Introduction to SQL: 4th lesson – Order By

So far, you've learned how to use several SQL clauses. For instance, you know how to use SELECT to pull specific columns from a table, along with WHERE to pull rows that meet specified criteria. You also know how to use aggregate functions like COUNT(), along with GROUP BY to treat multiple rows as a single group.

Now you'll learn how to change the order of your results using the ORDER BY clause, and you'll explore a popular use case by applying ordering to dates.

ORDER BY clause:

ORDER BY is usually the last clause in your query, and it sorts the results returned by the rest of your query. Notice that the rows are not ordered by the ID column. We can quickly remedy this with the query below.

query = “””

SELECT ID, Name, Animal

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

ORDER BY ID

“””

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

ID Name Animal

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

1 Dr. Harris Bonkers Rabbit

2 Moon Dog

3 Ripley Cat

4 Tom Cat

The ORDER BY clause also works for columns containing text, where the results show up in alphabetical order.

query = “””

SELECT ID, Name, Animal

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

ORDER BY Animal

“””

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

ID Name Animal

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

4 Tom Cat

3 Ripley Cat

2 Moon Dog

1 Dr. Harris Bonkers Rabbit

You can reverse the order using the DESC argument (short for 'descending'). The next query sorts the table by the Animal column, where the values that are last in alphabetic order are returned first.

query = “””

SELECT ID, Name, Animal

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

ORDER BY Animal DESC

“””

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

ID Name Animal

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

1 Dr. Harris Bonkers Rabbit

2 Moon Dog

3 Ripley Cat

4 Tom Cat

Dates:

There are two ways that dates can be stored in BigQuery: as a DATE or as a DATETIME. The DATE format has the year first, then the month, and then the day. It looks like this:

YYYY-[M]M-[D]D

1} YYYY: 4-digit year

2} [M]M: 1 or 2 digit month

3} [D]D: 1 or 2 digit day

So, 2019-01-10 is interpreted as January 10, 2019.

The DATETIME format is like the date format ... but with time added at the end.

EXTRACT function:

Often you'll want to look at part of a date, like the year or the day. You can do this with EXTRACT. The query below returns two columns, where column Day contains the day corresponding to each entry the Date column from the pets\_with\_date table:

query = “””

SELECT Name, EXTRACT(DAY from Date) AS Day

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

“””

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

ID Name Day

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

1 Dr. Harris Bonkers 18

2 Moon 16

3 Ripley 7

4 Tom 23

SQL is very smart about dates, and we can ask for information beyond just extracting part of the cell. For example, this query returns one column with just the week in the year (between 1 and 53) for each date in the Date column:

query = “””

SELECT Name, EXTRACT(DAY from Date) AS Day

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

“””

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

ID Name Week

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

1 Dr. Harris Bonkers 15

2 Moon 19

3 Ripley 1

4 Tom 7