*Trabajo práctico unidad 8 Tratamiento de Datos*

***BASES DE DATOS***



*Índice*

*Introducción—---------------------------------------------------página 03*

*Cuestiones sobre transacciones —---------------página 04,05,06*

*Operaciones SQL —------------------------------------------página 07*

*Conclusión —---------------------------------------------------página 08*

*Introducción*

En este trabajo ampliamos nuestro abanico de consultas a inserciones y modificaciones con consultas dentro. También ampliamos conocimiento con un par de preguntas de teoría.

*Cuestiones sobre transacciones*

*1. ¿Qué son las propiedades ACID?*

Son un conjunto de características que se utilizan para describir la fiabilidad y consistencia de una transacción en sistemas de gestión de bases de datos relacionales.

• Atomicidad : Una transacción es indivisible, o se ejecutan todas la sentencias o no se ejecuta ninguna.

• Consistencia : Después de una transacción la base de datos estará en un estado válido y consistente.

• Aislamiento : Garantiza que cada transacción está aislada del resto de transacciones y que el acceso a los datos se hará de forma exclusiva.

• Durabilidad : Los cambios que realiza una transacción sobre la base de datos son permanentes.

*2. ¿Cuáles son los tres problemas de concurrencia en el acceso a datos que pueden suceder cuando se realizan transacciones? Ponga un ejemplo para cada uno de ellos.*

Lectura sucia(dirty read): Una transacción lee datos que han sido escritos por otra transacción que aún no se ha confirmado.

Ejemplo: en una base de datos se actualiza la disponibilidad de un artículo a "agotado". Luego por otra consulta se busca el stock del articulo. Sin embargo, como la actualización anterior no se confirmó, si la transacción actual encuentra el valor "0", la información obtenida será errónea y podría afectar su toma de decisiones.

Lectura no repetible (Non-repeateable Read) : Se produce cuando una transacción lee un registro varias veces y obtiene diferentes valores cada vez.

Ejemplo: en una transacción X lee un registro que contiene información sobre el estado de un pedido, valor "pendiente". En ese momento, otra transacción Y actualiza el estado del mismo pedido a "enviado". Si la transacción X vuelve a leer el registro, obtendrá un valor diferente al que leyó en la primera lectura. En este caso, el valor obtenido por la transacción X cambió entre sus dos lecturas, debido a que la transacción Y actualizó el registro.

Lectura fantasma (Phantom Read) : Durante una transacción, se ejecutan dos consultas idénticas, y los resultados de la segunda no son iguales a los de la primera.

Ejemplo: Una transacción T1 realiza una consulta para obtener todos los clientes que han comprado productos en un rango de fecha específico. En ese momento, otra transacción T2 agrega un nuevo registro de cliente que cumple con ese criterio. Si la transacción T1 vuelve a realizar la misma consulta, obtendrá un conjunto de registros diferente al que obtuvo en la primera consulta, debido a que el nuevo registro añadido por la transacción T2 ahora aparece en los resultados.

*3. Cuando se trabaja con transacciones, el SGBD puede bloquear conjuntos de datos para evitar o permitir que sucedan los problemas de concurrencia comentados en el ejercicio anterior. ¿Cuáles son los cuatro niveles de aislamiento que se pueden solicitar al SGBD?*

READ UNCOMMITTED (no aislado): no garantiza ninguna protección contra los problemas de concurrencia, ya que permite que una transacción lea datos que han sido modificados por otra transacción que aún no se ha completado.

READ COMMITTED (aislamiento básico): garantiza que las transacciones solo pueden leer datos que han sido confirmados por otra transacción. Sin embargo, aún existe la posibilidad de que una transacción lea datos que han sido modificados por otra transacción que aún no se ha confirmado.

REPEATABLE READ (aislamiento medio): garantiza que una transacción siempre verá los mismos datos, incluso si otra transacción modifica los mismos datos en paralelo. Los datos se bloquean hasta que se completa la transacción, lo que significa que no se pueden modificar ni leer hasta que se completa la transacción.

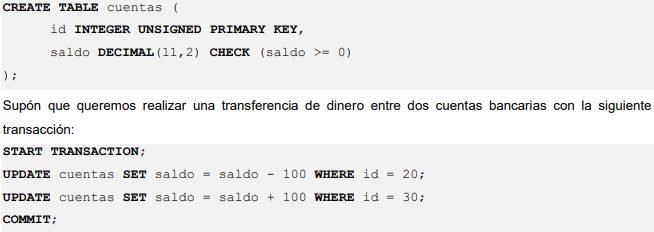
SERIALIZABLE (aislamiento completo): garantiza que las transacciones se ejecutan como si fueran una tras otra, en serie. Se bloquean los datos y se evitan los problemas de concurrencia, pero esto puede afectar el rendimiento del sistema si hay muchas transacciones simultáneas.

*4. ¿Cuál es el nivel de aislamiento que se usa por defecto en las tablas de Oracle? ¿Y en MySQL?*

En Oracle, el nivel por defecto es READ COMMITED.

En MySQL, el nivel de aislamiento por defecto es REPEATABLE READ

*5. Considera que tenemos una tabla donde almacenamos información sobre cuentas bancarias definida de la siguiente manera:*

**

*-¿Qué ocurriría si el sistema falla o si se pierde la conexión entre el cliente y el servidor después de realizar la primera sentencia UPDATE?*

Al no pasar por el commit no lo guarda

*-¿Qué ocurriría si no existiese alguna de las dos cuentas (id = 20 y id = 30)?*

Haría la transacción de forma erronea ya que al no existir ambas cuentas ejecutaría el commit dando error en la base de datos.

*-¿Qué ocurriría en el caso de que la primera sentencia UPDATE falle porque hay menos de 100 € en la cuenta y no se cumpla la restricción del CHECK establecida en la tabla?*

Restará la cantidad de saldo dicha y quedaria saldo 0 e insertaria 100$ a la otra cuenta, a pesar de que una de esas cuentas no tenia el saldo correspondiente para hacer bien la transaccion (100$)

*Operaciones SQL*

a.Inserta un jugador con codigo 11111, nombre juan pérez, de florida, altura 6-9 peso 85, posición F que sea del equipo de Kobe Bryant y cobre 50000



b.Inserta un jugador con codigo 2222 nombre pepe phone y que tenga el salario maximo de todos los jugadores



c.Hubo un error en los puntos por partido a lo largo de la carrera de Kobe bryant hay que sumar 10 puntos por partido a los partidos de Kobe Bryant



d.Los bulls tienen una penalización y se les resta 5 puntos por partidos en las estadísticas.



e.Por una sancion eliminamos los datos de estadisticas del equipo Bulls



*Conclusión*

En esta parte del trabajo pude aprender sobre las transacciones en SQL y el que pasa cuando algo se cae de por medio, es interesante el saber como funciona ya que necesitamos saber si hay algún error previsible.Y en la inserción , modificación y eliminación de datos es interesante ver que algo que ya hicimos antes podemos ampliar las posibilidades con consultas de por medio.