



***Universidad Nacional Autónoma  
de México***



***Facultad de Ingeniería***

**Ingeniería en Computación**

**Bases de datos.**

**Tarea 20.**

**Profesor:**

Ing. Fernando Arreola Franco

**Alumno:**

Santiago Martínez Ricardo

**N. Cuenta:**

318187251

**Grupo:**

04

## Tarea 20:

***Investigar los 4 niveles de aislamiento en una base de datos, así como algunos puntos del manejo de concurrencias en un DBMS.***

En realidad, no existen "4 niveles de aislamiento" como una clasificación estándar en las bases de datos. Sin embargo, hay un concepto llamado "ACID" que representa cuatro propiedades fundamentales para garantizar la integridad de las transacciones en las bases de datos. Estas propiedades son: atomicidad, consistencia, aislamiento y durabilidad.

El aislamiento en el contexto de las bases de datos se refiere a la capacidad de una transacción para operar de manera aislada y sin interferir con otras transacciones concurrentes. En otras palabras, el aislamiento garantiza que los cambios realizados por una transacción sean invisibles para otras transacciones hasta que se completen y se confirmen.

Para el aislamiento en las bases de datos, se definen cuatro niveles de aislamiento que proporcionan diferentes grados de control sobre la concurrencia de las transacciones:

1. Nivel de aislamiento READ UNCOMMITTED (No confirmado): Es el nivel más bajo de aislamiento. Permite que una transacción lea datos sin esperar a que se confirmen las modificaciones realizadas por otras transacciones. Esto puede resultar en problemas de consistencia si una transacción lee datos que finalmente se deshacen (rollback) debido a la cancelación de otra transacción.
2. Nivel de aislamiento READ COMMITTED (Confirmado): Este nivel de aislamiento garantiza que una transacción solo lea datos confirmados por otras transacciones. Evita la lectura de datos no confirmados, lo que reduce el riesgo de problemas de consistencia. Sin embargo, aún puede haber problemas de lectura sucia (dirty reads) si una transacción lee datos que luego son modificados o deshechos por otra transacción.
3. Nivel de aislamiento REPEATABLE READ (Lectura repetible): Este nivel de aislamiento asegura que una transacción siempre lea los mismos datos durante su ejecución, incluso si otros cambios ocurren en paralelo. Evita tanto la lectura sucia (dirty reads) como la lectura fantasma (phantom reads), donde una transacción recibe filas adicionales o faltantes en un conjunto de resultados debido a las modificaciones de otras transacciones.
4. Nivel de aislamiento SERIALIZABLE (Serializable): Es el nivel de aislamiento más alto y garantiza que las transacciones se ejecuten de manera secuencial, como si se estuvieran ejecutando una detrás de otra. Esto evita cualquier anomalía de concurrencia y garantiza la consistencia.

completa de los datos, pero puede tener un impacto en el rendimiento debido a la serialización de las transacciones.

Es importante destacar que los diferentes sistemas de gestión de bases de datos pueden implementar estos niveles de aislamiento de manera ligeramente diferente o pueden tener niveles adicionales más específicos.

### ***Manejo de concurrencia.***

El manejo de concurrencia en un DBMS (Sistema de Gestión de Bases de Datos) es un aspecto fundamental para garantizar la integridad y consistencia de los datos en entornos con múltiples transacciones concurrentes. Aquí hay algunos puntos importantes a tener en cuenta:

- **Concurrencia:** La concurrencia se refiere a la capacidad de varias transacciones para acceder y modificar simultáneamente la base de datos. Permite mejorar el rendimiento y la eficiencia del sistema al permitir que múltiples usuarios realicen operaciones de manera concurrente.
- **Problemas de concurrencia:** Las operaciones concurrentes pueden generar problemas como la lectura sucia (dirty reads), donde una transacción lee datos no confirmados por otras transacciones, y la escritura sucia (dirty writes), donde una transacción sobrescribe los datos modificados por otra transacción. También pueden ocurrir fenómenos como la lectura fantasma (phantom reads), donde una transacción obtiene filas adicionales o faltantes en un conjunto de resultados debido a modificaciones de otras transacciones.
- **Bloqueo (Locking):** El bloqueo es una técnica utilizada para gestionar la concurrencia. Los bloqueos se utilizan para garantizar que solo una transacción a la vez pueda acceder o modificar un objeto de la base de datos. Hay diferentes tipos de bloqueos, como el bloqueo exclusivo (write lock) y el bloqueo compartido (read lock), que controlan el acceso concurrente a los datos.
- **Control de concurrencia:** El control de concurrencia implica el uso de algoritmos y mecanismos para gestionar y coordinar las operaciones concurrentes de manera segura. Esto incluye la detección y resolución de conflictos entre transacciones, así como la garantía de la consistencia y aislamiento de las operaciones.
- **Aislamiento:** El aislamiento es una propiedad clave del control de concurrencia que garantiza que una transacción se ejecute de manera aislada y sin interferencias de otras transacciones concurrentes. Los diferentes niveles de aislamiento (como READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ y SERIALIZABLE) determinan el grado de aislamiento que se proporciona a las transacciones.

- Transacciones atómicas: Las transacciones deben ser atómicas, lo que significa que se deben ejecutar como una unidad indivisible. Si una transacción falla o se cancela en medio de su ejecución, los cambios realizados hasta ese momento deben deshacerse (rollback) para mantener la integridad de los datos.
- Optimización de concurrencia: Los DBMS suelen incluir optimizadores de concurrencia que buscan maximizar el rendimiento y la eficiencia al tomar decisiones sobre el acceso concurrente a los datos. Esto implica considerar factores como el tiempo de espera, la escalabilidad y la detección y resolución de conflictos.
- Control de versiones: Algunos sistemas de gestión de bases de datos utilizan el control de versiones para manejar la concurrencia. En lugar de bloquear los objetos de la base de datos, se mantienen múltiples versiones de los datos y las transacciones pueden trabajar con versiones específicas, evitando así conflictos y mejorando el rendimiento.
- Recuperación ante fallas: Además del manejo de concurrencia, un DBMS debe tener mecanismos adecuados para la recuperación ante fal.

## Referencias

- Ozsú, M. T., Valduriez, P. (2011). Principles of Distributed Database Systems (3rd ed.). Springer.
- Silberschatz, A., Korth, H. F., Sudarshan, S. (2010). Database System Concepts (6th ed.). McGraw-Hill.