

22

¿Cuáles de las siguientes planificaciones son serializables (por conflicto)? Por cada planificación serializable, determine las planificaciones en serie equivalentes.

a) $R_1(X); R_3(X); W_1(X); R_2(X); W_3(X);$

b) $R_1(X); R_3(X); W_3(X); W_1(X); R_2(X);$

c) $R_3(X); R_2(X); W_3(X); R_1(X); W_1(X);$

d) $R_3(X); R_2(X); R_1(X); W_3(X); W_1(X);$

23

Considere las transacciones T_1 , T_2 y T_3 , y las planificaciones S_1 y S_2 que se muestran a continuación. Dibuje los gráficos de serialización (precedencia) para S_1 y S_2 y explique si cada serialización es, o no, serializable. Si una planificación es serializable, escriba la(s) planificación(es) en serie equivalente(s).

T_1 : $R_1(X)$; $R_1(Z)$; $W_1(X)$;

T_2 : $R_2(Z)$; $R_2(Y)$; $W_2(Z)$; $W_2(Y)$;

T_3 : $R_3(X)$; $R_3(Y)$; $W_3(Y)$;

S_1 : $R_1(X)$; $R_2(Z)$; $R_1(Z)$; $R_3(X)$; $R_3(Y)$; $W_1(X)$; $W_3(Y)$; $R_2(Y)$; $W_2(Z)$; $W_2(Y)$;

S_2 : $R_1(X)$; $R_2(Z)$; $R_3(X)$; $R_1(Z)$; $R_2(Y)$; $R_3(Y)$; $W_1(X)$; $W_2(Z)$; $W_3(Y)$; $W_2(Y)$;

T1: select * from personas where color_pelo='RUBIO' ;

T2: update personas set salario=salario*1.1 ;

T3: update personas set salario=0 where color_pelo='RUBIO' ;
update personas set salario=20000 where color_pelo='NEGRO' ;

Tabla personas (DNI, SALARIO,COLOR_PELLO)

1	3000	RUBIO
2	4000	MORENO
3	5000	RUBIO
4	6000	NEGRO

Determina si los siguientes planes serían serializables:

P1: T1 lee tuplas 1 y 2, T2 escribe tuplas 1 y 2, T3 escribe tuplas 1 y 2, T1 lee tuplas 3 y 4, T2 escribe tuplas 3 y 4, T3 escribe las tuplas 3 y 4.

P2: T1 lee la tupla 1, T2 escribe tuplas 1, 2 y 3, T1 lee la tupla 2, T2 escribe la tupla 4, T3 escribe las tuplas 1, 2, 3 y 4, T1 lee las tuplas 3 y 4.

En el siguiente ejemplo, se utilizan cerrojos. ¿Obtenemos, gracias a ellos, planes serializables?

T1 lock_compartido(Y) read(Y) unlock(Y) lock_exclusivo(X) read(X) X:=X+Y write(X) unlock(X)	T2 lock_compartido(X) read(X) unlock(X) lock_exclusivo(Y) read(Y) Y:=Y+X write(Y) unlock(Y)
--	--

valores iniciales: X=20, Y=30

T1
read(X)
X:=X+1
write(X)

T2
X:=5
write(X)

T3
read(X)
X:=X+5
write(X)

inicio: X=10

¿Es serializable el siguiente plan? ¿Es posible, según el protocolo de bloqueo en 2 fases?

- 1.- T1 hace read(X) $X=10$
- 2.- T2 hace write(X) *escribe* $X=5$
- 3.- T3 hace read (X) *y lee* $X=5$
- 4.- T1 hace write(X) *y escribe* $X=11$
- 5.- T3 hace write(X) *y escribe* $X=10$

¿Puede haber algún problema?:

T1	T2
lock_exclusivo(X)	lock_exclusivo(Y)
lock_exclusivo(Y)	lock_exclusivo(X)
read(X); read(Y);	read(X); read(Y),...

¿Puede haber algún problema?:

T1	T2
lock_exclusivo(X)	lock_exclusivo(X)
lock_exclusivo(Y)	lock_exclusivo(Y)
read(X); read(Y);	read(X); read(Y),....

¿Alguna conclusión?

¿Puede haber algún problema con las siguientes sentencias?:

```
select *  
from billetes  
where estado='Libre';  
update billetes  
set estado='Reservado'  
where num=17;  
commit
```

Si es así, ¿cómo podría solucionarse?

¿Efecto de las siguientes transacciones?:

- 1) **insert into R values (1, 2);**
rollback;
insert into R values (3, 4);
commit;

- 2) **insert into R values (5, 6);**
savepoint my_sp_1;
insert into R values (7, 8);
savepoint my_sp_2;
insert into R values (9, 10);
rollback to my_sp_1;
insert into R values (11, 12);
commit;