Práctica 21

Transacciones en SQL Server

Objetivo

*En esta práctica se presentan varios ejemplos de transacciones mediante el empleo del lenguaje **Transact SQL** y el empleo del **SSMS**.*

Introducción

SQL Server funciona por defecto con ***Transacciones de confirmación automática***, es decir, cada instrucción individual es una transacción y se confirma automáticamente.

Es posible cambiar el entorno para que considere todas las modificaciones como transacciones implícitas, es decir, sin necesidad de declararlas. Por omisión, este modo se considera que está inactivo.

Para activar el modo de transacciones implícitas se debe ejecutar la instrucción.

```
SET IMPLICIT_TRANSACTIONS ON
```

Una vez que una primera transacción se ha confirmado o abortado, se inicia automáticamente una nueva transacción cuando se ejecute una instrucción para modificar datos.

La conexión continúa generando transacciones implícitas hasta que se desactiva el modo de transacciones implícitas.

En SQL Server se pueden anidar varias transacciones. Cuando se anidan varias transacciones, la instrucción COMMIT afectará a la última transacción iniciada, pero ROLLBACK afectará a todas las transacciones abiertas.

Un hecho para tener en cuenta es que, si se hace ROLLBACK de la transacción de nivel superior, se desharán también los cambios hechos por todas las transacciones internas, incluso si se ha realizado COMMIT en ellas.

Equipo necesario

Una computadora con sistema operativo ***Windows*** que cuente con ***SQL Server 2022*** y ***SSMS***.

Metodología

1. En el menú de programas de Windows, localice la aplicación ***SQL Server Management Studio 20***.
2. En la ventana emergente, oprima el botón ***Connect***.
3. En la barra de herramientas localice y oprima el botón de ***New Query***.
4. En la ventana del editor de consultas escriba la siguiente sentencia:

```
CREATE DATABASE banco
```

5. Oprima el botón ***Execute*** en la barra de herramientas.
6. En la barra de herramientas localice la lista desplegable ***Available Databases***, y seleccione ***banco***.
7. En el editor de consultas borre lo anterior y escriba la siguiente sentencia para crear una tabla de CUENTAS:

```
CREATE TABLE CUENTAS (Numero VARCHAR(7) PRIMARY KEY, Saldo
MONEY, Fecha DATE)
```

8. Debajo escriba la sentencia para crear otra tabla, donde se registren los movimientos de las cuentas.

```
CREATE TABLE MOVIMIENTOS (Numero VARCHAR(10), Saldo_Anterior
MONEY, Saldo_Nuevo MONEY, IMPORTE MONEY, Fecha DATE)
```

9. Y oprima ***Execute***, para crear ambas tablas.
10. Obtenga una imagen completa de la pantalla, mostrando las sentencias y el resultado de la ejecución y consérvela como evidencia.
11. Borre la sentencia anterior, y escriba las siguientes sentencias, para insertar las dos cuentas con las que se va a trabajar y listar la tabla:

```
INSERT INTO CUENTAS VALUES ('2024001', 2000.00, GETDATE())
INSERT INTO CUENTAS VALUES ('2024002', 2000.00, GETDATE())
SELECT * FROM CUENTAS
```

12. Oprima el botón ***Execute***.
13. Obtenga una imagen completa de la pantalla, mostrando las sentencias y el resultado de la ejecución y consérvela como evidencia.
14. Borre las sentencias anteriores, para escribir el siguiente script donde se realiza la transferencia de fondos entre cuentas, haciendo uso de una ***transacción explicita***.
15. Primer paso, se declaran las variables

```
DECLARE @importe MONEY =1500,
@CuentaOrigen VARCHAR(12) = '2024001',
@CuentaDestino VARCHAR(12) = '2024002'
```

16. Aquí inicia la transacción, se descuenta el importe del saldo de la cuenta origen, luego se registra el movimiento de retiro, a continuación, se hace el depósito del importe en la cuenta de destino y se registra el movimiento de depósito, finalmente se confirma la transacción:

```
BEGIN TRANSACTION
BEGIN TRY
UPDATE CUENTAS SET Saldo = Saldo - @importe WHERE Numero =
@CuentaOrigen
INSERT INTO MOVIMIENTOS SELECT Numero, Saldo + @importe, Saldo,
@importe, getdate() FROM CUENTAS WHERE Numero = @CuentaOrigen
UPDATE CUENTAS SET Saldo = Saldo + @importe WHERE Numero =
@CuentaDestino
INSERT INTO MOVIMIENTOS SELECT Numero, Saldo - @importe, Saldo,
@importe, getdate() FROM CUENTAS WHERE Numero = @CuentaDestino
COMMIT
END TRY
```

17. A continuación, se añade el bloque ***CATCH***, para cuando ocurra algún error.

```
BEGIN CATCH
ROLLBACK
PRINT 'Se ha producido un error!'
PRINT ERROR_MESSAGE()
END CATCH
```

18. Por último, ponga las sentencias para mostrar las tablas después de que se realice la transacción:

```
SELECT * FROM CUENTAS
SELECT * FROM MOVIMIENTOS
```

19. Oprima el botón ***Execute***.
20. Obtenga una imagen completa de la pantalla, mostrando el script completo y el resultado de la ejecución y consérvela como evidencia.
21. Es necesario evitar que la transacción se realice cuando el saldo final sea negativo en la cuenta de origen, modifique el script para que genere una excepción mediante ***RAISERROR***, para el caso cuando el saldo de la cuenta de origen sea insuficiente para transferir el importe solicitado.
22. Verifique su funcionamiento, oprima el botón ***Execute***, para provocar la excepción.
23. Obtenga una imagen completa de la pantalla, mostrando el script completo y el resultado de la ejecución y consérvela como evidencia.
24. Borre la transacción anterior, para escribir el siguiente ejemplo de ***transacciones anidadas***:

```
BEGIN TRAN
INSERT INTO CUENTAS VALUES ('2024003', 4000.00, GETDATE())
```

```
BEGIN TRAN
INSERT INTO CUENTAS VALUES ('2024004', 5000.00, GETDATE())
COMMIT
SELECT * FROM CUENTAS
ROLLBACK
SELECT * FROM CUENTAS
```

25. Oprima el botón ***Execute***.
26. Observe que a pesar de que se compromete la transacción interna, y que muestra la inserción de ambos registros, al cancelar la transacción externa, ningún registro permanece.
27. Obtenga una imagen completa de la pantalla, mostrando el script completo y el resultado de la ejecución y consérvela como evidencia.
28. Modifique la transacción anterior, de la siguiente manera, incluyendo un savepoint ***P1***:

```
BEGIN TRAN
INSERT INTO CUENTAS VALUES ('2024003', 2000.00, GETDATE())
INSERT INTO CUENTAS VALUES ('2024004', 2000.00, GETDATE())
SAVE TRANSACTION P1
DELETE FROM CUENTAS WHERE Numero = '2024001'
SELECT * FROM CUENTAS
ROLLBACK TRANSACTION P1
COMMIT
SELECT * FROM CUENTAS
```

29. Oprima el botón ***Execute***.
30. En este caso, ambas inserciones permanecen y el ROLLBACK parcial solamente cancela la operación de eliminación.
31. Genere una imagen que evidencie el contenido completo del script y el resultado de su ejecución, comente el resultado obtenido.
32. Cierre la aplicación SSMS.
33. Fin de la Práctica.

Evidencias

El alumno deberá enviar al instructor las evidencias requeridas durante la realización de la práctica.

Sugerencias didácticas

El instructor deberá orientar a los alumnos que tengan dificultades en la interpretación y la realización de las instrucciones de la práctica, pero sin darles las respuestas.

Resultados

Se demuestra haber adquirido suficiente conocimiento sobre la creación de Transacciones en SQL Server.

Bibliografía

- https://learn.microsoft.com/en-us/sql/ssms/user-assistance-in-sql-server-management-studio?view=sql-server-ver16