### Sorting Queries in SQL

select \*

FROM `my-data-project-35787.movie\_data.movies`

WHERE Genre = 'Comedy'

ORDER BY Release\_Date DESC;

To select all Comedy from the Dataset of movies.csv and sort / order them in Descending order of Release\_Date (by default it is sorted in Ascending order)

select \*

FROM `my-data-project-35787.movie\_data.movies`

WHERE Genre = 'Comedy'

AND Revenue > 300000000

ORDER BY Release\_Date DESC;

To give two conditions for filtering the data: Genre and Revenue

Problem 1: You are working on a project about music and have a table of genres you need to sort. The Genres table contains the columns GenreId and Name.

Write a SQL query to return the name of each genre from this table in alphabetical order.

Answer:

SELECT Name

FROM Genres

ORDER BY Name;

Problem 2: You are working on a project about music and have a table of tracks you need to sort.

The database you use contains a Tracks table. The table contains the following columns: TrackId, Name, AlbumId, MediaTypeId, GenreId, Composer, Milliseconds, Bytes, and UnitPrice.

Write a SQL query to pull all columns from the Tracks table for only tracks with Chris Cornell as the composer. Sort the results in descending order by GenreId.

Answer:

SELECT \*

FROM Tracks

WHERE Composer='Chris Cornell'

ORDER BY GenreId DESC;

SELECT \*

FROM `bigquery-public-data.sdoh\_cdc\_wonder\_natality.county\_natality`

LIMIT 1000;

To Display the first 1000 rows of the country\_natality table inside “sdoh\_cdc\_wonder\_natality” dataset

SELECT \*

FROM `bigquery-public-data.sdoh\_cdc\_wonder\_natality.county\_natality`

ORDER BY Births

LIMIT 10;

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.

SELECT \*

FROM `bigquery-public-data.sdoh\_cdc\_wonder\_natality.county\_natality`

ORDER BY Births

DESC

LIMIT 10;

Modify the query to sort in the other direction, returning the top 10 counties with the highest yearly birth counts between 2016-2018.

SELECT \*

FROM `bigquery-public-data.sdoh\_cdc\_wonder\_natality.county\_natality`

WHERE Year = '2018-01-01'

ORDER BY Births

DESC

LIMIT 10;

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.

Problem 3:

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?

SELECT \*

FROM `bigquery-public-data.sdoh\_cdc\_wonder\_natality.county\_natality`

WHERE Year = '2018-01-01'

ORDER BY Births

DESC;

Answer: Orange County, CA

### SQL Multiple Column ordering

Given the following **People** table:

FirstName | LastName | YearOfBirth

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

Thomas | Alva Edison | 1847

Benjamin | Franklin | 1706

Thomas | More | 1478

Thomas | Jefferson | 1826

If you execute the query below:

| SELECT \*  FROM People  ORDER BY  FirstName DESC,  YearOfBirth ASC; |
| --- |

The result set will look like this:

FirstName | LastName | YearOfBirth

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

Thomas | More | 1478

Thomas | Jefferson | 1826

Thomas | Alva Edison | 1847

Benjamin | Franklin | 1706

The meteorologists who you’re working with have asked you to get the temperature, wind speed, and precipitation for stations La Guardia and JFK, for every day in 2020, in descending order by date, and ascending order by Station ID. Use the following query to request this information:

SELECT

stn,

date,

-- Use the IF function to replace 9999.9 values, which the dataset

-- description explains is the default value when temperature is missing,

-- with NULLs instead.

IF(

temp=9999.9,

NULL,

temp) AS temperature,

-- Use the IF function to replace 999.9 values, which the dataset description explains is the default value when wind speed is missing, with NULLs instead.

IF(

wdsp="999.9",

NULL,

CAST(wdsp AS Float64)) AS wind\_speed,

-- Use the IF function to replace 99.99 values, which the dataset description explains is the default value when precipitation is missing, with NULLs instead.

IF(

prcp=99.99,

0,

prcp) AS precipitation

FROM

`bigquery-public-data.noaa\_gsod.gsod2020`

WHERE

stn="725030" -- La Guardia

OR stn="744860" -- JFK

ORDER BY

date DESC,

stn ASC;

The meteorologists also asked you a couple questions while they were preparing for the nightly news: They want the average temperature in June 2020 and the average wind\_speed in December 2020.

Instead of rewriting similar, but slightly different, queries over and over again, there is an easier approach: Save the results from the original query as a table for future queries.

“To save the query results to a dataset you can select more and save query results to destination dataset”

SELECT

AVG(temperature)

FROM

`my-data-project-35787.Derived\_Dataset.nyc\_weather`

WHERE

date BETWEEN '2020-06-01' AND '2020-06-30';

Now that you have the subset of this data saved in a new table, you can query it more easily. Use the following query to find the average temperature from the meteorologists first question:

You can also use this syntax to find the average wind\_speed or any other information from this subset of data you’re interested in. Try constructing a few more queries to answer the meteorologists’ questions!

The ability to save your results into a new table is a helpful trick when you know you're only interested in a subset of a larger complex dataset that you plan on querying multiple times, such as the weather data for just La Guardia and JFK. This also helps minimize errors during your analysis.