	Harris County, TX	48201	72420
5	2017-01-01	Harris County, TX	48201	68422
6	2018-01-01	Harris County, TX	48201	67095
7	2016-01-01	Cook County, IL	17031	66779
8	2017-01-01	Cook County, IL	17031	64374
9	2018-01-01	Cook County, IL	17031	61797
10	2016-01-01	Unidentified Counties, TX	48999	59168

Now, the query returns the 10 rows with the largest values in the Birth column. Los Angeles County takes up the top three spots.

Combine ORDER BY with WHERE clauses

Next, modify the query so that it returns the top 10 counties with the highest birth counts for 2018 only. To do this, add a WHERE clause to the query that specifies only rows that have a year value equal to 2018-01-01. Note how the ORDER BY clause comes after the WHERE clause.

Copy, paste, and run the following query:

```
SELECT
  *
FROM
  `bigquery-public-data.sdoh_cdc_wonder_natality.county_natality`
WHERE
  Year = '2018-01-01'
ORDER BY
  Births
DESC
LIMIT
  10
```

Your results should appear something like this:

Row	Year	County_of_Residence	County_of_Residence_FIPS	Births
1	2018-01-01	Los Angeles County, CA	06037	110271

The query worked. You successfully used both ORDER BY and WHERE clauses in the same query.

Confirmation and reflection

The last query you ran returned the top 10 counties with the highest birth counts for 2018 only. Remove the LIMIT statement and run the query again. What is the county with the 10th highest birth count?

- Orange County, CA
- Dallas County, TX
- Unidentified Counties, KY
- Miami-Dade County, FL

Correct

The county with the 11th highest birth count in 2018 is Orange County, CA. To find this answer, you ran a query with an ORDER BY clause and a WHERE clause. Going forward, you can use this knowledge of SQL to better organize and structure your data.

2. In this activity, you practiced sorting data using SQL queries with ORDER BY and WHERE clauses. In the text box below, write 2-3 sentences (40-60 words) in response to each of the following questions:

1 / 1 point

- How can the ORDER BY clause help you organize and structure your data?
- Why is it helpful to use the ORDER BY and WHERE clauses together when sorting and filtering data?
- Can you think of a business question that you could answer using this method?

How can the ORDER BY clause help you organize and structure your data?

Why is it helpful to use the ORDER BY and WHERE clauses together when sorting and filtering data?

Can you think of a business question that you could answer using this method?

Correct

Congratulations on completing this hands-on activity! A good response would include that sorting the data you return in your queries is a crucial tool for analyzing and understanding data.

You can also answer business questions by sorting the dataset according to a given metric. For instance, a store may want to know which products they sell the most or least. Sorting helps you answer business questions that involve phrases such as “how much,” “how many,” “best,” or “worst”—which will be a valuable skill in your career as a data analyst.