

# **BD Avanzada**

## **Tema 1: Gestión de Transacciones**

# Gestión de Transacciones

## Objetivo General:

- Al completar este tema el estudiante estará en la capacidad de manejar los conceptos sobre Gestión de Transacciones.

## Objetivos Específicos: Al completar este tema el estudiante deberá estar en la capacidad de discutir sobre los siguientes conceptos:

- Procesamiento de transacciones
- Atomicidad de las transacciones
- La consistencia, el aislamiento, la durabilidad y secuencialidad
- Control de concurrencia
- Técnicas que aseguran la secuencialidad, incluyendo los bloqueos, las marcas temporales
- Técnicas principales para asegurar la ejecución correcta de transacciones

# Gestión de Transacciones

- **Transacciones:** Conjunto de operaciones que forman una única unidad lógica de trabajo.
- **Propiedades de las transacciones:**
  - **Atomicidad.** O todas las operaciones de la transacción se realizan adecuadamente en la base de datos o ninguna de ellas.
  - **Consistencia.** La ejecución aislada de la transacción conserva la consistencia de la base de datos.
  - **Aislamiento.** Cada transacción ignora al resto de las transacciones que se ejecuten concurrentemente en el sistema.
  - **Durabilidad.** Tras la finalización con éxito de una transacción, los cambios realizados en la base de datos permanecen, incluso si hay fallos en el sistema

# Gestión de Transacciones

- **Atomicidad.**

- O todas las acciones de la transacción se ven reflejadas en la base de datos, o ninguna de ellas.
- **Estado inconsistente**, como resultado del fallo, el estado del sistema deja de reflejar el estado real del mundo que se supone que modela la base de datos.
- **La responsabilidad de asegurar la atomicidad** es del sistema de base de datos es del *componente de gestión de transacciones*.

- **Consistencia.**

- Si una base de datos es consistente antes de ejecutar una transacción, sigue siéndolo después de ejecutar dicha transacción.
- **La responsabilidad de asegurar la consistencia de una transacción** es del *programador de la aplicación* que codifica dicha transacción.

# Gestión de Transacciones

- **Aislamiento.**
  - El resultado obtenido al ejecutar concurrentemente las transacciones es un estado del sistema equivalente a uno obtenido al ejecutar una tras otra en algún orden.
  - La responsabilidad de asegurar la propiedad de aislamiento es del **componente de control de concurrencia**.
- **Durabilidad.**
  - Una vez que se completa con éxito una transacción, persisten todas las modificaciones realizadas en la base de datos, incluso si hay un fallo en el sistema después de completarse la ejecución de dicha transacción.
  - La responsabilidad de asegurar la durabilidad es del **componente de gestión de recuperaciones**.

# Gestión de Transacciones

- **Características de una transacción**
  - **Abortada**, cuando una transacción no termina su ejecución con éxito.
  - **Retrocedido**, una vez que se han deshecho los cambios efectuados por la transacción abortada.
  - **Comprometida**, una transacción que termina con éxito.
  - **Transacción compensadora**, forma de deshacer los cambios de una transacción comprometida.

# Gestión de Transacciones

- **Estados de una transacción**
  - **Activa**, estado inicial; la transacción permanece en este estado durante su ejecución.
  - **Parcialmente comprometida**, después de ejecutarse la última instrucción.
  - **Fallida**, tras descubrir que no puede continuar la ejecución normal.
  - **Abortada**, después de haber retrocedido la transacción y restablecido la base de datos a su estado anterior al comienzo de la transacción.
  - **Comprometida**, tras completarse con éxito.

# Gestión de Transacciones

- **Implementación de la atomicidad y la durabilidad**
  - **Copia en la sombra**
  - Copias de la base de datos.
  - En disco se mantiene un puntero llamado puntero\_bd que apunta a la copia actual de la base de datos.
  - Una transacción que quiera actualizar una base de datos crea primero una copia completa de dicha base de datos.
  - Todos los cambios se hacen en la nueva copia de la base de datos dejando la copia original inalterada.
  - Si en cualquier punto hay que abortar la transacción, la copia nueva se borra.
  - La copia antigua de la base de datos no se ve afectada.
  - Si la transacción se completa, en **primer lugar** se consulta al sistema operativo para asegurarse de que todas las páginas de la nueva copia de la base de datos se han escrito en disco.
  - **Después** de terminar esta orden se actualiza el puntero puntero\_bd para que apunte a la nueva copia de la base de datos; la nueva copia se convierte entonces en la copia de la base de datos actual.
  - La copia antigua de la base de datos se borra.
  - Esta implementación es extremadamente ineficiente en el contexto de grandes bases de datos,
  - La implementación no permite a las transacciones ejecutarse concurrentemente unas con otras.



# Gestión de Transacciones

- **Ejecuciones concurrentes**

- **Productividad y utilización de recursos mejoradas.**

- **Productividad** (*throughput*) del sistema, es decir, en el número de transacciones que puede ejecutar en un tiempo dado.
    - **Utilización** del procesador y del disco aumenta también; en otras palabras, el procesador y el disco están menos tiempo desocupados o sin hacer ningún trabajo útil.

- **Tiempo de espera reducido.**

- Si las transacciones operan en partes diferentes de la base de datos es mejor hacer que se ejecuten concurrentemente, compartiendo los ciclos de la UCP y los accesos a disco entre ambas.
    - **Tiempo medio de respuesta:** el tiempo medio desde que una transacción comienza hasta que se completa.

- **Esquemas de control de concurrencia.**

- El sistema de base de datos debe controlar la interacción entre las transacciones concurrentes para evitar que se destruya la consistencia de la base de datos.

- **Planificaciones.**

- Secuencias de ejecución.
    - Representan el orden cronológico en el que se ejecutan las instrucciones en el sistema.

- **Componente de control de concurrencia.**

- Componente del sistema de base de datos que realiza las ejecuciones concurrentes.

# Gestión de Transacciones

- **Secuencialidad**

- **Secuencialidad en cuanto a conflictos.** Si existen operaciones de diferentes transacciones sobre el mismo elemento de datos, y al menos una de esas instrucciones es una operación escribir.
- **Secuencialidad en cuanto a vistas.**
  - Buscan asegurar que cada transacción lee los mismos valores en ambas planificaciones y por tanto realizan los mismos cálculos.
  - Buscan asegurar que ambas planificaciones dan como resultado el mismo estado final del sistema.
- **Escrituras a ciegas.**
  - Cuando se realizan operaciones escribir sin haber realizado ninguna operación leer.

# Gestión de Transacciones

- **Recuperabilidad**

- **Planificación recuperable**, es aquella en la que para todo par de transacciones , tales que  $T_j$  lee elementos de datos que ha escrito previamente  $T_i$ , la operación comprometer de  $T_i$  aparece antes que la de  $T_j$ .
- **Retroceso en cascada**, este fenómeno en el cual un fallo en una única transacción provoca una serie de retrocesos de la transacción.
- **Planificaciones sin cascada**, es deseable restringir las planificaciones a aquellas en las que no puedan ocurrir retrocesos en cascada.

# Gestión de Transacciones

- **Definición de transacciones en SQL**
  - En la norma SQL se especifica el comienzo de una transacción explícitamente.
  - Las transacciones se terminan con una de las instrucciones SQL siguientes:
    - **Commit work** compromete la transacción actual y comienza una nueva.
    - **Rollback work** provoca que la transacción actual aborte.
- **Comprobación de la secuencialidad**
  - Se puede comprobar si una planificación es secuenciable en cuanto a conflictos construyendo el ***grafo de precedencia*** para dicha planificación y viendo que no hay ciclos en el grafo.