Introduction to SQL: 2nd lesson – Select, From & Where

Now that you know how to access and examine a dataset, you're ready to write your first SQL query! As you'll soon see, SQL queries will help you sort through a massive dataset, to retrieve only the information that you need. We'll begin by using the keywords SELECT, FROM, and WHERE to get data from specific columns based on conditions you specify.

SELECT … FROM clause:

* The most basic SQL query selects a single column from a single table by specifying the column after the word SELECT, and then specify the table after the word FROM.
* For instance, to select the Name column (from the pets table in the pet\_records database in the bigquery-public-data project), the query would appear as follows:

query = “””

SELECT Name

FROM `bigquery-public-data.pet\_records.pets`

“””

Note that when writing a SQL query, the argument we pass to FROM is not single or double quotation marks (‘ or “). It is in backticks (`).

WHERE clause:

* BigQuery datasets are large, so you'll usually want to return only the rows meeting specific conditions. You can do this using the WHERE clause.
* The query below returns the entries from the Name column that are in rows where the Animal column has the text 'Cat'.

query = “””

SELECT Name

FROM `bigquery-public-data.pet\_records.pets`

WHERE Animal = ‘Cat’

“””

More queries:

* If you want multiple columns, you can select them with a comma between the names.

query = “””

SELECT city, country

FROM `bigquery-public-data.openaq.global\_air\_quality`

WHERE country = ‘US’

“””

* You can select all columns with an asterisk like this:

query = “””

SELECT \*

FROM `bigquery-public-data.openaq.global\_air\_quality`

WHERE country = ‘US’

“””

Q&A – Notes on formatting:

The formatting of the SQL query might feel unfamiliar. If you have any questions, you can ask in the comments section at the bottom of this page. Here are answers to two common questions.

* ***Question: What's up with the triple quotation marks (""")?***

Answer: These tell Python that everything inside them is a single string, even though we have line breaks in it. The line breaks aren't necessary, but they make it easier to read your query.

* ***Question: Do you need to capitalize*** SELECT ***and*** FROM***?***

Answer: No, SQL doesn't care about capitalization. However, it's customary to capitalize your SQL commands, and it makes your queries a bit easier to read.