



**BASES DE DATOS**  
Trabajo Práctico N° 7  
**Protocolos para Control de Concurrencia**

## Ejercicios

1. Para las siguientes transacciones:  $T_1 = \text{Read}(C); \text{Read}(A); \text{Write}(A); \text{Read}(B)$  y  $T_2 = \text{Read}(C); \text{Read}(A); \text{Write}(A); \text{Read}(B)$ . Considere las siguientes planificaciones:

i)

T <sub>1</sub>	T <sub>2</sub>
	Read(B)
	Read(C)
	Read(A)
Read(C)	
Read(A)	
Write(A)	
	Write(B)
Read(B)	

ii)

T <sub>1</sub>	T <sub>2</sub>
	Read(B)
	Read(C)
Read(C)	
	Read(A)
Read(A)	
Write(A)	
	Write(B)
Read(B)	

iii)

T <sub>1</sub>	T <sub>2</sub>
	Read(B)
Read(C)	
	Read(C)
Read(A)	
	Read(A)
	Write(B)
Write(A)	
Read(B)	

iv)

T <sub>1</sub>	T <sub>2</sub>
1. Read(C)	
2.	Read(B)
3.	Read(C)
4. Read(A)	
5. Write(A)	
6.	Read(A)
7.	Write(B)
8. Read(B)	

Verifique si es posible obtener cada una de las planificaciones i), ii), iii), iv) al aplicar un protocolo de 2 Fases utilizando:

- a) Solo bloqueos exclusivos (lock-X).
- b) Bloqueos exclusivos y compartidos (lock-X y lock-S).
- c) Bloqueos exclusivos y compartidos con opción de Upgrade y Downgrade

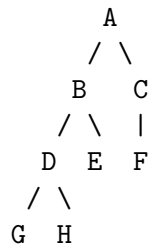
Para realizar los incisos anteriores simule la aplicación del protocolo agregando los bloqueos correspondientes y en caso de que la planificación no se pueda obtener indique en que punto la planificación no puede avanzar porque un bloqueo no fue concedido. **Note que a)** es mas restrictivo que **b)** que es mas restrictivo **c)**. Luego, si la planificación es posible en **a)** también es posible en **b)** y por lo tanto es posible en **c)**.

- d) ¿Alguna de las planificaciones anteriores resulto en un deadlock? en caso de que no, obtenga un ordenamiento de los bloqueos para la planificación iv) que produzca un deadlock.
- e) ¿Es posible obtener la planificación iv) utilizando bloqueos compartidos y exclusivos pero **sin imponer dos fases**? ¿la planificación es serializable?

- f) Suponga en la planificación iv) la transacción  $T_2$  comete en el instante 7. luego de ejecutar  $\text{write}(B)$  y la transacción  $T_1$  tiene una falla interna en el instante 8. antes de ejecutar  $\text{read}(B)$  y retrocede. ¿Que ocurre con la transacción  $T_2$ ? ¿como se conoce a este tipo de planificaciones? ¿Puede darse esta planificación bajo un protocolo de 2 fases estricto?
- g) Analice las diferentes alternativas de bloqueo para los protocolos de dos fases en cuanto a: serializabilidad de las planificaciones, nivel de concurrencia, posibilidad de deadlock, retrocesos en cascada y planificaciones no recuperables.

2. ¿Qué ventaja y desventaja proporcionan las alternativas de bloqueo de 2 fases estricto y riguroso?

3. Dada la siguiente transacción:  $T_0 = \text{read}(C); \text{read}(D); \text{write}(G)$  y el siguiente árbol



Mostrar la secuencia de locks necesarios para poder ejecutar  $T_0$ , respetando el protocolo de árbol.

4. Considere las siguientes planificaciones

$T_1$	$T_2$
Lock-X(C)	
	Lock-X(B)
Lock-X(D)	
Unlock(C)	
	Lock-X(C)
	Unlock(B)
Unlock(D)	
	Lock-X(A)
	Unlock(A)
	Lock-X(D)
	Unlock(C)
	Unlock(D)

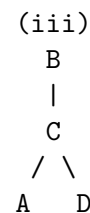
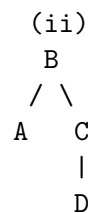
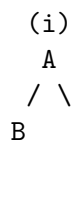
a)

$T_3$	$T_4$
Lock-X(A)	
	Lock-X(C)
	Lock-X(D)
	Unlock(C)
Lock-X(C)	
Unlock(A)	
	Unlock(D)
Lock-X(D)	
Unlock(C)	
Unlock(D)	

b)

$T_5$	$T_6$
Lock-X(B)	
	Lock-X(C)
	Lock-X(D)
	Unlock(C)
Lock-X(C)	
	Unlock(D)
Unlock(C)	
Lock-X(A)	
Unlock(B)	
Unlock(A)	

c)



Siguiendo el protocolo de árbol: ¿Con cual/es de los siguientes árboles es posible obtener cada planificación? Justifique.

5. Analice los protocolos basados en árbol en cuanto los siguientes puntos: Serializabilidad de las planificaciones, Nivel de concurrencia, Posibilidad de deadlock, retrocesos en cascada y planificaciones no recuperables.

6. Para las transacciones  $T_1$  y  $T_2$  que se muestran a continuación, con valor inicial de  $A = 100$ ,  $B = 200$ ,  $C = 300$  y  $D = 400$ .

$T_1 = \text{Read}(B); B = B * 4; \text{Write}(B); \text{Read}(C); \text{Read}(D); C = B + D; \text{Write}(C)$

$T_2 = \text{Read}(A); A = A + 100; \text{Write}(A); \text{Read}(B); \text{Read}(C); B = C; \text{Write}(B)$

- Dar una planificación concurrente serializable usando el protocolo de bloqueos de dos fases con locks compartidos y exclusivos que no caiga en deadlock. ¿Cuál es la serie equivalente? ¿Cuáles son los valores finales de A, B, C y D?
  - Dar una planificación concurrente serializable usando el protocolo de bloqueos de dos fases con locks compartidos y upgrade que caiga en deadlock.
  - ¿Es posible encontrar una planificación concurrente usando el protocolo de 2 fases estricto?
7. Para la siguiente planificación:

	$T_0$	$T_1$	$T_2$
1.	Read(A)		
2.		Write(A)	
3.	Write(A)		
4.			Write(A)

Con  $0 < ts(T_0) < ts(T_1) < ts(T_2)$  y  $R\text{-}ts(A) = R\text{-}ts(B) = R\text{-}ts(C) = W\text{-}ts(A) = W\text{-}ts(B) = W\text{-}ts(C) = 0$ .

- Verificar si es serializable en conflictos y en vistas.
- ¿Puede resultar esta planificación de aplicar el protocolo de dos fases?
- ¿Puede resultar esta planificación de aplicar el protocolo de estampillas de tiempo sin aplicar la regla de escritura de Thomas?
- ¿Puede resultar esta planificación de aplicar el protocolo de estampillas con la regla de escritura de Thomas?

**Nota:** Para los protocolos de estampillas, indique para cada instante de tiempo como se actualizan las estampillas correspondientes. Por ejemplo: 1.  $\langle A, R\text{-}ts=ts(T_0), W\text{-}ts=0 \rangle$ .

8. Considere la siguiente planificación para las transacciones  $T_0$ ,  $T_1$  y  $T_2$ , con  $ts(T_0) < ts(T_1) < ts(T_2)$  y suponga que inicialmente existen las siguientes versiones:  
 $\langle A_0, 11, R\text{-}ts=0, W\text{-}ts=0 \rangle$  y  $\langle B_0, 12, R\text{-}ts=0, W\text{-}ts=0 \rangle$

	$T_0$	$T_1$	$T_2$
1.	Read(A)		
2.	Read(B)		
3.		Read(A)	
4.		$A := A + 10$	
5.		Write(A)	
6.			Read(A)
7.	$B := A + B$		
8.	Write(B)		
9.		$B := A$	
10.		Write(B)	
11.			Read(B)

- Analizar el resultado de aplicar el protocolo de multiversión. Indique para cada instante de tiempo como se actualizan las estampillas y los valores de las versiones correspondientes. Por ejemplo: 1.  $\langle A_0, 11, R\text{-}ts=ts(T_0), W\text{-}ts=0 \rangle$

b) Encontrar, si el posible, otra planificación que al aplicar el protocolo de multiversión resulte en algún retroceso.

9. Para las transacciones  $T_0$ ,  $T_1$  y  $T_2$   $T_3$  y  $T_4$ , analizar el resultado de aplicar el protocolo de validación para a siguiente planificación, determinando que transacciones validan y cuales retroceden. **Nota:** Las operación Write modifica los datos locales de la transacción y la operacion *output* vuelca los valores de estos datos en la base de datos.

$T_0$	$T_1$	$T_2$	$T_3$	$T_6$
<i>start</i>				
Read(B)				
	<i>start</i>			
	Read(A)			
			<i>start</i>	
	Read(B)			
		<i>start</i>		
		Read(C)		
Write(B)				
			Read(C)	
	Write(C)			
		Write(C)		
<i>valid</i>				
<i>output(B)</i>				
<i>finish</i>				
	<i>valid</i>			
	<i>output(C)</i>			
	<i>finish</i>			
				<i>start</i>
				Read(B)
		<i>valid</i>		
			<i>valid</i>	
		<i>output(C)</i>		
		<i>finish</i>		
			<i>finish</i>	
				<i>valid</i>
				<i>finish</i>

10. Suponga que existen las transacciones  $T_1$ ,  $T_2$ ,  $T_3$ ,  $T_4$  y  $T_5$  los datos A, B, C y D. Los conjuntos de lectura (RS) y escritura (WS) para las transacciones son los siguientes:

$$\begin{aligned}
 RS(T_1) &= \{A, B\} & WS(T_1) &= \{A, B\} \\
 RS(T_2) &= \{B, C\} & WS(T_2) &= \{B, C\} \\
 RS(T_3) &= \{A, C\} & WS(T_3) &= \{C\} \\
 RS(T_4) &= \{D\} & WS(T_4) &= \{D\} \\
 RS(T_5) &= \{A, D\} & WS(T_5) &= \{\}
 \end{aligned}$$

Utilizaremos  $S_i, V_i$  y  $F_i$  para representar cuando una transacción  $T_i$  comienza, intenta validar y termina respectivamente. Para las siguientes secuencias de eventos:

a)  $\underline{S_1 S_2 V_1 F_1 S_4 S_3 V_2 S_5 F_2 V_3 V_5 F_3 V_4 F_5 F_4}_\rightarrow$

b)  $\underline{S_3 S_1 V_3 S_4 F_3 S_5 V_1 V_4 F_4 V_5 S_2 F_4 F_1 V_2 F_2}_\rightarrow$

Determine cuales transacciones validan y cuales retroceden, justificando en cada caso.

11. Simule la aplicación del protocolo de estampillas de tiempo pedido para las planificaciones dadas, completando las tablas que se presentan a continuación. Deberá mostrarse como se actualizan las estampillas de tiempo y los valores de los datos. En caso de producirse un retroceso por la violación del protocolo indique:

- en que punto se produce y porque.
- que transacciones retroceden y por que

a) Para las siguientes planificaciones simule la aplicación del Protocolo de estampillas de tiempo **sin** y regla de escritura de **Thomas** suponiendo que :

- los valores iniciales de los datos son: A= 11, B= 12 y C= 13
- las estampillas de tiempo son:  $0 < Ts(T1) < Ts(T2) < Ts(T3)$  e inicialmente  $R-ts(A) = R-ts(B) = R-ts(C) = W-ts(A) = W-ts(B) = W-ts(C) = 0$ .

i)

T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	Dato	Valor	R-Ts	W-ts
start						
read(A)						
A:=A+10						
write(A)						
		start				
		read(A)				
		A:=A+20				
	start					
	read(B)					
	B:= B+10					
		C:=10				
		write(A)				
		write(C)				
C:=100						
write(C)						
commit						
		commit				
	write(B)					
	commit					

ii)

T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	Dato	Valor	R-Ts	W-ts
start						
read(A)						
A:=A+10						
write(A)						
	start					
	read(B)					
		start				
		read(B)				
		read(A)				
		B:=A+B				
		write(B)				
B:=100						
write(B)						
commit						
		commit				
	B:=B+10					
	write(B)					
	commit					

b) Para la siguiente planificación simule la aplicación del protocolo de estampillas de tiempo **con** regla de escritura de **Thomas** suponiendo que :

- los valores iniciales de los datos son: A= 11, B= 12 y C= 13
- las estampillas de tiempo son:  $0 < Ts(T_1) < Ts(T_2) < Ts(T_3)$  e inicialmente  $R-ts(A) = R-ts(B) = R-ts(C) = W-ts(A) = W-ts(B) = W-ts(C) = 0$ .

i)

T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	Dato	Valor	R-Ts	W-ts
start						
read(A)						
A:=A+10						
write(A)						
		start				
		read(A)				
		A:=A+20				
	start					
	read(B)					
	B:= B+10					
		C:=10				
		write(A)				
		write(C)				
C:=100						
write(C)						
commit						
		commit				
	write(B)					
	commit					

c) Para las siguientes planificaciones simule la aplicación del Protocolo de mutiversión suponiendo que :

- las versiones iniciales de los datos son:  $\langle A_0, 11, R\text{-ts}=0, W\text{-ts}=0 \rangle$ ,  
 $\langle B_0, 12, R\text{-ts}=0, W\text{-ts}=0 \rangle$   $\langle C_0, 13, R\text{-ts}=0, W\text{-ts}=0 \rangle$
- las estampillas de tiempo son:  $0 < Ts(T1) < Ts(T2) < Ts(T3)$

i)

T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	Verisión	Valor	R-Ts	W-ts
start						
read(A)						
A:=A+10						
write(A)						
		start				
		read(A)				
		A:=A+20				
	start					
	read(B)					
	B:= B+10					
		C:=10				
		write(A)				
		write(C)				
C:=100						
write(C)						
commit						
		commit				
	write(B)					
	commit					

ii)

T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	Dato	Valor	R-Ts	W-ts
start						
read(A)						
A:=A+10						
write(A)						
	start					
	read(B)					
		start				
		read(B)				
		read(A)				
		B:=A+B				
		write(B)				
B:=100						
write(B)						
commit						
		commit				
	B:=B+10					
	write(B)					
	commit					

12. Dadas las siguientes transacciones:

$T_0 = \text{start}; \text{read}(B); \text{write}(B); \text{read}(A); \text{write}(A); \text{commit}$

$T_1 = \text{start}; \text{read}(B); \text{read}(C); \text{write}(A); \text{write}(C); \text{commit}$

$T_2 = \text{start}; \text{read}(A); \text{write}(A); \text{write}(B); \text{commit}$

Con  $0 < ts(T_0) < ts(T_1) < ts(T_2)$  y  $R\text{-ts}(A)=R\text{-ts}(B)=R\text{-ts}(C)=W\text{-ts}(A)=W\text{-ts}(B)=W\text{-ts}(C)=0$

- Encontrar, si es posible, una planificación concurrente donde al menos una de las transacciones retroceda aplicando el algoritmo de estampilla de tiempos.
- Encontrar, si es posible, una planificación concurrente donde ninguna transacción retroceda aplicando el algoritmo de estampilla de tiempos.
- Encontrar, si es posible, una planificación concurrente donde deba retroceder en cascada más de una transacción aplicando el algoritmo de estampilla de tiempos.
- Encontrar, si es posible, una planificación concurrente donde la falla de una transacción resulte en una planificación no recuperable.
- Encontrar, si es posible, una planificación donde se pueda aplicar la regla de escritura de Thomas.

Para mostrar las planificaciones encontradas utilice el siguiente formato:

	$T_0$	$T_1$	$T_2$	Estampillas
1.	start			
2.		start		
3.		Read(B)		$\langle B, R\text{-ts}=ts(T_1), W\text{-ts}=0 \rangle$
4.	...	...	...	...

13. Para las transacciones:

$T_0 = \text{read}(A); \text{read}(B); A:=A+10; \text{write}(A); B:=B+20; \text{write}(B)$

$T_1 = B:=22; \text{write}(B); C:=23; \text{write}(C)$

$T_2 = \text{read}(C); \text{read}(B); C:=C+10; \text{write}(C); B:=B+20; \text{write}(B)$

Encontrar una planificación concurrente (no en serie) resultante de aplicar:

- el protocolo de bloqueo de dos fases.
- el protocolo de estampillas.
- el protocolo de estampillas con regla de escritura de Thomas.
- el protocolo de árbol. Defina un árbol de precedencia a su elección.
- el protocolo de multiversión. Suponga que inicialmente existen las siguientes versiones:  
 $\langle A_0, 11, R\text{-ts}=0, W\text{-ts}=0 \rangle, \langle B_0, 12, R\text{-ts}=0, W\text{-ts}=0 \rangle, \langle C_0, 13, R\text{-ts}=0, W\text{-ts}=0 \rangle$
- el protocolo de validación.