



Tecnológico de Monterrey Campus Querétaro

Construcción de software

Laboratorio 25: Transacciones

Maestro/a:

Enrique Alfonso Calderon Balderas

Presenta:

Ricardo Alejandro Andrade Robledo / A01706813

En esta actividad me comprometo a aplicar mis conocimientos, a esforzarme en su desarrollo y a no servirse de medios no autorizados o ilícitos para realizarla.

Fecha de entrega: 14 de mayo de 2025

¿Qué es un Deadlock en Bases de Datos?

Un deadlock (bloqueo mortal) ocurre cuando dos o más transacciones se bloquean mutuamente porque cada una está esperando que la otra libere un recurso (como filas, tablas o locks), creando un ciclo de espera infinito.

Ejemplo Clásico de Deadlock

- Transacción A bloquea el Recurso 1 y solicita el Recurso 2.
- Transacción B bloquea el Recurso 2 y solicita el Recurso 1.
- Resultado: Ambas transacciones quedan eternamente esperando, sin progresar.

-- Transacción 1

BEGIN TRANSACTION;

UPDATE Cuenta SET saldo = saldo - 100 WHERE id = 1; -- Bloquea fila 1

-- Espera porque la Transacción 2 tiene bloqueada la fila 2

UPDATE Cuenta SET saldo = saldo + 100 WHERE id = 2;

COMMIT;

-- Transacción 2 (ejecutada en paralelo)

BEGIN TRANSACTION;

UPDATE Cuenta SET saldo = saldo - 50 WHERE id = 2; -- Bloquea fila 2

-- Espera porque la Transacción 1 tiene bloqueada la fila 1

UPDATE Cuenta SET saldo = saldo + 50 WHERE id = 1;

COMMIT;

Consecuencias del Deadlock

- Las transacciones involucradas no se completan.
- El sistema pierde rendimiento (espera inútil).
- Requiere intervención del DBMS (generalmente, abortando una transacción).

¿Cómo lo Manejan las Bases de Datos?

1. Detección:
 - a. El motor (MySQL, PostgreSQL, Oracle, etc.) monitorea dependencias cíclicas.
 - b. Usa algoritmos como Wait-for Graph.
2. Resolución:
 - a. Se elige una víctima (transacción a abortar) basado en prioridades (ej: la que menos lleva).
 - b. Se hace ROLLBACK de esa transacción y se liberan sus recursos.
 - c. La otra transacción puede continuar.
3. Prevención:

- a. Órdenes de acceso fijos (ej: siempre bloquear filas en orden ascendente por ID).
- b. Timeouts: Si una transacción espera demasiado, se aborta
- c. Niveles de aislamiento adecuados (evitar bloqueos innecesarios).