



Módulo 7: Práctica

Unidad 7 Trabajo Práctico

Alumna: Melisa Martellini

1. ¿Qué es la integridad de los datos?

La integridad asegura que los datos sean válidos y coherentes, y que cumplan con las reglas definidas en el esquema de la base

Tipos de restricciones:

- **Integridad de entidad:** Cada fila debe poder identificarse de forma única, mediante una clave primaria
- **Integridad referencial:** Impide que existan referencias a registros que no existen en otra tabla. Se logra mediante claves foráneas (foreign keys)
- **Integridad de dominio:** Controla los valores permitidos para una columna, a través del tipo de dato, restricciones NOT NULL, CHECK, etc

2. Consistencia

La consistencia garantiza que la base de datos pase siempre de un estado válido a otro estado válido, cumpliendo las restricciones definidas.

3. ¿Qué se entiende por **disponibilidad de la base de datos**? Describa los aspectos clave.

La disponibilidad se refiere a que la base de datos esté accesible y operativa en todo momento para los usuarios y aplicaciones

- Caídas del servidor
- Problemas de red
- Contención de recursos (por transacciones largas o mal gestionadas)

Parte 2: Transacciones y Propiedades ACID

1. ¿Qué es una transacción?

Una transacción en MySQL es una unidad lógica de trabajo que agrupa una o más operaciones SQL. Todas las operaciones de la transacción deben ejecutarse de forma completa (COMMIT) o no ejecutarse en absoluto (ROLLBACK), manteniendo la base de datos en un estado válido

2. Propiedades ACID

- **Atomicidad:** toda la transacción se ejecuta como una unidad indivisible. Si una parte falla, todo se revierte. Ejemplo: Se debita una cuenta pero no se acredita la otra
- **Consistencia:** La base de datos pasa de un estado válido a otro también válido. Ejemplo: Se inserta un alumno con carrera inexistente
- **Aislamiento:** Cada transacción actúa como si fuera la única en el sistema. Ejemplo Dos usuarios modifican el mismo saldo simultáneamente
- **Durabilidad:** Una vez confirmado un cambio, permanece en la base de datos, incluso si hay una caída del sistema. Ejemplo Corte de energía sin guardar cambios

3. BEGIN, COMMIT, ROLLBACK

- BEGIN: inicia la transacción
- COMMIT: confirma los cambios
- ROLLBACK: revierte todos los cambios si ocurre un error

Parte 3: Control de Concurrencia

1. ¿Qué es la concurrencia?

La concurrencia es la capacidad de ejecutar múltiples transacciones simultáneamente. Aunque mejora el rendimiento, puede ocasionar errores si no se controla correctamente.

2. Problemas comunes

- **Lecturas sucias:** Una transacción lee datos modificados por otra transacción que aún no se confirmó
- **Lecturas no repetibles:** Una transacción lee un mismo dato dos veces, y obtiene distintos resultados porque otra transacción lo modificó entre lecturas.
- **Lecturas fantasmas:** Una transacción ejecuta una consulta y luego, al repetirla, aparecen nuevos registros que antes no estaban, debido a otra transacción concurrente
- **Actualizaciones perdidas:** Dos transacciones modifican el mismo dato. Una sobrescribe los cambios de la otra sin saberlo.

3. Niveles de aislamiento

Nivel	Dirty Read	Non-repeatable Read	Phantom Read
READ UNCOMMITTED	✓ Sí	✓ Sí	✓ Sí
READ COMMITTED	✗ No	✓ Sí	✓ Sí
REPEATABLE READ (*)	✗ No	✗ No	✓ Sí
SERIALIZABLE	✗ No	✗ No	✗ No

4. Describa los **tres mecanismos de control de concurrencia** mencionados en el texto (*Bloqueo, Ordenamiento por marca de tiempo, Control de concurrencia de múltiples versiones*).

MySQL/InnoDB utiliza bloqueos a nivel de fila por defecto, lo que permite mayor concurrencia. Tipos:

- Bloqueo compartido (S): permite que otros lean pero no escriban.
- Bloqueo exclusivo (X): impide cualquier acceso de otras transacciones.

Control por marcas de tiempo (Timestamp Ordering):

Aunque no es el método principal en MySQL, el orden lógico de ejecución puede lograrse por medio de control de versiones y marcas de tiempo internas para decidir si una transacción debe esperar, abortar o continuar

MVCC (Control de concurrencia multiversión)

MySQL (con InnoDB) usa MVCC para permitir que las transacciones lean una versión consistente de los datos, sin bloquear.

- Las transacciones ven los datos como estaban al momento de comenzar, incluso si otro usuario los modifica después.
- Mejora el rendimiento porque permite lectura sin bloqueo.

5. ¿Qué es un bloqueo mutuo (*deadlock*) y cómo se resuelven?

Un deadlock ocurre cuando dos o más transacciones se bloquean mutuamente esperando recursos que la otra no libera. Ninguna puede avanzar

MySQL detecta automáticamente los deadlocks.

Cancela una de las transacciones y lanza un error: ERROR 1213 (40001): Deadlock found.

Parte 4: Aplicaciones y Investigación

1. Identifique y explique **tres aplicaciones del mundo real** donde las transacciones son críticas para mantener la integridad y consistencia de los datos.

- Transferencias de dinero entre cuentas.

Una transacción debe asegurar que si se descuenta dinero de una cuenta, el mismo monto se acredita en la otra. Si ocurre un fallo en medio del proceso, el sistema debe revertir la operación para evitar inconsistencias

-Procesamiento de pedidos y pagos en línea.

Cuando un cliente realiza una compra, la transacción debe incluir la reducción del stock, el registro del pago y la confirmación del pedido.

- Reserva de un vuelo

Una transacción asegura que un lugar en el vuelo no sea asignado a más de un cliente.

2. Investigación

Protocolos Pesimistas: previenen el problema presuponiendo que va a ocurrir.

Protocolos Optimistas: considerar el problema después que ocurrió

Cita en APA: Vitturini, M. M. (2015). *Elementos de bases de datos. Clase 16: Manejo de transacciones IV*. Universidad Nacional del Sur, Departamento de Ciencias e Ingeniería de la Computación. Recuperado de https://cs.uns.edu.ar/~wmg/ebd15/downloads/Teoria/EBD_16_2015_Manejo_de_Transacciones_IV.pdf

- MySQL cuenta con las características ACID solo cuando se usa con motores de almacenamiento o módulos de software en clústeres de InnoDB y NDB
- PostgreSQL brinda las características ACID en todas las configuraciones.

Cita en APA: Amazon Web Services. (s.f.). *Diferencia entre MySQL y PostgreSQL*. AWS. Recuperado de <https://aws.amazon.com/es/compare/the-difference-between-mysql-vs-postgresql/>

