



LU:.....Apellido y Nombres:..... Cant. Hojas (sin enun.):.....

ELEMENTOS DE BASES DE DATOS

Segundo Examen Parcial – 15 de Noviembre de 2011

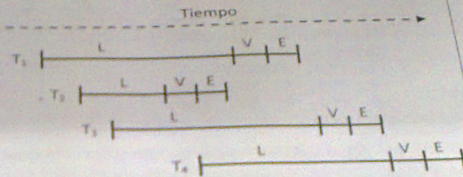
Atención! Realice los ejercicios en hojas separadas. Numere y nombre todas las hojas entregadas.

EJERCICIO 1: PROTOCOLOS DE GESTION DE CONCURRENCIA (Validación y TimeStamping)

Considere las siguientes transacciones y el siguiente diagrama de ejecución de las mismas:

T₁: read(A); read(B); write(B); write(D);
T₂: write(C); read(B); write(B);
T₃: read(D); write(D);
T₄: write(A); read(D);

Obs: En el diagrama la letra L significa que la transacción está en la fase de lectura, V en la fase de validación y E en la fase de escritura.



Considerando que las transacciones se ejecutan respetando el protocolo de validación según el diagrama planteado.

- Determine el estado final de cada transacción (commit o abort).
- Para cada transacción, justifique por qué valida o por qué debe abortar.
- Construya una planificación concurrente que utilice todas las transacciones dadas, que sea válida bajo el protocolo de estampilla de tiempo con regla de reescritura de Thomas, pero que no sea válida bajo el protocolo de estampilla de tiempo tradicional. Justifique.

EJERCICIO 2: PROTOCOLOS DE GESTION DE CONCURRENCIA (locking)

Tiempo	T ₁	T ₂	T ₃
1		read(Y)	
2		write(Y)	
3	read(X)		
4	write(X)		
5			read(Z)
6		read(X)	
7		write(X)	
8		commit	
9	read(Z)		
10	commit		
11			read(Y)
12			write(Y)
13			commit

- ¿Es posible obtener esta planificación utilizando un protocolo de bloqueo de dos fases (con bloqueos compartidos y exclusivos)? Justifique.
- ¿Es posible obtener esta planificación utilizando un protocolo de bloqueo de dos fases riguroso? Justifique.
- La planificación ¿Es recuperable?
- La planificación ¿Está libre de retrocesos en cascada?
- ¿Las expresiones “planificación serializable” y “planificación en serie” pueden ser usados indistintamente? Justifique.

EJERCICIO 3: RECUPERACION DE FALLOS

Considerando que se utiliza un esquema de modificación inmediata.

1. <T1, starta>
2. <T1, D, 20, 80>
3. <T1, commit>
4. <T3, starta>
5. <T3, C, 0, 12>
6. <checkpoint, ()>
7. <T4, starta>
8. <T4, B, 15, 35>
9. <T2, starta>
10. <T2, B, 35, 18>
11. <T4, A, 20, 45>
12. <T4, commit>
13. <T3, starta>
14. <T3, A, 45, 40>
15. <T2, D, 50, 35>
16. Fallo del sistema

- a) Complete el registro de bitácora de checkpoint.
- b) Suponga que en el instante 16 se produce un fallo del sistema. Especifique que acciones se realizan durante la recuperación del sistema, indicando:
 - Que listas se construyen y el contenido de cada una.
 - Las modificaciones y el orden en que se realizan, construyendo una tabla con el siguiente formato:

Dato	valor	Reg. de bitácora	Acción asociada
Y	nuevo	<Tj, Y, otro, nuevo>	Redo(Tj)
X	nuevo	<Ti, X, nuevo, otro>	Undo(Ti)

- c) ¿Qué valores tienen los datos en el instante 1 y qué valores tienen luego de la recuperación?
- d) Suponga que existe un sistema de bases de datos que nunca falla. ¿Sería necesario un administrador de recuperación (recovery manager) para un sistema así? Justifique.

EJERCICIO 4: BASES DE DATOS DISTRIBUIDAS

- i) Muestre con un ejemplo la utilidad de fragmentación vertical.
- ii) Mencione dos ventajas del uso de replicación. Justifique brevemente.
- iii) El algoritmo de escrituras perdidas (missing writes) establece que durante su operación normal funciona como ROWA, pero cuando detecta que algún sitio se cayó (enviando una escritura y el sitio no devuelve el correspondiente acuse de recibo <acknowledge>) cambia su modo de operación a falla operando según el algoritmo de Mayoría. Compare este algoritmo contra ROWA y Mayoría.
- iv) Suponga que se utiliza un algoritmo 3 de 6 (k de n), donde $Q_w=3$ y $Q_r=4$. Aunque respeta $Q_r + Q_w > (4 + 3 > 6)$, sabemos que no es correcto porque debería respetar que $Q_w > N/2$. Muestre con un ejemplo el problema de utilizar tal algoritmo.