## **Summary:**

## 

## **How to get to the BigQuery console**

In your browser, go to [console.cloud.google.com/bigquery](https://console.cloud.google.com/bigquery).

**Note:** Going to [console.cloud.google.com](https://console.cloud.google.com/) in your browser takes you to the main dashboard for the Google Cloud Platform. To navigate to BigQuery from the dashboard, do the following:

* Click the Navigation menu icon (Hamburger icon) in the banner.
* Scroll down to the **BIG DATA** section.
* Click **BigQuery** and select **SQL workspace**.

Watch the [How to use BigQuery](https://www.coursera.org/learn/data-preparation/lecture/YWn81/how-to-use-bigquery) video for an introduction to each part of the BigQuery SQL workspace.

## **(Optional) Explore a BigQuery public dataset**

You will be exploring a public dataset in an upcoming activity, so you can perform these steps later if you prefer.

* Refer to these [step-by-step instructions](https://cursive.io/shared/242bde9a6-5415-4ce0-bbae-7e875d14d927).

## **(Optional) Upload a CSV file to BigQuery**

These steps are provided so you can work with a dataset on your own at this time. You will upload CSV files to BigQuery later in the program.

* Refer to these [step-by-step instructions](https://cursive.io/shared/2dea0d610-ef6b-4ba8-8e44-d40dfeb0454b).

## **Getting started with other databases (if not using BigQuery)**

It is easier to follow along with the course activities if you use BigQuery, but if you are connecting to and practicing SQL queries on other database platforms instead of BigQuery, here are similar getting started resources:

* [Getting started with MySQL](https://dev.mysql.com/doc/mysql-getting-started/en/): This is a guide to setting up and using MySQL.
* [Getting started with Microsoft SQL Server](https://docs.microsoft.com/en-us/sql/relational-databases/tutorial-getting-started-with-the-database-engine?view=sql-server-ver15): This is a tutorial to get started using SQL Server.
* [Getting started with PostgreSQL](https://www.postgresql.org/docs/10/tutorial-start.html): This is a tutorial to get started using PostgreSQL.
* [Getting started with SQLite](https://www.sqlite.org/quickstart.html): This is a quick start guide for using SQLite.

--If you’re comfortable using queries to answer questions, try creating and running queries to answer any of the questions below:

SELECT \*

FROM bigquery-public-data.london\_bicycles.cycle\_hire

LIMIT 10;

--What are the names of the stations that bike\_id 1710 started from?

SELECT start\_station\_name

FROM `bigquery-public-data.london\_bicycles.cycle\_hire`

WHERE bike\_id = 1710;

--How many bike\_ids have ended at "Moor Street, Soho"?

SELECT count(bike\_id)

FROM `bigquery-public-data.london\_bicycles.cycle\_hire`

WHERE end\_station\_name = 'Moor Street, Soho';

--What is the station\_id for "Canton Street, Poplar"?

SELECT DISTINCT start\_station\_id

FROM `bigquery-public-data.london\_bicycles.cycle\_hire`

WHERE start\_station\_name='Canton Street, Poplar';

--What is the name of the station whose ID is 111?

SELECT DISTINCT start\_station\_name

FROM `bigquery-public-data.london\_bicycles.cycle\_hire`

WHERE start\_station\_id=111;

--How many distinct bike\_ids had trip durations greater than 2400 seconds (or 40 minutes)?

SELECT count(DISTINCT bike\_id)

FROM `bigquery-public-data.london\_bicycles.cycle\_hire`

WHERE duration>2400;

top five baby names for boys in the United States in 2014:

SELECT name,count

FROM `BabyNames.Names2014`

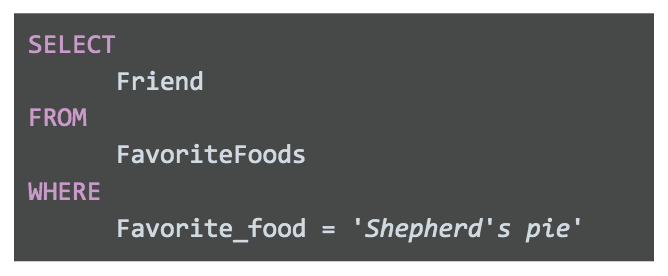
WHERE gender = 'M'

ORDER BY count DESC

LIMIT 5

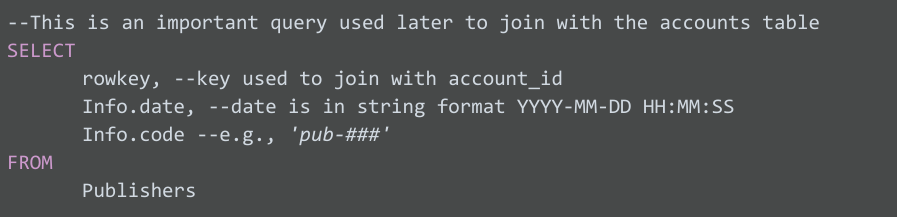
When writing SQL queries,

* capitalization and quotes do not usually matter but using a consistent style can make queries more professional.
* All caps should be used for clause starters and functions, column names should be lowercase, and table names should be in CamelCase.
* Some SQL dialects are case-sensitive, like BigQuery, and may need to be handled differently.
* Single quotes are generally preferred for strings, but double quotes may be necessary when the string contains apostrophes or quotation marks.



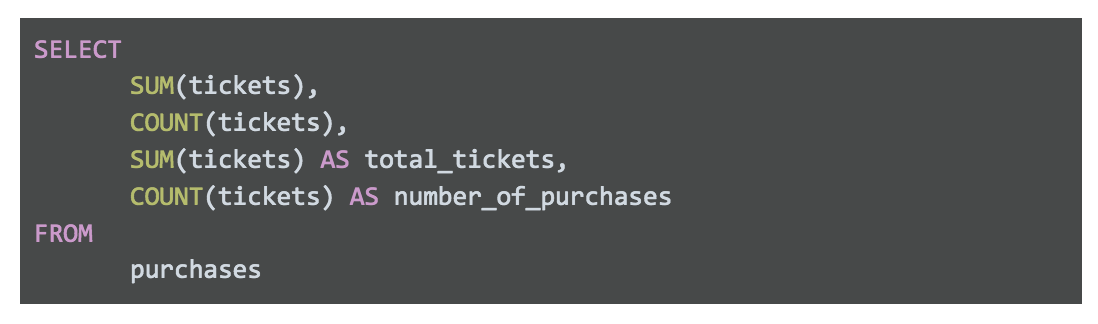
## **Comments as reminders**

As you get more comfortable with SQL, you will be able to read and understand queries at a glance. But it never hurts to have comments in the query to remind yourself of what you are trying to do. And if you share your query, it also helps others understand it.



## **Snake\_case names for columns**

It is important to always make sure that the output of your query has easy-to-understand names. If you create a new column (say from a calculation or from concatenating new fields), the new column will receive a generic default name (e.g., f0). For example:

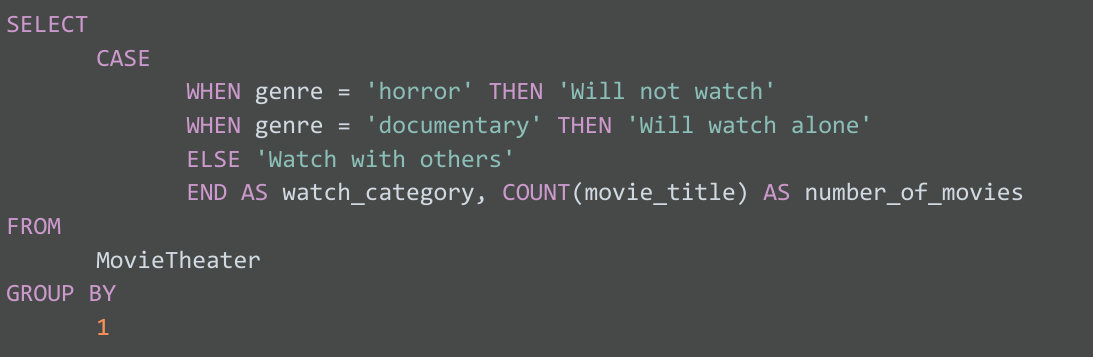


## **CamelCase names for tables**

You can also use CamelCase capitalization when naming your table. CamelCase capitalization means that you capitalize the start of each word, like a two-humped (Bactrian) camel. So the table TicketsByOccasion uses CamelCase capitalization. Please note that the capitalization of the first word in CamelCase is *optional;* camelCase is also used.

## **Indentation**

As a general rule, you want to keep the length of each line in a query <= 100 characters. This makes your queries easy to read. For example, check out this query with a line with >100 characters:



## **Multi-line comments**

