LABORATORIO 23: TRANSACCIONES

Angélica Güemes
A01421467

Revisa el contenido de la tabla clientes_banca desde la ventana que inicializaste como la segunda sesión.

SELECT * FROM CLIENTES_BANCA

¿Que pasa cuando deseas realizar esta consulta?

Puedo visualizar la tabla con los mismos datos que agregue en la otra conexion.

Inserta la siguiente transacción y ejecútala.

```
BEGIN TRANSACTION PRUEBA2
INSERT INTO CLIENTES_BANCA VALUES('004','Ricardo Rios Maldonado',19000;
INSERT INTO CLIENTES_BANCA VALUES('005','Pablo Ortiz Arana',15000);
INSERT INTO CLIENTES_BANCA VALUES('006','Luis Manuel Alvarado',18000);
```

Te habrás percatado que falta el COMMIT TRANSACTION

Revisa el contenido de la tabla clientes_banca desde la ventana que inicializaste como la primera sesión con la siguiente consulta.

SELECT * FROM CLIENTES_BANCA

Revisa el contenido de la tabla clientes_banca desde la ventana que inicializaste como la segunda sesión.

SELECT * FROM CLIENTES_BANCA

¿Qué pasa cuando deseas realizar esta consulta?

Cuando lo ejecuto desde la segunda sesión se queda cargando infinitamente. Nunca me muestra los datos. Cuando lo ejecuto desde la primera sesiçon que abrí no tiene problema. Me muestra los datos. Esto es ya que la transacción la escribí en la primera sesión.Si la hubiera escrito en la sesión dos la sesión uno sería la que no carga.

Intenta con la siguiente consulta desde la segunda sesión.

SELECT * FROM CLIENTES_BANCA where NoCuenta='001'

Explica por qué ocurre dicho evento.

No me quería mostrar nada porque el SGBD seguía intentando ejecutar la instrucción anterior. Tuve que presionar “cancel Running Statements” para ejecutar la nueva instrucción la cual cargo sin problema.

ejecuta:
ROLLBACK TRANSACTION PRUEBA2

Revisa nuevamente el contenido de la tabla clientes_banca desde la ventana que inicializaste como la segunda sesión.

SELECT * FROM CLIENTES_BANCA

¿Qué ocurrió y por qué?
Ya se ejecuta la consulta, sin embargo no aparecen los últimos 3 datos insertados ya que se eliminaron al ejecutar ROLLBACK TRANSACTION PRUEBA2

Inserta las siguientes transacciones

BEGIN TRANSACTION PRUEBA3
INSERT INTO TIPOS_MOVIMIENTO VALUES('A','Retiro Cajero Automatico');
INSERT INTO TIPOS_MOVIMIENTO VALUES('B','Deposito Ventanilla');
COMMIT TRANSACTION PRUEBA3

BEGIN TRANSACTION PRUEBA4
INSERT INTO MOVIMIENTOS VALUES('001','A',GETDATE(),500);
UPDATE CLIENTES_BANCA SET SALDO = SALDO -500
WHERE NoCuenta='001'
COMMIT TRANSACTION PRUEBA4

Posteriormente revisa si las tablas de clientes_banca y movimientos sufrieron algún cambio, es decir, si dio de alta el registro que se describe en la transacción y su actualización.
Sí se dieron de alta los datos ingresados en las transacciones.

```
--Captura y ejecuta la siguiente transacción:
BEGIN TRANSACTION PRUEBA5
INSERT INTO CLIENTES_BANCA VALUES('005','Rosa Ruiz Maldonado',9000);
INSERT INTO CLIENTES_BANCA VALUES('006','Luis Camino Ortiz',5000);
INSERT INTO CLIENTES_BANCA VALUES('007','Oscar Perez Alvarado',8000);

IF @@ERROR = 0
COMMIT TRANSACTION PRUEBA5
ELSE
BEGIN
PRINT 'A transaction needs to be rolled back'
ROLLBACK TRANSACTION PRUEBA5
```

```
END
```

¿Para qué sirve el comando @@ERROR revisa la ayuda en línea?
Para que se ejecute lo que sigue después de esa línea en caso de que la transacción no haya podido ser ejecutada exitosamente.

¿Explica qué hace la transacción?
Inserta 3 entradas a la tabla de Clientes_Banca, es decir da de alta 3 clientes nuevos., pero falta la linea de **COMMIT TRANSACTION** PRUEBA5 para que sea exitosa, Al omitirla estamos simultando algun problema externo, que pudiera causar que no se termine de ejecutar todo el bloque de una transacción.

¿Hubo alguna modificación en la tabla? Explica qué pasó y por qué.
No, No hubo ningun cambio.

Realiza los ejercicios con store procedures.

Una transacción que registre el retiro de una cajero. nombre del store procedure REGISTRAR_RETIRO_CAJERO que recibe 2 parámetros en NoCuenta y el monto a retirar.

```
CREATE PROCEDURE REGISTRAR_RETIRO_CAJERO
@NoCliente varchar(5),
@Monto numeric(10,2)
as
BEGIN TRANSACTION PRUEBA6
INSERT INTO Movimientos
  VALUES(@NoCliente,'A',GETDATE(),@Monto);
UPDATE Clientes_Banca
  SET Saldo = Saldo - @Monto
WHERE NoCliente=@NoCliente

IF @@ERROR = 0
COMMIT TRANSACTION PRUEBA6
ELSE
BEGIN
PRINT 'A transaction needs to be rolled back'
ROLLBACK TRANSACTION PRUEBA6
END
GO
```

Una transacción que registre el deposito en ventanilla. Nombre del store procedure REGISTRAR_DEPOSITO_VENTANILLA que recibe 2 parámetros en NoCuenta y el monto a depositar.

```
CREATE PROCEDURE REGISTRAR_DEPOSITO_VENTANILLA
@NoCliente varchar(5),
@Monto numeric(10,2)
as
BEGIN TRANSACTION PRUEBA7
INSERT INTO Movimientos
```

```
  VALUES(@NoCliente,'B',GETDATE(),@Monto);
UPDATE Clientes_Banca
  SET Saldo = Saldo + @Monto
WHERE NoCliente=@NoCliente

IF @@ERROR = 0
COMMIT TRANSACTION PRUEBA7
ELSE
BEGIN
PRINT 'A transaction needs to be rolled back'
ROLLBACK TRANSACTION PRUEBA7
END
GO
```