

Laboratorio 24: Manipulación de datos usando Transacciones en SQL Server

Comprobando la propiedad de aislamiento en una BD.

```
/* EN LA CONEXIÓN ORIGINAL */  
BEGIN;  
INSERT INTO Clientes_Banca VALUES('001', 'Manuel Rios Maldonado', 9000);  
INSERT INTO Clientes_Banca VALUES('002', 'Pablo Perez Ortiz', 5000);  
INSERT INTO Clientes_Banca VALUES('003', 'Luis Flores Alvarado', 8000);  
COMMIT;  
  
SELECT * FROM Clientes_Banca
```

	NoCue...	Nombre	Saldo
►	001	Manuel Rios Maldo...	9000.00
	002	Pablo Perez Ortiz	5000.00
	003	Luis Flores Alvarado	8000.00

¿Que pasa cuando deseas realizar la misma consulta desde la segunda conexión?

Aparecen exactamente los mismos datos que en la primer conexión

	NoCue...	Nombre	Saldo
	001	Manuel Rios Maldo...	9000.00
	002	Pablo Perez Ortiz	5000.00
	003	Luis Flores Alvarado	8000.00

```

/* EN LA SEGUNDA CONEXIÓN */
BEGIN ;
INSERT INTO Cientes_Banca VALUES('004','Ricardo Rios Maldonado',19000);
INSERT INTO Cientes_Banca VALUES('005','Pablo Ortiz Arana',15000);
INSERT INTO Cientes_Banca VALUES('006','Luis Manuel Alvarado',18000);

SELECT * FROM Cientes_Banca

```

¿Que pasa cuando deseas realizar la misma consulta desde la conexión original?

En la conexión original no se hizo la inserción de los clientes 4, 5 y 6.

NoCue...	Nombre	Saldo
001	Manuel Rios Maldo...	9000.00
002	Pablo Perez Ortiz	5000.00
003	Luis Flores Alvarado	8000.00

¿Que pasa cuando deseas realizar la consulta anterior desde la segunda conexión?

Se despliegan los 6 clientes, los tres clientes creados con la transacción anterior y los tres creados con la nueva desde esta misma conexión.

NoCue...	Nombre	Saldo
▶ 001	Manuel Rios Maldo...	9000.00
002	Pablo Perez Ortiz	5000.00
003	Luis Flores Alvarado	8000.00
004	Ricardo Rios Maldo...	19000.00
005	Pablo Ortiz Arana	15000.00
006	Luis Manuel Alvarado	18000.00

```

/* DESDE LA SEGUNDA CONEXIÓN */
SELECT * FROM Cientes_Banca where NoCuenta='001'

```

¿Qué pasa cuando deseas realizar esta consulta? Explica por qué ocurre dicho evento.

Despliega la cuenta con número 001 que pertenece a Manuel Ríos Maldonado.

	001	Manuel Rios Maldo...	9000.00

```

/* DESDE LA CONEXIÓN ORIGINAL */
ROLLBACK ;

/* DESDE LA SEGUNDA CONEXIÓN */
SELECT * FROM Clientes_Banca

```

¿Qué ocurrió y por qué?

Vuelve a desplegar sólo los tres primeros clientes ya que el comando ROLLBACK se encarga de revertir una transacción explícita o implícita al comienzo de la transacción, o a un punto de guardado dentro de la transacción. En este caso, mostrando sólo la primera transacción.

NoCue...	Nombre	Saldo
001	Manuel Rios Maldo...	9000.00
002	Pablo Perez Ortiz	5000.00
003	Luis Flores Alvarado	8000.00

Comprobando la propiedad de Atomicidad en una BD

```

/* Inserta la siguiente transacción y ejecútala */
BEGIN ;
INSERT INTO Tipos_Movimiento VALUES('A','Retiro Cajero Automatico');
INSERT INTO Tipos_Movimiento VALUES('B','Deposito Ventanilla');
COMMIT ;

BEGIN ;
INSERT INTO Movimientos VALUES('001','A',GETDATE(),500);
UPDATE Clientes_Banca SET SALDO = SALDO -500
WHERE NoCuenta='001';
COMMIT ;

```

Revisa si las tablas de clientes_banca y movimientos sufrieron algún cambio

NoCuenta	Nombre	Saldo
001	Manuel Rios Maldonado	8500.00
002	Pablo Perez Ortiz	5000.00
003	Luis Flores Alvarado	8000.00

NoCuenta	ClaveM	Fecha	Monto
001	A	2020-05-26 21:04:11	500.00

Manejando Fallas en una Transacción

```

/* Inserta la siguiente transacción y ejecútala */
BEGIN ;
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('001','Oscar Perez Alvarado',8000);

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

```

¿Para qué sirve el comando @@ERROR revisa la ayuda en línea?

Sirve para validar si existen o no errores en las transacciones realizadas.

Devuelve 0 si la instrucción Transact-SQL anterior no encontró errores.

Devuelve un número de error si la declaración anterior encontró un error. Si el error fue uno de los errores en la vista de catálogo sys.messages, entonces @@ ERROR contiene el valor de la columna sys.messages.message_id para ese error.

¿Qué hace la transacción?

Inserta nuevos clientes en la tabla definida y checa si existen errores para hacer commit en caso de que no ocurran o hacer rollback en caso de que sí.

¿Hubo alguna modificación en la tabla? Explica qué pasó y por qué.

No, porque el tercer INSERT INTO intenta insertar un cliente con la llave primaria duplicada es por eso que @@ERROR nos arroja un número diferente de 0 y en lugar de hacer un commit a la transacción se hace un rollback, es decir se revierte y terminamos con los datos originales.

Crea las siguientes transacciones dentro de un stored procedure:

- 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.

```
DROP PROCEDURE IF EXISTS `REGISTRAR_RETIRO_CAJERO` ;
DELIMITER //
CREATE PROCEDURE `REGISTRAR_RETIRO_CAJERO` (uNoCuenta VARCHAR(5), uMonto
DECIMAL(10,2))
BEGIN ;
    INSERT INTO Movimientos VALUES(uNoCuenta, 'A', GETDATE(), uMonto);
COMMIT ;
END //
DELIMITER ;
```

+ Opciones

NoCuenta	ClaveM	Fecha	Monto
001	A	2020-05-26 21:04:11	500.00
003	A	2020-05-27 00:00:00	750.00

- 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.

```

DROP PROCEDURE IF EXISTS `REGISTRAR_DEPOSITO_VENTANILLA`;
DELIMITER //
CREATE PROCEDURE `REGISTRAR_DEPOSITO_VENTANILLA` (uNoCuenta VARCHAR(5), uMonto
DECIMAL(10,5))
BEGIN ;
    INSERT INTO Movimientos VALUES(uNoCuenta, 'B', GETDATE(), uMonto);
COMMIT ;
END //
DELIMITER ;

```

NoCuenta	ClaveM	Fecha	Monto
001	A	2020-05-26 21:04:11	500.00
003	A	2020-05-27 00:00:00	750.00
001	B	2020-05-27 02:30:43	900.00