**Práctica 4: Bloqueos y Transacciones**

**Alumnos:**

* **Miguel Ángel Pérez Polo**
* **Miguel Ángel García Solano**

**Contenido del archivo comprimido:**

* Documentacion.pdf
* GIIC22.sql

Crea la tabla de cuentas:

cuentas1 para no borrar la tabla cuentas

CREATE TABLE cuentas1 (

numero number primary key,

saldo number not null

);

INSERT INTO cuentas1 VALUES (123, 400);

INSERT INTO cuentas1 VALUES (456, 300);

COMMIT;

Se abre otro SQL Developer y nos conectamos (2 sesiones simultáneas)

SET AUTOCOMMIT OFF

1. Abrir dos SQL Developer y conectarse a la base de datos desde ambos con tu usuario y contraseña. Llamaremos a dichas conexiones T1 y T2, y cada una poseerá sus propios bloqueos.
2. Poner SET AUTOCOMMIT OFF. Inhabilita la autoconfirmación de SQLDeveloper en Herramientas/Preferencias/Base de Datos/Avanzada. Deshabilitar ‘Confirmación Automática’.

**1. Bloqueos (update vs select)**

3. Desde la T1 ver el saldo de la cuenta 123. ¿Qué se ve? --> **400**

SELECT \*

FROM cuentas1 b

where numero=123

4. Desde T1 aumenta 100 euros el saldo de la cuenta 123.

update cuentas1

set saldo = saldo+100

where numero = 123

5. Desde la T1 ver el saldo de la cuenta 123. ¿Qué se ve? --> **500**

6. Desde la T2 ver el saldo de la cuenta 123. ¿Qué se ve? --> **400**

7. Desde la T1: COMMIT;

commit

8. Desde la T2 ver el saldo de la cuenta 123. ¿Qué se ve? --> **500**

Conclusión: T2 no ve los cambios de T1 hasta que no los confirme

**2. Bloqueos (update vs update)**

1. Desde T1 aumenta 100 euros el saldo de la cuenta 123.

update cuentas1

set saldo = saldo+100

where numero = 123

1. Desde la T2 aumenta 200 euros el saldo de la cuenta 123. ¿Se puede? --> **No**. ¿qué le pasa a la T2? --> **Se queda a la espera**

update cuentas1

set saldo = saldo+200

where numero = 123

3. T1: COMMIT; ¿qué le pasa a la T2? --> **T2 pone 800 y en T1 pone 600 por lo que también hago un commit en la T2 para que en ambas ponga 800**

commit

**3. Bloqueos (Deadlock)**

1. Desde T1 aumenta 100 euros el saldo de la cuenta 123.

update cuentas1

set saldo = saldo+100

where numero = 123

1. Desde T2 aumenta 200 euros el saldo de la cuenta 456.

update cuentas1

set saldo = saldo+200

where numero = 456

1. Desde T1 aumenta 300 euros el saldo de la cuenta 456. --> **A la espera**

update cuentas1

set saldo = saldo+300

where numero = 456

1. Desde T2 aumenta 400 euros el saldo de la cuenta 123. ¿Qué ha pasado?

update cuentas1

set saldo = saldo+400

where numero = 123

Espera un poco y mira en la consola si se ha detectado un ‘deadlock’. Confirma dicha transacción para que pueda terminar la otra. -->

set saldo = saldo+300

where numero = 456

Informe de error:

Error SQL: ORA-00060: detectado interbloqueo mientras se esperaba un recurso

00060. 00000 - "deadlock detected while waiting for resource"

\*Cause: Transactions deadlocked one another while waiting for resources.

\*Action: Look at the trace file to see the transactions and resources involved. Retry if necessary.

Hago un commit en T1 y luego en T2

**4. Niveles de aislamiento**

Explorar y explicar el comportamiento de las siguientes transacciones T1 y T2, en dos sesiones de SQL Plus con:

1. En T1: ALTER SESSION SET ISOLATION\_LEVEL=SERIALIZABLE; --> **faltaba el =**
2. En T1 : SELECT SUM(saldo) FROM cuentas1 --> **1800**
3. En T2 : UPDATE cuentas1 SET saldo=saldo +100; COMMIT;
4. En T1 : SELECT SUM(saldo) FROM cuentas1 --> **1800**

¿Qué ha pasado? --> **En T1 saldo es 1800 y en T2 es 2000 porque se realizó un bloqueo en el punto 1**

1. En T1: ALTER SESSION SET ISOLATION\_LEVEL=READ COMMITTED --> **faltaba el =**
2. En T1 : SELECT SUM(saldo) FROM cuentas1 --> **saldo 2000**
3. En T2 : UPDATE cuentas1 SET saldo=saldo +100; COMMIT;
4. En T1 : SELECT SUM(saldo) FROM cuentas1 --> **2200**

¿Qué ha pasado?. --> **Que se han realizado los cambios correctamente**

Explicar si hay alguna diferencia según los niveles de aislamiento. --> **con READ COMMITTED realiza el commit automáticamente**