Universidad Nacional Autónoma de México

Núñez Badillo Armando Adair

Bases de datos

Grupo 1

Tarea 11

Niveles de aislamiento en bases de datos

Semestre 2026-1

Profesor: Ing. Fernando Arreola

Fecha de entrega: 24/10/25

Niveles de aislamiento en base de datos relacionales

Cuando un proceso de aplicación accede a datos, el nivel de aislamiento determina el grado en que esos datos están bloqueados o aislados de otros procesos concurrentes. El nivel de aislamiento está en vigor durante una unidad de trabajo.

Por lo tanto, el nivel de aislamiento de un proceso de aplicación especifica:

- El grado en que las filas leídas o actualizadas por la aplicación están disponibles para otros procesos de aplicación que se ejecutan simultáneamente.
- El grado en que la actividad de actualización de otros procesos de aplicación que se ejecutan simultáneamente puede afectar a la aplicación.
- El nivel de aislamiento correspondiente a sentencias de SQL estático se especifica como un atributo de un paquete y se aplica a los procesos de aplicación que utilizan ese paquete.

Los cuatro niveles de aislamiento de bases de datos relacionales, del más bajo al más alto, son: Lectura no confirmada (READ UNCOMMITTED), Lectura confirmada (READ COMMITTED), Lectura repetible (REPEATABLE READ) y Serial (SERIALIZABLE). Estos niveles controlan cómo las transacciones se aíslan de otras transacciones simultáneas, afectando la integridad de los datos y el rendimiento.

Niveles de aislamiento

Lectura no confirmada (READ UNCOMMITTED):

Es el nivel más bajo. Permite leer datos que otras transacciones no han confirmado. Esto ofrece un rendimiento rápido, pero puede provocar problemas como lecturas sucias, lecturas no repetibles y lecturas fantasma.

Lectura confirmada (READ COMMITTED):

Una transacción solo puede leer datos que ya han sido confirmados por otras transacciones. Este nivel evita las lecturas sucias, pero todavía puede experimentar lecturas fantasma. Es un nivel común y a menudo predeterminado en sistemas como SQL Server.

Lectura repetible (REPEATABLE READ):

Este nivel garantiza que si una transacción lee una fila varias veces, obtendrá los mismos datos cada vez, evitando lecturas no repetibles. Sin embargo, aún puede ocurrir el problema de las

lecturas fantasma (una fila nueva que cumple con los criterios de la lectura puede aparecer en una nueva lectura).

Serial (SERIALIZABLE):

Ofrece el nivel de aislamiento más alto. Las transacciones se ejecutan como si se ejecutaran una tras otra, en serie, sin interrupciones por otras transacciones. Esto garantiza la máxima integridad de los datos, evitando lecturas sucias, lecturas no repetibles y lecturas fantasma, aunque puede afectar significativamente el rendimiento al requerir bloqueos durante toda la transacción.

Propiedades ACID

ACID es un acrónimo que describe las cuatro propiedades clave que garantizan que las transacciones en una base de datos se procesen de manera fiable. Una **transacción** es simplemente un conjunto de una o más operaciones de base de datos (como leer, escribir, actualizar o borrar datos) que se ejecutan como una sola unidad de trabajo.

Si se tiene una **transferencia bancaria** de $100 de una cuenta A a una cuenta B. Esta transacción en realidad son dos operaciones:

1. Restar $100 de la cuenta A.
2. Sumar $100 a la cuenta B.

Las propiedades ACID aseguran que esta operación ocurra de forma correcta.

1. Atomicidad (Atomicity)

Esta propiedad garantiza que la transacción es "todo o nada". O se completan **todas** las operaciones dentro de la transacción, o no se completa **ninguna** (se deshace todo, como si nunca hubiera ocurrido).

- **Ejemplo:** En la transferencia bancaria, si se logra restar $100 de la cuenta A, pero el sistema falla (por ejemplo, se va la luz) antes de poder sumar $100 a la cuenta B, la **atomicidad** revierte la operación de resta. El dinero vuelve a la cuenta A. Es imposible que el dinero "desaparezca" y se quede en el limbo.

2. Consistencia (Consistency)

La base de datos siempre debe pasar de un estado válido a otro estado válido. Las transacciones no pueden violar las reglas de integridad definidas (como llaves foráneas, restricciones NOT NULL, o reglas de negocio como "el saldo no puede ser negativo").

- **Ejemplo:** Si la cuenta A tiene $50, la transacción para transferir $100 no puede completarse. La **consistencia** lo impide porque violaría una regla de negocio (saldo negativo no permitido). La transacción se cancela y la base de datos permanece en su estado válido original (con $50 en la cuenta A y sin cambios en la B).

3. Aislamiento (Isolation)

Esta es la propiedad clave. El **aislamiento** asegura que las transacciones que se ejecutan al mismo tiempo (concurrentemente) no interfieran entre sí. Desde la perspectiva de una transacción, parece que es la única que se está ejecutando en el sistema en ese momento.

- **Ejemplo:** Imagina que tienes $500 en tu cuenta A. Al mismo tiempo, intentas hacer dos cosas:
    1. **Transacción 1:** Pagar un servicio de $400.
    2. **Transacción 2:** Consultar tu saldo para un préstamo. Sin **aislamiento**, la Transacción 2 podría leer tu saldo *después* de que se restaron los $400 pero *antes* de que la Transacción 1 se confirme. O peor, ambas transacciones podrían intentar usar los $500 al mismo tiempo. El aislamiento previene esto, haciendo que una transacción espere a que la otra termine o, al menos, no vea sus cambios a medias.

4. Durabilidad (Durability)

Una vez que una transacción ha sido confirmada (ha finalizado exitosamente), sus cambios son **permanentes**. Sobrevivirán a cualquier fallo del sistema, como un corte de energía, un reinicio del servidor o un fallo del software.

- **Ejemplo:** Completar la transferencia bancaria. El sistema muestra "Transferencia Exitosa". Justo en ese segundo, se va la luz en el banco. Gracias a la **durabilidad**, cuando el sistema se reinicie, tu transferencia seguirá completada. Los datos se escribieron en un registro (log) en el disco duro antes de que se confirmara la transacción.

A. Silberschatz, H. F. Korth, and S. Sudarshan, *Database System Concepts*, 7th ed. New York, NY, USA: McGraw-Hill, 2019.

R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, 7th ed. Harlow, Essex, UK: Pearson, 2017.