Transacciones en Sistemas Distribuidos

Prof. Alejandro Reyes Ortiz

Material basado en el libro: Coulouris, J. Dollimore and T. Kindberg.


¿Me lo compras?
Tú vas al cine, tú eliges todo desde Facebook y tus papás pagan con PayPal.
Entra a la Aplicación de Cinépolis ® Cineticket en Facebook.
www.facebook.com/yo
Cinépolis Cineticket
Selecciona el lugar, la película y el horario que quieres.
BAT
18 30
Elige la opción de pago "Me lo compras" con PayPal.
¿Me lo compras?
Elige dónde te quieres sentar.
papá@mail.com
Ingresa el correo de tu papá o tu mamá (ELLOS PAGAN).
Ellos reciben un correo de Cinépolis ® (con el lugar, la hora y la película que quieres ver).
Lo pagan Rápido, Fácil y Seguro con PayPal.
Pague con PayPal
Los boletos del cine llegan a tu mail.
DISFRUTA de tu PELÍCULA

Objetivos

- Identificar las definiciones básicas
- Listar las primitivas de transacciones
- Identificar la estructura de un Sistema de Manejo de Transacciones
- Identificar los conceptos de transacciones anidadas, transacciones distribuidas y la bitácora
- Crear transacciones en Base de datos: MySQL

Transacciones

- Provienen de los sistemas de Gestión de BD. En el contexto de las bases de datos distribuidas:

"Una transacción es la ejecución consiste y confiable de un conjunto de operaciones agrupadas como una unidad que acceden a una base de datos compartida"

Transacciones

"Una transacción es una secuencia de una o más operaciones agrupadas como una unidad"

El inicio y el final de la transacción definen los puntos de consistencia de los datos. Si una acción de la transacción no se puede ejecutar, entonces ninguna acción dentro de la secuencia que conforma la transacción tendrá efecto.

Las operaciones que contiene una transacción se van almacenando temporalmente, no a nivel de disco. Es hasta que termina la transacción que se tienen efecto de manera permanente o no.

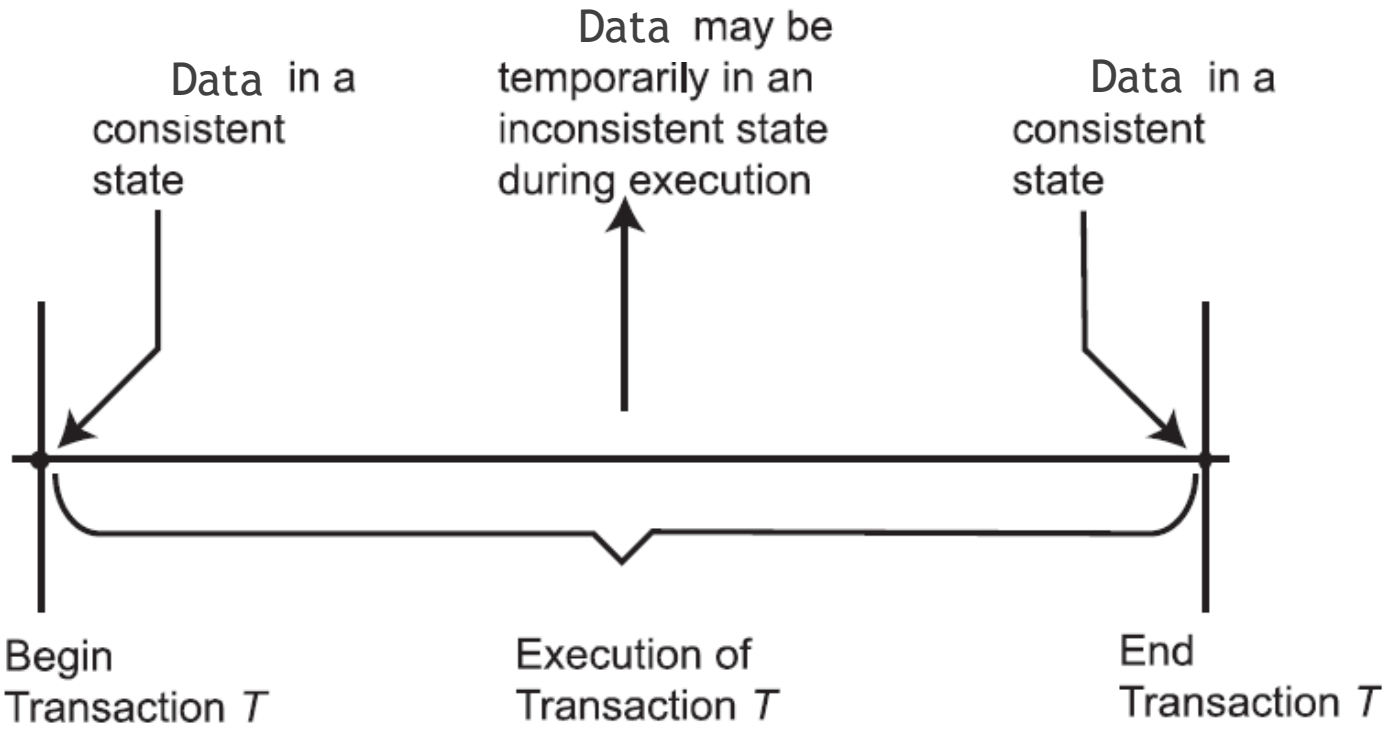
Transacciones

- En algunas situaciones, el cliente necesitan que una secuencia de solicitudes al servidor se ejecuten de manera atómica:
  - libres de interferencia por operaciones de otros clientes
  - Todas las operaciones se deben completar con éxito o no tener ningún efecto si el servidor falla.

Ejemplos: transferencia bancaria entre cuentas (retiro de cuenta A, abono a la cuenta B).

Transacciones

- Modelo de las transacciones



Transacciones

- Una ***transacción*** es una colección de acciones que hacen transformaciones de los estados de un sistema preservando la consistencia del sistema.

- El manejo de transacciones puede venir como parte del middleware que proporciona la especificación para un servicio de transacciones sobre objetos.

Transacciones

- Una transacción aplica a datos recuperables, puede estar formada por operaciones simples o compuestas y su intención es que sea atómica.

- Hay dos aspectos que se deben cumplir para lograr la atomicidad: **todo-o-nada, aislamiento**.

Transacciones: Condiciones de terminación

Una transacción siempre termina, aun en la presencia de fallas. Si una transacción termina de manera exitosa se dice que la transacción hace un ***commit*** (consumación).

Transacciones: Condiciones de terminación

Si la transacción se detiene sin terminar su tarea, se dice que la transacción ***aborta***.

Cuando la transacción es abortada, su ejecución se detiene y todas las acciones ejecutadas hasta el momento se deshacen (*undone*) regresando a la base de datos al estado antes de su ejecución.

A esta operación también se le conoce como *rollback*.

Propiedades de las transacciones

- Atomicidad (Atomicity)
- Consistencia (Consistency)
- Aislamiento (Isolation)
- Durabilidad (Durability)

Tipos de Transacciones

Clasificación de acuerdo a su estructura

- Transacciones planas: Estas transacciones tienen un punto de partida simple (**Begin_transaction**) y un punto simple de terminación (**End_transaction**)

```
Begin_transaction RESERVAR
begin
        EXEC SQL   UPDATE cuentas SET saldo=saldo-1000 WHERE id_cuenta=1111
        EXEC SQL   UPDATE cuentas SET saldo=saldo+1000 WHERE id_cuenta=222
end
```

Transacciones anidadas

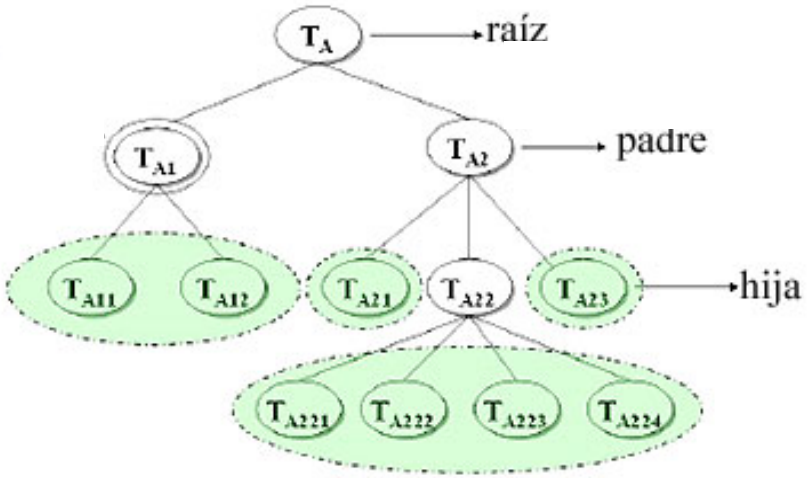
Las transacciones anidadas: las operaciones de una transacción anidada pueden incluir otras transacciones.

BeginTransaction Reservación
BeginTransaction Vuelo
...
EndTransaction {Vuelo}
BeginTransaction Hotel
...
endTransaction {Hotel}
BeginTransaction Car
...
endTransaction {Car}

EndTransaction {Reservación}

Transacciones anidadas

- Una transacción anidada dentro de otra transacción conserva las mismas propiedades que la de sus padres, esto implica, que puede contener así mismo transacciones dentro de ella.



Transacciones anidadas

- Existen restricciones para una transacción anidada:
  - Debe empezar después que su padre y debe terminar *antes* que él.
  - El *commit* de una transacción padre está condicionada al *commit* de sus transacciones hijas
  - Si alguna transacción hija aborta (*rollback*), la transacción padre también será abortada (*rollback*).

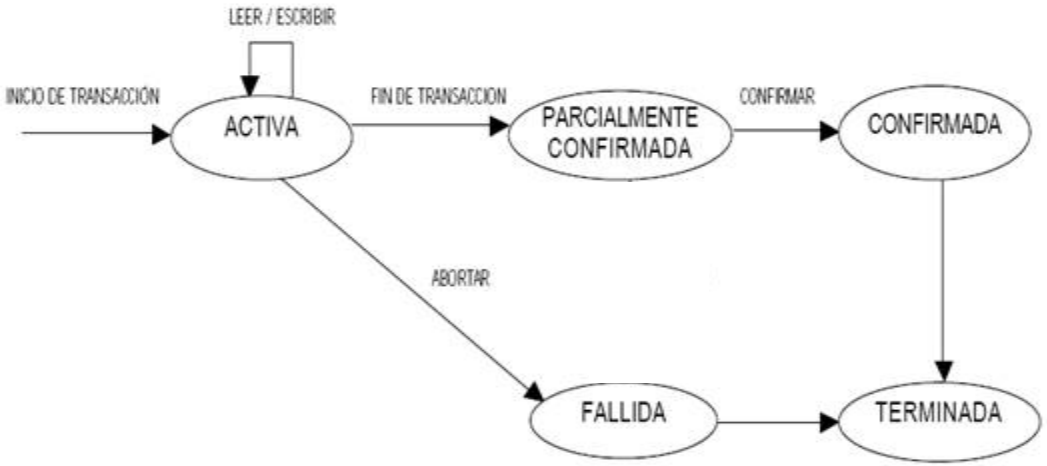
Estados de una transacción

- **Transacción Activa**: se encuentra en este estado justo después de iniciar su ejecución.
- **Transacción Parcialmente Confirmada**: en este punto, se han realizado las operaciones de la transacción pero no han sido almacenados de manera permanente.
- **Transacción Confirmada**: Ha concluido su ejecución con éxito y se almacenan de manera permanente.

Estados de una transacción

- **Transacción Fallida**: En este caso, es posible que la transacción deba ser cancelada.

- **Transacción Terminada**: indica que la transacción a abandonado el sistema.



Esquema de representación de los estados de una transacción

Transacciones: Bitácora

- Es un archivo que permite deshacer las operaciones realizadas sobre una o varias bases de datos en caso de que falle la transacción.

- Esto se hace con el fin de mantener la integridad de la información y que la transacción sea atómica.

Transacciones: Bitácora

```
x = 0;
y = 0;
BEGIN_TRANSACTION;
   x = x + 1;
   y = y + 2
   x = y * y;
END_TRANSACTION;
```

Bitácora	Bitácora	Bitácora
[x = 0 / 1]	[x = 0 / 1]	[x = 0 / 1]
	[y = 0/2]	[y = 0/2]
		[x = 1/4]

Recapitulación de la unidad

- Explica el concepto de transacción en sistemas distribuidos.
- ¿Cuáles son las condiciones de terminación de una transacción ?
- Mencione los tipos de transacciones
- ¿Qué es la bitácora y para qué sirve?
- Mencionar los estados de las transacciones
- Explicar las cuatro propiedades de las transacciones.

Primitivas para el manejo de transacciones

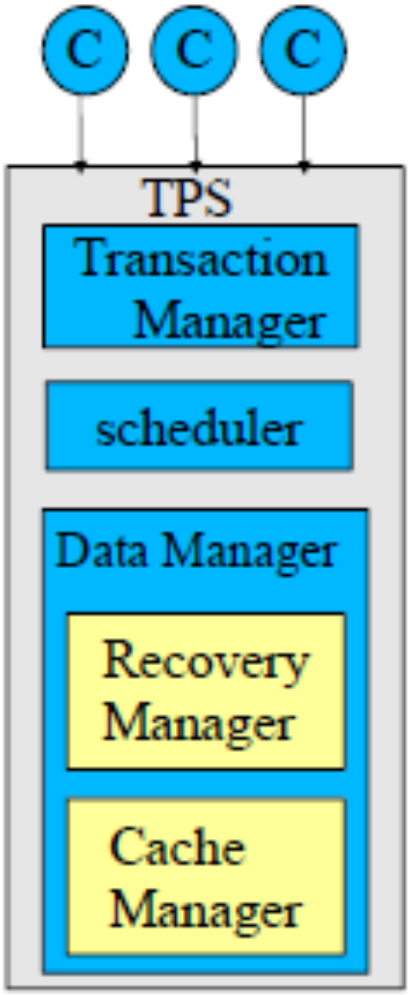
Las transacciones consisten de una secuencia de operaciones primitivas encerradas entre las palabras clave **Begin Transaction** y **End Transaction**. Por ejemplo:

BeginTransaction Reservación
. . .
EndTransaction {Reservación}

Primitivas para el manejo de transacciones

- ABORT_TRANSACTION (deshacer operación)
- READ (leer datos)
- WRITE (escribir datos)
- COMMIT (Consumación)

Estructura de un Sistema de Manejo de Transacciones



- El **Manejador de Transacciones** valida las peticiones de los clientes y pasa la transacción al planificador.
- El **Planificador** usa alguna estrategia para permitir una ejecución concurrente que sea secuencialmente equivalente.
- **Manejador de Datos**: transferir los datos a memoria principal, escribir actualizaciones, recuperarse ante fallas.