



Lab 24. Manipulación de datos usando Transacciones en SQL Server

▼ Class	Data Bases
🕒 Created	@Jun 02, 2020 9:17 AM
🔗 Materials	https://ejuarez.bitbucket.io/daw/laboratorios/lab15_transacciones.html
☑ Reviewed	<input type="checkbox"/>
▼ Type	Homework

Comprobando la propiedad de aislamiento en una BD

Inserta la siguiente transacción y ejecutala

```
/* PRIMERA CONEXION */
BEGIN TRANSACTION;
INSERT INTO ClientesBanca VALUES('001', 'Manuel Rios Maldonado', 9000);
INSERT INTO ClientesBanca VALUES('002', 'Pablo Perez Ortiz', 5000);
INSERT INTO ClientesBanca VALUES('003', 'Luis Flores Alvarado', 8000);
COMMIT;

SELECT * FROM ClientesBanca
```

NoCuenta	Nombre	Saldo
001	Manuel Rios Maldonado	9000.00
002	Pablo Perez Ortiz	5000.00
003	Luis Flores Alvarado	8000.00

¿Que pasa cuando deseas realizar esta consulta en una segunda conexión?

Aparecen los mismos datos

NoCuenta	Nombre	Saldo
001	Manuel Rios Maldonado	9000.00
002	Pablo Perez Ortiz	5000.00
003	Luis Flores Alvarado	8000.00
NULL	NULL	NULL

Inserta la siguiente transacción y ejecútala:

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

¿Qué pasa cuando realizas la consulta en la conexión original?

No se ven insertados los clientes 004, 005, 006

NoCuenta	Nombre	Saldo
001	Manuel Rios Maldonado	9000.00
002	Pablo Perez Ortiz	5000.00
003	Luis Flores Alvarado	8000.00
NULL	NULL	NULL

¿Qué pasa cuando realizas la consulta en la conexión dos?

Se despliegan los 6 clientes

NoCuenta	Nombre	Saldo
001	Manuel Rios Maldonado	9000.00
002	Pablo Perez Ortiz	5000.00
003	Luis Flores Alvarado	8000.00
004	Ricardo Rios Maldonado	19000.00
005	Pablo Ortiz Arana	15000.00
006	Luis Manuel Alvarado	18000.00
NULL	NULL	NULL

Intenta con la siguiente consulta desde la segunda sesión:

```
SELECT * FROM CLIENTES_BANCA where NoCuenta='001'
```

Aparece el Cliente Manuel Rios Maldonado

NoCuenta	Nombre	Saldo
001	Manuel Rios Maldonado	9000.00
NULL	NULL	NULL

Por último regresa a la ventana que mantiene activa tu primer sesión, agrega el siguiente comando a la pantalla y ejecútalo:

```
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?

NoCuenta	Nombre	Saldo
001	Manuel Rios Maldonado	9000.00
002	Pablo Perez Ortiz	5000.00
003	Luis Flores Alvarado	8000.00

Vuelve a desplegar solo los 3 primeros clientes porque ROLLBACK revierte las transacciones.

Comprobando la propiedad de Atomicidad en una BD

Inserta las siguientes transacciones y ejecutalas

```
START TRANSACTION;
INSERT INTO TiposMovimiento VALUES('A','Retiro Cajero Automatico');
INSERT INTO TiposMovimiento VALUES('B','Deposito Ventanilla');
COMMIT;
```

```
START TRANSACTION
INSERT INTO Movimientos VALUES('001','A',GETDATE(),500);
UPDATE ClientesBanca SET SALDO = SALDO -500
WHERE NoCuenta='001'
COMMIT
```

Verifica si hubo cambios en las tablas

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	2020-05-14 21:10:10	500.00
NULL	NULL	NULL	NULL

Manejando Fallas en una transacción

Captura y ejecuta la siguiente transacción:

```

BEGIN TRANSACTION PRUEBA5
INSERT INTO ClientesBanca VALUES('005','Rosa Ruiz Maldonado',9000);
INSERT INTO ClientesBanca VALUES('006','Luis Camino Ortiz',5000);
INSERT INTO ClientesBanca VALUES('001','Oscar Perez Alvarado',8000);

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

```

Ejecuta la transacción, esta se ejecuta de igual forma que realizas una consulta.

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

Para saber si la transacción mandó algún error, si es 0, no hubo. Pero si si hubo algún error, nos devuelve el id de ese error.

¿Qué hace la transacción?

Inserta clientes en la tabla ClientesBanca y si hay errores, hace rollback.

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

No, porque el tercero de los INSERT INTO tiene una llave primaria duplicada y el @@ERROR va a ser diferente de 0, y por ello se hace rollback

Crear las 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', date(), uMonto);
COMMIT ;
END //
DELIMITER ;

```

- **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`
CREATE PROCEDURE `REGISTRAR_DEPOSITO_VENTANILLA` (uNoCuenta VARCHAR(5), uMonto DECIMAL (10,2))
BEGIN

```

```
INSERT INTO Movimientos VALUES (uNoCuenta, 'B', date(), uMonto);  
COMMIT ;  
END //  
DELIMITER ;
```

Las transacciones deben de ser diseñadas para que en caso de que ocurra algún error la base de datos mantenga su consistencia.