Laboratorio #3
Introducción a SQL - *PAREJAS*

I. Modalidad y fecha de entrega

a) El laboratorio debe hacerse en parejas durante el período de clase asignado
b) Debe ser enviado antes de la fecha límite de entrega: Miércoles 27 de enero a las 21:40
c) Luego de la fecha y hora límites se restarán 10 puntos por cada hora de atraso en la entrega

II. Objetivo y descripción de la actividad

El objetivo de la actividad es que el estudiante profundice y practique sobre los conceptos de *queries* sobre múltiples tablas utilizando JOINs y la aplicación de agregación y las funciones de agregación. El estudiante debe conocer la sintaxis de SELECT introduciendo los conceptos de agregación (GROUP BY) y filtros sobre datos agregados (HAVING).

Introducciones generales y observaciones

Para completar este laboratorio deberá tener instalado localmente un motor de bases de datos PostgreSQL [1], así como un cliente por medio del cual ejecutar *queries* [2].

Se deberá entregar un documento PDF elaborado en Word que muestre la evidencia de cada instrucción ejecutada y su resultado. No se requiere mostrar todo el resultado de cada instrucción, pero sí lo suficiente para evidenciar que la instrucción se ejecutó correctamente.

Ejercicio 1: Uso de JOINs

Descargue y descomprima la base de datos **flights.backup** de a partir de este enlace: https://drive.google.com/file/d/1Y92GF-79SkJF-3MNOsE3fTjkthilWe6s/view?usp=sharing

Investigue el esquema de las tablas de la base de datos que contiene información acerca de los retrasos en vuelos ocurridos en Estados Unidos registrada por el *Bureau of Transportation Statistics*. Puede encontrar más información acerca de los campos en: https://www.transtats.bts.gov/DL_SelectFields.asp?Table_ID=236

A partir de esta base de datos, construya los *queries* necesarios para responder a las siguientes preguntas:

Pregunta 1: ¿Cuáles es el estado del que salió el vuelo con el mayor retraso en el tiempo de salida registrado? (10 puntos)

Iniciaremos respondiendo a esta pregunta en dos partes. Para comenzar, escriba un query para determinar el aeropuerto del que salió el vuelo con el mayor retraso de salida.

En su query muestre el aeropuerto de origen y el retraso en el tiempo de salida.

Dado que la base de datos contiene vuelos que no tuvieron su retraso en el tiempo de salida, debe aplicar un filtro a su consulta para no considerar aquellos vuelos cuyo `DepDelay` es `NULL`.

Su consulta no debe hacer uso de la función `MAX`. Investigue y utilice la clausula `LIMIT` para evitar cargar todos los registros de la tabla `ontime`.

Respuesta: El vuelo con mayor retraso de salida salió del aeropuerto `CLT` y tuvo un retraso de `2467` minutos.

A continuación determinaremos en la tabla `airports` a qué estado pertenece ese aeropuerto.

Escriba una consulta que le indique a qué estado pertenece el aeropuerto cuyo `iata` es igual al código de aeropuerto determinado anteriormente.

Su consulta debe mostrar el código `iata`, el nombre del aeropuerto, el estado y el país del aeropuerto.

Respuesta: El estado al que pertenece el aeropuerto es `NC`, Carolina del Norte.

Pregunta 2: Responda nuevamente a la pregunta 1 utilizando un solo query que que consulte simultáneamente ambas tablas dentro de la clausula `FROM` (5 puntos)

Su query debe mostrar el tiempo de retraso de salida, el aeropuerto de origen, el código iata del aeropuerto, el nombre del aeropuerto, la ciudad del aeropuerto y el estado del aeropuerto.

Pregunta 3: Responda finalmente a la pregunta 1 utilizando un solo query que haga uso de la sintaxis JOIN (5 puntos)

En la práctica suele ser más común la construcción `JOIN` para unir información de dos o más tablas.

Puede repasar la sintaxis para `JOIN`s en http://www.sqlitetutorial.net/sqlite-inner-join/

Pregunta 4: ¿Cuántos vuelos tuvieron un retraso de salida de más de 120 minutos saliendo del el estado de NY? (10 puntos)

Responda a esta pregunta utilizando un solo query.

Respuesta: Un total de 8,878 vuelos tuvieron un retraso de más de 120 minutos saliendo dle estado de NY

Ejercicio 2: Queries, JOINs y agregaciones

Pregunta 1: ¿Cuáles son los vuelos más largo y más corto en distancia registrados en la base de datos? (5 puntos)

Responda a esta pregunta en un solo query:

Respuesta: La distancia más larga recorrida es de 4,962 millas, mientras que la más corta es de 11 millas.

Pregunta 2: ¿Qué aeropuertos no han tenido vuelos de salida? (5 puntos)

Para esta pregunta debe hacer uso de su conocimiento sobre `OUTER JOIN`s y valores `NULL`.

Su respuesta debe mostrar el código IATA, nombre y estado del aeropuerto.

Hint: En la base de datos existen 3007 aeropuertos para los que no hay vuelos de salida

Pregunta 3: ¿Cuál es la temporada alta?

Escriba un query que muestre cuáles son los dos meses con más vuelos en la base de datos

Pregunta 4: ¿Cuáles son los peores días para viajar? Parte 1 (10 puntos)

Prepare un query que muestre el promedio de retraso en el tiempo de llegada (arrival) registrado para cada uno de los días de la semana utilizando funciones de agregación.

El esquema de la relación obtenida debe ser `(dia: int, retraso_promedio: float)` y debe mostrar los días ordenados por el peor día para viajar al mejor.

Respuesta:
5 10.675214410055972
7 9.294087238339747
4 8.246549198623555
1 8.030254747872275
2 7.29232332960935
3 6.379804071503802
6 5.682499626837823

Pregunta 5: ¿Cuáles son los peores días para viajar? Parte 2 (10 puntos)

Dado que el resultado anterior no necesariamente es significativo para un usuario final, es necesario unir la información sobre el día del vuelo con la tabla `weekdays` contenida en esta nueva versión de la base de datos.

Puede investigar más sobre esta tabla utilizando un Jupyter notebook paralelo a manera de _scratchpad_.

A continuación trabaje el query necesario para responder a la pregunta sobre los peores días para viajar de tal forma que su respuesta incluya el nombre del día y el promedio de tiempo de retraso, `(dia_nombre: VARCHAR, retraso_promedio: DOUBLE)`:

Respuesta:
Friday 10.675214410055972
Sunday 9.294087238339747
Thursday 8.246549198623555
Monday 8.030254747872275
Tuesday 7.29232332960935
Wednesday 6.379804071503802
Saturday 5.682499626837823

Pregunta 6: ¿Cuál es el vuelo más común? (15 puntos)

Considerando como vuelos repetidos aquellos que van de un mismo aeropuerto origen a un mismo aeropuerto destino, escriba un query que retorne los cinco vuelos más frecuentes registrados en la base de datos.

Por ejemplo, hay 11,224 vuelos registrados en la base de datos cuyo origen es `SAN` y cuyo destino es `LAX`.

Pregunta 7: ¿Cuál es el estado con más vuelos internos? (15 puntos)

Escriba un query que indique cuál es el estado con más vuelos internos.

Un vuelo interno es aquel cuyo aeropuerto origen y aeropuerto destino se encuentran en el mismo estado.

Hint: Necesitará hacer dos JOINs a la tabla `airport` con condiciones diferentes. Inicie preparando un query que le muestre 10 ejemplos de vuelos internos, para verificar que la lógica de su query devuelve tuplas válidas.

Respuesta: El estado con más vuelos internos es California (CA) con 302,086 vuelos internos

Pregunta 8: ¿Qué aerolíneas se retrasan más de 10 minutos regularmente? (10 puntos)

Escriba un query que retorne las aerolíneas cuyos vuelos se retrasan en promedio 10 minutos o más, ordenando de la que más se retrasa a la que menos.

El esquema de la relación obtenida debe ser de la forma `(airline_name: varchar, average_delay: float)`.

Hint: el uso de la cláusula `HAVING` puede resultar conveniente. Para la tabla de vuelos, utilice el atributo `UniqueCarrier` para asociar con el código de la aerolinea.

Respuesta:
American Airlines Inc. 12.202853434950445
PSA Airlines Inc. 11.404110178283158
Mesa Airlines Inc. 11.322566979170753
United Air Lines Inc. 11.001550560048052
JetBlue Airways 10.859381613638567
Continental Air Lines Inc. 10.809820575966226
ExpressJet Airlines Inc. (1) 10.320298523403915
ExpressJet Airlines Inc. 10.00033146217589

III. Temas a reforzar

- DML: SELECT, FROM, WHERE, JOIN, GROUP BY, Funciones de agregación

IV. Documentos a entregar

1. Un documento PDF por pareja, correctamente identificado que contenga:
    a. Pantallazos de cada instrucción SQL ejecutada y su resultado

V. Evaluación

- Ejercicio #1: 20 puntos
- Ejercicio #2: 80pts **Total: 100 puntos**