Introduction to SQL: 5th lesson – As & With

With all that you've learned, your SQL queries are getting pretty long, which can make them hard understand (and debug).

You are about to learn how to use AS and WITH to tidy up your queries and make them easier to read.

AS clause:

You learned in an earlier tutorial how to use AS to rename the columns generated by your queries, which is also known as aliasing. This is similar to how Python uses as for aliasing when doing imports like import pandas as pd or import seaborn as sns. To use AS in SQL, insert it right after the column you select. Here's an example of a query without an AS clause:

query = “””

SELECT Animal, COUNT(ID)

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

GROUP BY Animal

“””

-------- -----

Animal f0\_

-------- -----

Rabbit 1

Dog 1

Cat 2

And here's an example of the same query, but with AS.

query = “””

SELECT Animal, COUNT(ID) AS Number

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

GROUP BY Animal

“””

-------- ---------

Animal Number

-------- ---------

Rabbit 1

Dog 1

Cat 2

These queries return the same information, but in the second query the column returned by the COUNT() function will be called Number, rather than the default name of f0\_\_.

WITH … AS clause:

On its own, AS is a convenient way to clean up the data returned by your query. It's even more powerful when combined with WITH in what's called a "common table expression". A common table expression (or CTE) is a temporary table that you return within your query. CTEs are helpful for splitting your queries into readable chunks, and you can write queries against them.

For instance, you might want to use the pets table to ask questions about older animals in particular. So you can start by creating a CTE which only contains information about animals more than five years old like this:

query = “””

WITH Seniors AS

(

SELECT ID, name

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

WHERE Years\_old > 5

)

“””

--- -------

ID Name

--- -------

2 Moon

4 Tom

While this incomplete query above won't return anything, it creates a CTE that we can then refer to (as Seniors) while writing the rest of the query. We can finish the query by pulling the information that we want from the CTE. The complete query below first creates the CTE, and then returns all of the IDs from it.

query = “””

WITH Seniors AS

(

SELECT ID, name

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

WHERE Years\_old > 5

)

SELECT ID

FROM Seniors

“””

---

ID

---

2

4