* **¿Qué son las propiedades ACID?**

Las propiedades ACID son un conjunto de características que garantizan la fiabilidad de las transacciones en un sistema de bases de datos. ACID significa Atomicidad, Consistencia, Aislamiento y Durabilidad.

¿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.

* **Los tres problemas son:**

***Lectura sucia***: Una transacción lee un dato modificado por otra transacción que aún no se ha confirmado. Por ejemplo, si una transacción A actualiza un registro, pero aún no lo ha confirmado, y otra transacción B lee ese registro, estaría realizando una lectura sucia.

***Lectura no repetible***: Una transacción repite una lectura y obtiene resultados diferentes debido a modificaciones realizadas por otra transacción en el ínterin. Por ejemplo, si una transacción A lee un registro, luego otra transacción B lo actualiza, y si la transacción A vuelve a leer el mismo registro, obtendrá un valor diferente.

***Lectura fantasma***: Una transacción repite una consulta y obtiene un conjunto de resultados diferente debido a inserciones o eliminaciones realizadas por otra transacción. Por ejemplo, si una transacción A lee un conjunto de registros, luego otra transacción B inserta un nuevo registro que cumple con el criterio de la consulta de A, y si A vuelve a ejecutar la misma consulta, obtendrá un conjunto de resultados diferente.

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?**

Los cuatro niveles de aislamiento son: READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ y SERIALIZABLE.

* **¿Cuál es el nivel de aislamiento que se usa por defecto en las tablas InnoDB de MySQL?**

El nivel de aislamiento por defecto en las tablas InnoDB de MySQL es REPEATABLE READ.

* **¿Es posible realizar transacciones sobre tablas MyISAM de MySQL?**

Sí es posible realizar transacciones sobre tablas MyISAM en MySQL, pero no son completamente compatibles con las características ACID y pueden llevar a problemas de integridad de los datos.

* **¿Qué diferencias existen entre los motores InnoDB y MyISAM de MySQL?**

InnoDB es un motor de almacenamiento transaccional que admite características ACID, proporciona bloqueos a nivel de fila y es más adecuado para aplicaciones que requieren transacciones y concurrencia.

MyISAM es un motor de almacenamiento no transaccional que no soporta ACID, utiliza bloqueos a nivel de tabla y es más adecuado para aplicaciones que realizan principalmente lecturas y no necesitan transacciones.

* **¿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?**

Si el sistema falla o se pierde la conexión después de realizar la primera sentencia UPDATE, pero antes de que se ejecute la segunda sentencia UPDATE, *la transacción quedará incompleta*. En este caso, la primera cuenta bancaria habría sido debitada, pero la segunda cuenta no habría sido acreditada. Esto podría resultar en una inconsistencia en los saldos de las cuentas.

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

Si una de las dos cuentas (ya sea la cuenta de origen o la cuenta de destino) no existe, la *transacción no se realizaría* y se produciría un error. No se realizaría ningún cambio en las cuentas, y se debería manejar el error adecuadamente en la lógica de la aplicación.

* **¿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?**

Si la primera sentencia UPDATE falla debido a que la cuenta de origen tiene menos de 100 € y no cumple con la restricción CHECK, *la transacción se revertiría automáticamente* debido a la naturaleza de las transacciones ACID. Esto significa que el dinero no se transferiría y los cambios realizados por la primera sentencia UPDATE serían deshechos. En este caso, la transacción se consideraría fallida y se debería manejar el error adecuadamente en la lógica de la aplicación.