



Integridad de la base de datos y transacciones: garantizar la confiabilidad de los datos

Bienvenido al Módulo 7 de Base de Datos I. En esta presentación, exploraremos cómo las bases de datos mantienen la corrección, disponibilidad y consistencia de los datos incluso cuando múltiples usuarios acceden a ellas simultáneamente.

Agenda

Introducción a la integridad

Entender por qué la integridad de los datos es importante en los sistemas de bases de datos

Conceptos clave

Fundamentos de integridad, consistencia y disponibilidad

Transacciones y ACID

Los bloques de construcción de operaciones de base de datos confiables

Control de concurrencia

Gestionar múltiples usuarios que acceden a los datos simultáneamente

Aplicaciones del mundo real

Ejemplos prácticos y su importancia en los sistemas de producción

Objetivos del módulo



Al final de este módulo, usted:

- Entenderá los conceptos fundamentales de la integridad, consistencia y concurrencia de la base de datos
- Explicará cómo funcionan las transacciones y por qué son esenciales
- Describirá las propiedades ACID y su importancia
- Identificará problemas y soluciones de concurrencia comunes
- Aplicará estos conceptos a escenarios de base de datos del mundo real

¿Qué es la integridad de los datos?

La integridad de los datos se refiere a la precisión, coherencia y confiabilidad de los datos a lo largo de su ciclo de vida.

Asegura que los datos en una base de datos:

- Siguen las reglas y restricciones definidas
- Mantengan relaciones lógicas
- Eviten modificaciones no autorizadas
- Permanezcan en un estado válido en todo momento



Tipos de restricciones de integridad



Integridad de entidad

Garantiza que cada fila de una tabla sea identificable de forma única. Las claves principales deben ser únicas y no pueden contener valores NULL.

Ejemplo: El ID de estudiante debe ser único para cada registro de estudiante.



Integridad referencial

Mantiene las relaciones entre las tablas. Las claves foráneas deben coincidir con una clave principal existente o ser NULL.

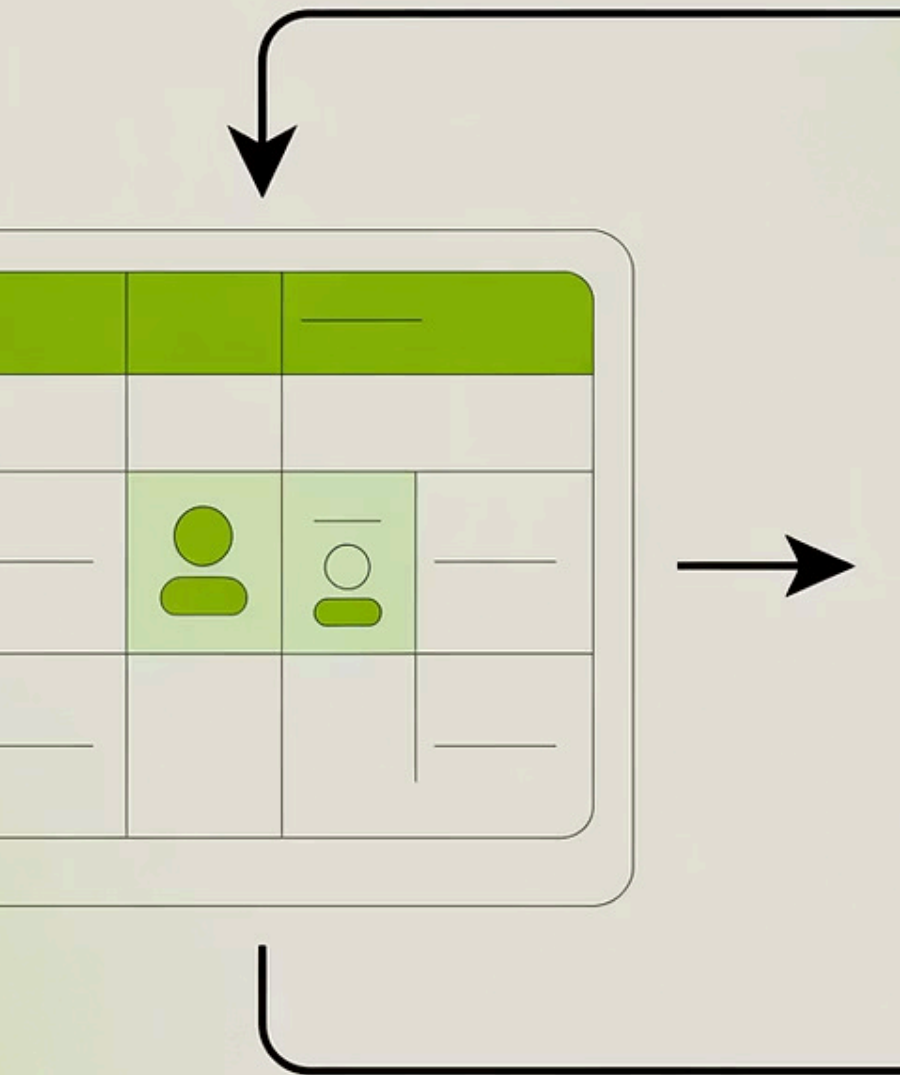
Ejemplo: Las matrículas de cursos deben hacer referencia a IDs de estudiantes válidos.



Integridad de dominio

Garantiza que los valores de una columna se ajusten a las restricciones definidas sobre el tipo de datos, el formato o el rango.

Ejemplo: La edad del estudiante debe ser un entero positivo entre 16 y 99.



Ejemplo visual: Integridad de la entidad

En esta tabla de estudiantes, la restricción de clave principal en ID de estudiante asegura:

- Cada estudiante tiene un identificador único (no se permiten duplicados)
- No se permiten valores NULL en el campo de ID
- La base de datos rechaza automáticamente los intentos de violar estas reglas

Entendiendo la Consistencia

¿Qué es la Consistencia de Base de Datos?

La consistencia asegura que una base de datos transite de un estado válido a otro estado válido, siguiendo todas las reglas y restricciones definidas.

Cuando decimos que una base de datos es consistente, queremos decir:

- Todas las restricciones de integridad se cumplen
- Las relaciones entre los datos permanecen válidas
- Las reglas de negocio se aplican



La consistencia es como una balanza equilibrada - todas las relaciones de datos deben permanecer en equilibrio después de cualquier operación.

Ejemplo de consistencia: Transferencia bancaria



Estado inicial

Cuenta A: \$1000

Cuenta B: \$500

Total: \$1500



Transferir \$300

A → B

Deducir de A

Agregar a B



Estado final

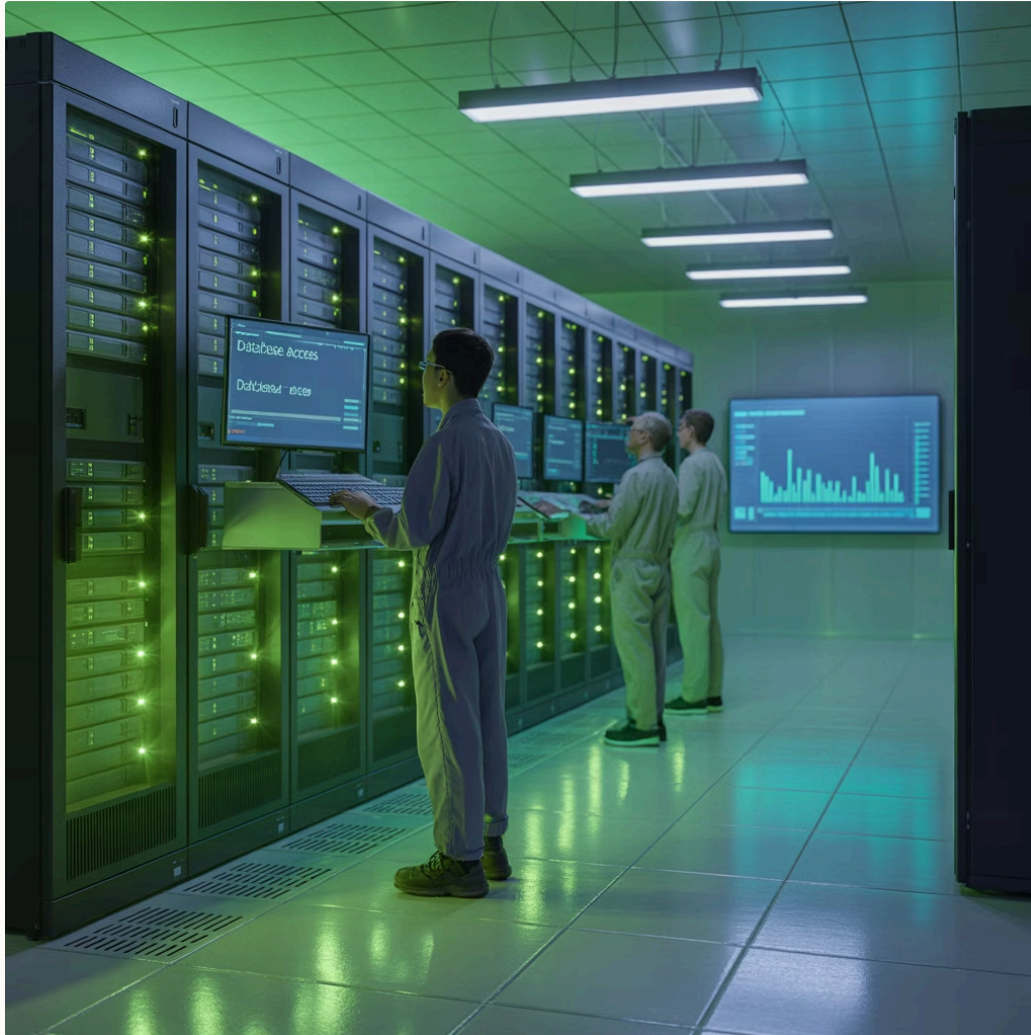
Cuenta A: \$700

Cuenta B: \$800

Total: \$1500

La consistencia asegura que la cantidad total de dinero en el sistema permanezca igual (\$1500) después de la transferencia. El dinero no puede desaparecer ni crearse durante la operación.

Disponibilidad de la base de datos



¿Qué es la disponibilidad de la base de datos?

La disponibilidad se refiere a la capacidad de la base de datos para permanecer accesible y operativa incluso cuando múltiples usuarios la están accediendo simultáneamente.

Aspectos clave:

- Tiempo de actividad y confiabilidad del sistema
- Tiempo de respuesta bajo carga
- Capacidad para manejar acceso concurrente
- Recuperación de fallas

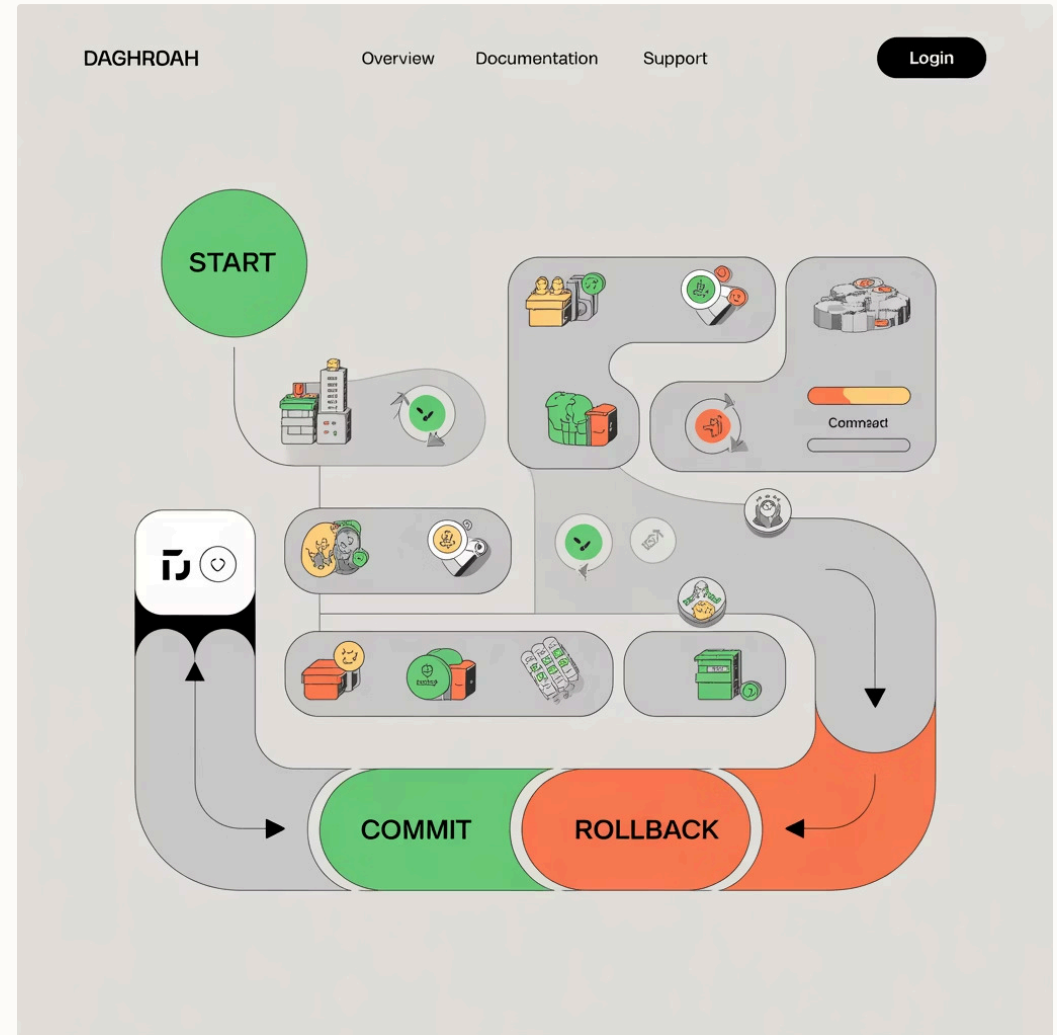
Introducción a las transacciones

¿Qué es una transacción de base de datos?

Una transacción es una unidad lógica de trabajo que contiene una o más operaciones de base de datos (lecturas/escrituras).

Características clave:

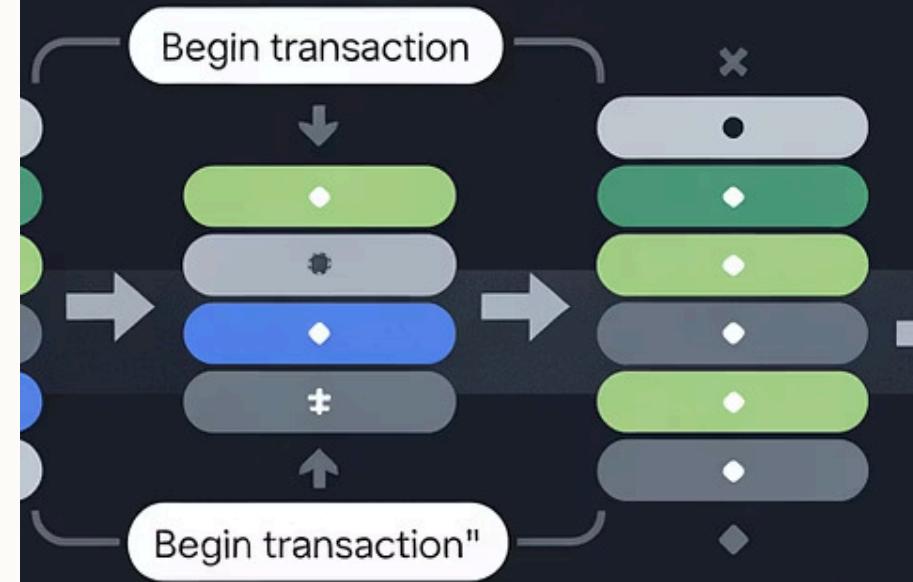
- Se ejecuta como una sola unidad indivisible
- Todas las operaciones se completan con éxito (confirmación) o ninguna lo hace (reversión)
- Mantiene la consistencia de la base de datos
- Aísla las operaciones concurrentes



Ejemplo de transacción en SQL

```
COMENZAR;  
ACTUALIZAR cuentas ESTABLECER saldo = saldo - 500  
DONDE id = 1;  
  
ACTUALIZAR cuentas ESTABLECER saldo = saldo + 500  
DONDE id = 2;  
CONFIRMAR;
```

Si cualquier parte falla (por ejemplo, la cuenta n.º 2 no existe), podemos usar DESHACER en lugar de CONFIRMAR para deshacer todos los cambios y mantener la coherencia.



Las propiedades ACID

Atomicidad

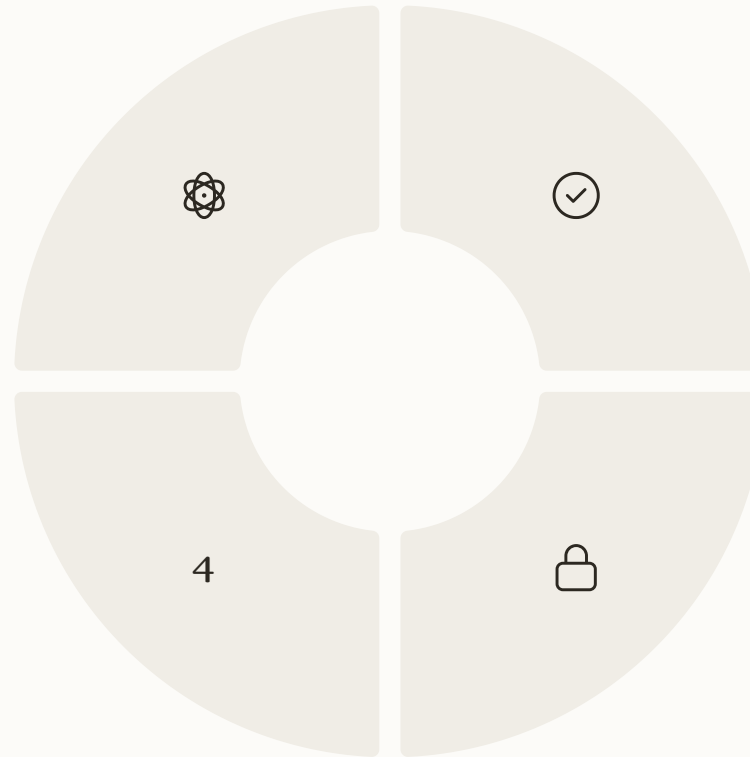
Todas las operaciones en una transacción tienen éxito o fallan. No hay transacciones parciales.

Al igual que un átomo, no se puede descomponer más.

Durabilidad

Una vez que se confirma una transacción, sus efectos persisten incluso a través de fallas del sistema.

Los cambios se almacenan permanentemente en memoria no volátil.



Consistencia

Una transacción transforma la base de datos de un estado válido a otro estado válido.

Se aplican todas las reglas, restricciones y disparadores.

Aislamiento

Las transacciones concurrentes se ejecutan como si fueran secuenciales.

Una transacción no puede ver los estados intermedios de otra transacción.

Análisis en profundidad: Atomicidad

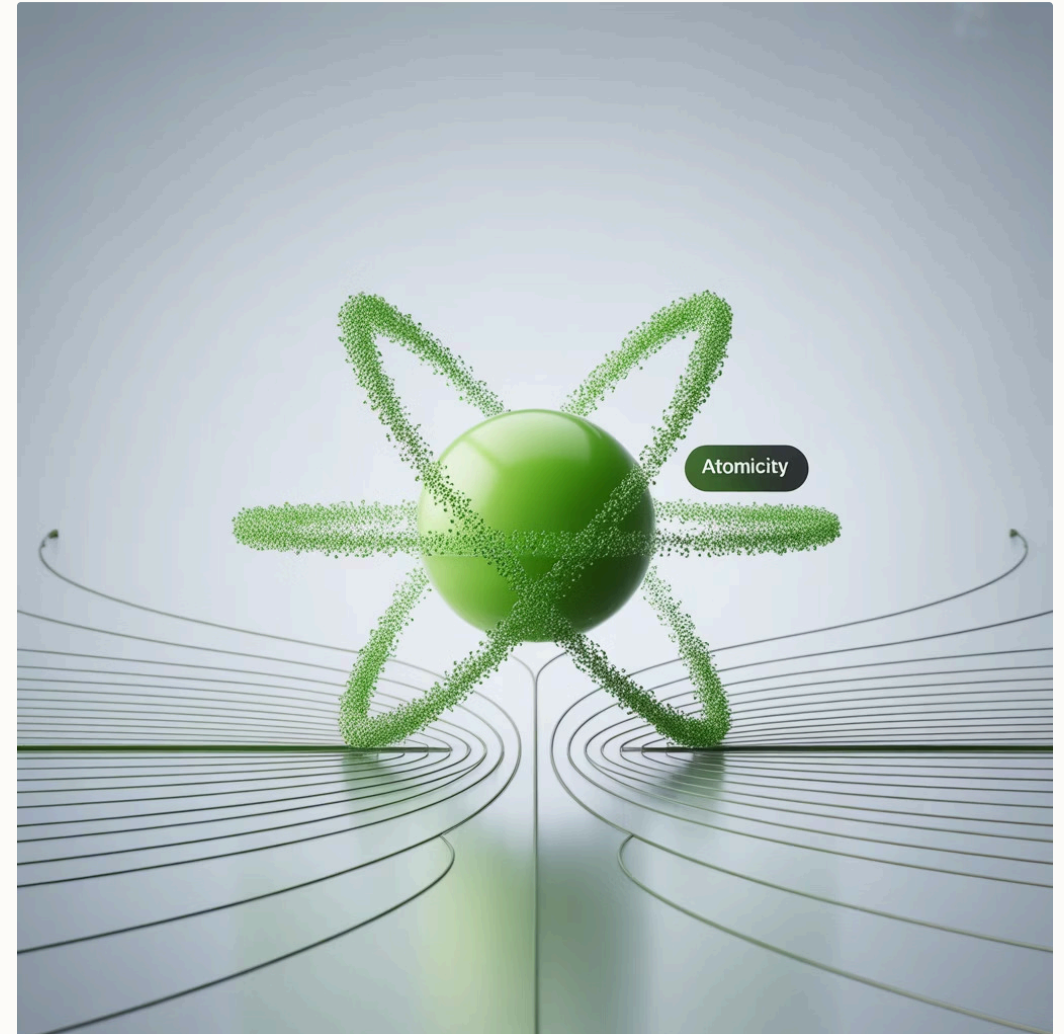
La atomicidad garantiza que una transacción se trate como una unidad de trabajo indivisible. Debe:

- Completarse por completo (todas las operaciones tienen éxito)
- Fallar por completo (ninguna operación surte efecto)

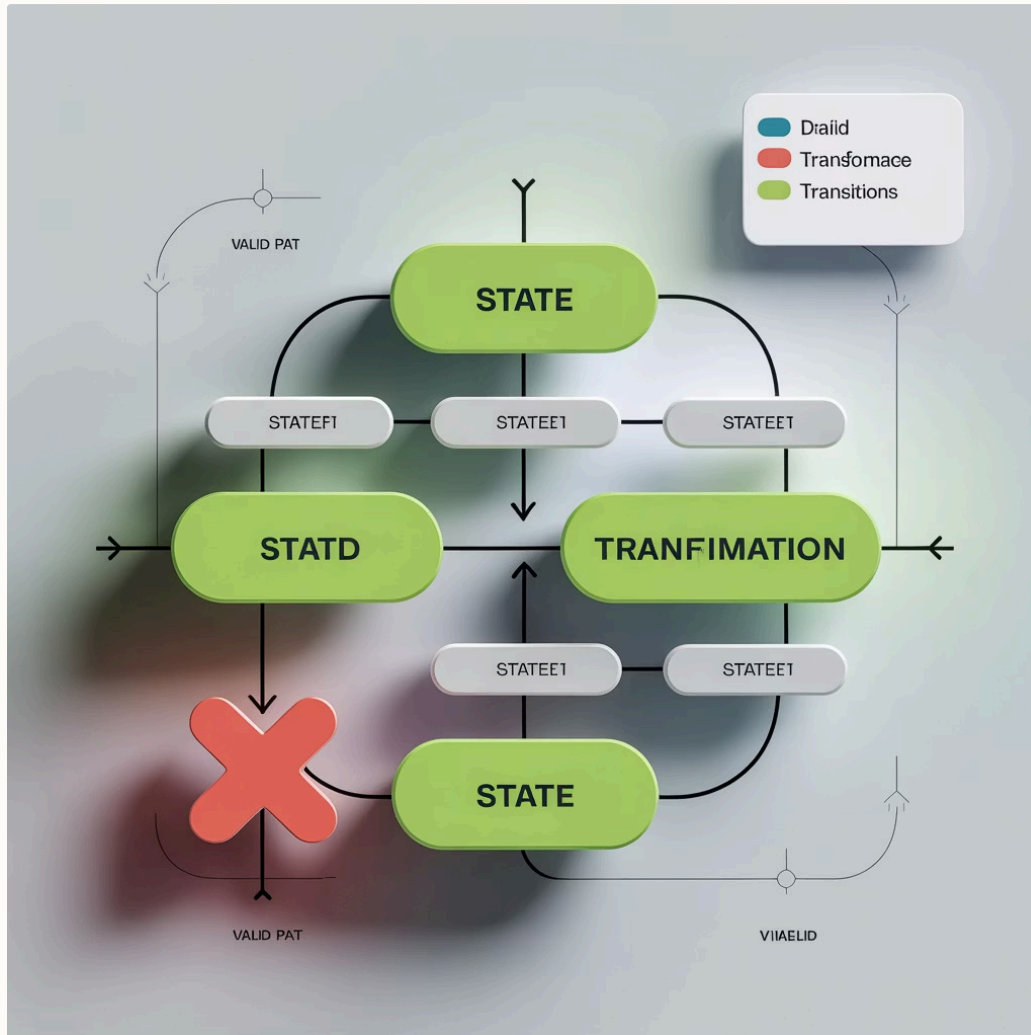
El sistema de base de datos realiza un seguimiento de todas las operaciones dentro de una transacción y puede "deshacer" si es necesario.

Mecanismos de implementación:

- Registro anticipado de escritura
- Paginación de sombra
- Segmentos de reversión



Análisis en profundidad: Consistencia



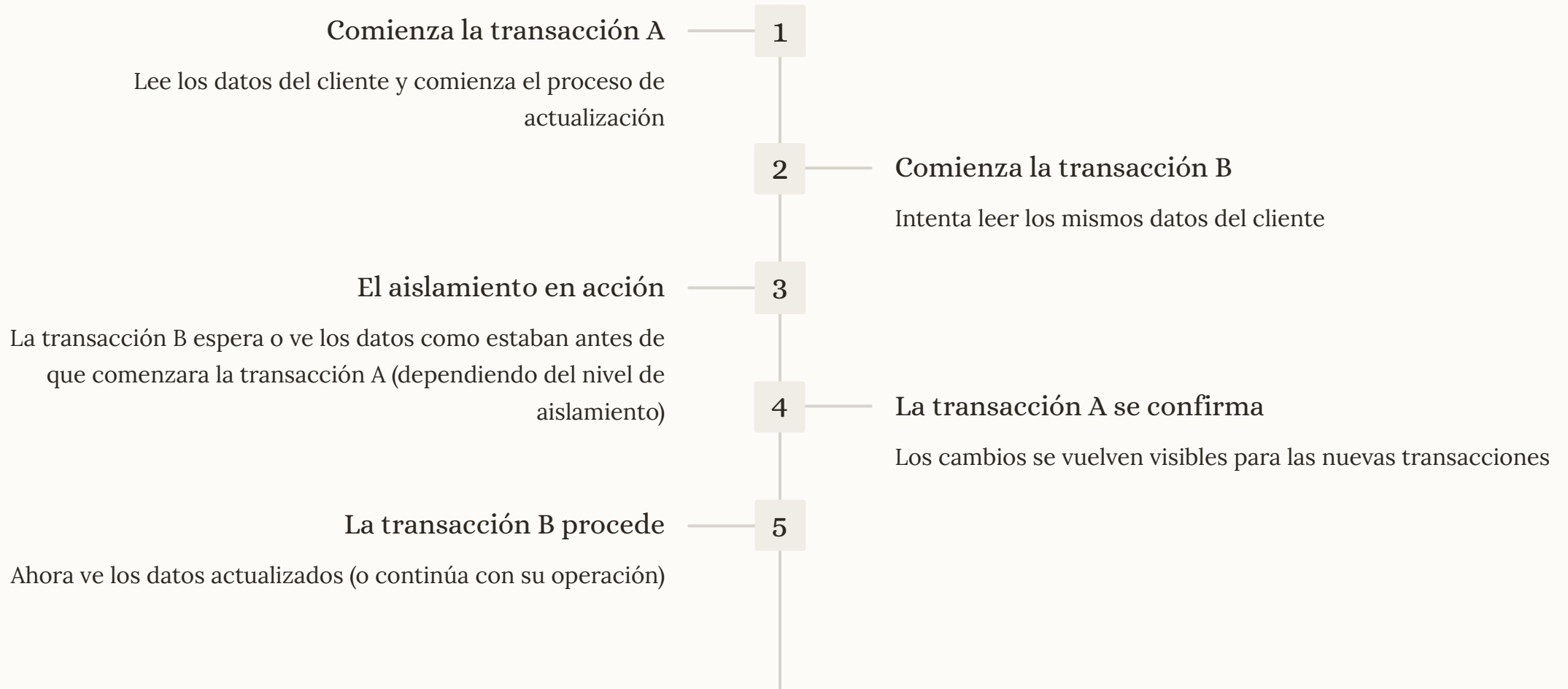
La consistencia garantiza que las transacciones solo transformen los datos de un estado válido a otro estado válido.

Esto significa:

- Se preservan todas las restricciones de integridad
- Se aplican las reglas de negocio definidas
- Se mantiene la integridad referencial
- Se respetan las restricciones personalizadas

El sistema de base de datos verifica la consistencia tanto durante la ejecución de la transacción como en el momento del compromiso.

Análisis en profundidad: Aislamiento



El aislamiento evita la interferencia entre las transacciones concurrentes, haciendo que parezcan ejecutarse secuencialmente incluso cuando se ejecutan simultáneamente.

Análisis en profundidad: Durabilidad

Las garantías de durabilidad aseguran que una vez que se ha confirmado una transacción, sus cambios son permanentes y sobrevivirán a:

- Fallos del sistema
- Cortes de energía
- Fallos de hardware
- Otros tipos de fallos del sistema

Implementado a través de:

- Registros de transacciones
- Puntos de control de la base de datos
- Escrituras en almacenamiento no volátil



Introducción a la Concurrency



¿Qué es la Concurrency?

La concurrencia se refiere a múltiples usuarios o procesos que acceden y potencialmente modifican los mismos datos simultáneamente.

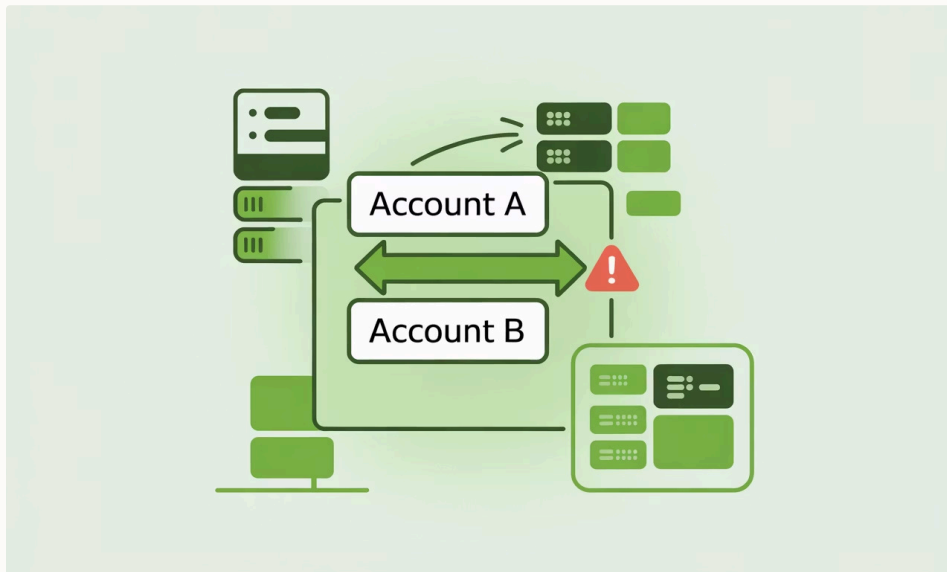
Desafíos en entornos concurrentes:

- Mantener la integridad de los datos
- Evitar interferencias entre transacciones
- Equilibrar el aislamiento con el rendimiento
- Evitar bloqueos y problemas de vivacidad

Problemas de concurrencia

1 Lecturas sucias

La transacción A lee datos que han sido modificados pero aún no confirmados por la transacción B. Si la transacción B se deshace, la transacción A ha leído datos no válidos.



3 Lecturas fantasma

La transacción A ejecuta una consulta dos veces y obtiene un conjunto de filas diferente porque la transacción B ha insertado o eliminado filas que coinciden con la condición de la consulta.

2 Lecturas no repetibles

La transacción A lee los mismos datos dos veces pero obtiene resultados diferentes porque la transacción B modificó y confirmó los datos entre las lecturas.

4 Actualizaciones perdidas

La transacción A y la transacción B leen y actualizan los mismos datos, y una de ellas sobrescribe los cambios de la otra sin incorporarlos.

Niveles de aislamiento

Nivel de aislamiento	Lecturas sucias	Lecturas no repetibles	Lecturas fantasma	Rendimiento
READ UNCOMMITTED	Posible	Posible	Posible	Más alto
READ COMMITTED	Prevenido	Posible	Posible	Alto
REPEATABLE READ	Prevenido	Prevenido	Posible	Medio
SERIALIZABLE	Prevenido	Prevenido	Prevenido	Más bajo

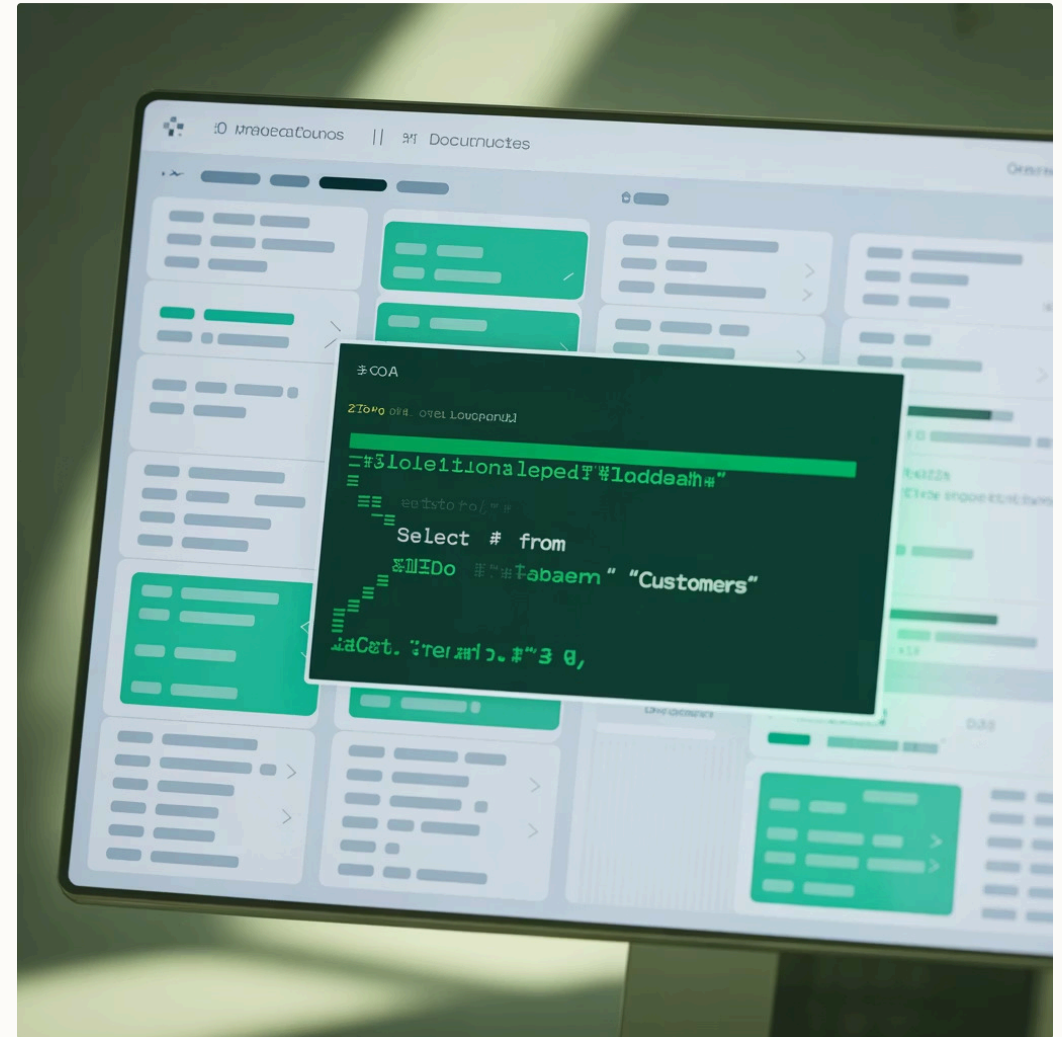
Los niveles de aislamiento más altos brindan más protección, pero a costa del rendimiento. El nivel apropiado depende de los requisitos específicos de su aplicación.

Establecer niveles de aislamiento en SQL

```
-- Establecer el nivel de aislamiento para la transacción actual
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;

-- Iniciar la transacción
BEGIN;
  SELECT * FROM accounts WHERE id = 1;
  -- Más operaciones...
COMMIT;
```

Los diferentes sistemas de bases de datos pueden tener una sintaxis ligeramente diferente, pero el concepto es el mismo.



El nivel de aislamiento determina qué tipos de anomalías de concurrencia se previenen durante la ejecución de la transacción.

Mecanismos de control de concurrencia



Bloqueo

Restringe el acceso a los elementos de datos que están siendo utilizados por otras transacciones.

- Bloqueos compartidos (lectura)
- Bloqueos exclusivos (escritura)
- A nivel de fila o a nivel de tabla



Ordenamiento por marca de tiempo

Asigna marcas de tiempo a las transacciones y ejecuta las operaciones en función del orden de las marcas de tiempo.

- Previene conflictos
- Garantiza una ejecución serializable



Control de concurrencia de múltiples versiones

Mantiene múltiples versiones de los elementos de datos para permitir operaciones de lectura sin bloqueo.

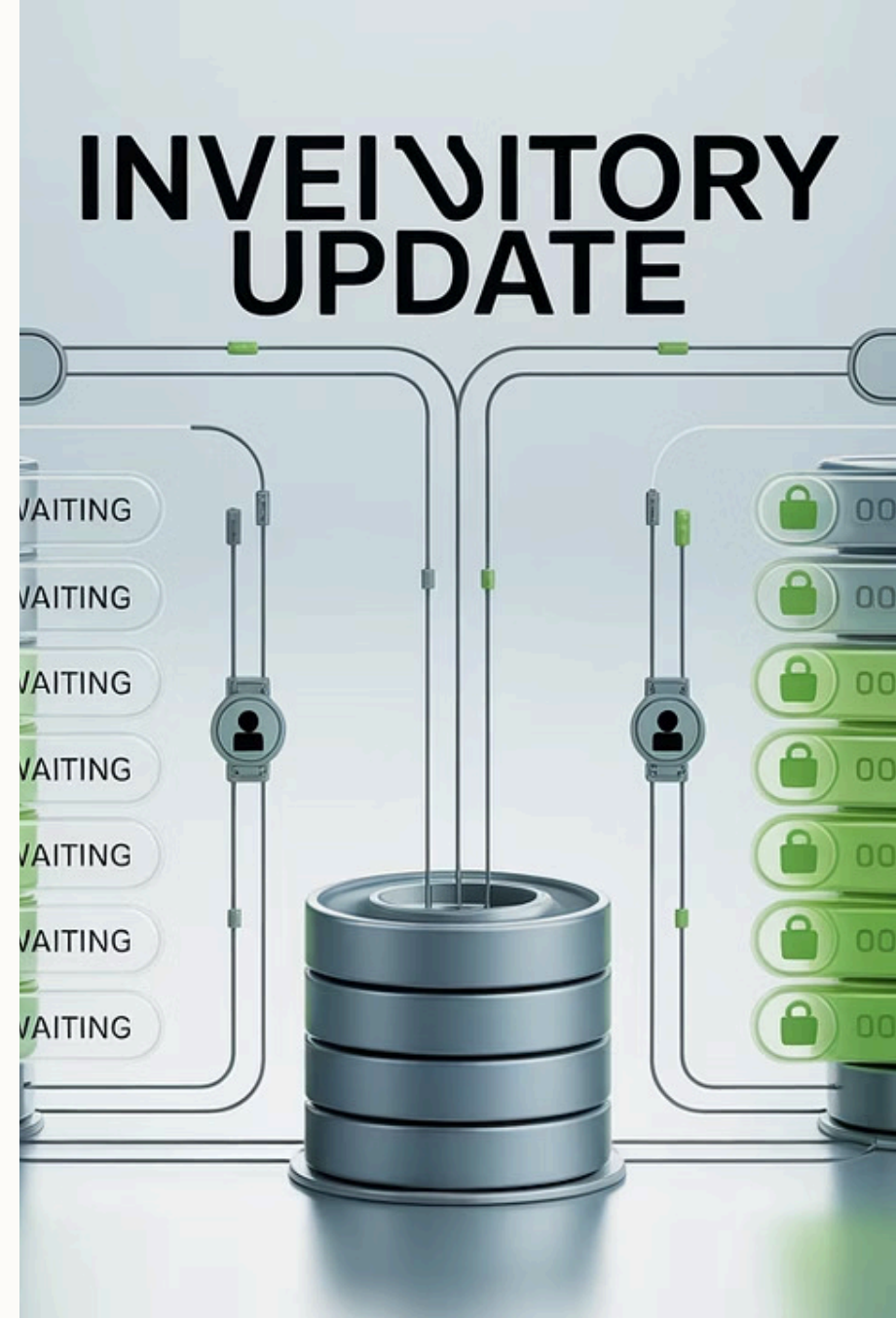
- Mayor concurrencia de lectura
- Aislamiento de instantánea

Ejemplo visual: el bloqueo en acción

En este escenario:

1. El usuario A comienza a actualizar el inventario de productos y adquiere un bloqueo exclusivo
2. El usuario B intenta actualizar el mismo producto, pero debe esperar
3. Una vez que el usuario A confirma su transacción, se libera el bloqueo
4. Ahora el usuario B puede proceder con su actualización

Esto evita el problema de la actualización perdida y asegura la coherencia de los datos.



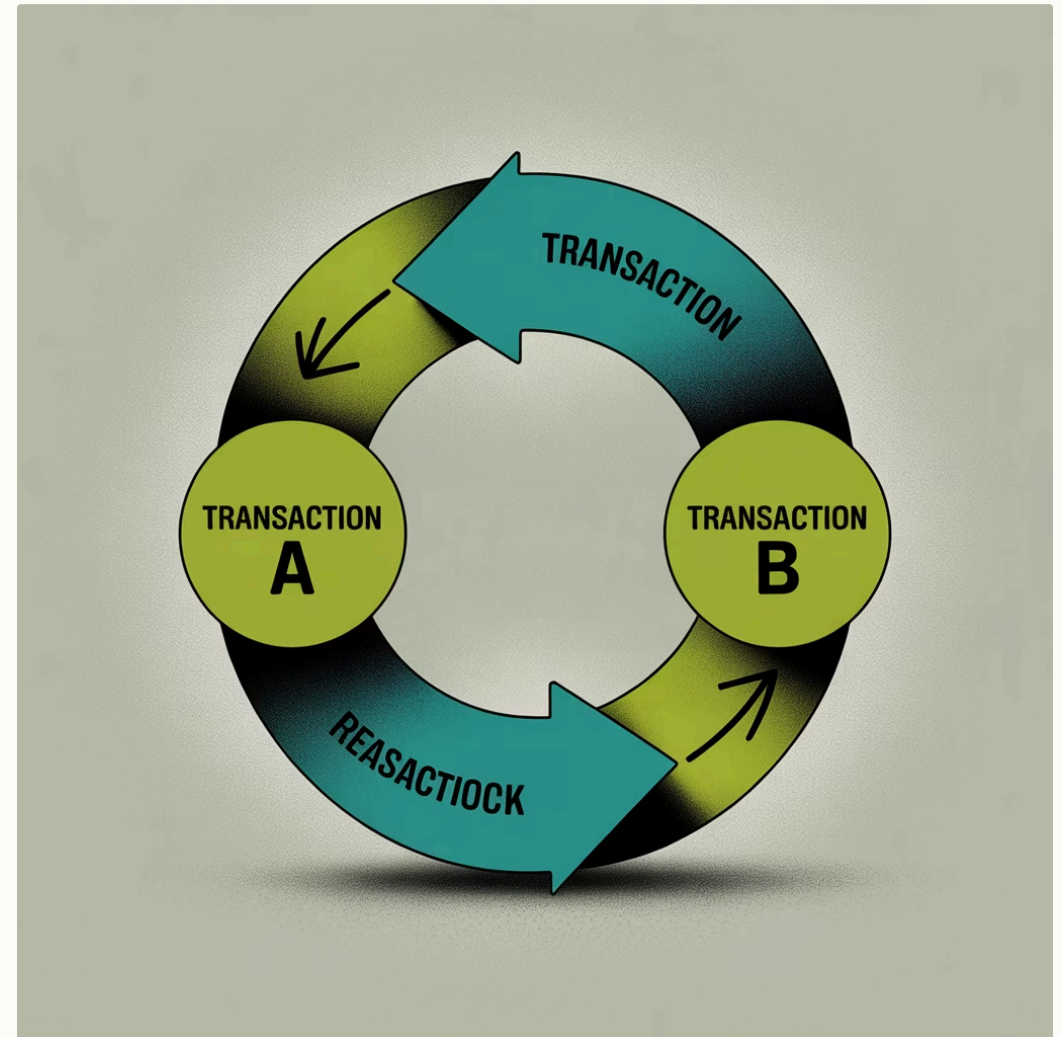
Bloqueos: Un desafío de concurrencia

¿Qué es un bloqueo?

Un bloqueo ocurre cuando dos o más transacciones están esperando que la otra libere los bloqueos, lo que resulta en un punto muerto donde ninguna puede proceder.

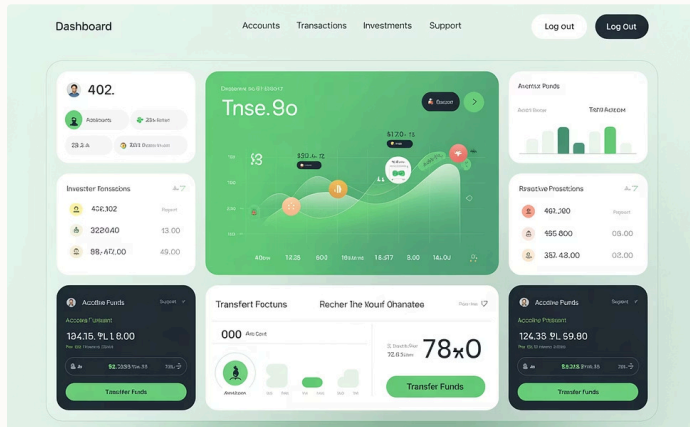
Resolución de bloqueos:

- Algoritmos de detección de bloqueos
- Selección de víctimas (qué transacción abortar)
- Retroceso y reintento automáticos
- Mecanismos de tiempo de espera



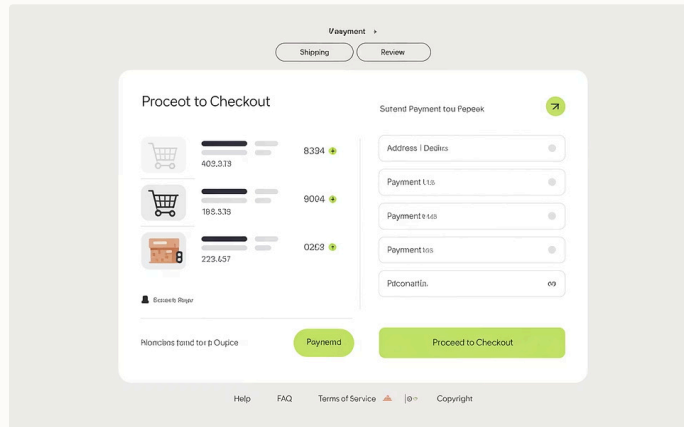
La Transacción A mantiene el Recurso 1 y espera el Recurso 2, mientras que la Transacción B mantiene el Recurso 2 y espera el Recurso 1. Ninguna puede proceder.

Aplicaciones del mundo real de las transacciones



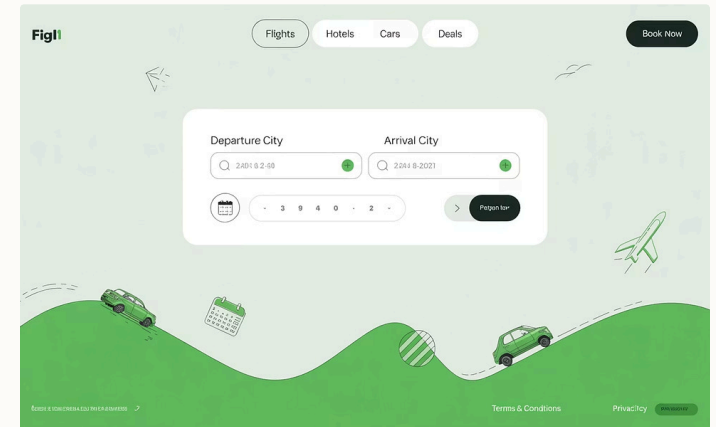
Sistemas bancarios

Cada transferencia de fondos debe ser atómica y duradera. Si un débito tiene éxito pero un crédito falla, el sistema debe revertirse para evitar que el dinero desaparezca.



Comercio electrónico

El procesamiento de pedidos implica actualizar el inventario, los registros de clientes y la información de pago como una sola transacción para mantener la coherencia.



Sistemas de reserva

Múltiples usuarios que reservan recursos limitados (asientos, habitaciones) requieren aislamiento para evitar la sobrecontratación y garantizar un acceso justo.

Puntos clave

1 Integridad y consistencia

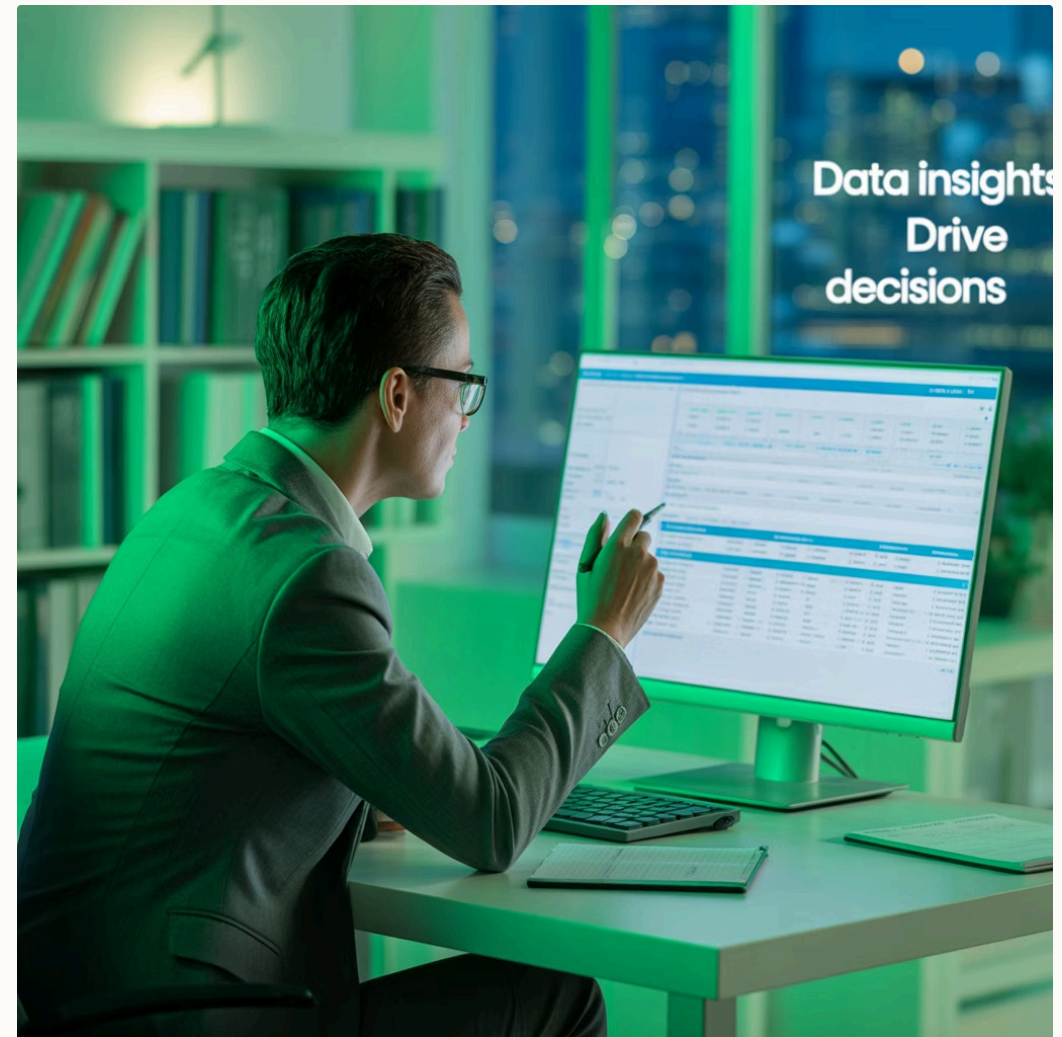
La integridad de la base de datos garantiza que los datos sigan todas las reglas definidas, mientras que la consistencia garantiza transiciones de estado válidas.

2 Transacciones y ACID

Las transacciones agrupan las operaciones como unidades atómicas, con las propiedades ACID que garantizan la confiabilidad en todas las circunstancias.

3 Control de concurrencia

Los niveles de aislamiento y los mecanismos de bloqueo adecuados evitan anomalías de datos cuando varios usuarios acceden a los datos simultáneamente.



Próximos pasos:

- Practicar la implementación de transacciones en SQL
- Explorar los niveles de aislamiento en su sistema de base de datos
- Analizar escenarios de transacciones del mundo real