

REPORTE PRACTICA 21

EVIDENCIAS INSTRUCCIÓN 10

```
CREATE TABLE CUENTAS (Numero VARCHAR(7) PRIMARY KEY, Saldo MONEY, FECHA DATE);
CREATE TABLE MOVIMIENTOS (Numero VARCHAR(10), Saldo_Anterior MONEY, Saldo_Nuevo MONEY, IMPORTE MONEY, Fecha DATE);
```

Messages

Commands completed successfully.

Completion time: 2025-04-13T22:54:40.5410674-06:00

EVIDENCIAS INSTRUCCION 13

SQLQuery1.sql - SA...GOLARA\reliv (64)*

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

Results

	Numero	Saldo	FECHA
1	2024001	2000.00	2025-04-13
2	2024002	2000.00	2025-04-13

EVIDENCIAS INSTRUCCION 20

SQLQuery1.sql - SA...GOLARA\reliv (64)*

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

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

Results

	Numero	Saldo	FECHA
1	2024001	-1000.00	2025-04-13
2	2024002	5000.00	2025-04-13

	Numero	Saldo_Anterior	Saldo_Nuevo	IMPORTE	Fecha
1	2024001	2000.00	500.00	1500.00	2025-04-13
2	2024002	2000.00	3500.00	1500.00	2025-04-13
3	2024001	500.00	-1000.00	1500.00	2025-04-13
4	2024002	3500.00	5000.00	1500.00	2025-04-13

EVIDENCIA INSTRUCCION 23

```

DECLARE @importe MONEY=1500, @CuentaOrigen VARCHAR(12)='2024001',
        @CuentaDestino VARCHAR(12)='2024002';
BEGIN TRANSACTION
BEGIN TRY
    IF EXISTS (
        SELECT 1 FROM CUENTAS
        WHERE Numero = @CuentaOrigen AND Saldo < @importe
    )
        BEGIN
            RAISERROR('Fondos insuficientes en la cuenta de origen.', 16, 1)
        END
    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

```

100 %

Results Messages

	Numero	Saldo	FECHA
1	2024001	-1000.00	2025-04-13
2	2024002	5000.00	2025-04-13

	Numero	Saldo_Anterior	Saldo_Nuevo	IMPORTE	Fecha
1	2024001	2000.00	500.00	1500.00	2025-04-13
2	2024002	2000.00	3500.00	1500.00	2025-04-13
3	2024001	500.00	-1000.00	1500.00	2025-04-13
4	2024002	3500.00	5000.00	1500.00	2025-04-13

EVIDENCIAS INSTRUCCIÓN 27

The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows a SQL query window with the following script:

```
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
```

The bottom pane shows the 'Results' tab with a table containing 4 rows of data. The first row is highlighted.

	Numero	Saldo	FECHA
1	2024001	-1000.00	2025-04-13
2	2024002	5000.00	2025-04-13
3	2024003	4000.00	2025-04-13
4	2024004	5000.00	2025-04-13

Below the first table, there is a second table with 2 rows of data:

	Numero	Saldo	FECHA
1	2024001	-1000.00	2025-04-13
2	2024002	5000.00	2025-04-13

EVIDENCIAS INSTRUCCION 31

```
BEGIN TRAN
INSERT INTO CUENTAS VALUES('2024003', 4000.00, GETDATE())
BEGIN TRAN
INSERT INTO CUENTAS VALUES('2024004', 5000.00, GETDATE())
SAVE TRANSACTION P1
DELETE FROM CUENTAS WHERE Numero='2024001'
SELECT * FROM CUENTAS
ROLLBACK TRANSACTION P1
COMMIT
ROLLBACK
```

100 %

Results Messages

	Numero	Saldo	FECHA
1	2024002	5000.00	2025-04-13
2	2024003	4000.00	2025-04-13
3	2024004	5000.00	2025-04-13

	Numero	Saldo	FECHA
1	2024001	-1000.00	2025-04-13
2	2024002	5000.00	2025-04-13