

## Práctica 04

Sobre elaboración de planificaciones concurrentes serializables y recuperables

### PARTE 1: TRANSACCIONES

#### Ejercicio 1

Dadas las siguientes tres transacciones T1, T2 y T3:

T1	T2	T3
leer (b)	c = 400	leer (c)
leer (a)	escribir (c)	leer (b)
b = a + b	b = 500	leer (a)
escribir (b)	escribir (b)	a = a + 100
a = b / 2	a = 100	escribir (a)
escribir (a)	escribir (a)	b = b + 1
		escribir (b)

Partiendo de la planificación secuencial T1 -> T2 -> T3, elabore una planificación secuenciable en términos de conflictos y en términos de vistas

#### Ejercicio 2

Dadas las siguientes transacciones T1, T2, T3 y T4:

T1	T2	T3	T4
A=50	C=20	leer (B)	B=300
escribir (A)	escribir (C)	leer (C)	escribir (B)
D=A*3	leer (A)	D=B+2*C	D=B/2
escribir (D)	leer (D)	escribir (D)	escribir (D)
leer (F)	F=A+D-C	C=D+2*B	
	escribir (F)	escribir (C)	
	D=F*2	F=C*4	
	escribir (D)	escribir (F)	

Donde los valores de partida (iniciales) de cada elemento de datos son:

A = 300  
B = 400  
C = 50  
D = 200  
F = 0

Suponga que estas 4 transacciones se ejecutan concurrentemente y necesitamos establecer una planificación para su ejecución que sea secuenciable. Asumiendo que partimos de la planificación secuencial T1 -> T2 -> T3 -> T4. Realice:

1. Elabore una planificación concurrente que sea secuenciable en términos de conflictos y vistas.

- Al final verifique que la planificación secuencial y la planificación secuenciable resultante son equivalentes, para ello realizar una prueba de escritorio para cada planificación y comprobar que los valores finales de cada elemento de dato son los mismos en ambas planificaciones.

### Ejercicio 3

Dadas las siguientes transacciones:

T1	T2	T3	T4
Read (X)	Read (Y)	Read (X)	Read (W)
Read (Y)	Write (Y)	Read (Y)	Write (X)
Read (Z)	Read (Z)	Write (Y)	Read (Y)
Write (Z)	Read (W)	Write (X)	
Write (U)			

Se pide:

- Partir de la planificación secuencial  $T1 > T2 > T3 > T4$
- Elaborar una planificación **secuenciable**
- Verificar que la planificación **sea recuperable**
- Verificar que la planificación **evite retrocesos en cascada**

### Ejercicio 4

Dadas las siguientes tres transacciones T1, T2, T3, T4 y T5:

T1	T2	T3	T4	T5
Leer (T)	Leer (W)	Escribir (W)	Leer (W)	Escribir (Q)
Leer (U)	Escribir (T)	Leer (T)	Leer (S)	Leer (T)
Escribir (U)	Leer (U)	Leer (U)	Escribir (U)	Escribir (W)
Escribir (V)	Escribir (S)	Leer (V)		Escribir (T)
		Escribir (V)		

Partiendo de la planificación secuencial  $T1 > T2 > T3 > T4 > T5$

- Elaborar una **planificación secuenciable**. Aplicar todos los conceptos de secuencialidad en cuanto a conflictos y en cuanto a vistas, de manera que se agoten todas las intercalaciones posibles.
- Verificar que la planificación anterior sea recuperable. Realizar los cambios necesarios y obtener una **planificación recuperable**.
- Verificar que la planificación del punto 2 evite retrocesos en cascada. Si es necesario realizar los cambios necesarios y obtener una **planificación que evite retrocesos en cascada**.

Nota. - Como resultado al final deben presentarse 4 planificaciones:

- Planificación secuencial
- Planificación secuenciable

- C. Planificación secuenciable recuperable
- D. Planificación secuenciable recuperable sin retroceso en cascada

## **PARTE 2: PREGUNTAS DE CONSULTA Y REPASO**

1. Explique ¿qué se entiende por transacción de datos? y ¿por qué son tan importantes para la integridad de la base de datos?
2. Con algún ejemplo propio explique los problemas que pueden ocurrir en un entorno multiusuario cuando se producen accesos simultáneos a la base de datos.
3. ¿Qué es un mecanismo bloqueo en bases de datos, y para que sirve?
4. Consulte cuales son los comandos SQL que usan para el manejo de transacciones en 3 bases de datos relacionales distintas a Oracle y MySQL (DB2, SQL Server, PostgreSQL, SQLite, etc.). Escoja usted los 3 motores a consultar.

.-