

Unidad V: Transacciones

5.1 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 añadírselos 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).

El concepto de transacción se desarrolló para atender los casos en los que el estado resultante de la base de datos depende del éxito completo en una serie de operaciones. 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.

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

5.2 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 (**Tx**) se ejecuta completamente ó de otra manera se eliminan los cambios parciales realizados.

Begin Transaction - Programa - End Transaction

Responsable: El método de recuperación, de no completar todas las operaciones, devuelve la BD a su estado anterior a empezar esa **Tx**(rollback).

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

Después de terminar una Transacción la Base de datos no viola ninguna de sus reglas: valores obligatorios, claves únicas, etc.

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

- **Aislamiento:** Los efectos de una **Tx** 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.

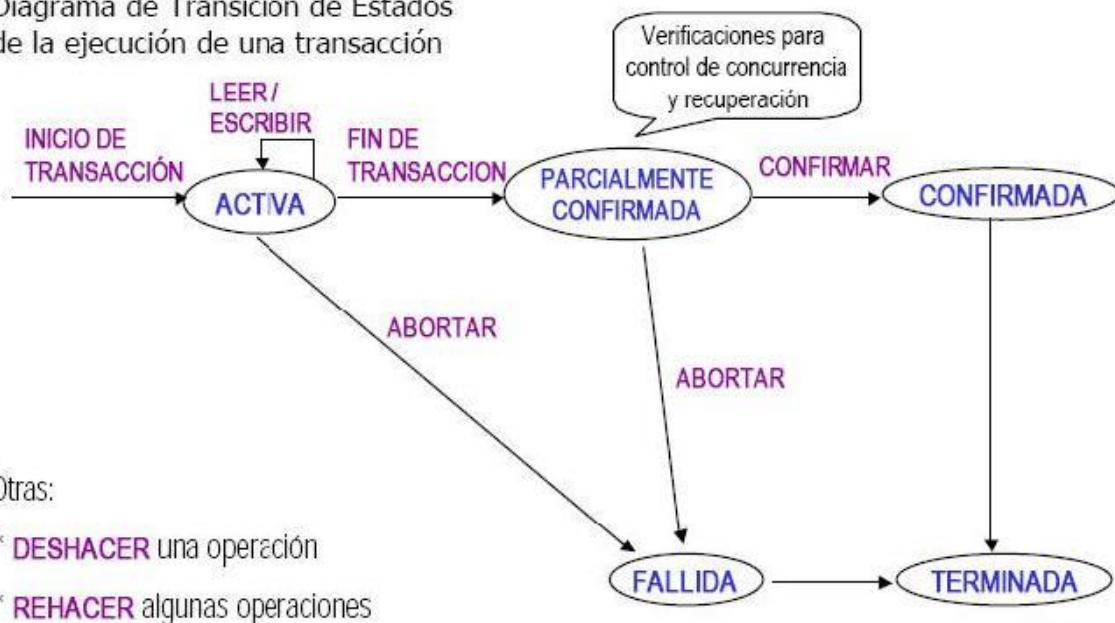
Más aun, si varias transacciones, se ejecutan concurrentemente, los resultados deben ser los mismos que si ellas se hubieran ejecutado secuencialmente. Esto se conoce como seriabilidad debido a que su resultado es la capacidad de volver a cargar los datos iniciales y reproducir una serie de transacciones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

Responsable: el **método de concurrencia:** mecanismos, reglas, protocolos

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

Responsable: el **método o gestor de recuperación.**

Diagrama de Transición de Estados de la ejecución de una transacción



5.3 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.

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

5.4 Niveles de aislamiento

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 cuanto a los efectos secundarios de la simultaneidad que se permiten, como las lecturas desfasadas o ficticias.

Control de los niveles de aislamiento de transacción:

- 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 empezó la instrucción o la transacción.
 - Lee la modificación de los datos no confirmados.

5.5 Commit y rollback

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

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

Lecturas consistentes

Por default, las tablas InnoDB ejecutan un lectura consistente (consistent read). Esto significa que cuando una sentencia SELECT es ejecutada, MySQL regresa los valores presentes en la base de datos hasta la transacción más reciente que ha sido completada. Si alguna transacción está en progreso, los cambios hechos por alguna sentencia INSERT o UPDATE no serán reflejados. Sin embargo, existe una excepción: las transacciones abiertas si pueden ver sus propios cambios. Para demostrar esto, necesitamos establecer dos conexiones al servidor MySQL.