



## Laboratorio #8 Índices y triggers - 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: 11 de octubre de 2022 a las 23:59
- c) Luego de la fecha límite se restarán 10 puntos por cada hora de atraso en la entrega

### II. Descripción de la actividad

Realice los siguientes ejercicios dejando evidencia de su resolución con al menos un pantallazo (*screenshot*) por cada subactividad. Algunas subactividades requerirán más de un pantallazo.

#### Ejercicio 1: selectividad de los atributos

Una empresa fabrica básculas inteligentes que le permiten a sus usuarios darle un seguimiento completo a su peso corporal. La empresa almacena la información capturada en dos tablas: usuario y detalle.

En la tabla de **usuario** se almacena un ID del usuario, fecha de inicio de uso del servicio, número de serie de la báscula, altura (en cm), peso inicial al momento del uso del servicio (en libras), fecha de nacimiento, país de nacimiento. En la tabla de **detalle** se almacena el ID del usuario, la fecha y hora de cada pesaje, el valor en libras del peso registrado.

Para cada una de las tablas ordene los campos según su selectividad (es decir, qué tan diferentes son los valores de una columna) de mayor a menor.

USUARIO	DETALLE

¿Qué sucedería con el rendimiento de un índice sobre los campos ID usuario y peso de la tabla **detalle** si eventualmente todos los usuarios llegaran y se mantuvieran en su peso ideal por tiempo indefinido?

#### Ejercicio 2: Declaración de índices

Utilizando la base de datos de vuelos utilizada en el Laboratorio #3 realice las siguientes actividades:

1.1 Investigue cómo enumerar los índices existentes en la base de datos y muéstrellos a continuación.

1.2 Defina un índice sobre los campos `flight.origin` y `flight.dest`

1.3 Elabore un query que muestre los códigos de aerolíneas que realizan vuelos que salen del aeropuerto **JFK** (sin duplicados), e indique cuánto tiempo toma la ejecución de su query.

1.4 Utilizando EXPLAIN muestre qué índice se utiliza para optimizar la ejecución de esa consulta

1.5 Elimine el índice y ejecute nuevamente el query elaborado en la actividad 1.2, anotando cuánto tiempo toma la ejecución del query sin el índice.

1.6 Cree nuevamente el índice eliminado, ya que le servirá para el siguiente ejercicio.

### Ejercicio 3: Selección de índices

El siguiente query reporta el promedio de retraso en el tiempo de llegada por cada aerolínea que vuela hacia el aeropuerto **JFK**:

```
SELECT a.airlinecode, AVG(f.arrdelay) AS arr_delay_average
FROM flight f
LEFT JOIN airline a ON f.reporting_airline = a.airlinecode
WHERE a.airlinecode IN (
  SELECT DISTINCT f.reporting_airline
  FROM flight f
  WHERE f.dest = 'JFK'
)
GROUP BY a.airlinecode
ORDER BY arr_delay_average DESC;
```

3.1 Ejecute el query e indique cuánto tiempo toma su ejecución.

3.2 Ejecute un EXPLAIN sobre este *query* e identifique el costo relativo del query completo reportado por PostgreSQL.

3.3 ¿Cuántos escaneos secuenciales reporta PostgreSQL en el *execution plan* de este *query*? ¿Cuáles son?

3.4 Conociendo los índices existentes en la base de datos y el *execution plan* propuesto por PostgreSQL cree el (los) índice(s) necesario(s) para mejorar el tiempo de respuesta del query, dejando evidencia de la instrucción de creación de cada índice.

3.5. Indique el nuevo costo relativo del query completo reportado por PostgreSQL y el nuevo tiempo de ejecución.

*Hint:* La resolución del catedrático eliminó todos los escaneos secuenciales del plan de ejecución, y redujo el tiempo de respuesta de 3.63s a 3.17 en su equipo local (es decir una reducción del 40%) creando dos índices.

### Ejercicio 4: consultas SARGABLES

Las consultas mostradas a continuación poseen índices pero no son utilizadas porque no son SARGABLES. Realice los cambios necesarios en cada una de las consultas para que estas utilicen los índices creados (sin utilizar la creación de índices sobre funciones) y muestre el código SQL de las nuevas consultas.

```
-- Esta consulta devuelve los estudiantes asignados por curso en el año 2022,
-- y posee un índice en la fecha de asignación, pero no es utilizado
```

```
SELECT id_curso, COUNT(*)
FROM asignacion
WHERE EXTRACT(YEAR, fecha_asignacion) = 2022
GROUP BY id_curso
```

-- Esta consulta devuelve los cursos que los estudiantes con apellido "Alvarado"  
-- se asignaron, y posee un índice en los apellidos, pero no es utilizado

```
SELECT id_curso, nombre_curso
FROM asignacion
      INNER JOIN estudiante on asignacion.carnet = estudiante.carnet
WHERE SUBSTRING(apellidos, 0, 8) = 'Alvarado'
```

-- Esta consulta devuelve las asignaciones cuyo tiempo total fueron más de 300 segundos  
-- La columna tiempo\_total almacena minutos y posee un índice que no es utilizado

```
SELECT id_asignacion, tiempo_total
FROM asignacion
WHERE tiempo_total*60 >= 300
```

### Ejercicio 5: Creación de triggers

Cree las siguientes tablas

```
ASIGNACION(carnet VARCHAR(12), codigo_curso VARCHAR(6), fecha_asignacion DATE)
ASIGNACION_HISTORICA(carnet VARCHAR(12), codigo_curso VARCHAR(6), fecha_asignacion
DATE, fecha_eliminacion DATE)
```

3.1 Cree un trigger sobre la tabla asignación que impida que una materia tenga más de 5 asignaciones para un mismo semestre.

3.2 Cree un trigger que registre en la tabla histórica de asignaciones las asignaciones que son eliminadas.

Para ambos triggers debe demostrar su funcionamiento.

### III. Temas a reforzar

- Declaración de índices
- Selección de campos para índices y revisión de ejecución de *query plans*
- Triggers

### IV. Documentos a entregar

- Documento PDF con la evidencia de la resolución de cada uno de los ejercicios (*screenshots*)

### V. Evaluación

- Ejercicio 1: 5 puntos
- Ejercicio 2: 15 puntos
- Ejercicio 3: 40 puntos
- Ejercicio 4: 10 puntos
- Ejercicio 5: 30 puntos
- **Total: 100 puntos**