

Prácticas de Administración de Bases de Datos

Grado en Ingeniería Informática

**PRÁCTICA 6**

Gestión y control de la concurrencia (Transacciones)

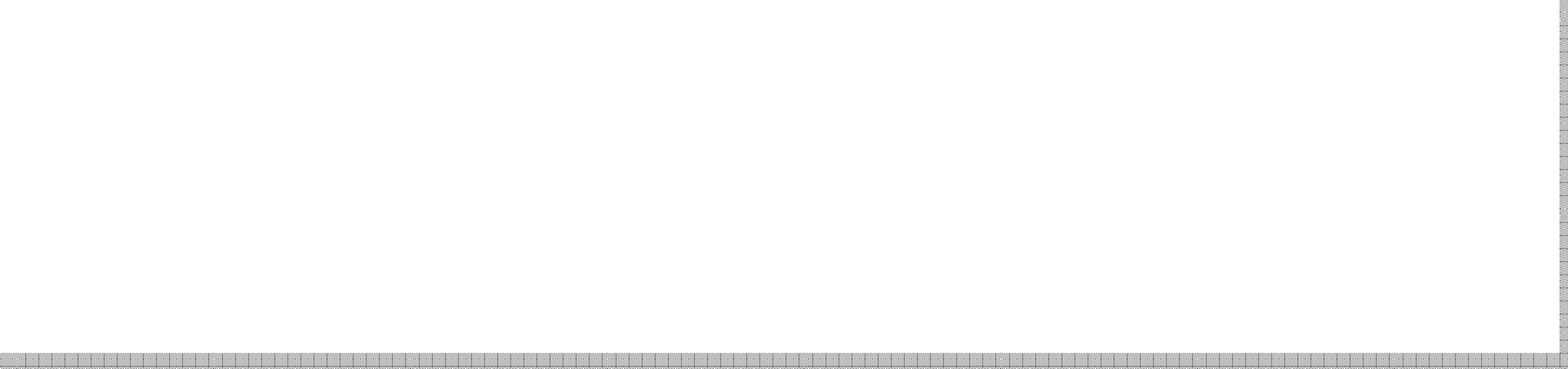
**Ejercicios**

1. Primero creo un nuevo usuario el cual contendrá todas las tablas.

CREATE USER cliente IDENTIFIED BY cliente;

GRANT CONNECT, RESOURCE TO nombre\_usuario;

1. Inicia una sesión y ejecuta las siguientes sentencias. Observa el resultado.



**CREATE TABLE T (id INT PRIMARY KEY, s VARCHAR(30), si NUMBER(5)); INSERT INTO T (id, s) VALUES (1, 'first');**

**COMMIT;**

**INSERT INTO T (id, s) VALUES (2, 'second'); SELECT \* FROM T;**

**ROLLBACK;**

**SELECT \* FROM T;**

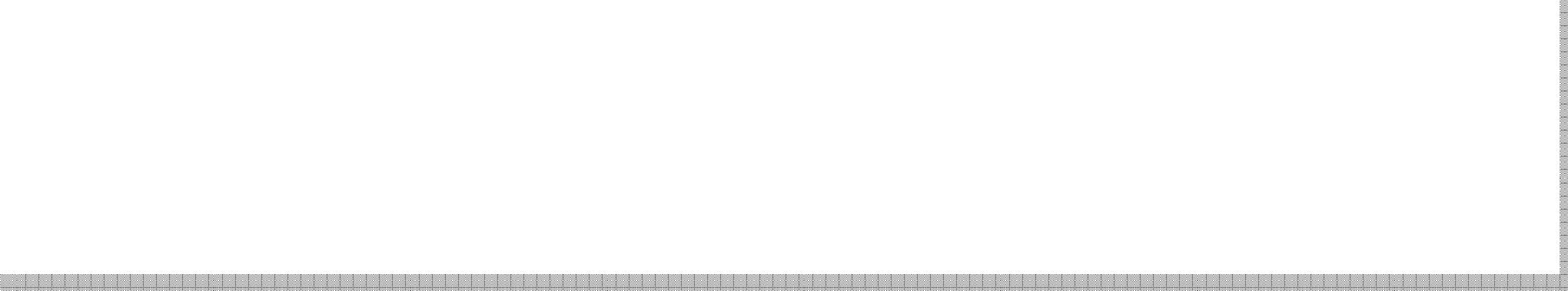
Añade las siguientes sentencias. ¿Cuál ha sido el efecto de ROLLBACK?

¿Ocurrió algún COMMIT implícito?

Texto

Descripción generada automáticamente

No ocurrió ningún commit implícito.



**INSERT INTO T (id, s) VALUES (2, 'Will this be committed?'); CREATE TABLE T2 (id INT);**

**INSERT INTO T2 (id) VALUES (1); SELECT \* FROM T2;**

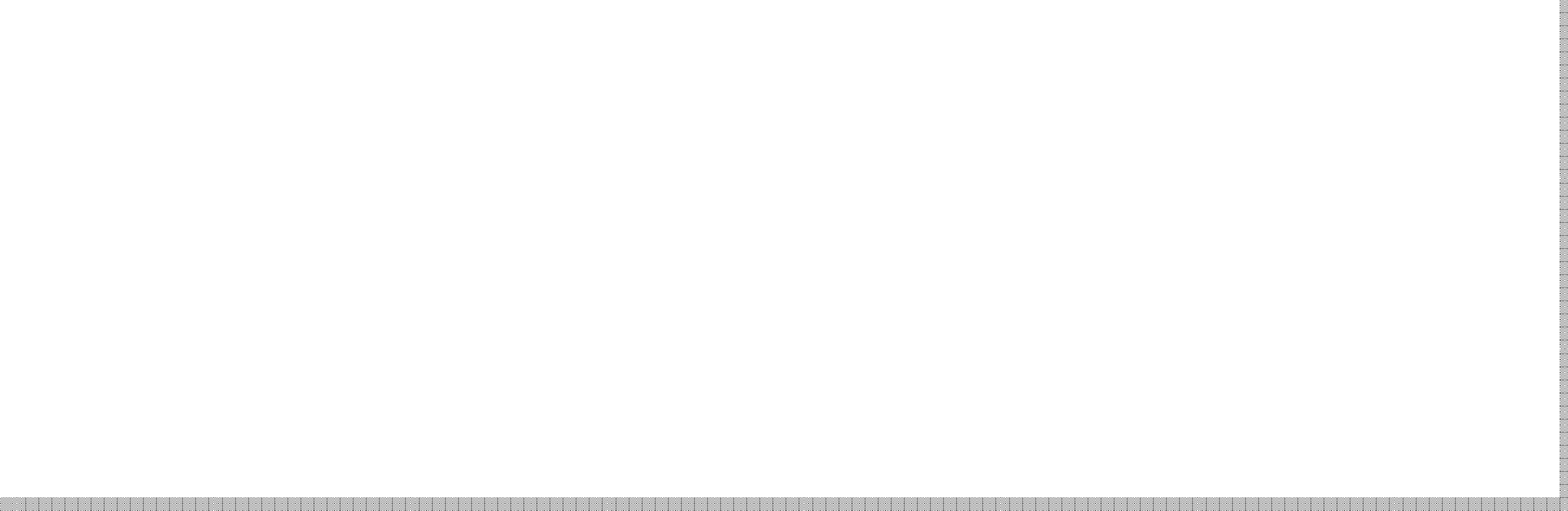
**ROLLBACK;**

Texto

Descripción generada automáticamente

Se crea un commit implícito cuando llama al créate table t2

Añade las siguientes sentencias. ¿Qué puede observarse?



**DELETE FROM T WHERE id > 1; COMMIT;**

**INSERT INTO T (id, s) VALUES (2, 'The test starts by this'); SELECT (1/0) FROM DUAL;**

**UPDATE T SET s = 'foo' WHERE id = 9999; DELETE FROM T WHERE id = 7777;**

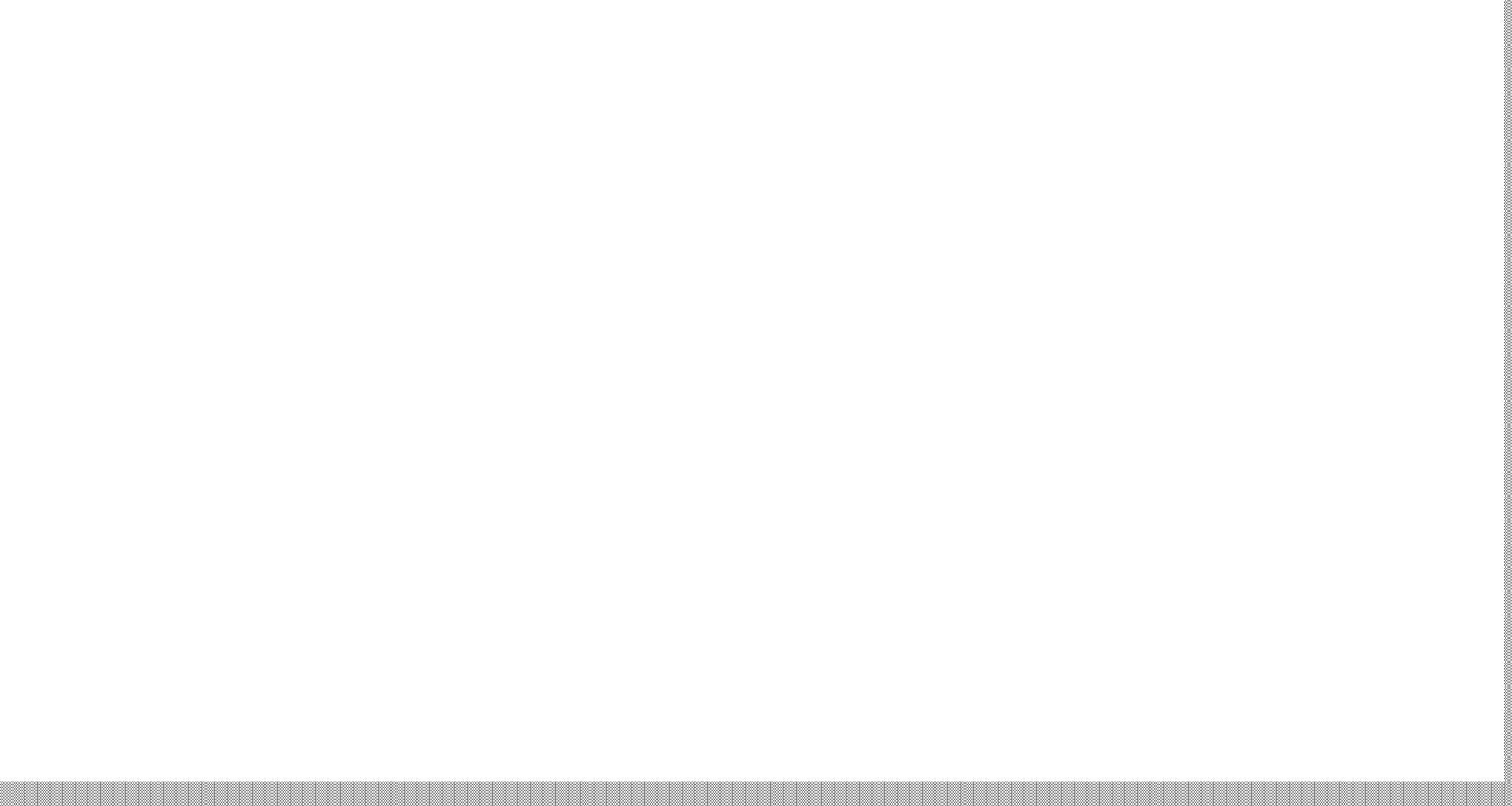
**COMMIT;**

**SELECT \* FROM T;**

Texto

Descripción generada automáticamente

Ahora añade las siguientes sentencias. ¿Qué se observa?



**UPDATE T SET s = 'Otro test' WHERE id = 2; INSERT INTO T (id, s)**

**VALUES (2, 'Soy un duplicado'); INSERT INTO T (id, s)**

**VALUES (3, 'Intentando insertar una cadena muy larga'); INSERT INTO T (id, s, si)**

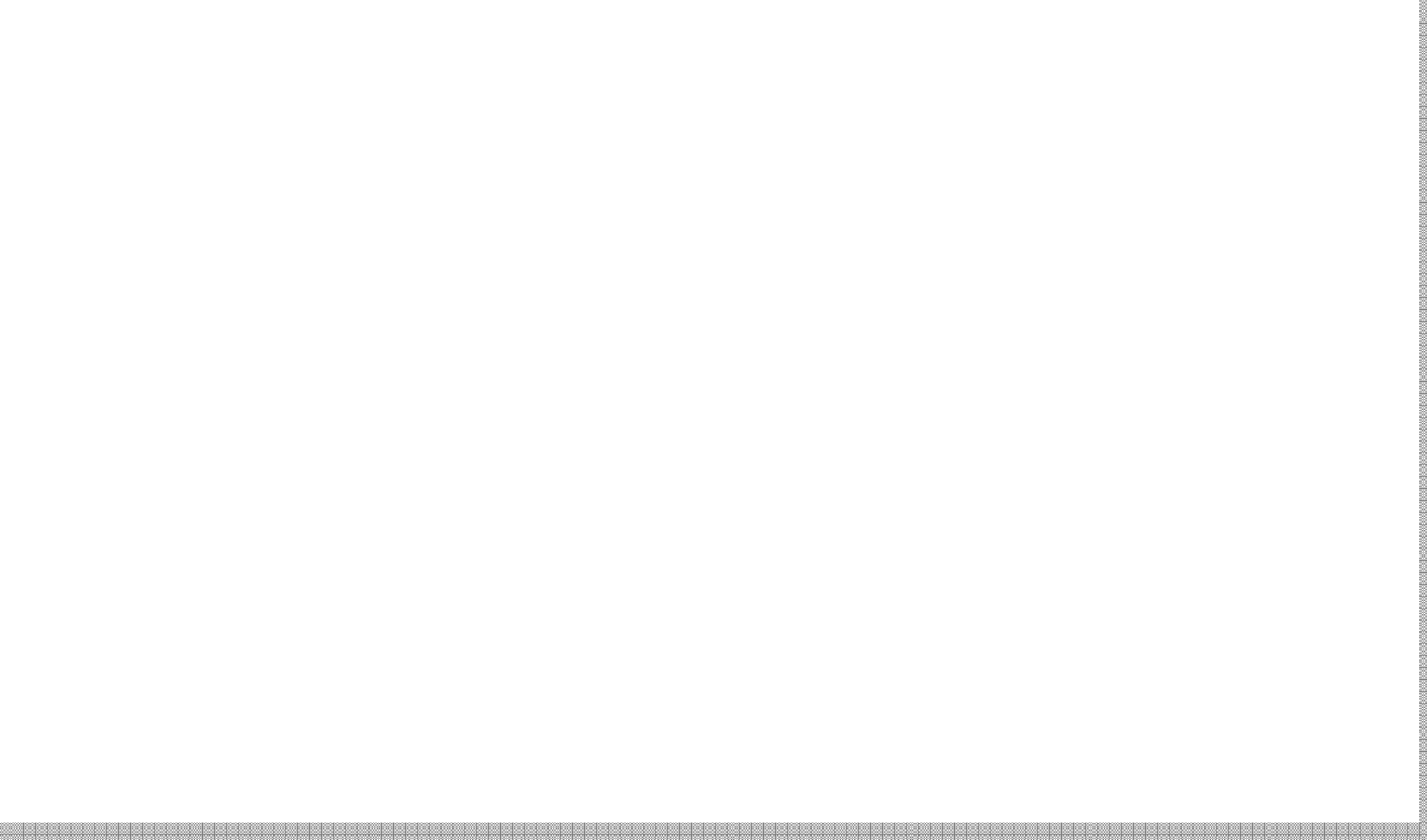
**VALUES (4, 'Se puede un 123456?', 123456); COMMIT;**

**SELECT \* FROM T;**

Texto

Descripción generada automáticamente

1. Ejecuta las siguientes sentencias. Entendiendo que se trata de transferencias de una cuenta a otra.. ¿hay algún problema en lo que se expresa a continuación? ¿Cómo podría solucionarse?



**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;**

**SELECT \* FROM Accounts;**

**UPDATE Accounts SET balance = balance - 100 WHERE acctID = 101; UPDATE Accounts SET balance = balance + 100 WHERE acctID = 202; COMMIT;**

**SELECT \* FROM Accounts;**

**UPDATE Accounts SET balance = balance - 2000 WHERE acctID = 101; UPDATE Accounts SET balance = balance + 2000 WHERE acctID = 202; COMMIT;**

**SELECT \* FROM Accounts;**

Texto

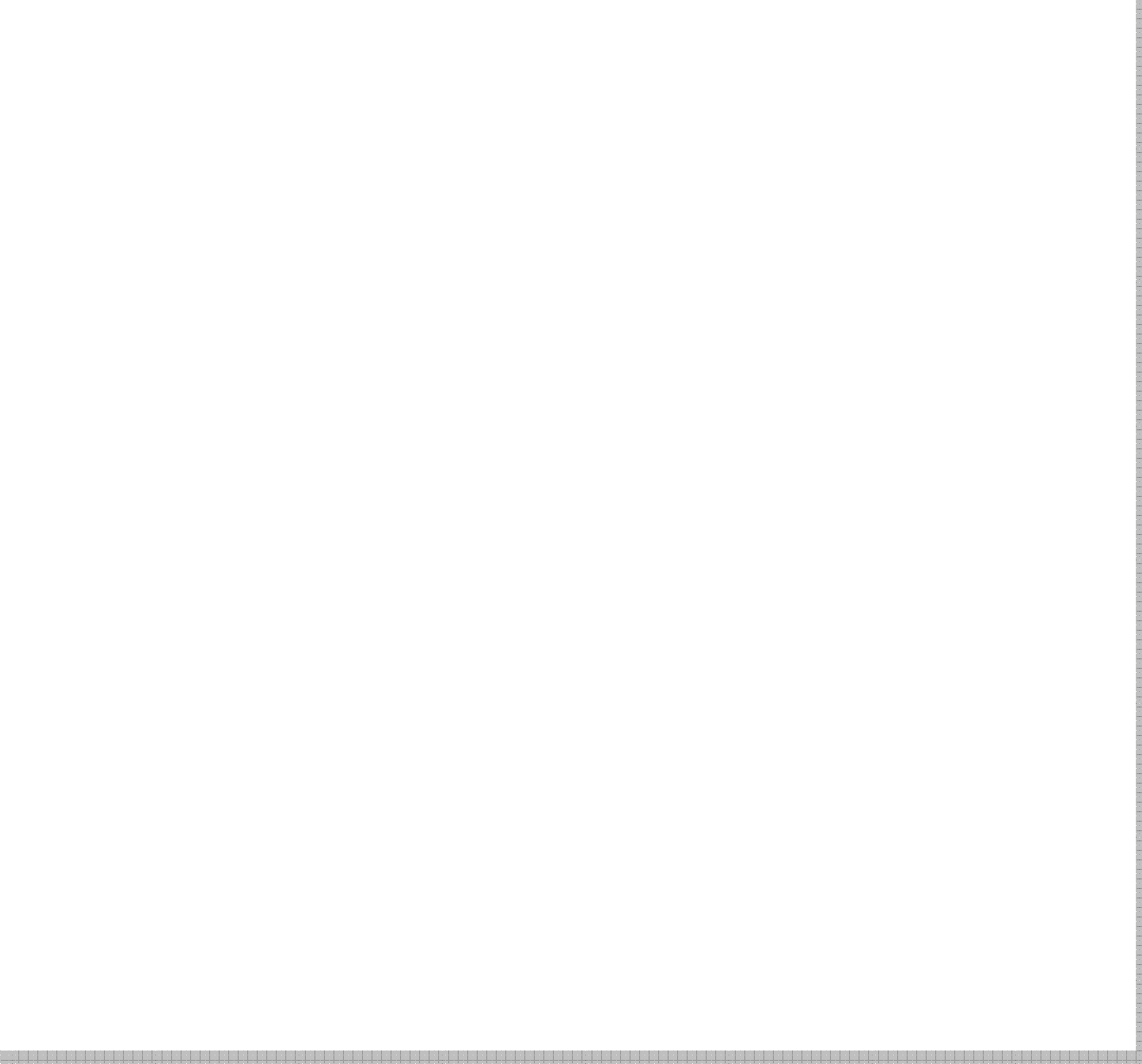
Descripción generada automáticamente Texto

Descripción generada automáticamente

El problema es que estamos quitandole a una cuenta bancaria una cantidad de dinero para ingresarle a otra, pero una de ellas no tiene suficiente como para cumplir la condicion de que sea mayor a 0, asique no debería realizarse ninguna de los dos updates, es decir, deberían ser una transición.

1. Abre dos sesiones en dos ventanas SQL\*Plus, correspondientes a los clientes A y

B. Ejecuta las correspondientes sentencias para cada cliente. ¿Observas alguna anomalía? ¿Alguna sugerencia?



**-- En cualquier cliente:**

**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;**

**SELECT \* FROM Accounts;**

**-- Cliente A comienza:**

**SELECT balance FROM Accounts WHERE acctID = 101;**

**-- se simula la lectura de 1000 en una variable**

**-- Cliente B comienza:**

**SELECT balance FROM Accounts WHERE acctID = 101;**

**-- se simula la lectura de 1000 en una variable**

**-- Cliente A continúa:**

**UPDATE Accounts SET balance = (1000 - 200) WHERE acctID = 101;**

**-- Cliente B continúa:**

**UPDATE Accounts SET balance = (1000 - 500) WHERE acctID = 101;**

**-- Cliente A continúa:**

**SELECT acctID, balance FROM Accounts WHERE acctID = 101; COMMIT;**

**-- Cliente B continúa:**

**SELECT acctID, balance FROM Accounts WHERE acctID = 101; COMMIT;**

Texto

Descripción generada automáticamente

Cliente A

Texto

Descripción generada automáticamente

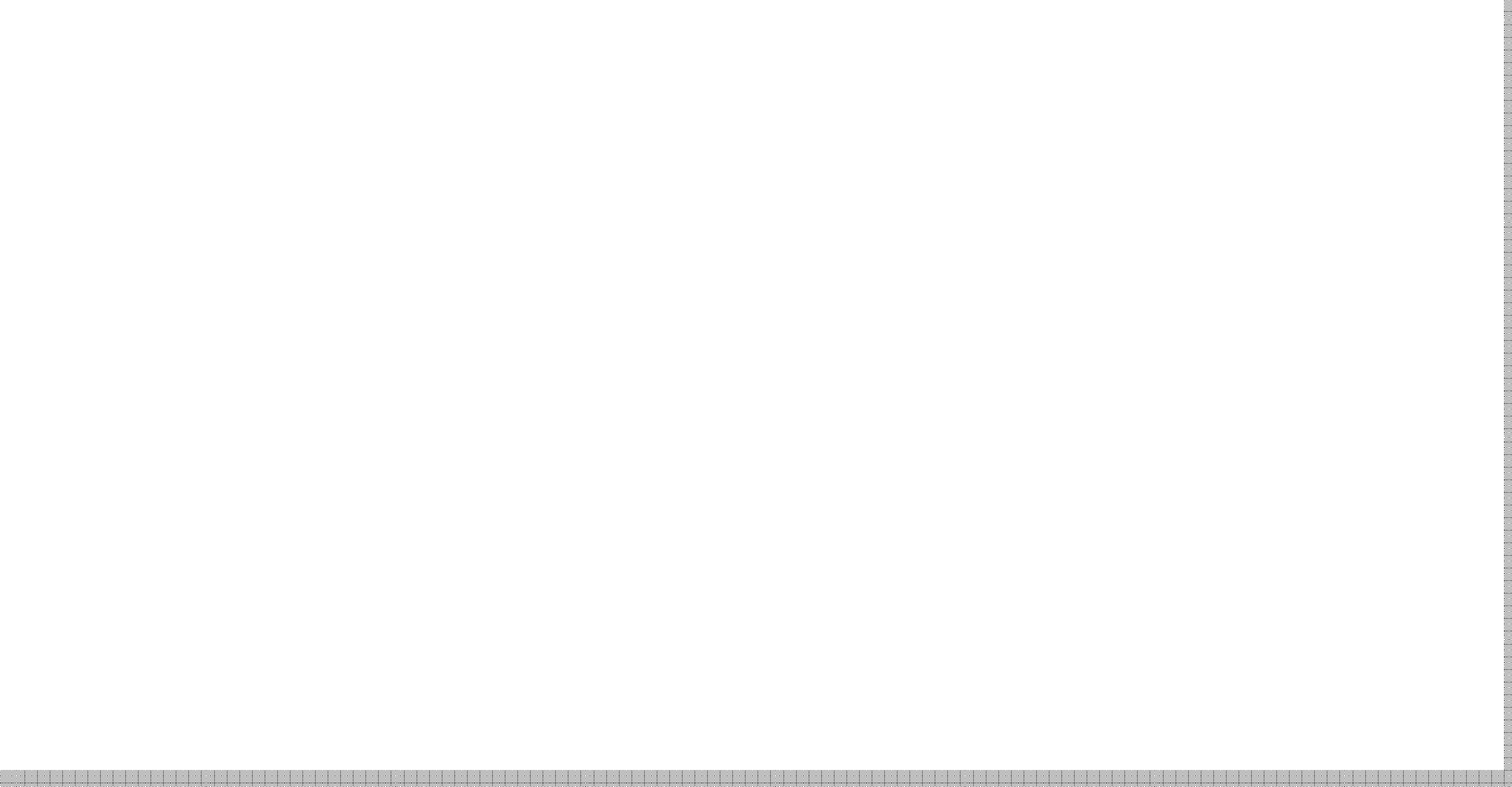
Cliente B

Texto

Descripción generada automáticamente

Hasta que el cliente A no hace Commit no permite al cliente B hacer el update, y se queda esperando.

**Ahora ejecuta las siguientes sentencias. ¿Qué ocurre?**



**-- En cualquier cliente:**

**DELETE FROM Accounts;**

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

**-- El cliente A transfiere 100 euros de la cuenta 101 a la 202**

**-- El cliente B transfiere 200 euros de la cuenta 202 a la 101**

**-- Cliente A:**

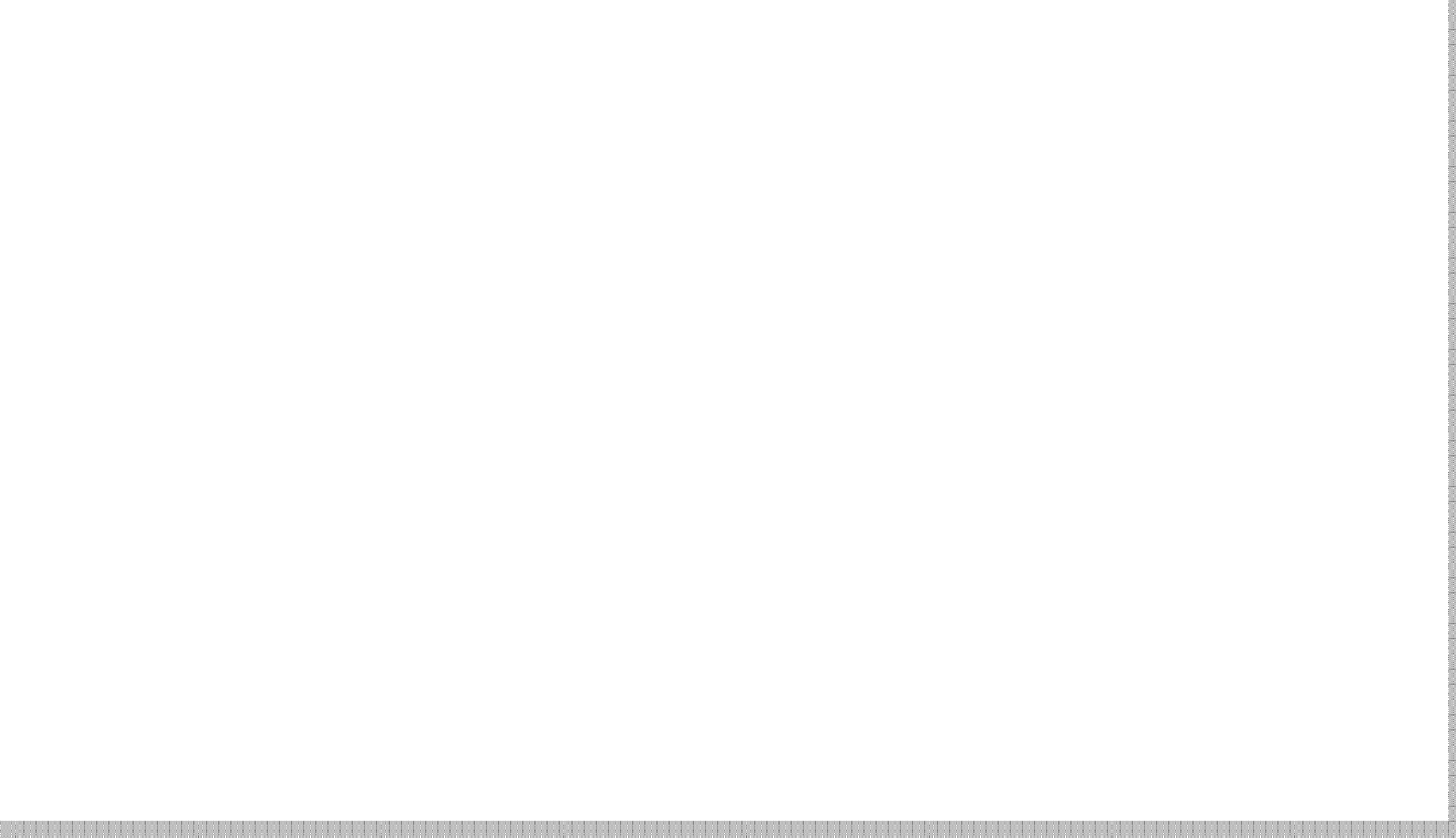
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **UPDATE Accounts** | **SET** | **balance** | **=** | **balance** | **-** | **100** | **WHERE** | **acctID** | **=** | **101;** |
| **-- Cliente B: UPDATE Accounts** | **SET** | **balance** | **=** | **balance** | **-** | **200** | **WHERE** | **acctID** | **=** | **202;** |
| **-- Cliente A: UPDATE Accounts** | **SET** | **balance** | **=** | **balance** | **+** | **100** | **WHERE** | **acctID** | **=** | **202;** |
| **-- Cliente B: UPDATE Accounts** | **SET** | **balance** | **=** | **balance** | **+** | **200** | **WHERE** | **acctID** | **=** | **101;** |
|  |  |  |  |  |  |  |  |  |  |  |

Texto

Descripción generada automáticamente

El cliente B espera a que hagas un commit, siguiendo las instrucciones.

**Ahora ejecuta las siguientes sentencias. ¿Observas alguna anomalía?**



**DELETE FROM Accounts;**

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

**-- Cliente A:**

**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;**

Texto

Descripción generada automáticamente

Texto

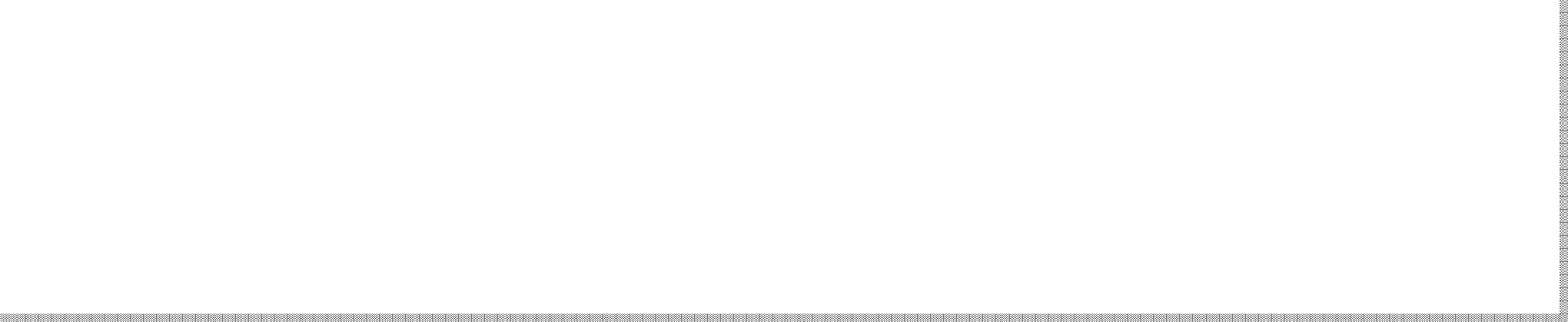
Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Lectura no repetible

1. Ejecuta las siguientes sentencias y analiza su resultado.



**DELETE FROM Accounts;**

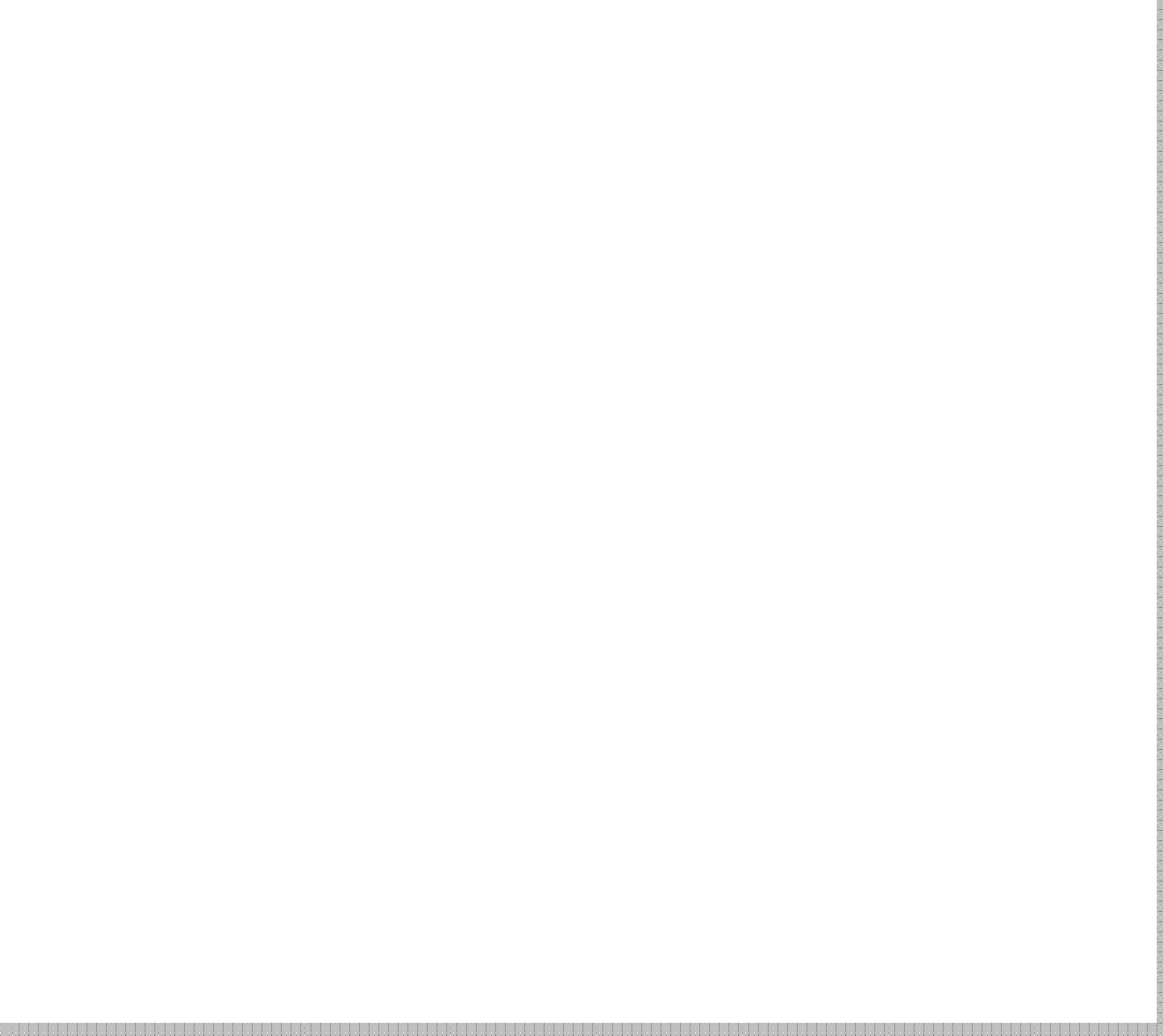
**INSERT INTO Accounts (acctID,balance) VALUES (101,1000); INSERT INTO Accounts (acctID,balance) VALUES (202,2000); COMMIT;**

**SELECT \* FROM Accounts;**

Texto

Descripción generada automáticamente

**A continuación se ejecutan las siguientes instrucciones, Obtenga conclusiones..**



**SELECT NAME, STATUS FROM V$TRANSACTION; SET TRANSACTION NAME 'ingresos\_1';**

**UPDATE Accounts SET balance = balance + 250 WHERE acctID = 101; SAVEPOINT tras\_ingreso\_250;**

**SELECT NAME, STATUS FROM V$TRANSACTION;**

**UPDATE Accounts SET balance = balance + 100 WHERE acctID = 202; SAVEPOINT tras\_ingreso\_100;**

**ROLLBACK TO SAVEPOINT tras\_ingreso\_250; SELECT \* FROM Accounts;**

**SELECT NAME, STATUS FROM V$TRANSACTION;**

**UPDATE Accounts SET balance = balance + 300 WHERE acctID = 202; ROLLBACK;**

**SELECT \* FROM Accounts;**

**SELECT NAME, STATUS FROM V$TRANSACTION;**

**SET TRANSACTION NAME 'ingresos\_2'; SELECT NAME, STATUS FROM V$TRANSACTION;**

**UPDATE Accounts SET balance = balance + 150 WHERE acctID = 101; UPDATE Accounts SET balance = balance + 200 WHERE acctID = 202;**

**SELECT NAME, STATUS FROM V$TRANSACTION; COMMIT;**

**SELECT \* FROM Accounts;**

**SELECT NAME, STATUS FROM V$TRANSACTION;**

Texto

Descripción generada automáticamente con confianza baja

Después de la primera transición.

Imagen que contiene Tabla

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Después del rollback al 250 y el select.

Tabla

Descripción generada automáticamente Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Después del rollback completo.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Antes del commit.

Texto

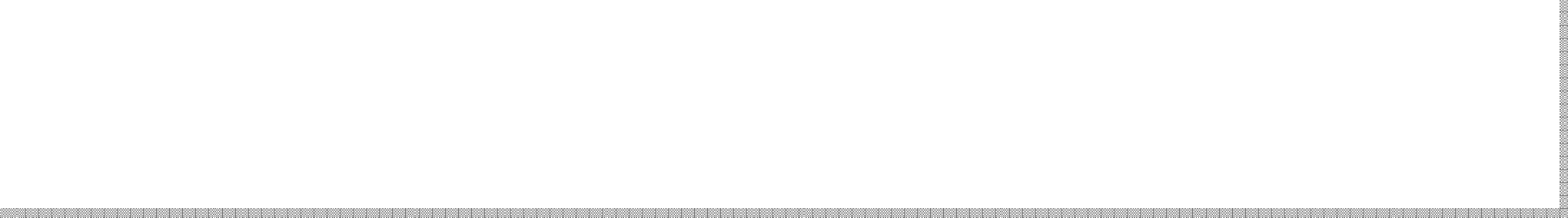
Descripción generada automáticamente

Final.

**A continuación se ejecutan las siguientes instrucciones, ¿se produce algún problema?. Obtenga conclusiones..**

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente Tiene que hacer commit primero



**SET TRANSACTION NAME 'uno';**

**UPDATE Accounts SET balance = balance + 250 WHERE acctID = 101; SET TRANSACTION NAME 'dos';**