

Unidad III: Transacciones, Concurrency y Recuperación.

Ejercicio 1

Examinar la planificación que se muestra a continuación:

T1	T2	T3
0		start
1		READ tax
2		tax := tax + 1
3 start		
4 READ salary		
5 salary := salary + 1		
6		WRITE tax
7		commit
8	start	
9	READ tax	
10	READ salary	
11	tax := tax + salary	
12	WRITE tax	
13	commit	
14 ----- checkpoint start -----		
15 READ tax		
16 tax := tax + salary		
17 WRITE salary		
18 ----- checkpoint end -----		
19 commit		

Hay tres transacciones T1, T2 y T3, e inicialmente tax=2 y salary=1.

- Mostrar los registros del fichero undo/redo log que se debería de crear para este plan. Para cada entrada del log, indicar también que línea de arriba genera ese registro.
- Asumir que el algoritmo de recuperación undo/redo log con checkpoint se utiliza. La base de datos falla inmediatamente después de la sentencia 7. (Asumir que los registros del log están en el disco).
 - Qué transacciones tendrían que ser deshechas?
 - Qué transacciones tendrían que ser rehechas?
- Asumir de nuevo que se utiliza el mismo algoritmo que en el apartado b), pero ahora la base de datos falla después de la sentencia 18.
 - Qué transacciones deberían ser deshechas?
 - Qué transacciones deberían ser rehechas?

Ejercicio 2

Examinar la planificación que se muestra a continuación:

T1	T2	T3	T4
1			READ tax
2 READ salary			
3			WRITE tax
4	READ tax		
5	WRITE tax		
6 READ tax			
7 WRITE salary			
8		READ salary	
9 WRITE tax			
10		WRITE salary	
11			READ salary
12			WRITE salary

Se pide:

- Dibujar el grafo de precedencia para esta planificación.
- Cuál es el orden equivalente serie para esta planificación?
- Asumir que la planificación T4 no está presente. Cual es el grafo de precedencia ahora?
- Cuál es el orden equivalente serie para esta segunda planificación?

Ejercicio 3

Considerar la siguiente base de datos almacenada en disco:

Elemento	Valor
A	13
B	40
C	35
D	4
E	18

Para cada uno de los siguientes estados de log que se muestran a continuación, determinar si esos dos log podrían ser un redo log cuyas acciones resultaran que la base de datos tuviese esos valores en disco.

- | | |
|--|--|
| (a) <START T1>
<T1,C,35>
<T1,D,450>
<START T2>
<T2,C,18>
<T2,B,40>
<COMMIT T1>
<START CKPT (T2)>
<END CKPT>
<T2,D,18>
<START T3>
<T3,C,35>
<T3, E,18>
<T2,A,13>
<COMMIT T3>
<COMMIT T2> | (b) <START T1>
<T1,D,4>
<START T2>
<T2,E,6>
<T1,A,5>
<START CKPT (T1,T2)>
<T1,E,18>
<START T3>
<T3,C,35>
<T3,A,13>
<COMMIT T2>
<T3,B,40>
<COMMIT T3>
<END CKPT>
<T1,A,11>
<COMMIT T1> |
|--|--|

Ejercicio 4

Asumir que una base de datos, usando undo/redo log y checkpoint, falla con los registros en disco que se muestra a continuación:

```

<START T1>
<T1, X, 14, 28>
<T1, Y, 15, 5>
<START T2>
<T2, Z, 20, 10>
<COMMIT T1>
<START CKPT (T2)>
<T2, W, 4, 7>
<START T3>
<END CKPT>
<T3, X, 28, 17>
<COMMIT T2>

```

- Cuales son todos los posibles valores en disco de cada elemento W, X, Y y Z?
- Qué transacciones serían rehechas en el proceso de recuperación?
- Si el registro <END CKPT> no estuviese presente en el log, cuales serían ahora las respuestas para las cuestiones a) y b)?

Ejercicio 5

Dada la siguiente planificación:

T1	T2	T3
	leer (A)	
		escribir (A)
leer (B)		
leer (A)		
	escribir (A)	
escribir (A)		
		leer (B)

¿Es secuenciable en conflictos o en vistas?

Ejercicio 6

Dadas las siguientes transacciones T1, T2 y T3, donde antes de empezar a ejecutarse las transacciones A vale 1000, B vale 500 y C vale 2000.

T1	T2	T3
Leer(A)	C=0	Leer(A)
Leer(B)	Leer(A)	Leer(C)
C=A+B	A=C+1000	C=A-C
C=C+A*0.1	Escribir(A)	Escribir(C)
A=A-A*0.1		
Escribir(A)		
B=B-C		
Escribir(C)		
D=A-1		
Escribir(B)		

Se pide:

- Realizar una planificación concurrente secuenciable en conflictos utilizando el protocolo de bloqueo de dos fases refinado.
- Mostrar el estado del registro histórico después de ejecutarse la planificación del apartado a) utilizando el algoritmo de modificación inmediata y los valores de los elementos de datos en la base de datos.

Ejercicio 7

En un sistema no distribuido hay transacciones que están esperando datos que tienen otras transacciones:

- transacción T1 está esperando datos que tiene transacción T3
- transacción T2 está esperando datos que tiene transacción T4
- transacción T4 está esperando datos que tiene transacción T5
- transacción T5 está esperando datos que tiene transacción T1

- transacción T6 está esperando datos que tiene transacción T1 y T4
- transacción T7 está esperando datos que tiene transacción T1

- Determine el grafo de espera para demostrar que no hay interbloqueos.
- ¿Cual sería tres modificaciones que producirían interbloqueos?

Ejercicio 8

Dada la siguiente planificación de transacciones que se ejecutan concurrentemente

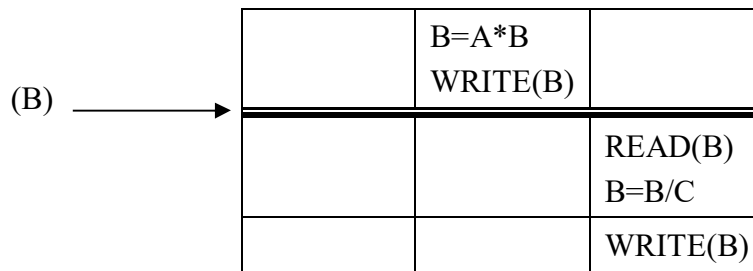
T1	T2	T3	
			Punto de revisión
leer (A)			
C= 0.5			
		leer (D)	
A=A*C			
	leer (B)		
		leer (B)	
	leer (A)		
		D=D + (B*1000)	
		escribir(D)	
	B= A+ B		
	escribir (B)		
escribir (A)			fallo

- Construir el Registro Histórico para esta planificación utilizando modificación inmediata de la Base de Datos sabiendo que antes de empezar a ejecutarse las transacciones A valía 1.100.000, B valía 200 y D valía 3.500.000.
- Suponiendo que hay un fallo en el punto indicado (en el que T2 y T3 ya habían terminado) indicar cómo se realiza la recuperación de la B.D. en ese caso.

Ejercicio 9

Se tiene la siguiente planificación, donde inicialmente A=100, B=200 y C=300. Suponer que cada transacción acaba justo con la última sentencia de ella.

	T1	T2	T3	
			READ(C) A=0.5*C	
	READ(B) B=B+10			
		A=10 WRITE(A) READ(B)		
(A) →			WRITE(A)	
		READ(A)		
	WRITE(B)			← Checkpoint



1. Mostrar el estado del registro histórico para esta planificación utilizando los esquemas (R) redo y (U-R) undo-redo log
2. Si el sistema se cae en el punto (A), para cada caso ¿Qué estado tendría la base de datos en memoria y en disco y cómo se recuperaría el sistema?
3. Si el sistema se cae en el punto (B), para cada caso ¿Qué estado tendría la base de datos en memoria y en disco y cómo se recuperaría el sistema?
4. Si no estuviese el checkpoint presente, qué efecto tendría en los puntos (A) y (B).
5. ¿Es secuenciable en conflictos y en vistas la planificación anterior? ¿Por qué? Si lo es, ¿Cuáles serían las posibles planificaciones serie?
6. ¿Tiene rollbacks en cascada la planificación anterior? ¿Por qué?
7. ¿Es recuperable la planificación anterior? ¿Por qué?
8. Utilizando un protocolo de bloqueo de dos fases (normal), ¿Cuál sería el orden de ejecución de las transacciones?

Ejercicio 10

El siguiente fragmento muestra las operaciones y los *intentos* de bloqueo que realizan una serie de transacciones (T1 a T6) sobre un conjunto de elementos de datos (A a E). Los pasos representados **no** indican que se adquiriera el bloqueo, sino sólo que se intenta realizar el mismo (XLOCK significa bloqueo exclusivo y SLOCK significa bloqueo compartido):

Secuencia	Transacción	Acción
1	T1	XLOCK A
2	T2	XLOCK E
3	T1	SLOCK B
4	T4	SLOCK C
5	T5	SLOCK B
6	T3	SLOCK D
7	T3	XLOCK A
8	T6	XLOCK A
9	T4	SLOCK B
10	T2	XLOCK D
11	T1	SLOCK E
12	T5	READ B

Suponiendo que inicialmente no hay establecido ningún bloqueo en el sistema, determinar:

1. El grafo de espera de las transacciones.
2. Las transacciones que se encuentran bloqueadas al terminar dicho fragmento.
3. Las transacciones que no se encuentran bloqueadas.

4. Al final del fragmento anterior, ¿qué bloqueos serían concedidos a las transacciones y sobre qué elementos de datos?
5. ¿Existe alguna situación de interbloqueo (*deadlock*)?. Si se produce esa situación, ¿cómo se podría deshacer el mismo? Razonar la respuesta.

Ejercicio 11

Considerar los planes P_1 y P_2 que se muestran a continuación:

P_1 : $r_1(X)$, $r_2(Z)$, $r_1(Z)$, $r_3(X)$, $r_3(Y)$, $w_1(X)$, $w_3(Y)$, $r_2(Y)$, $w_2(Z)$, $w_2(Y)$, c_1 , c_2 , c_3

P_2 : $r_1(X)$, $r_2(Z)$, $r_3(X)$, $r_1(Z)$, $r_2(Y)$, $r_3(Y)$, $w_1(X)$, c_1 , $w_2(Z)$, $w_3(Y)$, $w_2(Y)$, c_3 , c_2

Donde:

$r_n(A)$, indica la lectura de la transacción n del elemento de datos A ,

$w_n(A)$, indica la escritura de la transacción n del elemento de datos A .

c_n , indica la operación COMMIT de la transacción n .

Justificar razonando:

- a) Si cada plan es o no es serializable en cuanto a conflictos y en cuanto a vistas. En caso de ser serializables obtener los posibles planes equivalentes serie.
- b) Si cada plan tiene o no tiene rollbacks en cascada.
- c) Si cada plan es o no es recuperable.

Ejercicio 12

Considerar el siguiente log de una base de datos:

1	<START T1>
2	<T1, A, 22,17>
3	<START T2>
4	<START T3>
5	<T2,C,42,31>
6	<T3,E,9,14>
7	<T1,B,13,12>
8	<T1,A,17,8>
9	<T2,D,6,3>
10	<COMMIT T1>
11	<START T4>
12	<T4,F,7,5>
13	<T2,D,3,1>
14	<START CHECKPOINT (T2,T3,T4)>
15	<T2,C,31,2>
16	<T4,F,5,16>
17	<START T5>
18	<T3,E,14,15>
19	<T5,G,29,30>

20	<COMMIT T2>
21	<END CHECKPOINT>
22	<T5,G,30,32>
23	<ABORT T3>
24	<T4,H,22,11>
25	<COMMIT T5>
26	<COMMIT T4>

Considerar los siguientes escenarios de fallo:

- a. El sistema falla justo antes de la escritura de la línea 9 en disco
 - b. El sistema falla justo antes de la escritura de la línea 16 en disco
 - c. El sistema falla justo antes de la escritura de la línea 20 en disco
 - d. El sistema falla justo antes de la escritura de la línea 21 en disco.
 - e. El sistema falla justo antes de la escritura de la línea 23 en disco.
 - f. El sistema falla justo antes de la escritura de la línea 25 en disco.
 - g. El sistema falla justo antes de la escritura de la línea 26 en disco.
 - h. El sistema falla justo después de la escritura de la línea 26 en disco.
-
- a) ¿Qué valores contendrá la base de datos para cada elemento de datos A,B,C,D,E,F,G y H en disco después de que se realice con éxito la recuperación del sistema?.
 - b) Obtener el grafo de precedencia del fragmento del log anterior y todas las planificaciones equivalentes en serie.