

Diseño y Gestión de Bases de Datos

Solución Ejercicios Tema 4

1. $x = 4, y = 2$

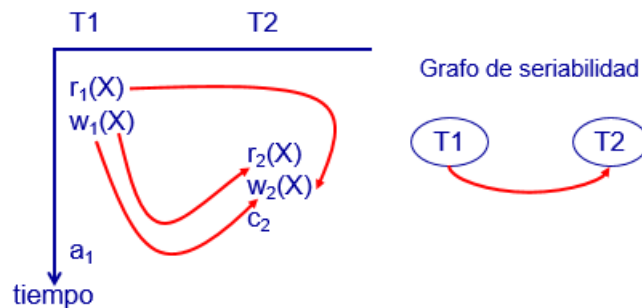
T1: $r(x)$
 $r(y)$
 Si $x > 5$ entonces $x := x - y$
 Si no $x := x + y$
 $w(x)$
 c

T2: $r(x)$
 $x := x + 3$
 $w(x)$
 c

- Plan T1-T2 $\rightarrow x=9, y=2$
- Plan T2-T1 $\rightarrow x=5, y=2$

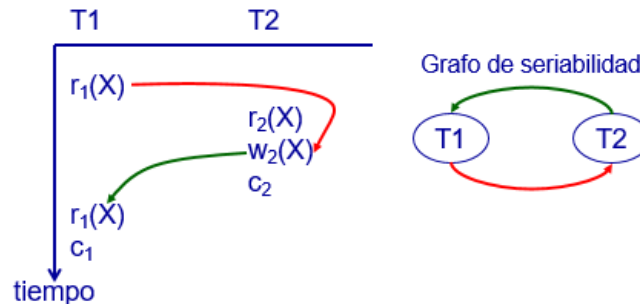
2.

Lectura sucia: $P = r_1(x), w_1(x), r_2(x), w_2(x), c_2, a_1$



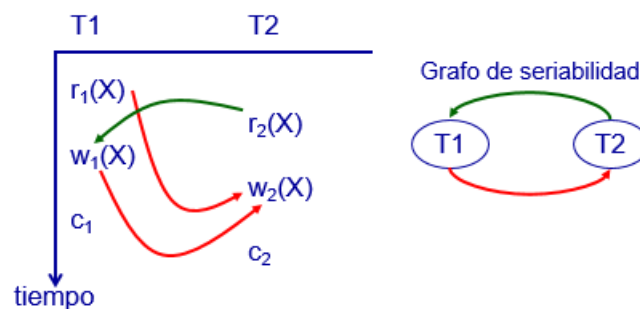
El grafo de seriabilidad no tiene ningún ciclo, el plan es serializable. El problema de la 'lectura sucia' reside en el peligro de la 'anulación en cascada'. Al anular T1 se debe anular en cascada T2 ya que ha leído (lectura sucia) un dato actualizado por T1, que finalmente no ha sido confirmado.

Lectura no repetible: $P = r_1(x), r_2(x), w_2(x), c_2, r_1(x), c_1$



El grafo tiene un ciclo, luego el plan P no es serializable.

Pérdida de actualizaciones: $P = r_1(x), r_2(x), w_1(x), w_2(x), c_1, c_2$

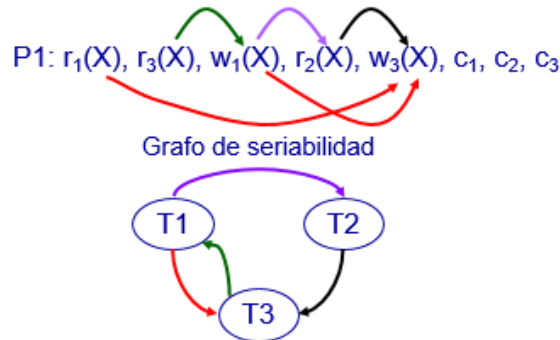


El grafo tiene un ciclo, luego el plan P no es serializable.

Concluyendo, en un plan serializable por conflictos podría aparecer la anomalía de *lectura sucia*, pero nunca la de *lectura no repetible* o *pérdida de actualizaciones*. En estos dos últimos casos, el grafo de seriabilidad siempre tiene un ciclo, y cuando el grafo tiene un ciclo, entonces el plan no es serializable por conflictos.

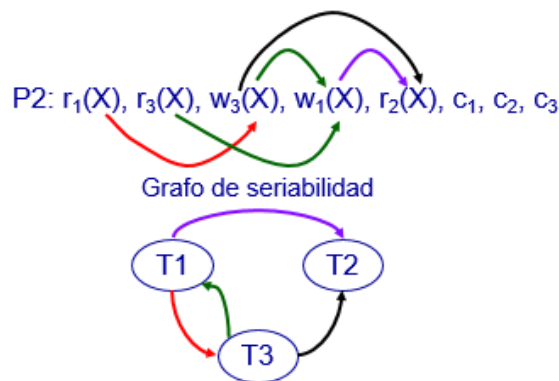
3.

Plan P1: Operaciones en conflicto y grafo de seriabilidad en el plan P1:



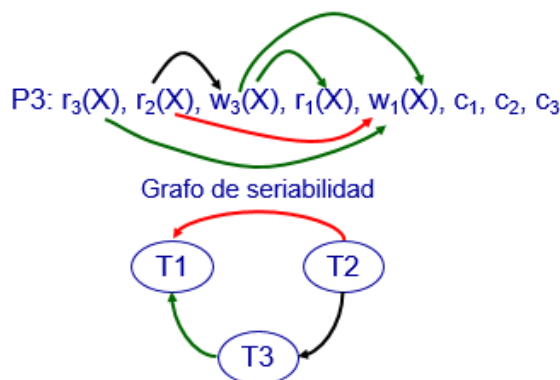
Observamos que hay un ciclo. El plan no es serializable por conflictos.

Plan P2: Operaciones en conflicto y grafo de seriabilidad en el plan P2:



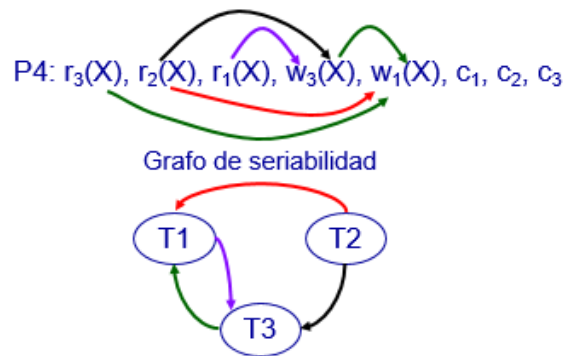
Observamos que hay un ciclo. El plan no es serializable por conflictos.

Plan P3: Operaciones en conflicto y grafo de seriabilidad en el plan P3:



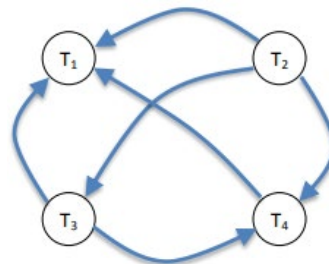
Observamos que no hay ciclos. El plan es serializable por conflictos. El único plan en serie equivalente es: T2 – T3 – T1

Plan P4: Operaciones en conflicto y grafo de seriabilidad en el plan P4:



Observamos que hay un ciclo. El plan no es serializable por conflictos.

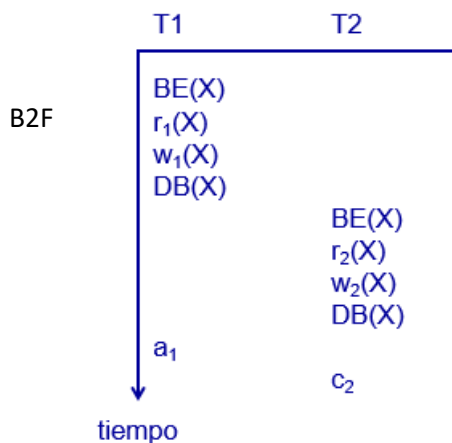
Plan P5: Operaciones en conflicto y grafo de seriabilidad en el plan P5:



Se puede comprobar que el grafo es acíclico, por tanto, el plan es serializable por conflictos. Por otro lado, analizando el grafo se puede deducir que el plan serial equivalente por conflictos es $T2 \rightarrow T3 \rightarrow T4 \rightarrow T1$. En cuanto a las anomalías de la concurrencia que se dan en el plan, únicamente se presenta la lectura sucia, como se puede ver en las correspondientes operaciones en conflicto que se dan en el plan, $w_3(y) \rightarrow r_1(y)$, $w_3(y) \rightarrow r_4(y)$ y $w_4(z) \rightarrow r_1(z)$

4.

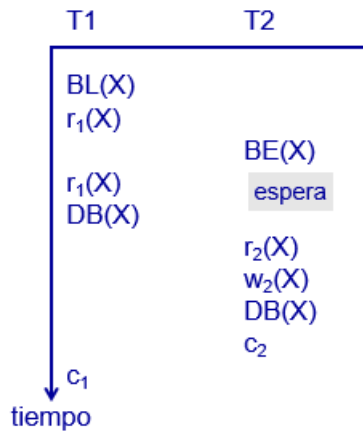
Lectura sucia: $P = r_1(x), w_1(x), r_2(x), w_2(x), a_1, c_2$



Como se puede observar en el ejemplo, el protocolo de bloqueo admite la anomalía de la 'lectura sucia' ya que es posible que una transacción desbloquee elementos de datos antes de que finalice, permitiendo, de este modo, que otras transacciones lean los datos modificados por ella aún no confirmados,.

Si el bloqueo fuera implícito no sería posible el desbloqueo anterior a la finalización de la transacción, y no sería posible la lectura sucia.

Lectura no repetible: $P = r_1(x), r_2(x), w_2(x), c_2, r_1(x), c_1$

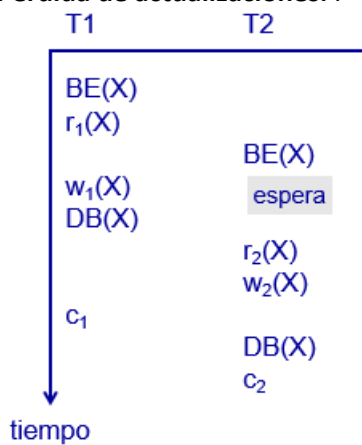


Con B2F no se produce la lectura no repetible.

Como se puede observar al realizar el bloqueo de lectura T1 impide que T2 pueda bloquear para escritura el dato X. De esta forma, al volver a leerlo, T1 obtiene el mismo valor y no se produce la lectura no repetible. El plan que se ejecuta (controlado por B2F) es distinto al planteado:

$P' = r_1(x), r_1(x), r_2(x), w_2(x), c_2, c_1$

Pérdida de actualizaciones: $P = r_1(x), r_2(x), w_1(x), w_2(x), c_1, c_2$



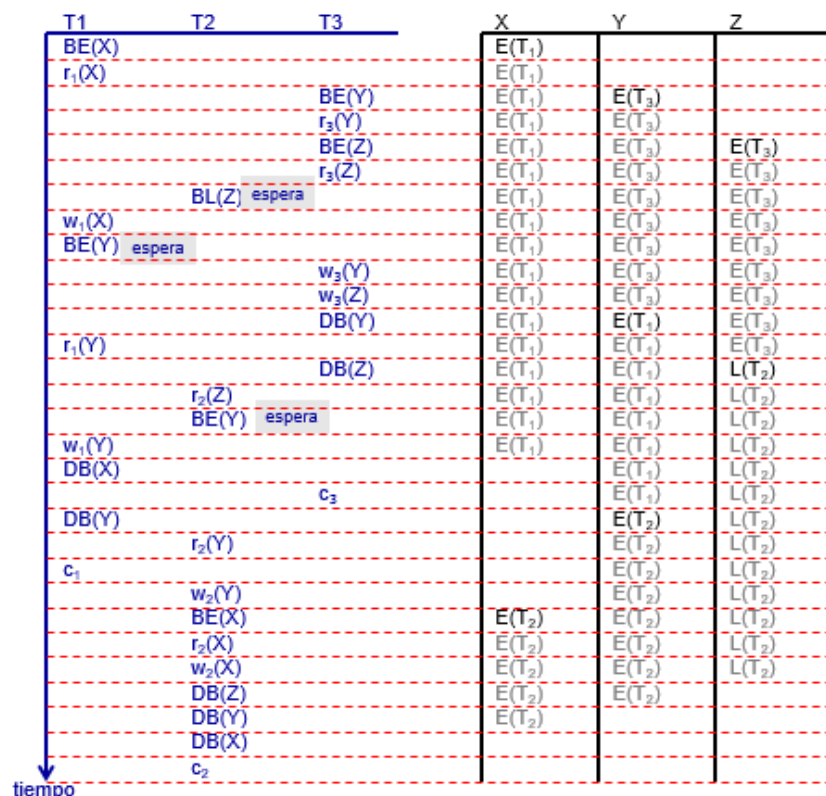
Con B2F no se produce la pérdida de actualizaciones.

Si se realizaran primero BL para luego promoverlos a BE, se produce el problema del bloqueo mortal.

Como se puede observar al realizar el bloqueo de escritura T1 impide que T2 pueda leer el dato X antes de que T1 lo haya modificado. Cuando T1 desbloquea el dato X, T2 lee el valor modificado por T1, impidiendo de esta forma que se pierda la actualización de T1. El plan que se ejecuta (controlado por B2F) es distinto al planteado:

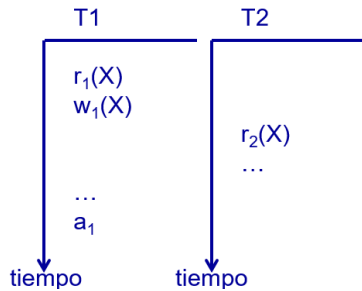
$P' = r_1(x), w_1(x), r_2(x), w_2(x), c_1, c_2$

5. Cuando una transacción vaya a leer y escribir un dato, directamente se solicitará el BE sin pasar por el BL de esta forma se evitan bloqueos mortales.

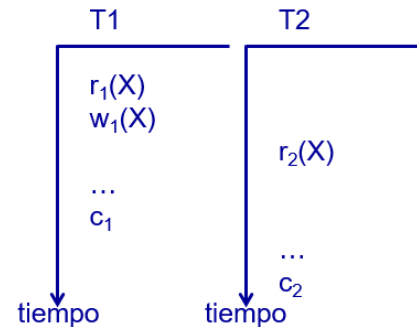


Hay que tener en cuenta que la solución que se plantea no es única, por ejemplo T1 podría haber empezado con un bloqueo en lectura para X y promocionar ese bloqueo a escritura cuando lo necesite porque va a escribir X.

6.



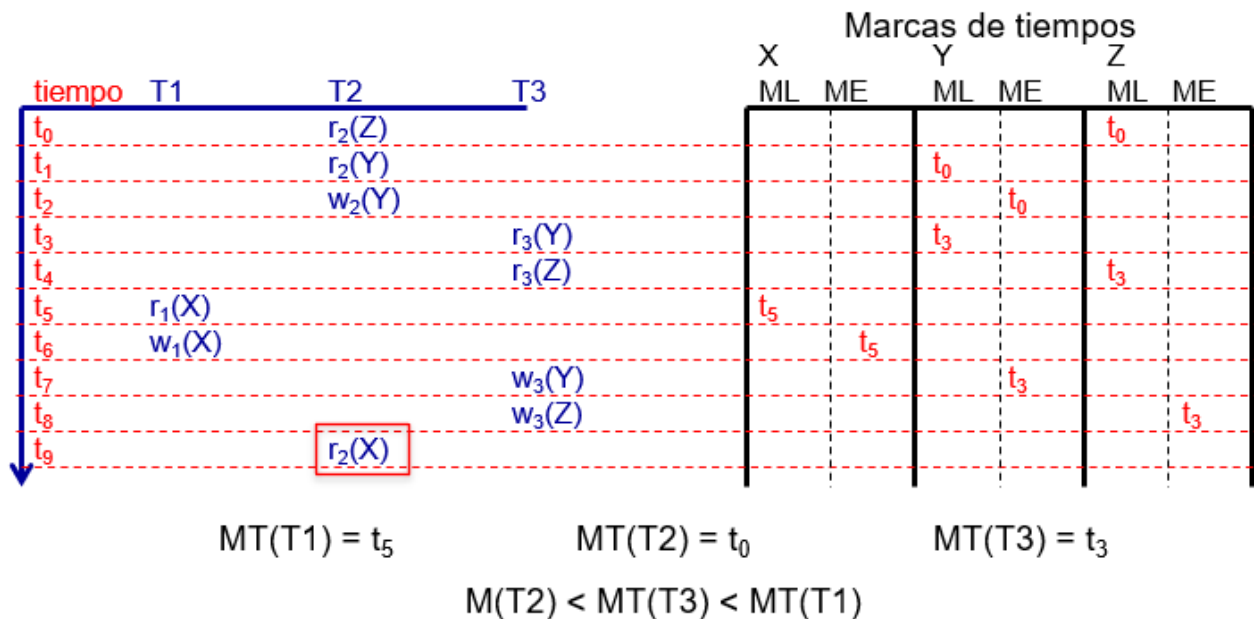
Plan A: con problema de anulación en cascada. T2 hace una lectura sucia de T1, al anular esta última tiene que ser anulada en cascada T2.



Plan B: sin problema de anulación en cascada. T2 hace una lectura sucia de T1, pero como esta última termina con confirmación la lectura sucia no representa ningún problema.

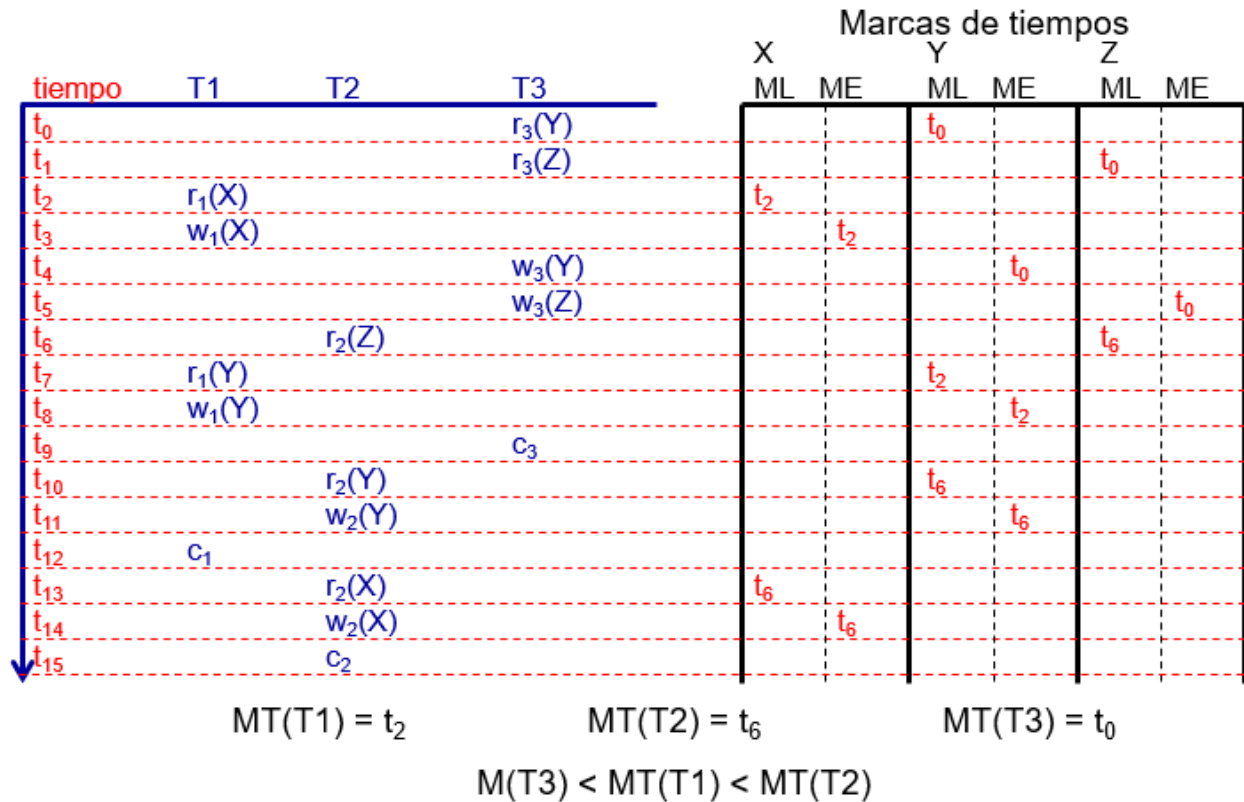
7. Supongo que los elementos de datos X, Y, Z tienen marcas de tiempo de lectura y escritura menores que las marcas de tiempo de las tres transacciones incluidas en el plan.

Plan P1:



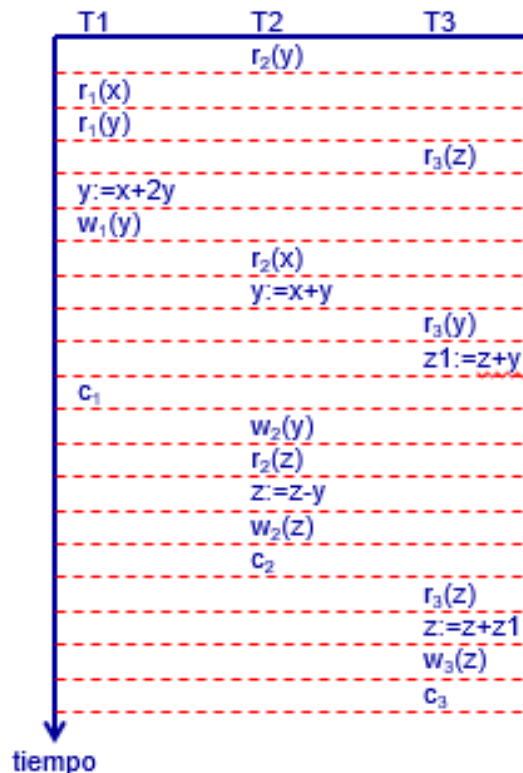
En el instante t_9 , la operación $r_2(X)$ no puede realizarse, ya que X tiene una marca de escritura que es mayor que la marca de tiempo de T2 (i.e. ha sido escrito por una transacción más joven que T2). Por este motivo el SGBD anularía T2 ejecutándola más adelante. El plan de ejecución P1 no estaría permitido con el protocolo OMT.

Plan P2:



En este caso el plan de ejecución P2 sí que estaría permitido por el protocolo OMT.

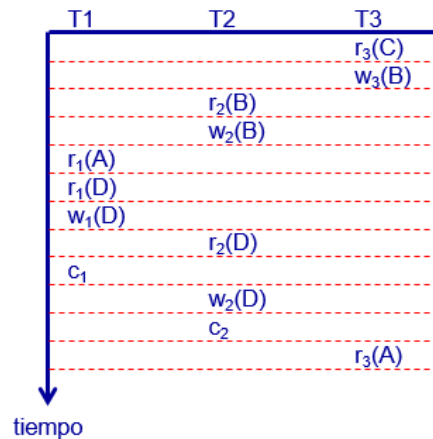
8. El plan visto a lo largo del tiempo sería:



Los protocolos tendrían el siguiente comportamiento:

- **Protocolo B2F:** este protocolo no permitiría el plan ya que T1 quedaría en espera al solicitar el bloqueo en escritura de y para la operación $w_1(y)$ porque T2 tiene que haber bloqueado para lectura Y antes de la operación $r_2(y)$. Es decir, la confirmación de T1 nunca podría ocurrir antes de que T2 desbloquee Y.

- Protocolo **OMT**: en este protocolo T2 sería anulada al solicitar la operación $w_2(y)$. En ese instante $MT_lectura(Y) = MT(T3)$ y $MT_escritura(Y) = MT(T1)$ y dado que $MT(T2) < MT(T1) < MT(T3)$ la transacción T2 sería revertida.
 - Protocolo **MV**: en este protocolo la transacción T2 sería anulada al solicitar la operación de escritura $w_2(y)$ ya que una transacción más joven (T1) ya ha leído Y.
9. Veamos como es el diagrama temporal de ejecución de las transacciones:



Si nos fijamos en las transacciones que leen de otras, podremos deducir el tipo de plan de ejecución desde el punto de vista de la recuperación. Podemos ver que T2 está leyendo de T3 (elemento B) y que T2 también está leyendo de T1 (elemento D). Además, T2 confirma después de T1 y antes que T3. Este segundo hecho hace que el plan no sea recuperable (existe una transacción en el plan que lee de otra (lectura sucia) y que confirma antes de que esta última finalice).

Proceso de recuperación: se debe deshacer las actualizaciones de la transacción T3 que ha sido interrumpida por el fallo. Deshacer T3 implica deshacer T2 que hizo una lectura sucia de T3 y que ya fue confirmada (plan no recuperable). Se deberá rehacer T1 que ha sido confirmada antes del fallo, y se supone, después del último punto de control.

10.

- $MT(T1) = t_3$, $MT(T2) = t_1$, $MT(T3) = t_0$
- T3-T2-T1. Es el plan en serie cronológico: plan en serie en el que las transacciones se ejecutan en el orden de sus marcas de tiempo.
- t_0 : $MT_L(X) = t'$, $MT_E(X) = t_0$
 t_1 : $MT_L(X) = t_1$, $MT_E(X) = t_0$
 t_2 : $MT_L(X) = t_1$, $MT_E(X) = t_1$
 t_3 : $MT_L(X) = t_3$, $MT_E(X) = t_1$
 t_4 : $MT_L(X) = t_3$, $MT_E(X) = t_1$
 t_5 : $MT_L(X) = t_3$, $MT_E(X) = t_3$
 t_6 : La operación leer(X) de T3 se aborta ya que $MT(T3) < MT_E(X)$ ($t_0 < t_3$). T3 debe revertirse e introducirse de nuevo en el sistema.
 En t_6 el protocolo ha detectado que T3 está intentando leer un dato modificado por una transacción que en el plan en serie equivalente cronológico es más joven que ella.

11.

PROTOCOLO B2F Explícito**BL = Bloqueo de Lectura PB = Promoción del bloque de lectura a bloqueo de escritura D = Desbloqueo**

t	T ₁	T ₂	T ₃
t ₁₀	BL(x)		
t ₁₁	r ₁ (x)		
t ₁₂			BL(y)
t ₁₃			r ₃ (y)
t ₁₄	PB(x)		
t ₁₅	w ₁ (x)		
t ₁₆	BL(y)		
t ₁₇	D(x)		
t ₁₈		BL(x)	
t ₁₉		r ₂ (x)	
t ₂₀	r ₁ (y)		
t ₂₁	D(y)		
t ₂₂			PB(y)
t ₂₃			w ₃ (y)
t ₂₄	C ₁		
t ₂₅		PB(x)	
t ₂₆		w ₂ (x)	
t ₂₇		D(x)	
t ₂₈			BL(x)
t ₂₉			r ₃ (x)
t ₃₀		C ₂	
t ₃₁			D(x)
t ₃₂			D(y)
t ₃₃			C ₃

El protocolo B2F Explícito sí que acepta este plan de ejecución.

PROTOCOLO B2F Implícito

t	T ₁	T ₂	T ₃
t ₁₀	r ₁ (x) [implica BL(x)]		
t ₁₁			r ₃ (y) [implica BL(y)]
t ₁₂	w ₁ (x) [implica PB(x)]		
t ₁₃		r ₂ (x) Pasa a ESPERA	

En el protocolo B2F Implícito, el plan no es posible ya que la lectura de x por T₂ no podrá ocurrir antes de que se confirme T₁ liberando entonces x.

PROTOCOLO Ordenamiento por Marcas de tiempo (OMT)

Supongo que tanto x como y tienen unas marcas iniciales a t_0 y que si $i < j$ entonces $t_i < t_j$. También supongo que el plan empieza en t_{10} .

				Marcas de tiempo			
				X		Y	
t	T ₁	T ₂	T ₃	ML	ME	ML	ME
t ₉				t ₀	t ₀	t ₀	t ₀
t ₁₀	r ₁ (x)			t ₁₀	t ₀	t ₀	t ₀
t ₁₁			r ₃ (y)	t ₁₀	t ₀	t ₁₁	t ₀
t ₁₂	w ₁ (x)			t ₁₀	t ₁₀	t ₁₁	t ₀
t ₁₃		r ₂ (x)		t ₁₃	t ₁₀	t ₁₁	t ₀
t ₁₄	r ₁ (y)			t ₁₃	t ₁₀	t ₁₁	t ₀
t ₁₅			w ₃ (y)	t ₁₃	t ₁₀	t ₁₁	t ₁₁
t ₁₆	c ₁			t ₁₃	t ₁₀	t ₁₁	t ₁₁
t ₁₇		w ₂ (x)		t ₁₃	t ₁₃	t ₁₁	t ₁₁
t ₁₈			r₃(x)				

El protocolo de ordenación por marcas de tiempo no acepta el plan ya que en el instante t_{18} la transacción T_3 intenta leer un elemento de datos que ha sido escrito por una transacción más joven ($MT_E(x) > MT(T_3)$).

PROTOCOLO Multiversión (MV)

Supongo que tanto x_1 como y_1 tienen unas marcas iniciales a t_0 y que si $i < j$ entonces $t_i < t_j$. También supongo que el plan empieza en t_{10} .

				Marcas de tiempo									
				X ₁		Y ₁		X ₂		Y ₂		X ₃	
				ML	ME	ML	ME	ML	ME	ML	ME	ML	ME
t	T ₁	T ₂	T ₃	t ₀	t ₀	t ₀	t ₀						
t ₁₀	r ₁ (x)			t ₁₀	t ₀	t ₀	t ₀						
t ₁₁			r ₃ (y)	t ₁₀	t ₀	t ₁₁	t ₀						
t ₁₂	w ₁ (x)			t ₁₀	t ₀	t ₁₁	t ₀	t ₁₀	t ₁₀				
t ₁₃		r ₂ (x)		t ₁₀	t ₀	t ₁₁	t ₀	t ₁₃	t ₁₀				
t ₁₄	r ₁ (y)			t ₁₀	t ₀	t ₁₁	t ₀	t ₁₃	t ₁₀				
t ₁₅			w ₃ (y)	t ₁₀	t ₀	t ₁₁	t ₀	t ₁₃	t ₁₀	t ₁₁	t ₁₁		
t ₁₆	C ₁			t ₁₀	t ₀	t ₁₁	t ₀	t ₁₃	t ₁₀	t ₁₁	t ₁₁		
t ₁₇		w ₂ (x)		t ₁₀	t ₀	t ₁₁	t ₀	t ₁₃	t ₁₀	t ₁₁	t ₁₁	t ₁₃	t ₁₃
t ₁₈			r ₃ (x)	t ₁₀	t ₀	t ₁₁	t ₀	t ₁₃	t ₁₀	t ₁₁	t ₁₁	t ₁₃	t ₁₃
t ₁₉		C ₂		t ₁₀	t ₀	t ₁₁	t ₀	t ₁₃	t ₁₀	t ₁₁	t ₁₁	t ₁₃	t ₁₃
t ₂₀			C ₃	t ₁₀	t ₀	t ₁₁	t ₀	t ₁₃	t ₁₀	t ₁₁	t ₁₁	t ₁₃	t ₁₃

El protocolo de multiversión sí que acepta el plan. Fijaos que en la lectura $r_3(x)$ (en la que fallaba el plan en el protocolo OMT) el valor que lee T_3 es x_2 (que es la versión de x creada por una transacción más joven anterior a T_3), la marca de lectura de esta versión de x no es actualizada porque ha sido leído por una transacción más joven que T_3 (T_2).