

Congratulations! You passed!
Grade received 100% To pass 100% or higher



1. 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 efficiently 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.

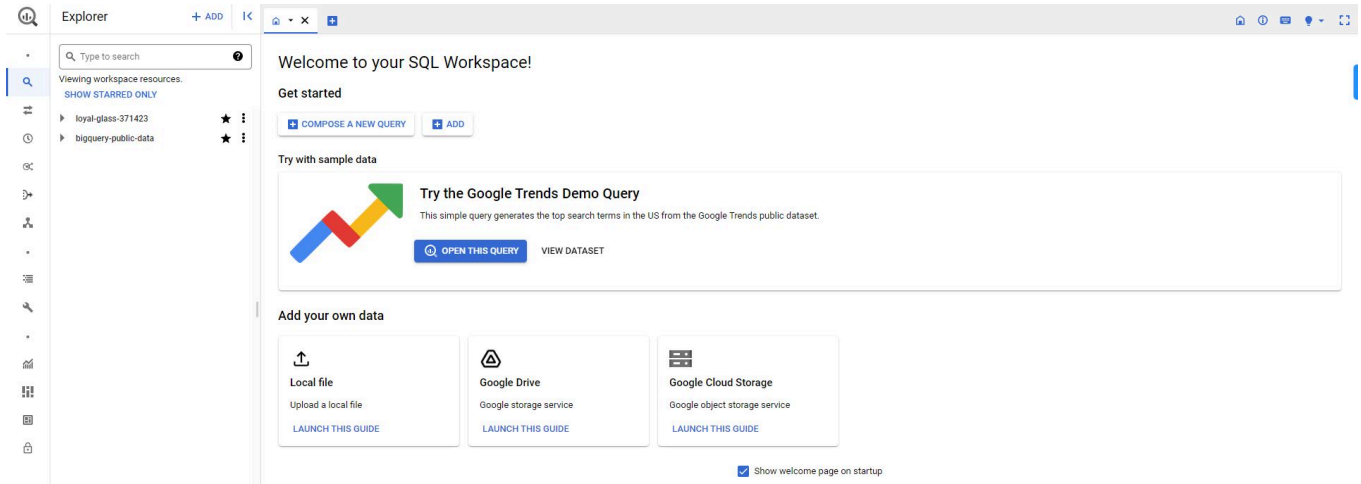
Load the dataset

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 on 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 drop-down in the data box to the right of the *Google Cloud* icon in the top left corner and selecting NEW PROJECT.

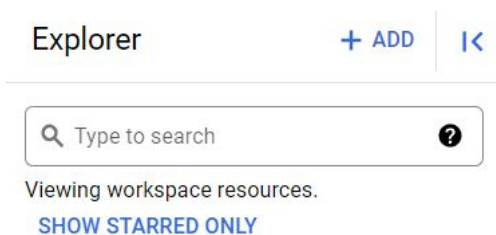


3. Name your project something that will help you identify it later. You can give it a unique project ID or use the auto-generated name that is created for you in the data box. Within the *Location* section, keep the default option set to No organization.

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 at the top of the Explorer menu pane, which will call up the Add menu of various types of public data repositories.



6. Navigate down the list in the Add window to select the Public Datasets option.



7. In the Marketplace search bar, type sdoh_cdc_wonder_natality.

 Marketplace

sdoh_cdc_wonder_natality

Marketplace > Datasets

Filter Type to filter

Category ^

- Maps (7)
- Big data (18)
- Analytics (21)
- Databases (6)
- Machine learning (4)


Type


Datasets


Price ^


Free (201)


201 results


**About COVID-19 Public Datasets**
BigQuery Public Datasets Program
Getting started with COVID-19 Public Datasets


**About Cymbal: Google Cloud's demo brand**
Cymbal Group
Synthetic datasets across industries showcasing Google Cloud.


**AFSC Open Data Portal**
NOAA
Fisheries research data for the Alaska region


**American Community Survey (ACS)**
United States Census Bureau
Detailed US demographic data at various geographic resolutions

**Area Deprivation Index (ADI)**
BroadStreet
ADI: An index of socioeconomic status for communities

**Austin Crime Data**
City of Austin
City of Austin crime data for 2014 and 2015

**Band Protocol Data**
Cloud Public Datasets - Finance
Band Protocol data loaded into BigQuery

**Births Data Summary**
Centers for Disease Control
Natality Data from CDC Births

**Bitcoin Cash Cryptocurrency Dataset**
Bitcoin Cash
The Bitcoin Cash blockchain loaded to BigQuery

8. In the resulting window, click the CDC Births Data Summary. After reviewing the detailed dataset info, click on the VIEW DATASET button.

[←](#) Product details

Births Data Summary

[Centers for Disease Control](#)

Nativity Data from CDC Births

[VIEW DATASET](#) [↗](#)[OVERVIEW](#)[SAMPLES](#)[RELATED PRODUCTS](#)

Overview

This dataset contains natality data based on CDC-collected statistics for live births occurring within the United States to U.S. residents.

The data capture a range of maternal demographic information, such as state and county of residence, mother's age and race, ethnicity and country of origin, marital status, and education. It also includes health and medical data on these mothers, including prior birth history, prenatal care visits, WIC enrollment, tobacco use, method of delivery, method of payment, and congenital anomalies and other morbidity data.

Beyond maternal characteristics, this dataset also illustrates both paternal and infant information that may be relevant to understanding certain social determinants of health. Paternal characteristics include age, race and ethnicity (including country of origin), education. Infant characteristics: gender, birth weight, delivery, congenital abnormalities.

For researchers and population health teams, this data can be used to identify localities that have had higher-than-average complicated and high-cost births and give insight into possible targeting strategies based on population characteristics. The data are derived from birth certificates, reported to the CDC. For more information, see [here](#) [↗](#).

Additional details

Type: [Datasets](#)Category: [Datasets for COVID-19 research](#), [Healthcare](#)Dataset source: [CDC](#) [↗](#)

Cloud service: BigQuery

Expected update frequency: Annually

9. This will bring you back to the BigQuery Sandbox interface in a new tab. This will also open up the bigquery-public-data drop-down list in the Explorer menu. You can use this to browse datasets and tables.

Explorer

+ ADD

sdoh_cdc_wonder_natality

sdoh_cdc_wonder_natality

Dataset info

Dataset ID	bigquery-public-data.sdoh_cdc_wonder_natality
Created	Mar 26, 2020, 8:28:31 PM UTC-5
Default table expiration	Never
Last modified	Sep 20, 2022, 2:58:15 AM UTC-5
Data location	US
Description	
Default collation	
Default rounding mode	ROUNDING_MODE_UNSPECIFIED
Case insensitive	false
Labels	freebqcovid : freebqcovid
Tags	

SHOW MORE

10. Navigate back to the Editor tab, and click on the blue + button to the right of the Dataset info tab, which will open up a new workspace.



11. This is where you will generate a new query from the public data list. Copy, paste, and run the following query to display the first 1,000 rows of the county_natality table:

```
1 SELECT
2   *
3 FROM
4   bigquery-public-data.sdoh_cdc_wonder_natality.county_natality
5 LIMIT
6   1000
```

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

UNSAVE... CANCEL SAVE SCHEDULE MORE

```

1 SELECT
2 *
3 FROM
4 `bigquery-public-data.sdo_h_cdc_wonder_natality.county_natality`
5 LIMIT
6 1000
7

```

Query results

[SAVE RESULTS](#)
[EXPLORE DATA](#)

Query complete (0.3 sec elapsed, 168.4 KB processed)

[Job information](#)
[Results](#)
[JSON](#)
[Execution details](#)

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

Note: In some cases, pasting the SQL syntax into the FROM line *without* the *tick marks* will most likely work. If you receive an error, try adding tick marks to the ends of the Table ID syntax, and this may fix the error. Follow this method for the rest of the *copy + paste* code chunks for the remainder of the activity.

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.

Copy, paste, and run the following query:

```
1 SELECT
2   *
3 FROM
4   bigquery-public-data.sdohc_dc_wonder_natality.county_natality
5 ORDER BY
6   Births
7 LIMIT
8   10
```

The results of your query should appear like this:

*UNSAVE...

COUNTY...

CANCEL

SAVE

SCHEDULE

MORE

```

1  SELECT
2    *
3  FROM
4    `bigquery-public-data.sdo_h_cdc_wonder_natality.county_natality`
5  ORDER BY
6    Births
7  LIMIT
8    10
9
    
```

Query results

SAVE RESULTS
 EXPLORE DATA

Query complete (0.1 sec elapsed, cached)

Job information
 Results
 JSON

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:

```

1  SELECT
2    *
3  FROM
4    bigquery-public-data.sdo_h_cdc_wonder_natality.county_natality
5  ORDER BY
6    Births
7  ASC
8  LIMIT
9    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.

*UNSAVE...

COUNTY...

CANCEL

SAVE

SCHEDULE

MORE

```

1 SELECT
2   *
3 FROM
4   `bigquery-public-data.sdo_h_cdc_wonder_natality.county_natality`
5 ORDER BY
6   Births
7 LIMIT
8   10
9
    
```

Query results

SAVE RESULTS
 EXPLORE DATA

Query complete (0.1 sec elapsed, cached)

Job information
 Results
 JSON

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:

```

1 SELECT
2   *
3 FROM
4   bigquery-public-data.sdo_h_cdc_wonder_natality.county_natality
5 ORDER BY
6   Births
7 DESC
8 LIMIT
9   10
    
```

Your results should appear like this:

```

1 SELECT
2   *
3 FROM
4   `bigquery-public-data.sdo_h_cdc_wonder_natality.county_natality`
5 WHERE
6   Year = '2018-01-01'
7 ORDER BY
8   Births
9   DESC
10  LIMIT
11   10

```

Query results

SAVE RESULTS
 EXPLORE DATA

Query complete (0.2 sec elapsed, 168.4 KB processed)

Job information
 Results
 JSON
 Execution details

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:

```

1 SELECT
2   *
3 FROM
4   bigquery-public-data.sdo_h_cdc_wonder_natality.county_natality
5 WHERE
6   Year = '2018-01-01'
7 ORDER BY
8   Births
9   DESC
10  LIMIT
11   10

```

Your results should appear something like this:

*UNSAVE...

COUNTY...

RUN

SAVE

SCHEDULE

MORE

```

1 SELECT
2 *
3 FROM
4 `bigquery-public-data.sdo_h_cdc_wonder_natality.county_natality`
5 WHERE
6   Year = '2018-01-01'
7 ORDER BY
8   Births
9 DESC
10 LIMIT

```

Query results

SAVE RESULTS

EXPLORE DATA

Query complete (0.3 sec elapsed, 168.4 KB processed)

Job information

Results

JSON

Execution details

Row	Year	County_of_Residence	County_of_Residence_FIPS	Births
1	2018-01-01	Los Angeles County, CA	06037	110271
2	2018-01-01	Harris County, TX	48201	67095
3	2018-01-01	Cook County, IL	17031	61797
4	2018-01-01	Unidentified Counties, TX	48999	56295
5	2018-01-01	Maricopa County, AZ	04013	51854
6	2018-01-01	Unidentified Counties, GA	13999	41000
7	2018-01-01	San Diego County, CA	06073	40070
8	2018-01-01	Kings County, NY	36047	38574
9	2018-01-01	Unidentified Counties, VA	51999	38308
10	2018-01-01	Dallas County, TX	48113	38182

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 11th 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:

- 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?

The ORDER BY clause allows you to sort data by specific columns in ascending or descending order, making it easier to identify patterns, rank entries, and structure information based on numerical or categorical values. This organization is crucial for quick insights, especially when looking for top or bottom values in large datasets.

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

Using ORDER BY with WHERE enables targeted sorting by first filtering specific criteria, making analysis more focused and efficient. It allows you to retrieve relevant data points within a particular subset, reducing data clutter and making insights easier to interpret by focusing on specific conditions.

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

A business could ask, "Which products had the highest sales in 2023 within the electronics category?" This can be answered by filtering with WHERE to limit data to 2023 and the electronics category, and using ORDER BY in descending order to list the top-selling products.

✓ 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.