## Ejercicios de Transacciones y Control Concurrencia

===== EJ-TCC0 ===================================
---

Qué pasaría si un trigger actualiza filas y después el procedimiento que provocó su ejecución abortara e hiciera un rollback?

Describe la causa y el problema producido por la concurrencia en esta situación (SIN Control Concurrencia):

$T_1$	T <sub>2</sub>
Leer(X)	
X := X - N	
	Leer(X)
	X := X + M
Escribir(X)	
Leer(Y)	
	Escribir(X)
Y := Y + K	

Sean dos transacciones de modo que: (CON Control Concurrencia)

- La primera hace la transferencia de 100 € de una cuenta A a una cuenta B y
- La segunda hace un reintegro de 20 € de cuenta B y un reintegro de 50 € de la cuenta A.

Suponiendo que se pueden ejecutar simultáneamente varias instancias de estas transacciones, se pide:

- a) Describir, mediante una secuencia de operaciones, una situación de interbloqueo.
- b) Describir, mediante una secuencia de operaciones, una situación de datos inseguros

Suponiendo que las transacciones A y B se ejecutan en el nivel de aislamiento READ COMMITTED. Que valores devuelve cada SELECT ?

```
INSERT
INTO Primes VALUES (2);

SELECT * FROM A.Primes;
INSERT INTO A.Primes VALUES (2003);
SELECT * FROM A.Primes;

COMMIT;
SELECT * FROM A.Primes;

SELECT * FROM Primes;
```

Ejemplos de Teoría Héctor Gómez Gauchía

Dadas las tablas T1, T2 y T3 compartidas por todos los usuarios(tienen permisos) y la tabla T4 que solo pertenece al usuario (los otros usuarios no tienen permisos) que está ejecutando un bloque PL/SQL con la siguiente secuencia de instrucciones:

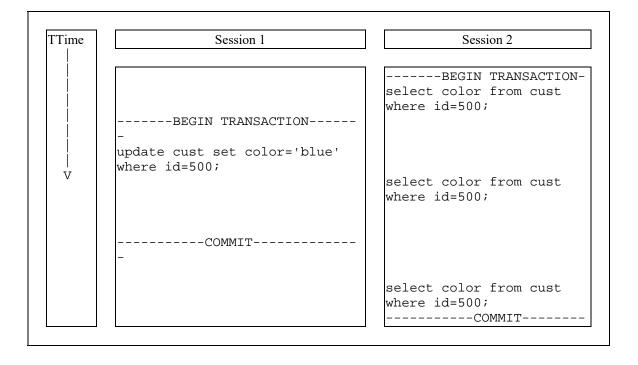
```
lock table T1 in exclusive mode;
select .... from T1 .....; --1
update table T2 ....; --2
insert into T3 ...; --3
commit;
select .... from T1 .....; --4
update table T2 ....; --5
commit;
lock table T1 in exclusive mode;
select ... from T1 ....; --6
lock table T4 in share mode;
select ... from T4 ....; --7
select ... from T4 ....; --7
```

Se desea ejecutar todas las instrucciones en *una sola* transacción sin permitir que otras transacciones puedan alterar el resultado durante la ejecución. Quiero permitir que otras transacciones puedan consultar las tablas que *no* actualizo. Para ello, modifica o incluye los *commit* y *lock* necesarios. Además, borra los que sobren.

## 

Dada la tabla **cust (id, color)** y las dos sesiones concurrentes ambas sobre dicha tabla. Ambas tienen el nivel de aislamiento READ COMMITTED. Responde a las siguientes preguntas:

- a) Cuál es el resultado de cada consulta? (asume que color = 'red' al comienzo)
- **b)** Qué problema has observado?
- c) Cómo resuelves el problema manejando los niveles de aislamiento?



Ejemplos de Teoría Héctor Gómez Gauchía