

Prácticas de

Administración de Bases de Datos

Grado en Ingeniería Informática

**PRÁCTICA 7**

Gestión y control de la concurrencia (Niveles de aislamiento)

**Ejercicios**

1. Inicia dos sesiones (con el mismo usuario) para los clientes A y B, y ejecuta las siguientes sentencias. Observar el resultado y comentar los resultados obtenidos.

**DROP TABLE Accounts; CREATE TABLE Accounts (**

**acctID INTEGER NOT NULL PRIMARY KEY,**

**balance INTEGER NOT NULL**

**CONSTRAINT unloanable\_account CHECK (balance >= 0)**

**);**

**INSERT INTO Accounts (acctID, balance) VALUES (101, 1000); INSERT INTO Accounts (acctID, balance) VALUES (202, 2000); COMMIT;**

**-- Cliente A:**

**SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;**

**SELECT \* FROM Accounts WHERE balance > 500;**

**-- Cliente B:**

**UPDATE Accounts SET balance = balance - 500 WHERE acctID = 101; UPDATE Accounts SET balance = balance + 500 WHERE acctID = 202; COMMIT;**

**-- Cliente A:**

**SELECT \* FROM Accounts WHERE balance > 500; COMMIT;**

**SELECT \* FROM Accounts WHERE balance > 500;**

Texto

Descripción generada automáticamente

Primeros comandos.

Texto

Descripción generada automáticamente

Cliente A comandos, como es serializable no cambia el select hasta que hace commit.

Texto

Descripción generada automáticamente

Cliente B comandos

Ahora ejecuta las siguientes sentencias y analiza su resultado.

**-- Cliente A:**

**INSERT INTO Accounts (acctID,balance) VALUES (303,3000); INSERT INTO Accounts (acctID,balance) VALUES (404,4000); COMMIT;**

**SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;**

**SELECT \* FROM Accounts WHERE balance > 2500;**

**-- Cliente B:**

**DELETE FROM Accounts WHERE acctID = 404; SELECT \* FROM Accounts;**

**-- Cliente A:**

**SELECT \* FROM Accounts WHERE balance > 2500;**

**INSERT INTO Accounts (acctID,balance) VALUES (505,5000);**

**-- Cliente B: COMMIT;**

Texto

Descripción generada automáticamente Texto

Descripción generada automáticamente

**-- Cliente A:**

**SELECT \* FROM Accounts WHERE balance > 2500;**

**INSERT INTO Accounts (acctID,balance) VALUES (505,5000); COMMIT;**

**SELECT \* FROM Accounts WHERE balance > 2500;**

**INSERT INTO Accounts (acctID,balance) VALUES (505,5000); COMMIT;**

**SELECT \* FROM Accounts WHERE balance > 2500;**

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

¿Qué hay que hacer para cambiar ISOLATION LEVEL para toda una sesión?

-- Establecer el nivel de aislamiento READ COMMITTED para la sesión actual

SET TRANSACTION ISOLATION LEVEL READ COMMITTED;

1. Inicia dos sesiones (con el mismo usuario) para los clientes A y B. Ejecuta las siguientes sentencias.

**CREATE TABLE TRENES (**

**TIPO VARCHAR(15) PRIMARY KEY, VELOCIDAD INT**

**);**

**CREATE TABLE ENLACES (**

**ID VARCHAR(10) PRIMARY KEY, TIPO REFERENCES TRENES, NUMPASAJEROS INT,**

**RUTA VARCHAR(30)**

**);**

**CREATE TABLE PASAJEROS ( NOMBRE VARCHAR(40),**

**ID REFERENCES ENLACES, PRIMARY KEY(NOMBRE, ID)**

**);**

**INSERT INTO TRENES VALUES ('Talgo AVRIL', 380); INSERT INTO TRENES VALUES ('Talgo 350', 350); INSERT INTO TRENES VALUES ('Talgo XXI', 220);**

**INSERT INTO ENLACES**

**VALUES ('ALVIA 215', 'Talgo 350', 186, 'HUELVA-MADRID'); INSERT INTO ENLACES**

**VALUES ('AVE 154', 'Talgo AVRIL', 298, 'SEVILLA-MADRID'); COMMIT;**

Texto

Descripción generada automáticamente

**-- Cliente A:**

**SELECT TIPO, COUNT(\*) FROM ENLACES GROUP BY TIPO;**

**-- Cliente B:**

**INSERT INTO ENLACES**

**VALUES ('ALVIA 212', 'Talgo 350', 170, 'HUELVA-CORDOBA');**

**-- Cliente A:**

**SELECT TIPO, COUNT(\*) FROM ENLACES GROUP BY TIPO;**

**-- Cliente B:**

**COMMIT;**

**-- Cliente A:**

**SELECT TIPO, COUNT(\*) FROM ENLACES GROUP BY TIPO;**

¿Qué situación se ha producido?

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Lectura no repetible y fantasma.

Ahora ejecuta estas sentencias y analiza el resultado.

**-- Cliente A:**

**DELETE ENLACES WHERE ID = 'ALVIA 212'; COMMIT;**

**SET TRANSACTION READ ONLY;**

**SELECT TIPO, COUNT(\*) FROM ENLACES GROUP BY TIPO;**

**-- Cliente B:**

**INSERT INTO ENLACES**

**VALUES ('ALVIA 212', 'Talgo 350', 170, 'HUELVA-CORDOBA');**

**-- Cliente A:**

**SELECT TIPO, COUNT(\*) FROM ENLACES GROUP BY TIPO;**

**-- Cliente B:**

**COMMIT;**

**-- Cliente A:**

**SELECT TIPO, COUNT(\*) FROM ENLACES GROUP BY TIPO; DELETE ENLACES;**

**COMMIT;**

**SELECT TIPO, COUNT(\*) FROM ENLACES GROUP BY TIPO;**

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Tras ejecutar las siguientes sentencias, ¿se produce algún problema? ¿Por qué?

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamenteNo ocurre error, porque las inserciones aunque son en la misma tabla, no son sobre la misma tupla.

**-- Cliente A:**

**SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;**

**UPDATE ENLACES SET NUMPASAJEROS = NUMPASAJEROS+1 WHERE ID = ‘ALVIA 215’;**

**-- Cliente B:**

**SET TRANSACTION ISOLATION LEVEL SERIALIZABLE; INSERT INTO TRENES VALUES ('Talgo 360', 360);**

**-- Cliente A:**

**COMMIT;**

**-- Cliente B:**

**UPDATE ENLACES SET NUMPASAJEROS = NUMPASAJEROS+1 WHERE ID = 'AVE 154'; COMMIT;**