Laboratorio #4
SQL Parte 3 - *INDIVIDUAL*

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: martes 9 de agosto a las 23:59
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 investigue y aplique los conceptos de restricciones (*constraints*) pertinentes a la creación de una base de datos de acuerdo con las reglas del dominio aplicables.

Adicionalmente se busca que el estudiante profundice y practique sobre los conceptos de *queries* utilizando el conjunto completo de características de consultas SQL, incluyendo JOINs, *subqueries* y agregaciones, para responder a consultas avanzadas sobre bases de datos relacionales en PostgreSQL.

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]. Adicionalmente deberá tener cargada la base de datos de vuelos utilizada durante el Laboratorio #3.

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.

Se deberá entregar también un *dump* de la base de datos creada para el ejercicio 1.

Ejercicio 1: DDLs y constraints (30)

En esta parte del laboratorio investigaremos de qué forma podemos utilizar `FOREIGN KEY`s y `UNIQUE`s como un tipo de *constraint* dentro del esquema de una base de datos, usando como dominio un sistema de registro académico universitario simplificado.

1.1 Preparación

Inicie creando una base de datos `lab04` en su cliente de SQL. A continuación escriba y ejecute los queries necesarios para construir tablas con los siguientes esquemas:

`estudiante(dpi: VARCHAR(30), fechaNacimiento: DATE, nombres: VARCHAR(50), apellidos: VARCHAR(50))`

(Observe que en esta tabla, el campo `dpi` debe estar especificado como llave primaria).

`curso(codigo: VARCHAR(30), nombre: VARCHAR(50))`

En esta tabla, el campo `codigo` debe estar especificado como llave primaria.

Seguidamente escriba los `INSERT`s necesarios para registrarse a usted y a otros estudiantes y a este curso dentro de las tablas.

Recuerde que el código del presente curso es `CC3088`.

Note que, dado que el campo `estudiante.dpi` es llave, la base de datos le impide insertarse a usted dos veces.

1.2 Creación de constraints e integridad referencial (10 puntos)

A continuación, utilizando `FOREIGN KEY`s y `UNIQUE`s defina el esquema de una tabla `asignacion` que contenga el dpi de un estudiante, el código de un curso y la fecha de la asignación, y **que no permita ingresar tuplas con un estudiante o curso inexistente**.

Adicionalmente no se debe permitir tener dos veces al mismo estudiante en el mismo curso con la misma fecha de asignación. Esto puede lograrse con `PRIMARY KEY`s, pero en este ejercicio debe lograrlo por medio de constraints `UNIQUE`.

1.3 Verificación de constraints (5 puntos)

A fin de dejar evidencia que las políticas implementadas se cumplen, construya y ejecute queries para:

1.3.1. Su asignación y la de otro estudiante a este curso el día de hoy:

1.3.2. El ingreso de una asignación con código de estudiante estudiante DPI `2413 94996 0108` a este curso (este estudiante no existe en la tabla de estudiantes, por lo que la inserción debe resultar en un error)

1.3.2. Su asignación a un curso con código `FF2017-435` (este curso no existe en la tabla de cursos, por lo que la inserción debe resultar en un error)

Su entrega debe dejar evidencia de que el primer query fue aceptado y los otros dos rechazados.

1.4 Comportamientos derivados (15)

De acuerdo con las reglas de negocio de la universidad de este ejemplo, si un estudiante se retira debe eliminarse de la tabla de estudiantes y además se deben eliminar todas sus asignaciones. Si ahora mismo trata de eliminarse de la tabla de estudiantes verificará que la base de datos se lo impide, ya que hay una asignación que hace referencia a su tupla de estudiante.

Investigue sobre las operaciones CASCADE en las llaves foráneas y modifique su esquema para que la eliminación de un estudiante resulte en la eliminación automática de sus asignaciones.

Hint: un cambio en un *constraint* FOREIGN KEY en PostgreSQL requiere primero eliminar del *constraint* original y luego crearlo, como se muestra aquí.

Su entrega debe dejar evidencia de que la eliminación de un estudiante resulta en la eliminación de todas sus asignaciones.

Ejercicio 2: Consultas avanzadas en SQL

Pregunta 2.1: ¿Qué aerolíneas de las que vuelan desde SFO se retrasan más en llegar? (10 puntos)

Responderemos a esta pregunta en dos partes:

2.1.1

Para comenzar, escriba un query que retorne el listado de aerolíneas que realizaron al menos un vuelo desde el aeropuerto `SFO`.

Su respuesta debe incluir únicamente el código de la aerolínea.

Respuesta: Las 15 aerolíneas que realizaron al menos un vuelo desde el aeropuerto SFO son F9, OO, US, AS, CO, UA, B6, MQ, HA, AA, FL, XE, DL, WN y NW

2.1.2

A continuación utilice el operador `IN` dentro de una cláusula `WHERE` para identificar los vuelos cuya aerolínea se encuentra dentro del resultado del query anterior 2.1.2.

Finalmente compute el promedio de retraso de todas las aerolíneas cuyo nombre se encuentre dentro del resultado del query 2.1.2, agrupando por aerolínea.

Su respuesta debe incluir el código de la aerolínea y el promedio en el tiempo de llegada.

Hint: Utilice un *subquery* como parte de la cláusula `WHERE`.

Respuesta:

AA: 12.6071940357
…
…
…

Pregunta 2.2: ¿Qué aerolíneas vuelan de California a Nueva York? (10 puntos)

Su respuesta debe mostrar el nombre de la aerolínea.

Calcule su respuesta utilizando *subqueries*.

Hint: Deben ser cuatro aerolíneas

Respuesta:
American Airlines Inc.
JetBlue Airways
Delta Air Lines Inc.
United Air Lines Inc.

Pregunta 2.3: ¿Qué aerolíneas han realizado vuelos del estado de California al de Nueva York, y también de California a Nuevo México? (10 puntos)

Su respuesta debe mostrar el nombre de las aerolíneas.

De su respuesta en términos de subqueries.

Respuesta: United Air Lines Inc.

Pregunta 2.4. ¡Quiero maximizar mis millas! (15 puntos)

Suponga que en un futuro no muy lejano le gustaría visitar tanto Portland (PDX) como Eugene (EUG), saliendo desde SFO. Con el objetivo de maximizar mis millas de viajero frecuente, me gustaría utilizar la misma aerolínea para ambos vuelos.

Escriba un *query* que muestre los nombres de las aerolíneas que realizaron tanto vuelos SFO -> PDX como vuelos SFO -> EUG. Asegúrese de mostrar el nombre de las aerolíneas sin duplicados.

Respuesta: United Air Lines Inc., SkyWest Airlines Inc.

Pregunta 2.5: ¿Existen aerolíneas que hagan únicamente vuelos internos? (10 puntos)

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

De su respuesta en términos de `EXCEPT`.

Respuesta: En la base de datos no hay ninguna aerolínea que haga únicamente vuelos internos.

Pregunta 2.6: ¿Qué proporción de aerolíneas se retrasan en llegar un promedio 10 minutos o más con respecto al total de aerolíneas que han operado vuelos? (15 puntos)

En el laboratorio 3 usted desarrolló un query para mostrar qué aerolíneas se retrasan en llegar un promedio 10 minutos o más.

Escriba ahora un query que retorne la proporción de las aerolíneas que se retrasan en promedio 10 minutos o más, con respecto a la cantidad de aerolíneas que operaron vuelos.

Recuerde no quemar (*hardcode*) la cantidad de aerolíneas disponibles, sino obtener la cantidad a partir de un query.

Nota: la función `COUNT(*)` en retorna valores de tipo *integer*, por lo que deberá investigar e incluir en su query una conversión (*casting*) de valores para obtener una proporción correcta en *float*.

Respuesta: 40%

III. Temas a reforzar

- DML: JOIN, Agregaciones y suqueries

IV. Documentos a entregar

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

V. Evaluación

- Ejercicio #1: 35 puntos
- Ejercicio #2: 65 puntos

Total: 100 puntos