



“TRANSACCIONES, BLOQUEOS Y NIVELES DE AISLAMIENTO”

TALLER DE BASES DE DATOS

INGENIERÍA EN SISTEMAS COMPUTACIONALES PRESENTA:

EMMANUEL SALDAÑA ÁLVAREZ

LEONARDO MARTINEZ GONZALEZ

JIQUILPAN, MICHOACÁN, NOVIEMBRE DE 2024



Contenido

¿QUÉ SON LAS TRANSACCIONES?	3
¿CÓMO APLICO LAS TRANSACCIONES A MI PROYECTO?	3
¿TIPOS DE BLOQUEOS?	4
¿NIVELES DE AISLAMIENTO Y CÓMO FUNCIONAN?	4

¿QUÉ SON LAS TRANSACCIONES?

Las transacciones representan un grupo de operaciones SQL que se llevan a cabo como una unidad independiente. En una operación, o todas las transacciones se finalizan de manera adecuada, o ninguna se realiza. Este concepto se denomina ACID:

Atomicidad: La transacción es indivisible, lo que significa que se realiza completamente o no sucede.

Coherencia: Las operaciones transfieren la base de datos de un estado válido a otro estado válido.

Aislamiento: Garantiza que las transacciones de una transacción permanezcan ocultas a otras hasta que se verifiquen.

Durabilidad: Una vez confirmada una transacción, las modificaciones permanecen inalterables, incluso si el sistema se bloquea.

¿CÓMO APLICO LAS TRANSACCIONES A MI PROYECTO?

Para aplicar transacciones en el proyecto, estas se usan principalmente para asegurar que un conjunto de operaciones en la base de datos se complete de manera correcta y coherente, especialmente en casos en que varios cambios deben ocurrir simultáneamente en diferentes tablas o registros.

Por ejemplo, supongamos que se debe registrar un préstamo de libros en la base de datos Biblioteca_mangas. Este proceso podría requerir varios pasos: registrar el préstamo en una tabla de `Prestamos` y luego registrar detalles específicos en otra tabla, como la cantidad de libros en `DetallePrestamos`. Para que los datos sean consistentes, es importante que ambos registros se realicen juntos o no se realice ninguno. Si uno de los pasos falla (como un error al escribir en una de las tablas), los cambios de ambos pasos deberían deshacerse.

Las transacciones ayudan a implementar este comportamiento, iniciándose con **BEGIN TRANSACTION**. Si todas las operaciones se ejecutan correctamente, el comando **COMMIT** finaliza la transacción y confirma los cambios. Si ocurre un error, el comando **ROLLBACK** revierte todos los cambios realizados en la transacción, manteniendo la base de datos en un estado consistente.

¿TIPOS DE BLOQUEOS?

SQL Server emplea bloqueos para regular el acceso simultáneo a la información y prevenir disputas. Los bloqueos principales incluyen:

Bloqueo compartido (S): Facilita que varias operaciones lean un recurso (como una secuencia o una página) sin realizar cambios.

Bloqueo exclusivo (X): Facilita que una única operación altere un recurso y previene que otras operaciones lean o se registren en el recurso bloqueado.

Bloqueo de actualización (U): Se aplica cuando una operación requiere la actualización de un recurso, para prevenir un "deadlock" entre bloqueos compartidos y exclusivos.

Bloqueo intencional: SQL Server emplea bloqueos intencionales (IX, IS, IU) en niveles de tabla o página para señalar bloqueos situados en nivel de fila, garantizando uniformidad en los niveles de jerarquía.

¿NIVELES DE AISLAMIENTO Y CÓMO FUNCIONAN?

Los niveles de aislamiento establecen el control de las interacciones entre transacciones simultáneas y evitan inconvenientes como lecturas sucias, lecturas no repetitivas y lecturas fantasmales. Los grados de aislamiento en SQL Server son los siguientes:

Read Uncommitted: Las transacciones tienen la capacidad de leer información no confirmada por otras operaciones, lo que facilita lecturas veloces, pero con el peligro de lecturas sucias.

Read Committed: Facilita la lectura únicamente de datos confirmados por otras transacciones, previniendo lecturas sucias, pero facilitando lecturas no repetitivas.

Repeatable Read: Garantiza que, si una operación lee un valor, ese valor permanecerá inalterable durante la operación, previniendo lecturas no repetibles, pero puede provocar lecturas fantasmas.

Serializable: El nivel de aislamiento más elevado; garantiza que las lecturas no sean alteradas por otras operaciones, evitando así lecturas no repetibles y lecturas fantasma.