

## LAS TRANSACCIONES

### CONCEPTOS BÁSICOS:

Desde el punto de vista del usuario la interacción con la base de datos se lleva a cabo mediante operaciones con significado en el modelo semántico (por ejemplo, una transferencia de fondos en un banco).

Desde el punto de vista de la base de datos estas operaciones pueden estar formadas por varias operaciones elementales (por ejemplo, quitar fondos de una cuenta y añadirse los a otra)

Se llama Transacción a una colección de operaciones que forman una unidad lógica de trabajo en una BD realizada por una o más sentencias SQL estrechamente relacionadas.

Una transacción es una unidad de la ejecución de un programa que lee y escribe datos a y desde la Base de Datos. Puede consistir en varias operaciones de acceso a la base de datos. Una Transacción está delimitada por instrucciones de inicio transacción y fin transacción (la transacción consiste en todas las operaciones que se ejecutan entre inicio transacción y fin transacción).

Este concepto vio la luz debido a que varias operaciones sucesivas pueden modificar el resultado de operaciones anteriores. En esos casos, si alguna operación produce un error, el estado resultante puede ser indeterminado.

Para solucionar este problema, las transacciones agrupan una serie de operaciones de manera que es posible garantizar la integridad del resultado final. O todas las operaciones se ejecutan con éxito y se confirman (se escriben en la base de datos), o toda la transacción se considera no realizada. La acción de cancelar una transacción se denomina deshacer la transacción. Deshacer una transacción permite anular los cambios y recuperar el estado de la base de datos previo a la transacción.

Por ejemplo, en una transacción bancaria automatizada, si un banco transfiere dinero desde la cuenta A a la cuenta B, la retirada de fondos de A y el depósito en B deben producirse con éxito para procesar los fondos correctamente, de lo contrario la transacción entera debe cancelarse.

El proceso de transacciones tenemos: se ejecutan todas las operaciones que componen la transacción, o no se realiza ninguna.

En SQL	
Éxito	Fracaso
<pre>Begin transacction Instrucción 1 Instrucción 2 ... Commit work End transacction</pre>	<pre>Begin transacction Instrucción 1 Instrucción 2 ... Rollback work End transacction</pre>

Las transacciones son un conjunto de órdenes que se ejecutan formando una unidad de trabajo, es decir, en forma indivisible.

Podemos contar con varias operaciones pero las más comunes son:

BEGIN: Especifica que va a empezar una transacción.

COMMIT: Le indica al motor que puede considerar la transacción completada con éxito.

ROLLBACK: Indica que se ha alcanzado un fallo y que debe restablecer la base al punto de integridad sin realizar ningún cambio dentro de la base de datos.

### **PROPIEDADES DE LAS TRANSACCIONES:**

Una unidad lógica de trabajo debe exhibir cuatro propiedades, conocidas como propiedades ACID (atomicidad, coherencia, aislamiento y durabilidad), para ser calificada como transacción.

Atomicity: Una Transacción (X) se ejecuta completamente o de otra manera se eliminan los cambios parciales realizados.

Begin : comentado anteriormente es donde inicia la transacción.

Programa: realizas los cambios que deseas en la base de datos.

End Transaccion: termina la transacción que acabas de realizar.

Responsable: El método de recuperación, de no completar todas las operaciones, devuelve la BD a su estado anterior a empezar esa X (rollback) se utiliza rollback cuando los cambios que utilizamos no son los requeridos en la base de datos, dejándolos en un estado original como si no se hubiera realizado ningún movimiento o cambio.

Coherencia:Asegura que los datos que observamos no cambian (por otros usuarios) hasta que acabemos la Transacción.

Responsable:los programadores mediante la definición adecuada de la integridad referencial: check, triggers, primary key, foreign key,...

Aislamiento: Los efectos de una X no son visibles a otros usuarios mientras no se confirmen.

Una Transacción en ejecución no puede revelar sus resultados a otras transacciones concurrentes antes de finalizar.

Más aún, si varias transacciones, se ejecutan consecuentemente los resultados deben ser los mismos que si ellas se hubieran ejecutado secuencialmente. Esto se conoce como seriabilidad debido a que su resultado es la capacidad de volver a cargar los datos iniciales y reproducir una serie de transacciones para finalizar con los datos en el mismo estado en que estaban después de realizar transacciones originales.d

Durabilidad: Si el sistema falla no debe permitir que se pierdan las operaciones realizadas por X ya confirmadas.



**COMMIT:** Finaliza la transacción actual y hace permanentes (confirma) los cambios realizados dentro de la base de datos.

**ROLLBACK:** Finaliza la transacción actual y deshace los cambios realizados si los datos ingresados no son los que se requerían dentro de la base de datos.



## GRADOS DE CONSISTENCIA:

Consistencia es un término más amplio que el de integridad. Podría definirse como la coherencia entre todos los datos de la base de datos. Cuando se pierde la integridad también se pierde la consistencia. Pero la consistencia también puede perderse por razones de funcionamiento.

Una transacción finalizada (confirmada parcialmente) puede no confirmarse definitivamente (consistencia).

Si se confirma definitivamente el sistema asegura la persistencia de los cambios que ha efectuado en la base de datos.

Si se anula los cambios que ha efectuado son deshechos gracias al rollback.

La ejecución de una transacción debe conducir a un estado de la base de datos consistente (que cumple todas las restricciones de integridad definidas).

Si se confirma definitivamente el sistema asegura la persistencia de los cambios que ha efectuado en la base de datos.

Si se anula los cambios que ha efectuado son deshechos.

Una transacción que termina con éxito se dice que está comprometida (committed), una transacción que haya sido comprometida llevará a la base de datos a un nuevo estado consistente que debe permanecer incluso si hay un fallo en el sistema. En cualquier momento una transacción sólo puede estar en uno de los siguientes estados.

Activa (Active): el estado inicial; la transacción permanece en este estado durante su ejecución.

Parcialmente comprometida (Uncommitted): Después de ejecutarse la última transacción, ya que los cambios que hemos realizado todavía no son los correctos.

Comprometida (Committed): tras completarse con éxito y una vez ejecutada la instrucción no hay cambios.

Fallida (Failed): Esto se realiza tras descubrir que no se puede continuar la ejecución normal ya que probablemente ocurrió un error durante la ejecución del comando.

Abortada (Rolled Back): después de haber retrocedido la transacción y restablecido la base de datos a su estado anterior al comienzo de la transacción.

Ahora refiriéndonos a niveles de aislamiento podemos encontrar lo siguiente:

El estándar SQL trataba de establecer los niveles de aislamiento que permitirían a varios grados de consistencia para queries ejecutadas en cada nivel de aislamiento. Las lecturas repetibles "REPEATABLE READ" es el nivel de aislamiento que garantiza que un query un resultado consistente. En la definición SQL estándar, la lectura comprometida "READ COMMITTED" no regresa resultados consistentes, en la lectura no comprometida "READ UNCOMMITTED" las sentencias SELECT son ejecutadas sin realizar bloqueos, pero podría usarse una versión anterior de un registro. Por lo tanto, las lecturas no son consistentes al usar este nivel de aislamiento.

## **COMMIT Y ROLLBACK:**

Ejemplo básico en MySQL precio del dolar con respecto al peso 1995 al 2012.

```
1 CREATE TABLE IF NOT EXISTS dolar (  
2   fecha DATE,  
3   precio DECIMAL (8,4),  
4   PRIMARY KEY (fecha)  
5 ) ENGINE = InnoDB DEFAULT CHARSET=latin1;
```

Commit (acción de comprometer) se refiere a la idea de consignar un conjunto de cambios "tentativos, o no permanentes". Un uso popular es al final de una transacción de base de datos.

Una sentencia COMMIT en SQL finaliza una transacción de base de datos dentro de un sistema gestor de base de datos relacional (RDBMS) y pone visibles todos los cambios a

otros usuarios. El formato general es emitir una sentencia BEGIN, una o más sentencias SQL, y entonces la sentencia COMMIT. Alternativamente, una sentencia ROLLBACK se puede emitir, la cual deshace todo el trabajo realizado desde que se emitió BEGIN. Una sentencia

En términos de transacciones, lo opuesto de commit para descartar los cambios "en tentativa" de una transacción, es un rollback.

un rollback es una operación que devuelve a la base de datos a algún estado previo. Los Rollbacks son importantes para la integridad de la base de datos, a causa de que significan que la base de datos puede ser restaurada a una copia limpia incluso después de que se han realizado operaciones erróneas. Son cruciales para la recuperación de crashes de un servidor de base de datos; realizando rollback(devuelto) cualquier transacción que estuviera activa en el tiempo del crash, la base de datos es restaurada a un estado consistente.

En SQL, ROLLBACK es un comando que causa que todos los cambios de datos desde la última sentencia BEGIN, o START TRANSACTION sean descartados por el sistema de gestión de base de datos relacional (RDBMS), para que el estado de los datos sea "rolled back"(devuelto) a la forma en que estaba antes de que aquellos cambios tuvieran lugar.

En muchos dialectos de SQL, ROLLBACKs son específicos de la conexión. Esto significa que si se hicieron dos conexiones a la misma base de datos, un ROLLBACK hecho sobre una conexión no afectará a cualesquiera otras conexiones. Esto es vital para el buen funcionamiento de la Concurrency.

La funcionalidad de rollback está normalmente implementada con un Log de transacciones, pero puede también estar implementada mediante control de concurrencia multiversión.