APARTADO 1

SALDO desde T2: 400

SALDO desde T2: 500

APARTADO 2

No se puede, está en continuo funcionamiento

Al hacer el commit en T1 me sentí como Dios jugando a ser el rey porque T2 terminaba y ejecutaba el update.

SALDO desde T1: 600

SALDO desde T1: 800

APARTADO 3

Igual que en la primera parte del apartado dos no se ejecutan las instrucciones 3 y 4 puesto que las instrucciones 1 y 2 han actualizado las mismas cuentas y todavía no se ha hecho commit.

Se produce un interbloqueo en la instrucción 3 al hacer el commit en T1; puesto que este commit permite la realización de la instrucción 4.

La tabla acaba con los datos:

Cuenta 123----- > 1300

Cuenta 456----- > 500

APARTADO 4

1-Session alterado.

2-SELECT SUM(saldo) FROM cuentas; -------> 1800

3- 2 filas actualizadas. Confirmación terminada.

4- SUM(SALDO)----- >1800

La suma del saldo permanece invariante, aunque a primera vista hubiéramos pensado que tomaría el valor 2000; pero al aislar T1 del resto, no le afectan los cambios confirmados de otras transacciones.

5- Session alterado.

6- SUM(SALDO)----- >2000 (se consideran ya las actualizaciones de T2)

7-2 filas actualizadas. Confirmación terminada.

8- SUM(SALDO)----- >2200

APARTADO 5

1-Table BUTACAS creado. Table RESERVAS creado.

Sequence SEQ\_BUTACAS creado. Sequence SEQ\_RESERVAS creado.

2- 3 filas insertadas. Confirmación terminada.

3-

INFO: Se intenta reservar.

Procedimiento PL/SQL terminado correctamente.

SCRIPT\_COL

--------------------------------

"C:\hlocal\MlC\preguntar.sql"

V\_ERROR

--------------------------------------------------------------------------------

false

'¿Confirmar la reserva?'

s

INFO: Localidad reservada.

Procedimiento PL/SQL terminado correctamente.

Confirmación terminada.

4-

ERROR: La localidad ya está reservada.

Procedimiento PL/SQL terminado correctamente.

SCRIPT\_COL

--------------------------------

"C:\hlocal\MlC\preguntar.sql"

V\_ERROR

--------------------------------------------------------------------------------

true

'¿Confirmar la reserva?'

s

INFO: No se ha reservado la localidad.

Procedimiento PL/SQL terminado correctamente.

Confirmación terminada.

5- ERROR: No existe esa localidad.

Procedimiento PL/SQL terminado correctamente.

SCRIPT\_COL

--------------------------------

"C:\hlocal\MlC\preguntar.sql"

V\_ERROR

--------------------------------------------------------------------------------

true

'¿Confirmar la reserva?'

s

INFO: No se ha reservado la localidad.

Procedimiento PL/SQL terminado correctamente.

Confirmación terminada.

6-

7-

INFO: Se intenta reservar.

Procedimiento PL/SQL terminado correctamente.

SCRIPT\_COL

--------------------------------

"C:\hlocal\MlC\preguntar.sql"

V\_ERROR

--------------------------------------------------------------------------------

false

'¿Confirmar la reserva?'

s

INFO: Localidad reservada.

Procedimiento PL/SQL terminado correctamente.

Confirmación terminada.

8- Me permite confirmar la reserva del punto 6.

INFO: Se intenta reservar.

Procedimiento PL/SQL terminado correctamente.

SCRIPT\_COL

--------------------------------

"C:\hlocal\MlC\preguntar.sql"

V\_ERROR

--------------------------------------------------------------------------------

false

'¿Confirmar la reserva?'

s

INFO: Localidad reservada.

Procedimiento PL/SQL terminado correctamente.

Confirmación terminada.

9- SET TRANSACTION ISOLATION LEVEL = SERIALIZABLE;

Me permite hacer la reseva propuesta en el punto 7

Transaction ISOLATION correcto.

INFO: Se intenta reservar.

Procedimiento PL/SQL terminado correctamente.

SCRIPT\_COL

--------------------------------

"C:\hlocal\MlC\preguntar.sql"

V\_ERROR

--------------------------------------------------------------------------------

false

'¿Confirmar la reserva?'

s

INFO: Localidad reservada.

Procedimiento PL/SQL terminado correctamente.

Confirmación terminada.

Al confirmar la reserva del punto 6 salta el error:

begin

if '&v\_confirmar'='s' and :v\_error='false' then

insert into reservas values (Seq\_Reservas.NEXTVAL,'&v\_evento','&v\_fila','&v\_columna');

dbms\_output.put\_line('INFO: Localidad reservada.');

else

dbms\_output.put\_line('INFO: No se ha reservado la localidad.');

end if;

end;

Informe de error -

ORA-08177: no se puede serializar el acceso para esta transacción

ORA-06512: en línea 3

08177. 00000 - "can't serialize access for this transaction"

\*Cause: Encountered data changed by an operation that occurred after

the start of this serializable transaction.

\*Action: In read/write transactions, retry the intended operation or

transaction.

Confirmación terminada

SET AUTOCOMMIT OFF

CREATE TABLE cuentas (

numero number primary key,

saldo number not null

);

INSERT INTO cuentas VALUES (123, 400);

INSERT INTO cuentas VALUES (456, 300);

COMMIT;

-- 1

UPDATE cuentas SET saldo=saldo + 100 WHERE numero=123;

COMMIT;

-- 2

UPDATE cuentas SET saldo=saldo + 100 WHERE numero=123;

COMMIT;

select saldo from cuentas WHERE numero=123;

select saldo from cuentas WHERE numero=123;

-- 3

UPDATE cuentas SET saldo=saldo + 100 WHERE numero=123;

UPDATE cuentas SET saldo=saldo + 300 WHERE numero=456;

COMMIT;

-- 4

ALTER SESSION SET ISOLATION\_LEVEL = SERIALIZABLE;

SELECT SUM(saldo) FROM cuentas;

SELECT SUM(saldo) FROM cuentas;

ALTER SESSION SET ISOLATION\_LEVEL = READ COMMITTED;

SELECT SUM(saldo) FROM cuentas;

SELECT SUM(saldo) FROM cuentas;

--5

CREATE TABLE butacas(id number(8) primary key,

evento varchar(30),

fila varchar(10),

columna varchar(10)) ;

CREATE TABLE reservas(id number(8) primary key,

evento varchar(30),

fila varchar(10),

columna varchar(10)) ;

CREATE SEQUENCE Seq\_Butacas INCREMENT BY 1 START WITH 1

NOMAXVALUE;

CREATE SEQUENCE Seq\_Reservas INCREMENT BY 1 START WITH 1

NOMAXVALUE;

INSERT INTO butacas VALUES (Seq\_Butacas.NEXTVAL,'Circo','1','1');

INSERT INTO butacas VALUES (Seq\_Butacas.NEXTVAL,'Circo','1','2');

INSERT INTO butacas VALUES (Seq\_Butacas.NEXTVAL,'Circo','1','3');

COMMIT;