



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

Facultad de Ingenieria

## TAREA 6

Niveles de Aislamiento en Bases de Datos

Grupo 5

Semestre 2020-2

BASES DE DATOS

Profesor: Ing. Fernando Arreola Franco

**Integrantes:**

Vivanco Quintanar, Diego Armando

26 de marzo de 2020

# 1. Introducción

## 1.1. ¿Qué es una transacción en bases de datos?

Una transacción es una unidad de trabajo compuesta por diversas tareas, cuyo resultado final debe ser que se ejecuten todas o ninguna de ellas. En un sistema de base de datos todas las operaciones relacionadas entre sí que se ejecuten dentro un mismo flujo lógico de trabajo, deben ejecutarse en bloque. De esta manera si todas funcionan la operación conjunta de bloque tiene éxito, pero si falla cualquiera de ellas, deberán retrocederse todas las anteriores que ya se hayan realizado. De esta forma evitamos que el sistema de datos quede en un estado incongruente.

Por ejemplo, si vamos al banco y ordenamos una transferencia para pagar una compra que hemos realizado por Internet, el proceso en sí está formado por un conjunto (o bloque) de operaciones que deben ser realizadas para que la operación global tenga éxito:

1. Comprobar que nuestra cuenta existe es válida y está operativa.
2. Comprobar si hay saldo en nuestra cuenta.
3. Comprobar los datos de la cuenta del vendedor (que existe, que tiene posibilidad de recibir dinero, etc...).
4. Retirar el dinero de nuestra cuenta.
5. Ingresar el dinero en la cuenta del vendedor.

Dentro de este proceso hay cinco operaciones, las cuales deben tener éxito o fallar conjuntamente. En el caso de las operaciones 4 y 5 que modifican datos es algo obvio que no puede funcionar una y fallar la otra. Si se retira el dinero de nuestra cuenta en el paso 4 y hay algún problema que evita que pueda continuar el proceso, el dinero habrá salido de nuestra cuenta pero no se ha anotado en la cuenta de destino porque se ha producido un error. De repente hay un dinero que ha desaparecido y la base de datos se encuentra en un estado inconsistente. Es evidente que esto no puede ocurrir.

Precisamente para evitar este tipo de situaciones existen las transacciones: marcan bloques completos de operaciones y comprueban que, o se realizan todas, o que si hay algún problema se deshacen todas.

En nuestro ejemplo de transferencia fallida, al producirse un error en el paso 5 se habría deshecho automáticamente la operación de retirada de dinero del paso 4 y toda la información habría quedado como antes de comenzar el proceso.

Otro tema importante relacionado con las transacciones es la gestión de la concurrencia y los bloqueos. En el ejemplo del banco los 3 primeros pasos realmente no implican modificación alguna de datos, por lo que si fallan no dejan la base de datos en un estado inconsistente ¿o quizá sí?. ¿Qué pasaría si al mismo tiempo que se está realizando nuestra transferencia, entra en nuestra cuenta un cargo diferido que teníamos pendiente? Sería posible que de repente nos quedásemos sin saldo para realizar la operación actual o, peor aún, que se anotasen mal ambos cargos en cuenta de modo que se "pisasen"dejando un saldo inconsistente. O puede que entre los pasos 3 y 5 la cuenta del vendedor de repente se haya cancelado, justo en medio de la operación. El dinero iría a parar a una cuenta no válida.

Para controlar el comportamiento de las transacciones en estos casos se definen diferentes niveles de aislamiento de una transacción.

## 2. Niveles de Aislamiento en Bases de Datos

Las transacciones especifican un nivel de aislamiento que define el grado en que se debe aislar una transacción de las modificaciones de recursos o datos realizadas por otras transacciones. Los niveles de aislamiento se describen en función de los efectos secundarios de la simultaneidad que se permiten, como las lecturas de datos sucios o las lecturas fantasmas.

Los niveles de aislamiento de transacciones controlan lo siguiente:

- Controla si se realizan bloqueos cuando se leen los datos y qué tipos de bloqueos se solicitan.

- Duración de los bloqueos de lectura.
- Si una operación de lectura que hace referencia a filas modificadas por otra transacción:
  - Se bloquea hasta que se libera el bloqueo exclusivo de la fila.
  - Recupera la versión confirmada de la fila que existía en el momento en el que se inició la instrucción o la transacción.
  - Lee la modificación de los datos no confirmada.

Un nivel de aislamiento menor significa que los usuarios tienen un mayor acceso a los datos simultáneamente, con lo que aumentan los efectos de la simultaneidad que pueden experimentar, como las lecturas de datos sucios (Las lecturas sucias (o lecturas no confirmadas) son lecturas de filas que están siendo modificadas por una transacción abierta.) o la pérdida de actualizaciones. Por el contrario, un nivel de aislamiento mayor reduce los tipos de efectos de simultaneidad, pero requiere más recursos del sistema y aumenta las posibilidades de que una transacción bloquee a otra. El nivel de aislamiento apropiado depende del equilibrio entre los requisitos de integridad de los datos de la aplicación y la sobrecarga de cada nivel de aislamiento.

El nivel de aislamiento superior, que es serializable, garantiza que una transacción recuperará exactamente los mismos datos cada vez que repita una operación de lectura, aunque para ello aplicará un nivel de bloqueo que puede afectar a los demás usuarios en los sistemas multiusuario. El nivel de aislamiento menor, de lectura sin confirmar, puede recuperar datos que otras transacciones han modificado pero no confirmado. En este nivel se pueden producir todos los efectos secundarios de simultaneidad, pero no hay bloqueos ni versiones de lectura, por lo que la sobrecarga se reduce.

Los niveles de aislamiento que nos ofrece SQL Server son:

- **SERIALIZABLE:** No se permitirá a otras transacciones la inserción, actualización o borrado de datos utilizados por nuestra transacción. Los bloquea mientras dura la misma.
- **REPEATABLE READ:** Garantiza que los datos leídos no podrán ser cambiados por otras transacciones, durante esa transacción.
- **READ COMMITED:** Una transacción no podrá ver los cambios de otras conexiones hasta que no hayan sido confirmados o descartados.
- **READ UNCOMMITTED:** No afectan los bloqueos producidos por otras conexiones a la lectura de datos.
- **SNAPSHOT:** Los datos seleccionados en la transacción se verán tal y como estaban al comienzo de la transacción, y no se tendrán en cuenta las actualizaciones que hayan sufrido por la ejecución de otras transacciones simultáneas.

Las transacciones se deben ejecutar en un nivel de aislamiento de lectura repetible, al menos, para evitar las pérdidas de actualizaciones que pueden producirse cuando dos transacciones recuperan la misma fila, y a continuación la actualizan según los valores recuperados originalmente. Si las dos transacciones actualizan las filas con una única instrucción UPDATE y no basan la actualización en los valores recuperados previamente, la pérdida de las actualizaciones no puede producirse en el nivel de aislamiento predeterminado de lectura confirmada.

### 3. Referencias

- campusMVP. (2014). Fundamentos de SQL: Transacciones. 17/04/2020, de campusMVP Sitio web: <https://www.campusmvp.es/recursos/post/Fundamentos-de-SQL-Transacciones.aspx>
- Microsoft. (S.A.). Descripción de los niveles de aislamiento. 17/04/2020, de Microsoft Sitio web: <https://docs.microsoft.com/es-es/sql/connect/jdbc/understanding-isolation-levels?view=sql-server-ver15>