

## Laboratorio 23 — Transacciones

¿Qué pasa cuando deseas realizar esta consulta desde la segunda sesión?

```
SELECT * FROM Clientes_Banca
```

La consulta se realiza con éxito.

¿Qué pasa cuando deseas realizar esta consulta desde la segunda sesión (después de la segunda transacción)?

```
SELECT * FROM Clientes_Banca
```

La consulta no funciona desde la segunda sesión iniciada.

Explica por qué ocurre dicho evento (Ejecutar `SELECT * FROM CLIENTES_BANCA where NoCuenta='001'`)

La entrada con el número de cuenta “001” ya está registrada dentro de la tabla, mientras que los registros de la segunda transacción todavía se encuentran en espera de confirmación.

¿Qué ocurrió y por qué? (Después de realizar `ROLLBACK TRANSACTION PRUEBA2`)

```
SELECT * FROM Clientes_Banca
```

La consulta se realiza con éxito. El Rollback deshace los cambios que estaban pendientes por la segunda transacción, por lo que la segunda sesión ya puede acceder a la tabla en su totalidad.

¿Para qué sirve el comando `@@ERROR`?

```
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('001', '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 checar si se produce un error con la instrucción anterior.

Explica qué hace la transacción

Se intenta insertar nuevas entradas a la tabla Clientes\_Banca. Sin embargo, una de estas nuevas entradas contiene un ID que ya existe en la base de datos, por lo que se produce un error y se rechaza la transacción. Se utiliza el comando `@@ERROR` para imprimir un mensaje en pantalla cuando el error ocurre.

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

No hay modificación porque, aunque sólo hay un conflicto con las nuevas entradas, ninguna sucede hasta que no existe un error y la transacción completa sea exitosa, haciendo un commit.

### Procedimientos con Transacciones

```

IF EXISTS (SELECT name FROM sysobjects
           WHERE name = 'REGISTRAR_RETIRO_CAJERO' AND type = 'P')
    DROP PROCEDURE REGISTRAR_RETIRO_CAJERO
GO

CREATE PROCEDURE REGISTRAR_RETIRO_CAJERO
    @noCuenta VARCHAR(5),
    @monto NUMERIC(10,2)
AS
    BEGIN TRANSACTION Retiro
    UPDATE Clientes_Banca SET Saldo = Saldo - @monto WHERE
NoCuenta = @noCuenta
    INSERT INTO Realizan VALUES(@noCuenta, 'A', GETDATE(),
@monto)

    IF @@ERROR = 0
    COMMIT TRANSACTION Retiro
    ELSE
    BEGIN
    PRINT 'Error en la transacción'
    ROLLBACK TRANSACTION Retiro
    END
GO

IF EXISTS (SELECT name FROM sysobjects
           WHERE name = 'REGISTRAR_DEPOSITO_VENTANILLA' AND type = 'P')
    DROP PROCEDURE REGISTRAR_DEPOSITO_VENTANILLA
GO

CREATE PROCEDURE REGISTRAR_DEPOSITO_VENTANILLA
    @noCuenta VARCHAR(5),
    @monto NUMERIC(10,2)
AS
    BEGIN TRANSACTION Deposito
    UPDATE Clientes_Banca SET Saldo = Saldo + @monto WHERE
NoCuenta = @noCuenta
    INSERT INTO Realizan VALUES(@noCuenta, 'B', GETDATE(),
@monto)

    IF @@ERROR = 0
    COMMIT TRANSACTION Deposito
    ELSE
    BEGIN
    PRINT 'Error en la transacción'
    ROLLBACK TRANSACTION Deposito
    END
GO

```