

## Secuenciabilidad de transacciones

### Ejemplo explicado paso a paso

Supongamos que el SGBD recibe simultáneamente las siguientes 3 transacciones

T1	T2	T3
Leer(A) C = A Escribir(C) leer(B) A = A + B Escribir(A)	Leer(C) A = 50 Escribir(A) B = B + C Escribir(B)	Leer(A) A = A * 1.12 Escribir(A)

Las cuales llegaron en el siguiente orden: T1 -> T2 -> T3.

Requerimos entonces elaborar una planificación que permita la ejecución concurrente de estas 3 transacciones y que sea secuenciable respecto a T1 -> T2 -> T3.

Para ello partimos de la planificación secuencial establecida, que sería:

#### Planificación secuencial T1 -> T2 -> T3

T1	T2	T3
Leer(A) C = A Escribir(C) leer(B) A = A + B Escribir(A)	Leer(C) A = 50 Escribir(A) B = B + C Escribir(B)	Leer(A) A = A * 1.12 Escribir(A)

A efectos de elaborar una planificación secuenciable solo nos interesa evaluar las operaciones de lectura y escritura:

#### Planificación secuencial T1 -> T2 -> T3

T1	T2	T3
Leer(A) Escribir(C) leer(B) Escribir(A)	Leer(C) Escribir(A) Escribir(B)	Leer(A) Escribir(A)

#### Planificación secuenciable

El mantener una planificación de ejecución secuencial aunque es más simple y asegura la consistencia de la BD, no es lo más óptimo en cuanto a tiempos de respuesta, en este caso la planificación T1 tardaría mucho menos que la planificación T3.

Lo ideal es encontrar una planificación que permita la ejecución concurrente de las tres transacciones pero que además asegure la consistencia de la BD. A este tipo de planificaciones se la llama SECUENCIABLE, ya que es equivalente a la planificación secuencial en cuanto a que lleva a la BD al mismo estado de consistencia final; pero al ser una ejecución concurrente mejora los tiempos de respuesta promedio.

La idea entonces es que a partir de la planificación secuencial, vayamos intercalando la ejecución de operaciones consecutivas de distintas transacciones (siempre manteniendo la atomicidad y aislamiento de cada transacción), mientras el resultado final sea consistente (equivalente a la planificación secuencial). Para ello nos valemos de los tipos de secuencialidad existentes:

- Secuencialidad en términos de conflictos
- Secuencialidad en términos de vistas

### Secuencialidad en cuanto a conflictos

La secuencialidad en cuanto a conflictos nos dice que dos instrucciones consecutivas de diferentes transacciones se pueden intercalar si no existe conflicto entre ellas. Y existe conflicto cuando las dos operaciones acceden al mismo elemento de dato y al menos una de ellas escribe.

Para nuestro ejemplo:

En P0 que es la planificación secuencial con la que hemos partido ESCRIBIR(A) de T1 y LEER(C) de T2 son operaciones consecutivas de diferentes transacciones. En este caso como acceden a diferente elemento de dato (A y C) no existe conflicto y por lo tanto se puede intercalar su ejecución en el tiempo, tal como lo muestra P1.

P0

T1	T2	T3
Leer(A)		
Escribir(C)		
leer(B)		
<b>Escribir(A)</b>		
	<b>Leer(C)</b>	
	Escribir(A)	
	Escribir(B)	
		Leer(A)
		Escribir(A)

P1

T1	T2	T3
Leer(A)		
Escribir(C)		
<b>leer(B)</b>		
	<b>Leer(C)</b>	
Escribir(A)		
	Escribir(A)	
	Escribir(B)	
		Leer(A)
		Escribir(A)

En P1 a su vez se aplica lo mismo para LEER(B) de T1 y LEER(C) de T2, dando como resultado P2:

P2

T1	T2	T3
Leer(A)		
<b>Escribir(C)</b>		
	<b>Leer(C)</b>	
leer(B)		
<b>Escribir(A)</b>		
	<b>Escribir(A)</b>	
	<b>Escribir(B)</b>	
		<b>Leer(A)</b>
		Escribir(A)

P3

T1	T2	T3
Leer(A)		
<b>Escribir(C)</b>		
	<b>Leer(C)</b>	
leer(B)		
<b>Escribir(A)</b>		
	<b>Escribir(A)</b>	
		<b>Leer(A)</b>
	<b>Escribir(B)</b>	
		<b>Escribir(A)</b>

En P2, ESCRIBIR(C) de T1 y LEER(C) de T2 son operaciones consecutivas, pero en este caso acceden al mismo elemento de dato y una de ellas escribe, por tanto hay conflicto y no se pueden intercalar. Lo mismo ocurre con ESCRIBIR(A) de T1 y ESCRIBIR(A) de T2. Pero entre ESCRIBIR(B) de T2 y LEER(A) de T3 no existe conflicto, por lo tanto se intercalan dado como resultado P3.

Y así sucesivamente en rojo aparecen las operaciones con conflicto y en azul las que no tienen conflicto. Finalmente llegamos a la planificación final PF, en la cual ya no existen intercalaciones posibles:

PF		
T1	T2	T3
Leer(A)		
Escribir(C)		
	Leer(C)	
leer(B)		
Escribir(A)		
	Escribir(A)	
		Leer(A)
		Escribir(A)
	Escribir(B)	

A esta última entonces es a la que llamamos PLANIFICACION SECUENCIABLE EN CUANTO A CONFLICTOS.

#### Planificación secuenciable en cuanto a conflictos

T1	T2	T3
Leer(A)		
Escribir(C)		
	Leer(C)	
leer(B)		
Escribir(A)		
	Escribir(A)	
		Leer(A)
		Escribir(A)
	Escribir(B)	

#### Secuencialidad en cuanto a vistas

La secuencialidad en cuanto a vistas se basa en tres reglas, a saber:

Partiendo de una planificación secuencial P0, decimos que una planificación concurrente es secuenciable en términos de vistas respecto a P0 si se cumple que:

- R1. En ambas planificaciones (secuencial y secuenciable) debe cumplirse que el valor original de un mismo elemento de dato debe ser accedido o leído por la misma transacción (en caso de que exista una operación de lectura).
- R2. Si en la planificación secuencial existen dos instrucciones de diferentes transacciones donde la una escribe un elemento de datos y la otra lee el valor del mismo elemento que fue escrito por la anterior, el orden de estas dos instrucciones debe mantenerse en la planificación secuenciable equivalente.
- R3. En ambas planificaciones la transacción que escribe el valor final de un elemento de datos debe ser la misma.

**Toda planificación secuenciable en términos de conflictos, lo es también en términos de vistas.**

Por lo tanto debemos asumir que para nuestro ejemplo la planificación secuenciable en cuanto a conflictos que obtuvimos, lo es también en cuanto vistas, y se comprueba verificando el cumplimiento de las reglas así.

**P0**

T1	T2	T3
Leer(A)		
Escribir(C)		
leer(B)		
Escribir(A)		
	Leer(C)	
	Escribir(A)	
	Escribir(B)	
		Leer(A)
		Escribir(A)



**PF**

T1	T2	T3
Leer(A)		
Escribir(C)		
	Leer(C)	
leer(B)		
Escribir(A)		
	Escribir(A)	
		Leer(A)
		Escribir(A)
	Escribir(B)	

- En P0, T1 lee el valor inicial de A, lo mismo ocurre en PF
- En P0, T1 lee el valor inicial de B, lo mismo ocurre en PF
- En P0, T2 lee el valor de C que previamente escribió T1, lo mismo ocurre en PF
- En P0, T3 lee el valor de A que previamente escribió T2, lo mismo ocurre en PF
- En P0, T1 escribe el valor final de C, lo mismo ocurre en PF
- En P0, T2 escribe el valor final de B, lo mismo ocurre en PF
- En P0, T3 escribe el valor final de A, lo mismo ocurre en PF

Con ello se comprueba que efectivamente nuestra planificación secuenciable en cuanto a conflictos también es secuenciable en cuanto a vistas.

Ahora bien, sabemos que en cuanto a conflictos ya no es posible realizar mas intercalaciones. La pregunta es:

Se pueden realizar más intercalaciones en términos de vistas?

La respuesta para nuestro ejemplo es SI, y tiene que ver con la tercera regla (R3).

Lo que se puede hacer por ejemplo es que ESCRIBIR(A) de T1 se ejecute después de LEER(A) de T3, así:

**PF2**

T1	T2	T3
Leer(A)		
Escribir(C)		
	Leer(C)	
leer(B)		
	Escribir(A)	
		Leer(A)
Escribir(A)		
		Escribir(A)
	Escribir(B)	

Con ello las tres reglas se siguen cumpliendo, incluidas R2 y R3 para el elemento de datos A que pudieron haberse visto comprometidas con el cambio:

P0

T1	T2	T3
Leer(A)		
Escribir(C)		
leer(B)		
Escribir(A)		
	Leer(C)	
	Escribir(A)	
	Escribir(B)	
		Leer(A)
		Escribir(A)



PF2

T1	T2	T3
Leer(A)		
Escribir(C)		
	Leer(C)	
leer(B)		
	Escribir(A)	
		Leer(A)
Escribir(A)		
		Escribir(A)
	Escribir(B)	

- En P0, T1 lee el valor inicial de A, lo mismo ocurre en PF2
- En P0, T1 lee el valor inicial de B, lo mismo ocurre en PF2
- En P0, T2 lee el valor de C que previamente escribió T1, lo mismo ocurre en PF2
- En P0, T3 lee el valor de A que previamente escribió T2, lo mismo ocurre en PF2
- En P0, T1 escribe el valor final de C, lo mismo ocurre en PF2
- En P0, T2 escribe el valor final de B, lo mismo ocurre en PF2
- En P0, T3 escribe el valor final de A, lo mismo ocurre en PF2

Como vemos el cumplimiento de las reglas se mantiene. Dicho de otra manera, en nuestro ejemplo no importa si T1 escribe A después de T2, mientras T3 lo escriba al final y mientras T3 lea el valor que escribió T2, no hay problema de inconsistencia.

Hasta ahí llegamos, no es posible realizar más intercalaciones que no impliquen romper alguna de las 3 reglas. Por tanto nuestra planificación concurrente final nos queda:

#### Planificación secuenciable en términos de vistas

T1	T2	T3
Leer(A)		
Escribir(C)		
	Leer(C)	
leer(B)		
	Escribir(A)	
		Leer(A)
Escribir(A)		
		Escribir(A)
	Escribir(B)	

#### Recuerde:

- El orden de las operaciones en cada transacción individual, no debe ser alterado en ningún caso.
- El resto de operaciones que conforman cada transacción y que no corresponden a lecturas o escrituras se ejecutan igual en el mismo orden.