

TECNOLÓGICO NACIONAL DE MÉXICO
CAMPUS JIQUILPAN
INGENIERÍA EN SISTEMAS COMPUTACIONALES



MATERIA:

TALLER DE BASES DE DATOS

UNIDAD: 4

GRUPO: B

TEMA:
TRANSACCIONES

ALUMNO:
GUTIÉRREZ SÁNCHEZ OSVALDO

PROFESOR:
MARTÍNEZ GONZÁLEZ LEONARDO

Preguntas

1.- ¿Qué es una transacción en bases de datos? En una base de datos, una transacción es un conjunto de operaciones que se ejecutan como una sola unidad de trabajo. Es decir, todas las operaciones de una transacción deben realizarse correctamente o ninguna se realiza.

2.- ¿Cómo puedo aplicar transacciones en mi proyecto? Se implementaron transacciones SQL para garantizar la atomicidad de las operaciones críticas, asegurando así la integridad de los datos. Se utilizó el mecanismo BEGIN TRANSACTION-COMMIT para agrupar múltiples actualizaciones, permitiendo un rollback completo en caso de fallo.

3.- ¿Cuáles son los tipos de bloqueos en bases de datos? Los bloqueos en bases de datos son mecanismos fundamentales para garantizar la consistencia y la integridad de los datos en entornos multiusuario. Estos bloqueos controlan el acceso concurrente a los recursos de la base de datos, evitando conflictos y garantizando que las transacciones se ejecuten correctamente.

Tipos de bloqueos:

Según la granularidad:

Bloqueos a nivel de tabla: Afectan a toda la tabla, impidiendo que otras transacciones modifiquen cualquier fila de esa tabla mientras el bloqueo está activo.

Bloqueos a nivel de página: Afectan a una página de datos, que es una unidad física de almacenamiento dentro de la tabla.

Bloqueos a nivel de fila: Son los más granulares y afectan solo a una fila específica de una tabla.

Según el modo de bloqueo:

Bloqueo exclusivo (X): Impide que otras transacciones lean o escriban en el recurso bloqueado. Solo la transacción que posee el bloqueo exclusivo puede realizar modificaciones.

Bloqueo compartido (S): Permite que múltiples transacciones lean el recurso bloqueado simultáneamente, pero ninguna puede modificarlo.

Bloqueo de actualización (U): Es un tipo de bloqueo intermedio que permite a una transacción leer el recurso y posteriormente actualizarlo. Sin embargo, impide que otras transacciones modifiquen el recurso mientras está bloqueado en modo de actualización.

Bloqueo de intención: Indica que una transacción tiene la intención de establecer un bloqueo de modo más restrictivo en el futuro cercano.

Otros tipos de bloqueos:

Bloqueo de esquema: Se aplica a objetos de esquema como tablas, índices y vistas.

Bloqueo de espacio de tabla: Se aplica a un espacio de tabla completo, que es una unidad de almacenamiento lógico.

4.- ¿Niveles de atracción y cómo funcionan?

- **Lecturas sucias:** Una transacción lee un valor que aún no ha sido confirmado por otra transacción.
- **Lecturas no repetibles:** Una transacción lee el mismo dato varias veces y obtiene resultados diferentes en cada lectura.
- **Pérdida de actualizaciones:** Los cambios realizados por una transacción se pierden debido a los cambios realizados por otra transacción.
- **Bloqueos fantasma:** Una transacción no encuentra una fila que existía cuando comenzó la transacción, pero que fue eliminada por otra transacción.

Los principales niveles de atracción:

- **Read Uncommitted (Lectura no confirmada):** El nivel de aislamiento más bajo. Permite a una transacción leer datos no confirmados por otras transacciones, lo que puede llevar a lecturas sucias, no repetibles y pérdidas de actualizaciones.
- **Read Committed (Lectura confirmada):** Garantiza que una transacción solo leerá datos que ya han sido confirmados por otras transacciones, evitando lecturas sucias. Sin embargo, aún puede ocurrir el problema de lecturas no repetibles.
- **Repeatable Read (Lectura repetible):** Evita tanto lecturas sucias como lecturas no repetibles. Una vez que una transacción ha leído un conjunto de datos, otros no pueden modificar esos datos hasta que la transacción se complete. Sin embargo, puede ocurrir el problema de bloqueos fantasma.
- **Serializable:** El nivel de aislamiento más alto. Garantiza que las transacciones se ejecuten como si se ejecutaran secuencialmente, evitando cualquier tipo de anomalía de concurrencia.