



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN
IIC2413 - BASES DE DATOS

Interrogación 2 PAUTA

Fecha: 29 de mayo de 2024

1º semestre 2024 - Profesores: N. Alvarado - A. Reyes - E. Bustos - C.Álvarez

INSTRUCCIONES

- I Al finalizar la Prueba debe digitalizarla y subirla a Canvas correctamente. No seguir esta instrucción los expone a un descuento de hasta 5 décimas.
- II Cada hoja debe contener su Nombre Completo y Número de Lista en la esquina superior izquierda.
- III No separe las hojas.
- IV Debe firmar la lista de asistencia.
- V Duración 2 horas.

Pregunta 1: SQL y Dependencias

- I **(3pt)** El Estado ha decidido integrar todas sus operaciones portuarias a través de la empresa Cochrane Ports. Fuiste contratado por esta empresa para liderar el diseño de la nueva base de datos unificada de Cochrane Ports. A la continuación se muestra el esquema:
 - Región(rid INT PRIMARY KEY, nombre VARCHAR)
 - Ciudad(cid INT PRIMARY KEY, nombre VARCHAR, rid INT, FOREIGN KEY (rid) REFERENCES Región(rid))
 - Puerto(pid INT PRIMARY KEY, nombre VARCHAR, cid INT, FOREIGN KEY (cid) REFERENCES Ciudad(cid))
 - Instalación(iid INT PRIMARY KEY, nombre VARCHAR, capacidad INT, pid INT, FOREIGN KEY (pid) REFERENCES Puerto(pid), jefe VARCHAR, FOREIGN

KEY (jefe) REFERENCES Trabajador(rut))

- Muelle(mid INT PRIMARY KEY, FOREIGN KEY (mid) REFERENCES Instalación(iid), tipo_barco VARCHAR)
- Astillero(aid INT PRIMARY KEY, FOREIGN KEY (aid) REFERENCES Instalación(iid), largo max FLOAT, ancho max FLOAT, peso max FLOAT)
- Trabajador(rut VARCHAR PRIMARY KEY, nombre VARCHAR, edad INT, sexo VARCHAR, iid INT FOREIGN KEY (iid) REFERENCES Instalación(iid))

Un Puerto tiene Instalaciones y éstas pueden ser Muelles o un Astilleros.

Escriba las siguientes consultas SQL:

- a) Muestre la edad promedio de los trabajadores de cada Puerto.
- b) Muestre todos los jefes de las instalaciones del puerto con nombre 'Mejillones'.

II (3pt) Justifique por qué el esquema se encuentra en BCNF o 3NF.

Solución: Importante: Esta es una solución propuesta, recordar considerar cualquier suposición hecha por el estudiante al momento de contestar esta pregunta. Deben ser flexibles con la corrección considerando lo anterior y además la reduncia en diversas consultas SQL.

I Consultas SQL

- Muestre la edad promedio de los trabajadores de cada Puerto.

```
SELECT Puerto.pid, AVG(Trabajador.edad)
FROM Puerto, Instalación, Trabajador
WHERE Puerto.pid = Instalación.pid AND Instalación.iid = Trabajador.iid
GROUP BY Puerto.pid;
```

- Muestre todos los jefes de las instalaciones del puerto con nombre 'Mejillones'.

```
SELECT Trabajador.rut, Trabajador.nombre
FROM Puerto, Instalación, Trabajador
WHERE Puerto.pid = Instalación.pid
AND Instalación.jefe = Trabajador.rut
AND Puerto.nombre = 'Mejillones';
```

II Vemos que para cada tabla, la única dependencia no trivial es su llave primaria, y en cada caso la llave primaria implica todos los demás atributos, por lo que podemos decir que esto está en BCNF. Por otro lado, notemos que en términos de modelación, decidimos usar el rut de la persona como identificador del trabajador, pero en términos de implementación, probablemente vamos a querer agregar un id de trabajador (digamos tid) que sea un entero para hacer las consultas más eficientes. En ese caso, al tener el tid en la tabla de trabajador junto con el rut, la tabla ya no estaría en BCNF.

La razón de esto último es que *tid* es la llave primaria, pero tenemos la dependencia **rut** \rightarrow nombre, edad, sexo, iid

□

Pregunta 2: Lógica en la BD

Usted ha sido contratado para asesorar a la empresa *Cliente Feliz* en la adecuación de su base de datos para que pueda participar en el próximo Black Friday. *Cliente Feliz* tiene un negocio de arriendo de carritos de comida, teniendo los siguientes tipos de carrito: completos, hamburguesas, churros y pop corn.

El cliente puede contratar un carrito de comida para sus eventos, a través de la página web de *Cliente Feliz*. Si el cliente quiere contratar 2 carritos, debe realizar 2 transacciones independientes. Al seleccionar una fecha, es posible ver cuantos carritos por tipo están disponibles.

El esquema de la Base de Datos es el siguiente:

- Tipo_Carro (*id_tipo* INT PRIMARY KEY, *nombre* VARCHAR, *Descripcion* VARCHAR)
- Arriendo_Carro (*id_carro* INT PRIMARY KEY, *id_tipo* INT, FOREIGN KEY (*id_tipo*) REFERENCES Tipo_Carro(*id_tipo*))
- Evento (*id_evento* INT PRIMARY KEY, *id_carro* INT, *id_cliente* INT, *fecha_evento* DATE, *direccion_evento* VARCHAR, FOREIGN KEY *id_carro* REFERENCES Arriendo_Carro(*id_carro*), FOREIGN KEY *id_cliente* FOREIGN KEY Cliente(*id_cliente*))
- Cliente (*id_cliente* INT PRIMARY KEY, *nombre_cliente* VARCHAR, *apellido_cliente* VARCHAR, *RUT* VARCHAR, *email_cliente* VARCHAR, *telefono_cliente* INT)
- Pago (*id_pago* INT PRIMARY KEY, *id_evento* INT, *fecha_pago* DATE, *monto* INT, *pago_aceptado* BOOLEAN, FOREIGN KEY *id_evento* REFERENCES Evento(*id_evento*))

- I (3pt) Para que en la página web sea posible visualizar la cantidad de carros disponibles por tipo, ¿Qué tipos de lógica a nivel de Base de Datos usaría? Justifique su respuesta
- II (3pt) Debido a la alta demanda esperada, se ha decidido que los carros se reserven solo por 5 minutos, mientras realiza el proceso de pago. Para ello, se agregarán los campos *fecha_reserva*, *hora_reserva* y *estado* a la tabla de Evento. *estado* especifica que es una reserva en proceso de pago (0), confirmada (1) o expirada (-1). Existe un proceso automático y desatendido que actualizará la tabla Evento con todas las compras que no se completaron en el lapso de 5 minutos, dejándolas en estado expirada (-1). Cree un Trigger para actualizar la tabla Evento con el estado confirmada (1) cuando un registro es insertado en la tabla de Pago y su pago ha sido aceptado (*pago_aceptado* TRUE).

El esquema de la tabla *evento* actualizado es el siguiente:

```
Evento (id_evento INT PRIMARY KEY,
id_carro INT, FOREIGN KEY id_carro REFERENCES Arriendo_carro(id_carro),
id_cliente INT, FOREIGN KEY id_cliente REFERENCES Cliente(id_cliente),
fecha_evento DATE,
direccion_evento VARCHAR,
estado INT,
fecha_reserva DATE,
hora_reserva TIME )
```

Solución: I Según el contenido visto en clase sobre **lógica en la bd**, tenemos 3 alternativas: vistas, procedimientos almacenados y triggers. Cómo en este caso se refiere a obtener información de la BD tenemos 2 alternativas posibles:

- Vista: podemos crear una vista que sea una consulta a las tablas del esquema y entregue como información la cantidad de carros disponibles por tipo.
- Procedimiento almacenado: podemos crear un procedimiento almacenado que contenga el mismo código de la consulta de la vista, entregado como resultado la cantidad de carros disponibles por tipo. El procedimiento podría no tener parámetros o recibir como parámetro el tipo de carro y entregar la cantidad disponible para ese tipo.

II El código del trigger es el siguiente:

```
CREATE TRIGGER update_evento
AFTER CREATE ON Pago
FOR EACH ROW
BEGIN
    UPDATE Evento
    SET estado = 1
    WHERE NEW.id_evento = Evento.id_evento
    AND NEW.pago_aceptado IS TRUE
END
```

□

Pregunta 3: Transacciones y Schedules

I (3pt) A qué tipo de conflicto de transacciones corresponden (Lectura sucia, Lectura no repetible o Actualización perdida) las siguientes definiciones:

- a) Ocurren cuando una transacción accede a datos que han sido modificados por otra transacción que aún no se ha terminado.
R: Lectura Sucia
- b) Ocurren cuando una fila es leída dos veces y el valor cambia entre ambas lecturas. Esto se debe a que otra transacción actualiza o modifica la fila entre las dos lecturas individuales.
R: Lectura no repetible
- c) Ocurren cuando dos transacciones que intentan actualizar la misma fila son procesadas en un tiempo que permite que una de las actualizaciones sobrescriba a la otra.
R: Actualización Perdida

II (3pt) Dado el schedule serial, **Schedule 1**, indique si el **Schedule 2** y/o el **Schedule 3** son equivalentes al Schedule 1. Justifique su respuesta. R: Solo 2, En S3, T2 el W(B) produce que el R(B) de la T1 entregue otro valor

| Schedule 1 | | Schedule 2 | | Schedule 3 | |
|-------------------|----------|-------------------|----------|-------------------|----------|
| T1 | T2 | T1 | T2 | T1 | T2 |
| Read(A) | | Read(A) | | Read(A) | |
| Write(A) | | Write(A) | | Write(A) | |
| Read(B) | | | Read(A) | | Read(A) |
| Write(B) | | | Write(A) | | Write(A) |
| | Read(A) | Read(B) | | | Read(B) |
| | Write(A) | Write(B) | | | Write(B) |
| | Read(B) | | Read(B) | Read(B) | |
| | Write(B) | | Write(B) | Write(B) | |

Pregunta 4: Arquitectura de Sistemas y Recuperación de fallas

- I **(3pt)** Explique cómo se gestionan las fallas de transacciones en una base de datos utilizando el Undo Logging. Incluya en su respuesta las reglas de Undo Logging y un ejemplo de recuperación de una transacción abortada.
- II **(2pt)** Describa el proceso de recuperación con Redo Logging. ¿Cómo se utilizan los checkpoints en este método?
- III **(1pt)** ¿Cuál es el beneficio de usar Nonquiescent Checkpoints en lugar de checkpoints tradicionales?

Solución: I La gestión de fallas de transacciones en una base de datos utilizando *Undo Logging* se basa en registrar en un log las operaciones de una transacción antes de que se realicen. Esto permite revertir los cambios de una transacción que no se completa correctamente, garantizando la consistencia y la integridad de la base de datos.

- Antes de modificar un valor en la base de datos, se debe escribir en el log el valor antiguo.
- Después de escribir en el log, se debe escribir el nuevo valor en la base de datos.
- Antes de terminar la transacción (commit), se debe escribir un registro de commit en el log.

- II El proceso de recuperación con Redo Logging se basa en la idea de registrar en un log las operaciones de una transacción después de que se realicen, de manera que se puedan volver a aplicar en caso de fallo para asegurar la consistencia de la base de datos.

Los checkpoints son puntos de verificación que se escriben periódicamente en el log para indicar un estado consistente de la base de datos. Los checkpoints ayudan a reducir el tiempo de recuperación, ya que permiten limitar la cantidad de log que debe ser leído y procesado en caso de fallo.

- III Minimización del tiempo de inactividad, mayor concurrencia reducción del impacto en el rendimiento y recuperación más eficiente.

□