

Recuperación y control de concurrencia

Alex Di Genova

06/06/2022

Contenidos

- Repaso (Transacciones y ACID)
- Recuperación y control de concurrencia.

Transacciones

Transacciones

- DDL (Data Definition Language)
 - CREATE TABLE...
- DML (Data Manipulation Language)
 - SELECT ...
- TCL (Transaction Control Language)
 - Las transacciones son una parte fundamental de cómo las bases de datos relacionales protegen la integridad y confiabilidad de los datos que contienen.
 - Las transacciones se utilizan automáticamente en todos los comandos DDL y DML.
- Transacción?
 - Una transacción puede involucrar la actualización de un solo valor hasta un procedimiento que puede insertar múltiples valores en múltiples tablas (**update**).
 - Es la unidad básica de cambio en BD (Operaciones parciales no son permitidas)
 - Si dos clientes cambian el mismo registro al mismo tiempo?
 - (Control de concurrencia)
 - Si estamos realizando una transferencia entre bancos y se corta la luz. ¿Cuál es el estado de la base de datos?
 - (Durabilidad, recuperación, consistencia)

Transacciones

Realizar una transferencia de dinero

- Como podemos tranferir fondos de manera segura a otra cuenta bancaria.

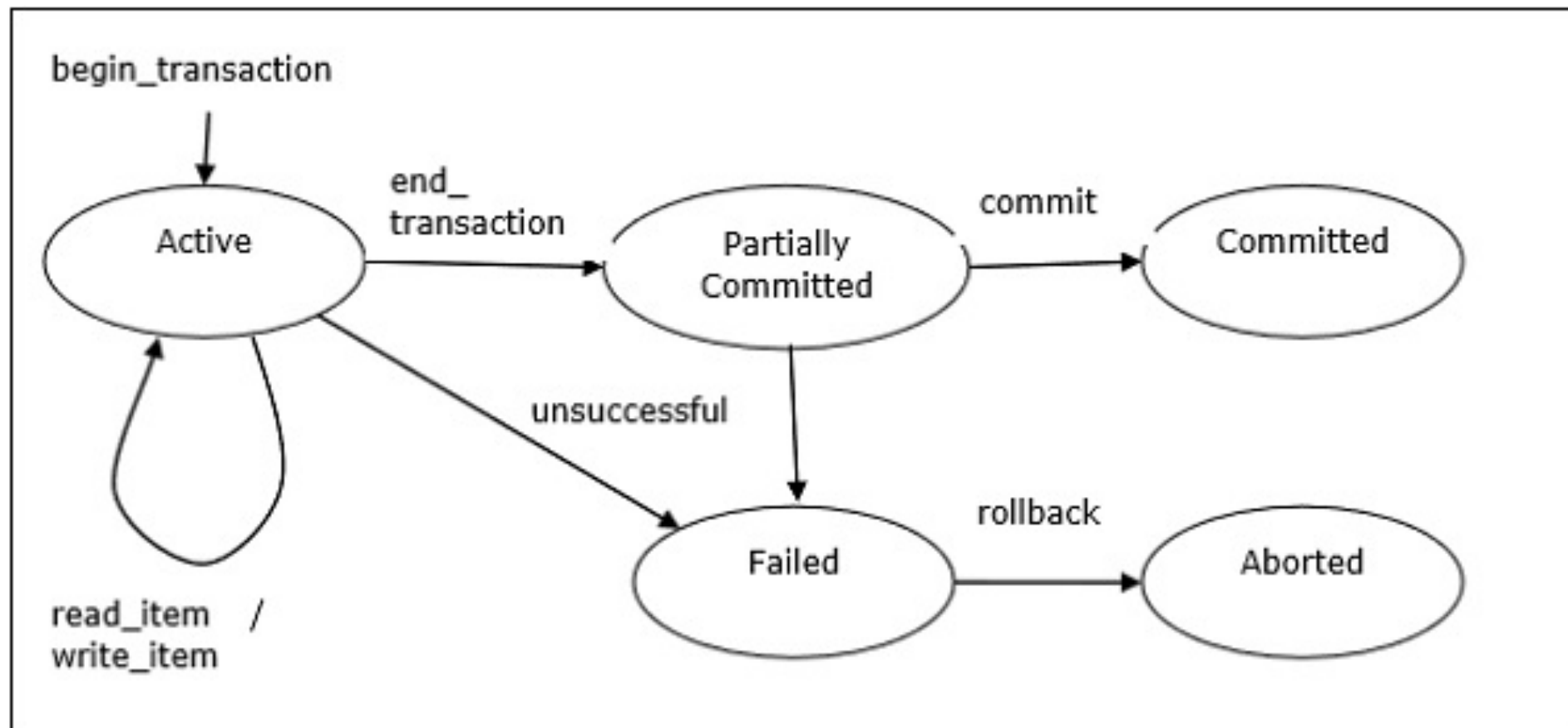
Pasos	Cliente1	Cliente2
Chequear	X	-
Subtraer	X	-
Agregar	-	X



- Todos los pasos deberían tener éxito por completo, dando como resultado que el saldo se transfiera correctamente, o los tres pasos deberían fallar por completo, dando como resultado que ambas cuentas no se modifiquen.
- Cualquier otro resultado, donde un paso tiene éxito y el otro falla, conlleva a un error.

Transacciones

Etapas de una transacción



- **BEGIN**
 - **COMMIT** (transacción ok) -> DBMS almacena todos los cambios
 - **ABORT** (transacción falla) -> todos los cambios se deshacen de modo que es como si el txn nunca se ejecutó en absoluto.
- **El concepto de transacción, se utiliza para representar una unidad lógica de procesamiento de base de datos que debe completarse en su totalidad para garantizar su exactitud.**

Propiedades ACID

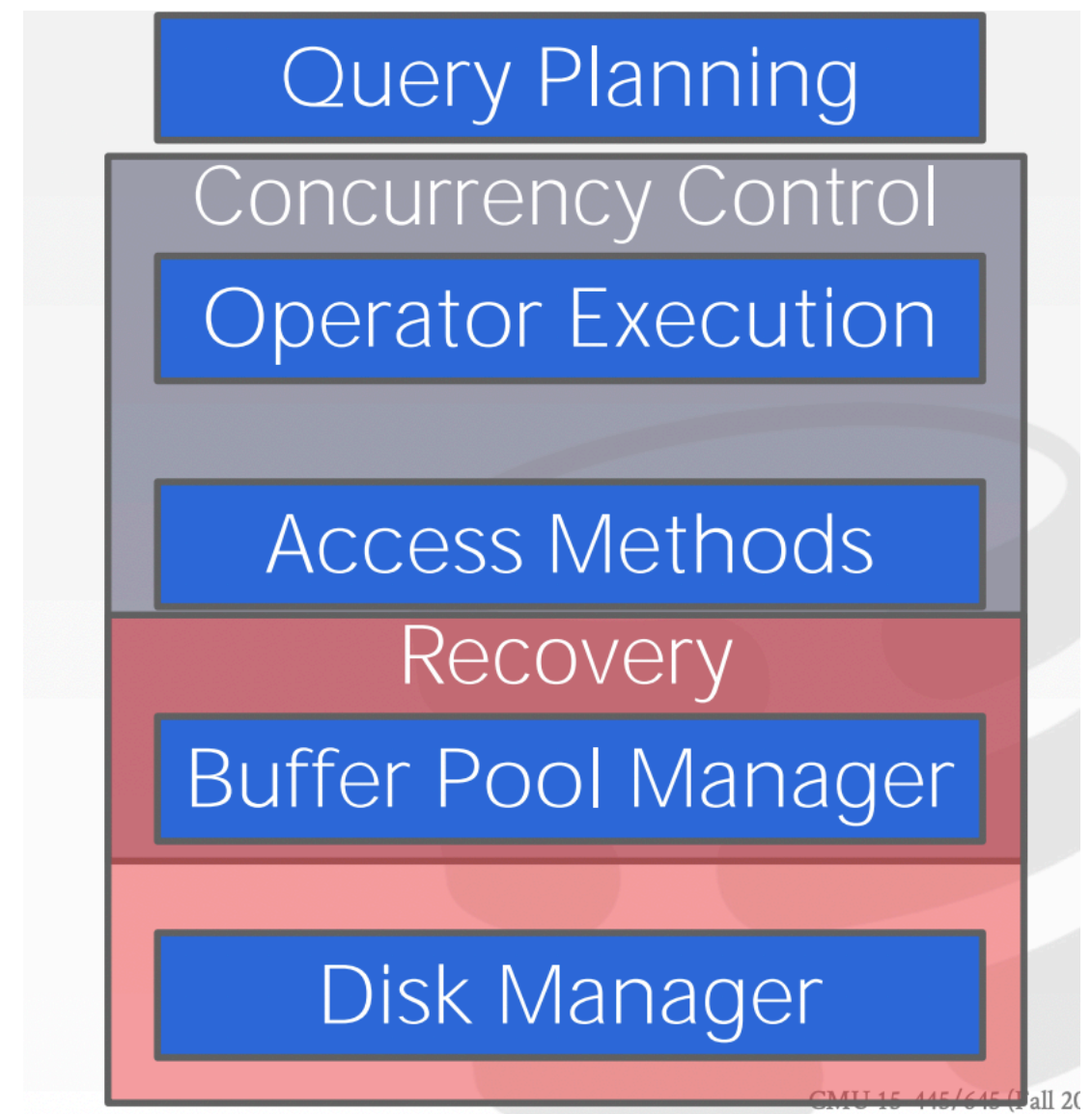
Propiedades ACID

- El estándar para transacciones confiables y sólidas es la prueba ACID.
- ACID significa **Atómica, Consistente, Aislada y Duradera**
- **Atómica:** Una transacción debe ser atómica, en el sentido de que el cambio no se puede dividir en partes más pequeñas. **“Todo o nada”**
- **Consistente:** Suponiendo que una base de datos se inicia en un estado coherente, la aplicación de una transacción debe mantener la base de datos coherente. **“Me parece correcto”**
- **Aislada:** Los cambios no deben ser visibles para ningún otro sistema que acceda a la base de datos, ni deben integrarse en el registro permanente de la base de datos hasta que se confirme toda la transacción. **“Como si estuviera solo”**
- **Duradera:** Una vez que se devuelve un estado de éxito, no debería importar si el proceso se cancela, el sistema pierde energía o el sistema de archivos de la base de datos desaparece; al reiniciar, los cambios confirmados deben estar presentes en la base de datos. **“Sobrevivir a fallas”**
- **Las cuatro propiedades deben cumplirse para garantizar la integridad general de la base de datos.**

Concurrencia y recuperación

Concurrencia y recuperación

- El control de concurrencia de un DBMS y los componentes de recuperación están a lo largo del diseño de toda su arquitectura.



Concurrencia y recuperación

Motivación

- Ambos modificamos el mismo registro en un tabla al mismo tiempo.

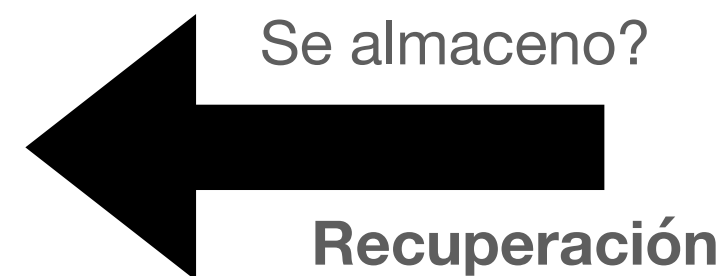
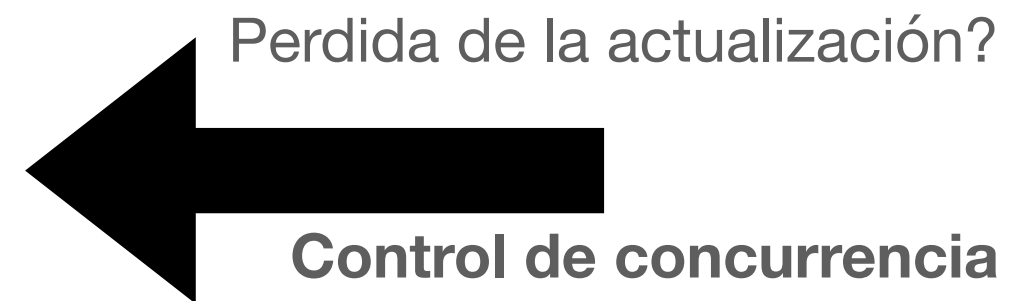
- ¿Cómo evitar y controlar esto?

- Transferi 1M entre banco pero hay un corte de energía.

- ¿Cuál es el estado correcto de la base de datos?

- **Propiedades valiosas de los DBMS.**

- **Basado en concepto de transacciones con propiedades ACID**

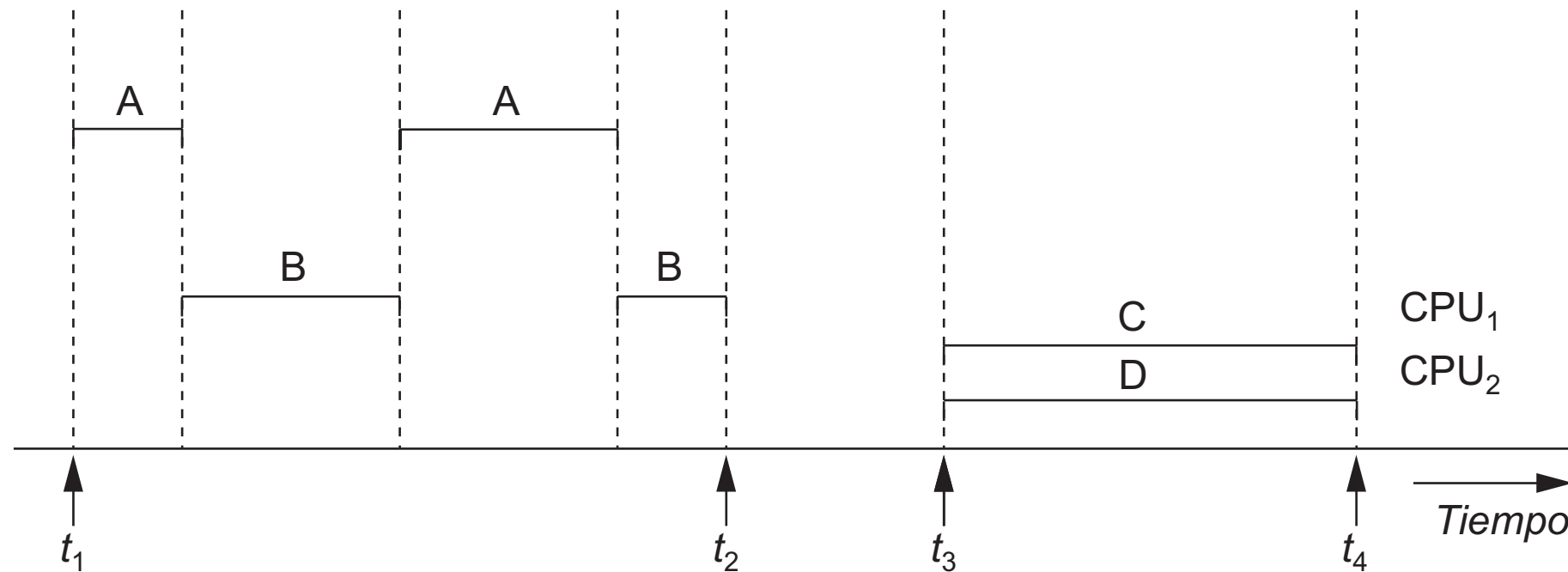


Concurrencia

- Los usuarios envían transacciones(tsc), y cada tsc se ejecuta como si fuera unica (aislada).
- El problema de controlar la concurrencia, surge cuando varias tsc enviadas por varios usuarios interfieren entre sí de un modo que produce resultados incorrectos.
- Un protocolo de concurrencia es como el DBMS decide la ejecucion apropiada de las operaciones de multiples tsc.
- Dos categorías de protocolos:
 - → Pesimista: No permitir que surjan problemas (anticipar).
 - → Optimista: Asumir que los conflictos son raros y tratarlos después de que sucedan.

Concurrencia

Interpolado o paralelo



- `read_item(X)`. Lee un elemento de base de datos denominado X y lo almacena en una variable de programa.
- `write_item(X)`. Escribe el valor de la variable de programa X en un elemento de base de datos denominado X.

(a)

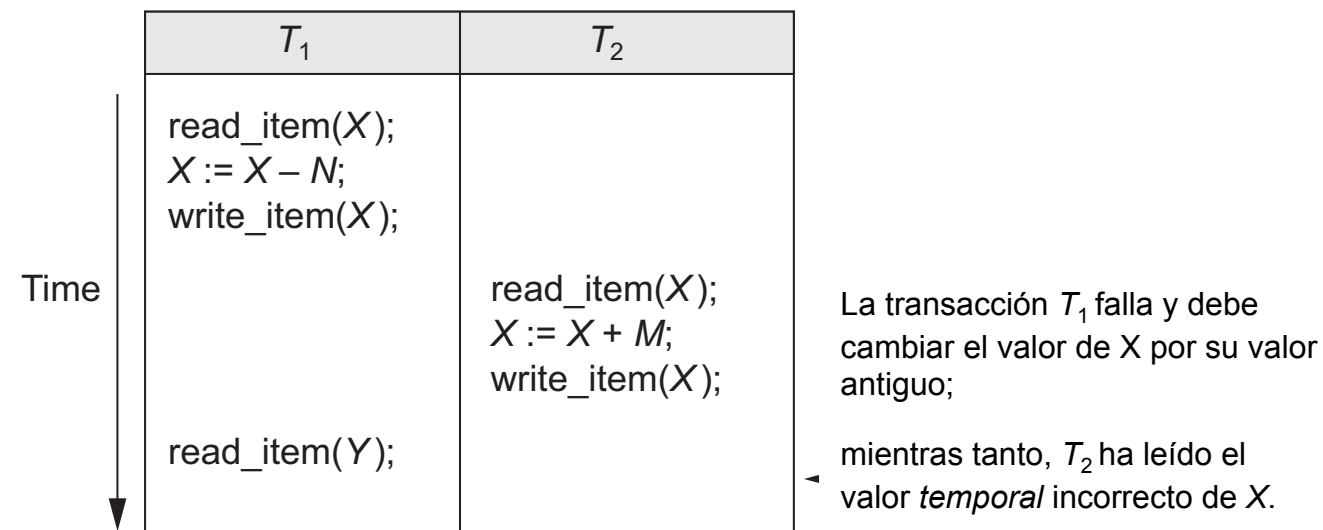
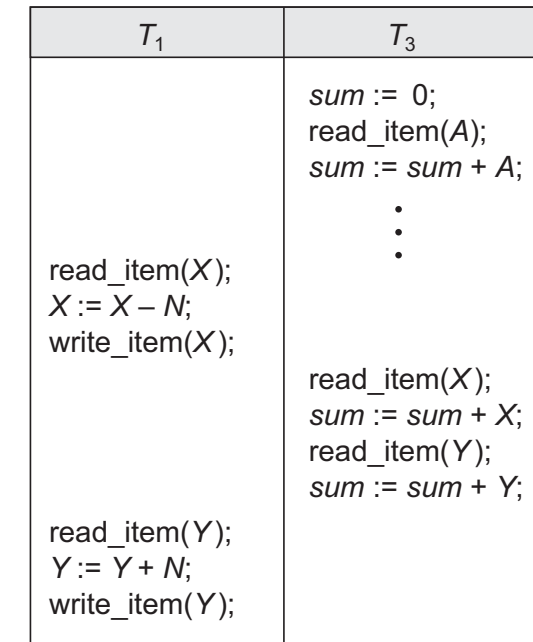
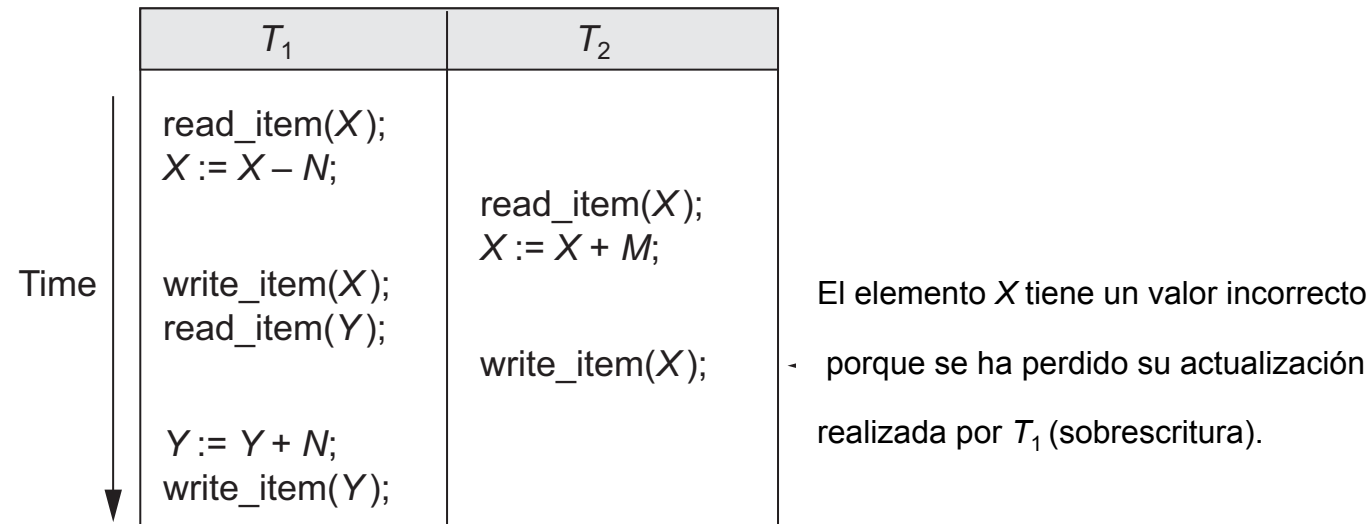
T_1
<code>read_item(X);</code> <code>X := X - N;</code> <code>write_item(X);</code> <code>read_item(Y);</code> <code>Y := Y + N;</code> <code>write_item(Y);</code>

(b)

T_2
<code>read_item(X);</code> <code>X := X + M;</code> <code>write_item(X);</code>

Concurrencia

Problemas



- Problema por pérdida de actualización
- Problema de la actualización temporal (o lectura sucia)
- El problema de la suma incorrecta

Concurrencia

Protocolos

- Bloqueo en dos fases (Pesimista)
 - Determina el orden de serialización de operaciones mientras se ejecutan tsc.
- Ordenamiento por Timestamp (Optimista)
 - Determina el orden de serialización de tsc antes de que se ejecuten.

Consultas?

Consultas o comentarios?

Muchas gracias