

### Primera Parte

I. Realizar un interbloqueo sencillo como lo explicado en clases y desbloquear con el nolock

1. Usando la misma base de datos anterior creada de Empresa (use Empresa), hacer un select de la tabla Employee para visualizarla

```
-- Create database Empresa
use empresa

-- create table employee(
empno int primary key,
e_name varchar(25),
e_last_name varchar(25),
salary money,
bonus money
)
```

```
-- insert into employee(empno, e_name, e_last_name, salary, bonus)
values(000060, 'Mark', 'Polk', 2000, 150),
      (000070, 'Joel', 'Loop', 1600, 200)

select * from employee
```

150 %

Results Messages

	empno	e_name	e_last_name	salary	bonus
1	60	Mark	Polk	2000.00	150.00
2	70	Joel	Loop	1600.00	200.00

2. Iniciar una transacción (begin transaction), actualizar el campo bonus sumando 1 y usando una condición de where con un número de empleado que ud. escoja, ejecutar y confirmar que se ejecute el cambio

```
BEGIN TRANSACTION
USE EMPRESA
SELECT * FROM EMPLOYEE
UPDATE EMPLOYEE
SET BONUS = BONUS +1
WHERE EMPNO = '000070'

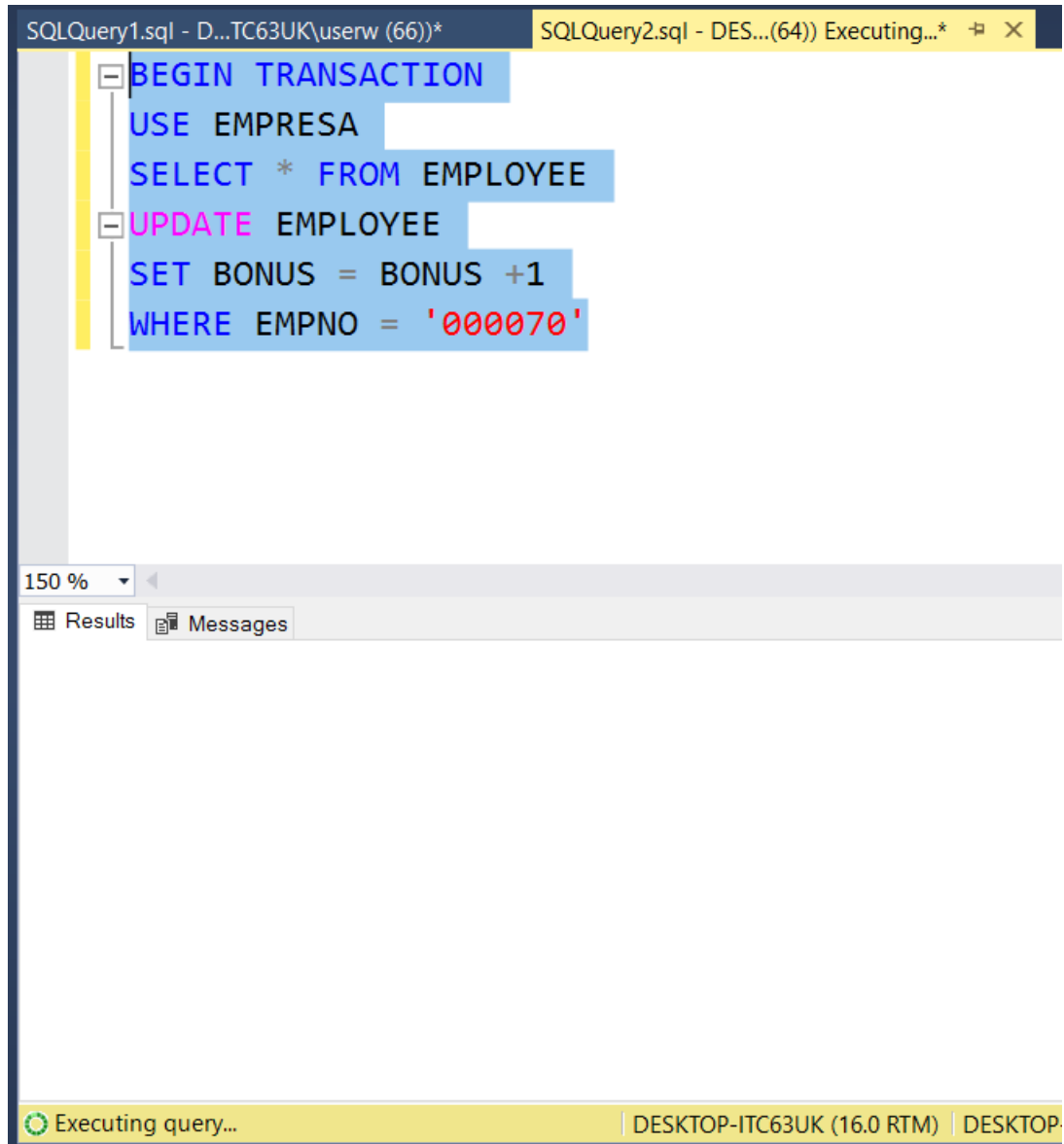
SELECT * FROM EMPLOYEE
```

150 %

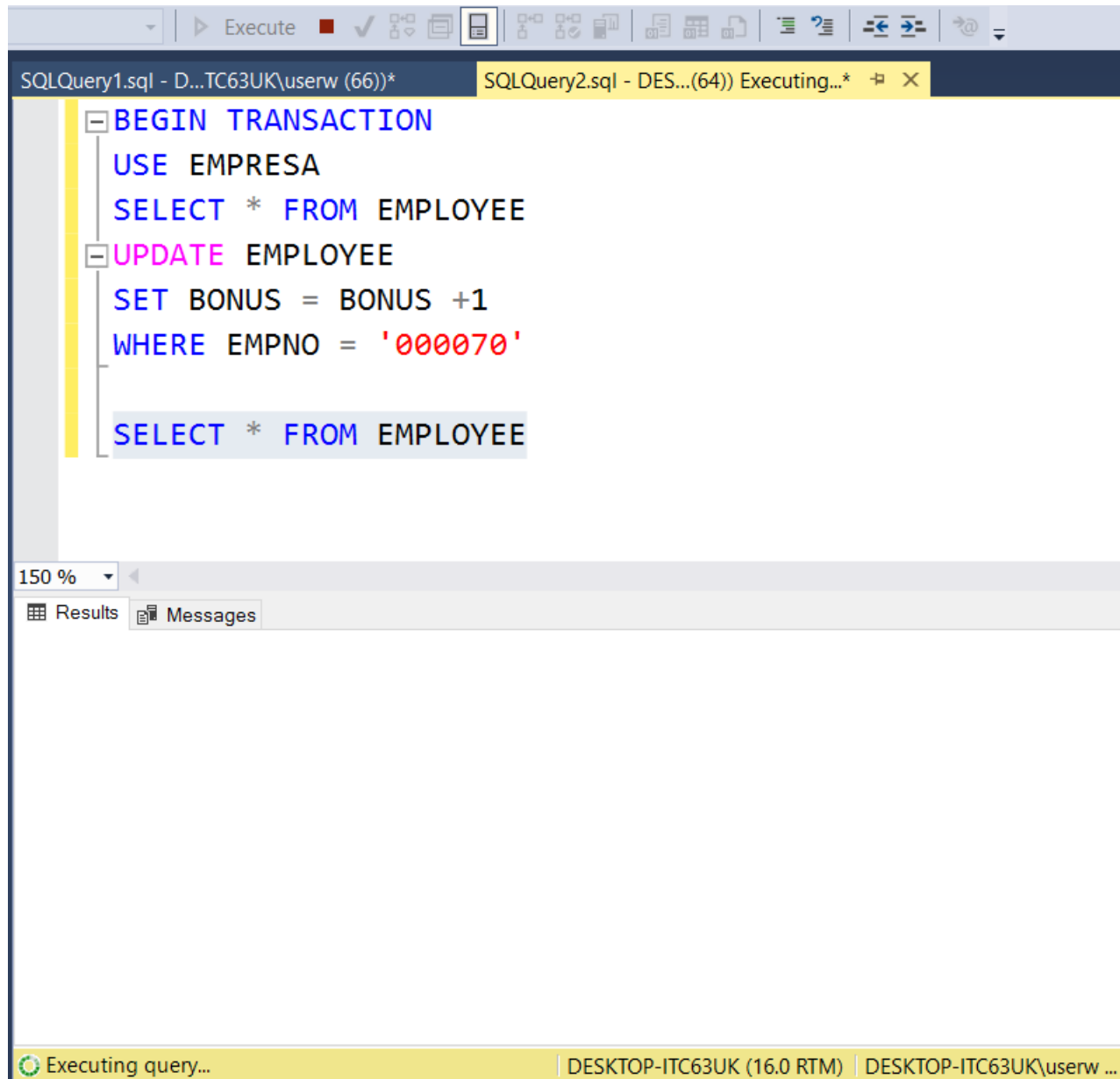
Results Messages

	empno	e_name	e_last_name	salary	bonus
1	60	Mark	Polk	2000.00	150.00
2	70	Joel	Loop	1600.00	201.00

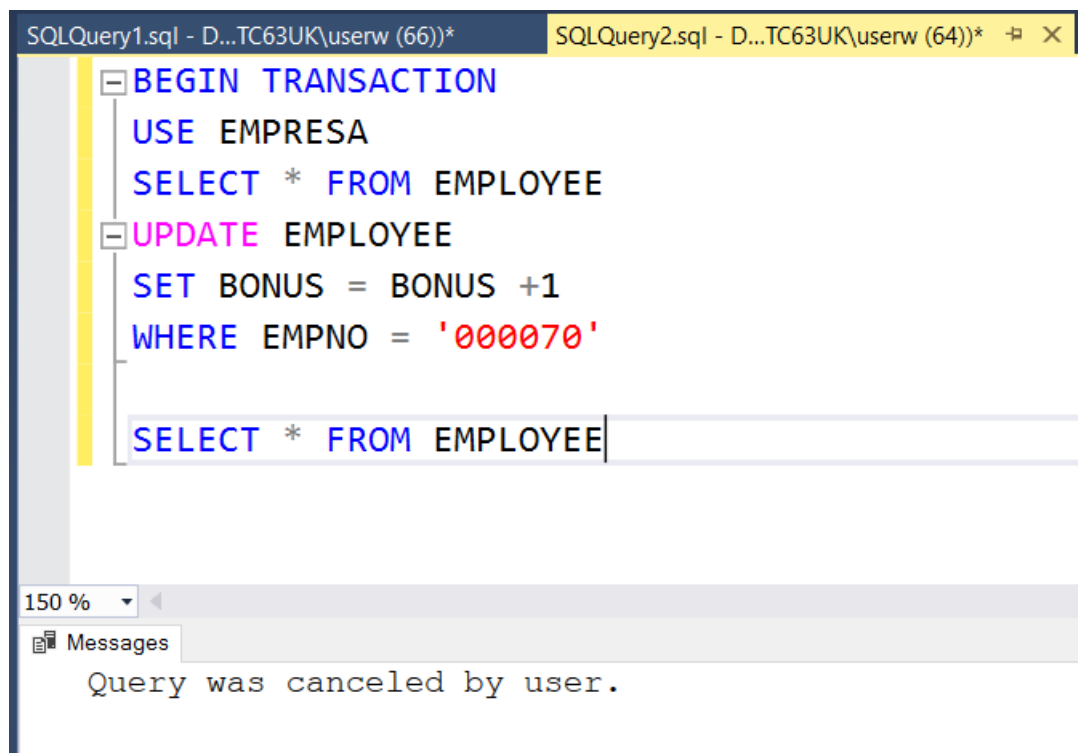
3. Abrir otra consulta y realizar la misma transacción para validar que el interbloqueo se de



4. Hacer un select nuevamente para visualizar la tabla y al no poder continuar, primero cancelar y luego probar la visualización con el (nolock)



No es posible realizar el select



The screenshot shows a SQL Server Enterprise Manager interface. At the top, there are two tabs: 'SQLQuery1.sql - D...TC63UK\userw (66))\*' and 'SQLQuery2.sql - D...TC63UK\userw (64))\*'. The main window displays a SQL script with the following content:

```
BEGIN TRANSACTION
USE EMPRESA
SELECT * FROM EMPLOYEE
UPDATE EMPLOYEE
SET BONUS = BONUS +1
WHERE EMPNO = '000070'

SELECT * FROM EMPLOYEE
```

Below the script, there is a zoom level dropdown set to '150 %' and a 'Messages' pane. The 'Messages' pane contains the text: 'Query was canceled by user.'

La consulta ha sido cancelada

5. Observar el comportamiento y si es posible ahora revisar la consulta.

The screenshot shows a SQL Server Enterprise Manager interface. At the top, there are two tabs: 'SQLQuery1.sql - D...TC63UK\userw (66))\*' and 'SQLQuery2.sql - D...TC63UK\userw (64))\*'. The active window displays the following SQL code:

```
SET BONUS = BONUS +1
WHERE EMPNO = '000070'

select * from employee

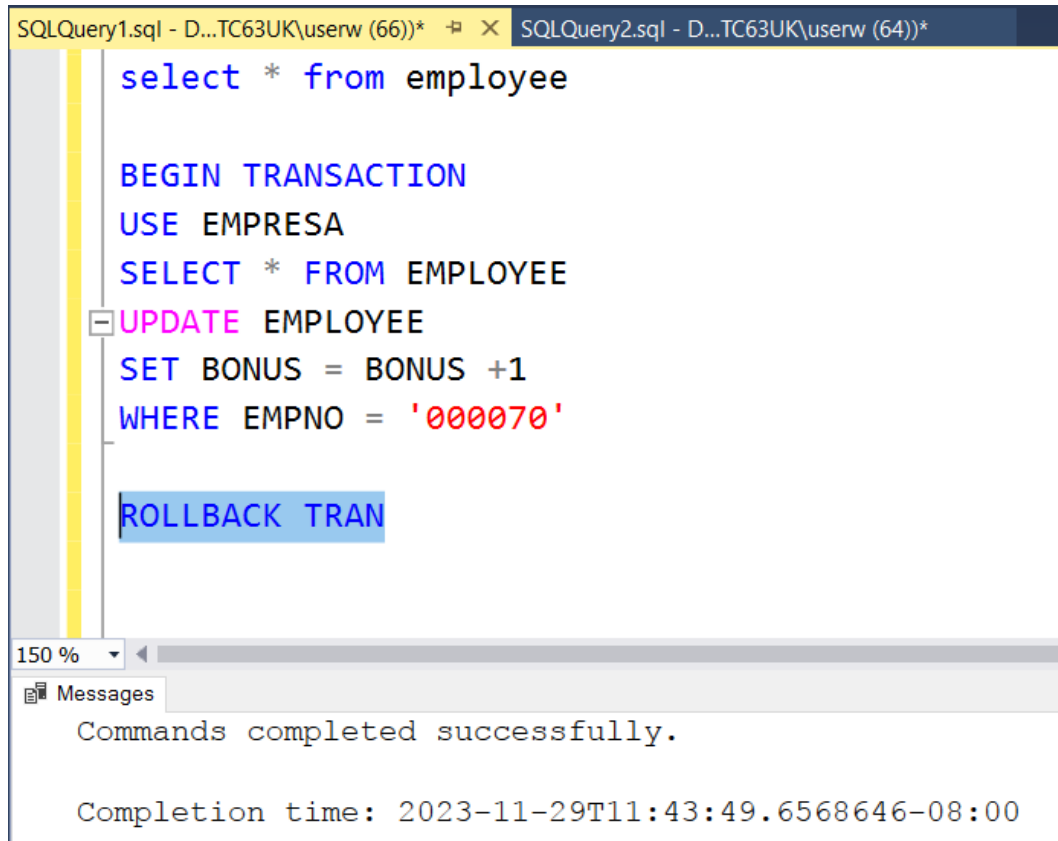
select * from employee WITH (NOLOCK)
```

Below the query window, there is a 'Results' tab and a 'Messages' tab. The 'Results' tab is active, showing a table with 6 columns: 'empno', 'e\_name', 'e\_last\_name', 'salary', and 'bonus'. The table contains two rows of data:

	empno	e_name	e_last_name	salary	bonus
1	60	Mark	Polk	2000.00	150.00
2	70	Joel	Loop	1600.00	201.00

Es posible realizar la consulta mediante el uso de nolock

6. Regresar al query inicial y deshacer el cambio con ROLLBACK y observar el comportamiento al visualizar nuevamente la tabla, si es necesarios cierre todos los query y abra uno nuevo haciendo el select de la tabla y verá que no se hizo el cambio porque la transacción había quedado abierta, ahora que le ejecutó el rollback se revirtió



The screenshot shows a SQL Server Enterprise Manager interface. At the top, there are two tabs: 'SQLQuery1.sql - D...TC63UK\userw (66))\*' and 'SQLQuery2.sql - D...TC63UK\userw (64))\*'. The active window displays a SQL script with the following commands:

```
select * from employee

BEGIN TRANSACTION
USE EMPRESA
SELECT * FROM EMPLOYEE
UPDATE EMPLOYEE
SET BONUS = BONUS +1
WHERE EMPNO = '000070'

ROLLBACK TRAN
```

Below the script, there is a 'Messages' window showing the following output:

```
Commands completed successfully.

Completion time: 2023-11-29T11:43:49.6568646-08:00
```

SQLQuery1.sql - D...TC63UK\userw (66))\* SQLQuery2.sql - D...TC63UK\userw (64))\*

```
USE EMPRESA
SELECT * FROM EMPLOYEE
UPDATE EMPLOYEE
SET BONUS = BONUS +1
WHERE EMPNO = '000070'

ROLLBACK TRAN

select * from employee
```

150 %

Results Messages

	empno	e_name	e_last_name	salary	bonus
1	60	Mark	Polk	2000.00	150.00
2	70	Joel	Loop	1600.00	200.00

Tras aplicar el rollback y realizar el select el bono del empleado 70 se reestablecio a 200.

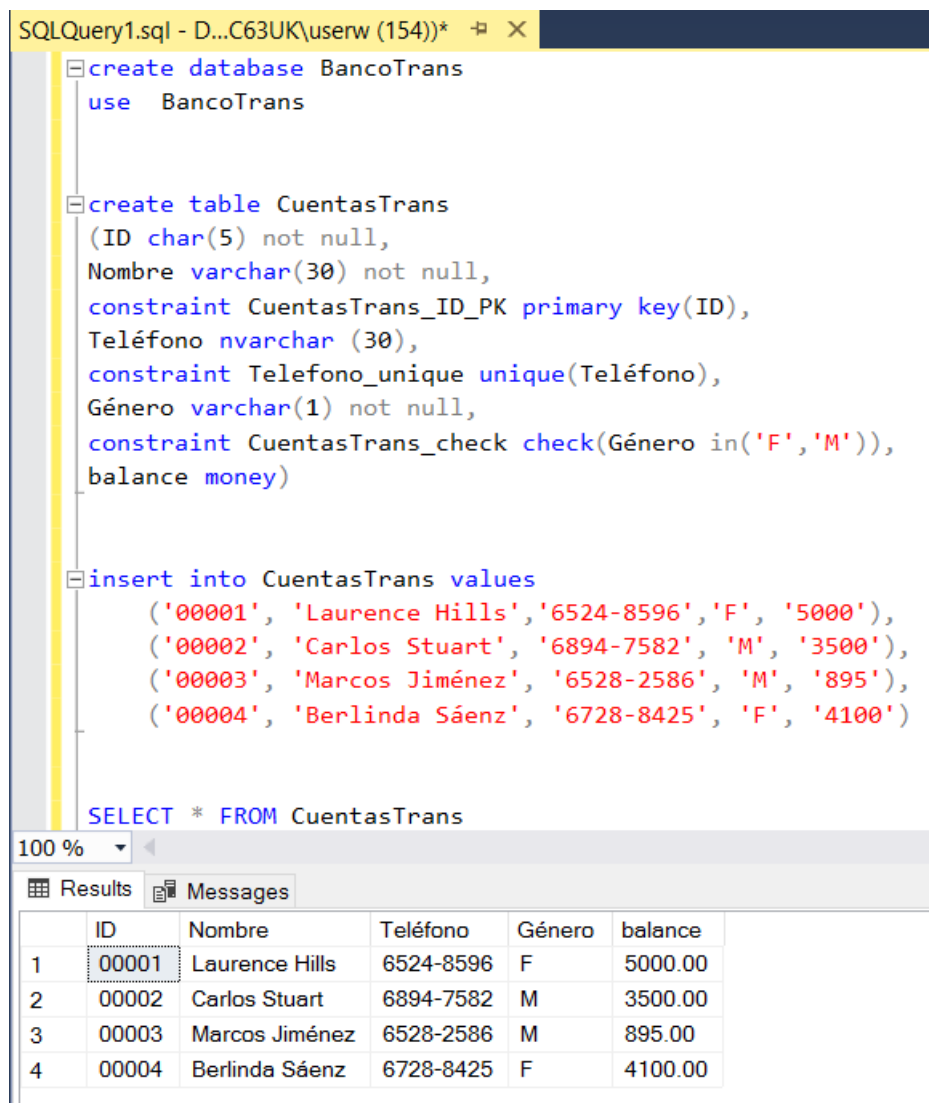


## Segunda parte (ejemplo de transacción entre cuentas): Problema:

De las cuentas que hay en el banco, se quiere pasar un dinero de una cuenta a otra. Laurence y Marcos son amigos desde hace muchos años y tienen cuenta en el mismo banco, Marcos le pide a Laurence que le preste \$1300 por medio de una transacción.

### PASOS:

1. Ejecutar el query proporcionado para correr: base de datos, tabla y datos
  - 1.1 Revisar y comprender el código entregado antes de ejecutarlo, usted debe poder hacerlo solo, pero para agilizar el tiempo y nos rinda en el lab. Se le ha proporcionado.
2. Luego visualizar la tabla de cuentasTrans



The screenshot shows a SQL query window titled "SQLQuery1.sql - D...C63UK\userw (154))\*". The query contains the following SQL code:

```
create database BancoTrans
use BancoTrans

create table CuentasTrans
(ID char(5) not null,
Nombre varchar(30) not null,
constraint CuentasTrans_ID_PK primary key(ID),
Teléfono nvarchar (30),
constraint Telefono_unique unique(Teléfono),
Género varchar(1) not null,
constraint CuentasTrans_check check(Género in('F','M')),
balance money)

insert into CuentasTrans values
('00001', 'Laurence Hills', '6524-8596', 'F', '5000'),
('00002', 'Carlos Stuart', '6894-7582', 'M', '3500'),
('00003', 'Marcos Jiménez', '6528-2586', 'M', '895'),
('00004', 'Berlinda Sáenz', '6728-8425', 'F', '4100')

SELECT * FROM CuentasTrans
```

Below the query, the "Results" tab is selected, showing a table with 6 columns: ID, Nombre, Teléfono, Género, and balance. The table contains 4 rows of data.

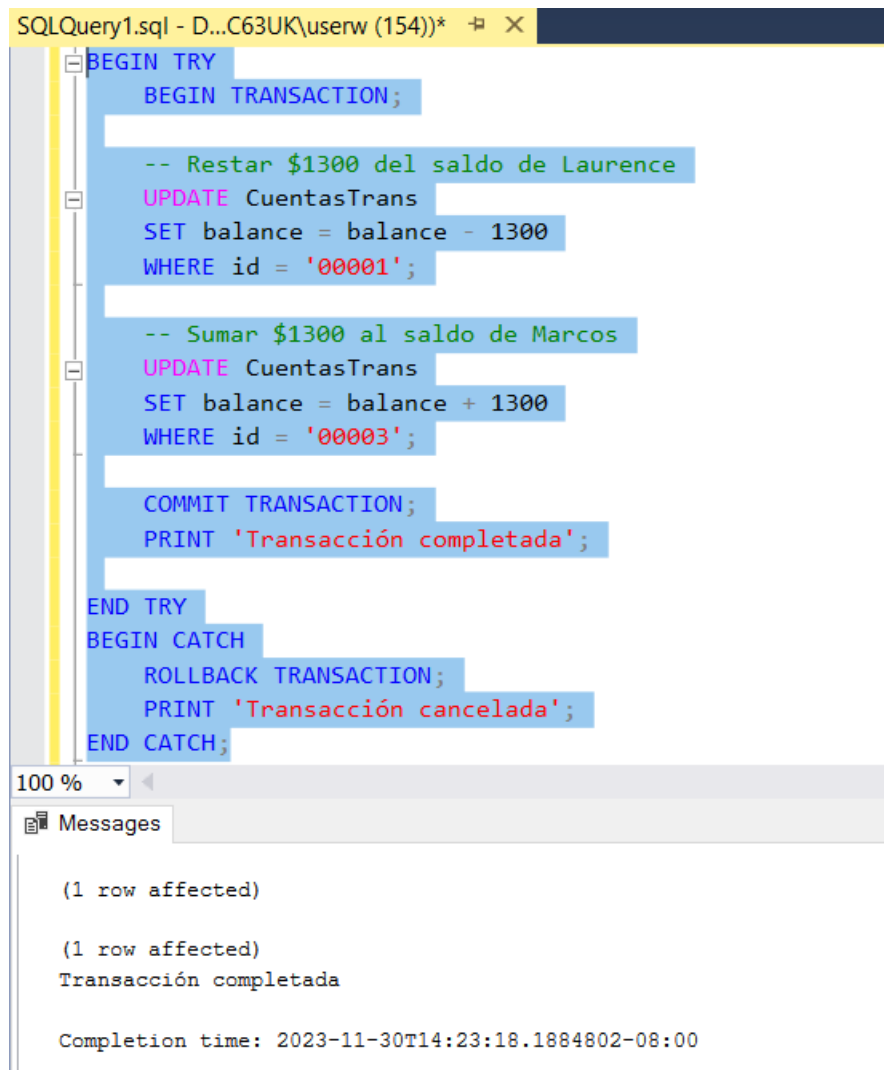
	ID	Nombre	Teléfono	Género	balance
1	00001	Laurence Hills	6524-8596	F	5000.00
2	00002	Carlos Stuart	6894-7582	M	3500.00
3	00003	Marcos Jiménez	6528-2586	M	895.00
4	00004	Berlinda Sáenz	6728-8425	F	4100.00

3. El banco debe asegurarse de restarle a Laurence los \$1300 de su cuenta y abonarlos a la cuenta de Marcos, en caso contrario, deshacer la transacción y devolverle los \$1300 a Laurence, esto lo hace por medio usando el *begin try, end try, begin catch, end catch*

Usar la siguiente guía:

```
BEGIN TRY
BEGIN TRANSACTION
***** hacer las sentencias necesarias para dicho cambio *****
COMMIT TRANSACTION
PRINT 'Transacción completada'
END TRY BEGIN CATCH
ROLLBACK TRANSACTION
PRINT 'Transacción cancelada'
END CATCH
```

Al usar **BEGIN TRY** en nuestro código lo que estamos haciendo es la prueba de errores, es decir, indicamos que ejecute todo el script que hay dentro de **BEGIN TRY** hasta **END TRY**, si no ha surgido ningún problema en ninguna de las sentencias, entonces que ejecute el **COMMIT** e imprime en pantalla el mensaje. Si hubiera existido algún error o falla en la transacción entonces se lanzaría el **CATCH** y por ende se dispararía el **ROLLBACK** y su respectivo mensaje.



The screenshot shows a SQL query window titled "SQLQuery1.sql - D...C63UK\userw (154))\*" containing a T-SQL script. The script uses a TRY-CATCH block to perform a transaction. It begins with "BEGIN TRY", followed by "BEGIN TRANSACTION;". Then, it has two UPDATE statements: one to subtract 1300 from the balance of account '00001' (Laurence) and another to add 1300 to the balance of account '00003' (Marcos). After these, it executes "COMMIT TRANSACTION;" and prints "Transacción completada;". The TRY block ends with "END TRY", followed by a CATCH block that rolls back the transaction and prints "Transacción cancelada;". The window is zoomed to 100%. Below the query window, the "Messages" pane shows the execution results: "(1 row affected)" for each UPDATE, "Transacción completada", and the completion time "2023-11-30T14:23:18.1884802-08:00".

```
BEGIN TRY
    BEGIN TRANSACTION;

    -- Restar $1300 del saldo de Laurence
    UPDATE CuentasTrans
    SET balance = balance - 1300
    WHERE id = '00001';

    -- Sumar $1300 al saldo de Marcos
    UPDATE CuentasTrans
    SET balance = balance + 1300
    WHERE id = '00003';

    COMMIT TRANSACTION;
    PRINT 'Transacción completada';
END TRY
BEGIN CATCH
    ROLLBACK TRANSACTION;
    PRINT 'Transacción cancelada';
END CATCH;
```

100 %

Messages

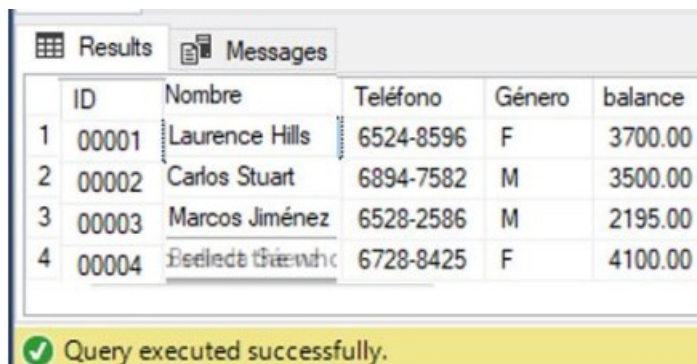
(1 row affected)

(1 row affected)

Transacción completada

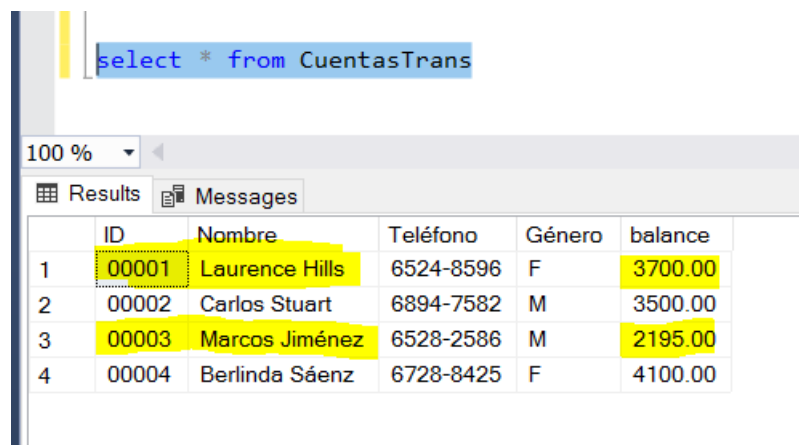
Completion time: 2023-11-30T14:23:18.1884802-08:00

4. EJECUTAR el código completo, visualizar la tabla para ver los cambios y debe salir así:



	ID	Nombre	Teléfono	Género	balance
1	00001	Laurence Hills	6524-8596	F	3700.00
2	00002	Carlos Stuart	6894-7582	M	3500.00
3	00003	Marcos Jiménez	6528-2586	M	2195.00
4	00004	Berlinda Sáenz	6728-8425	F	4100.00

Query executed successfully.



```
select * from CuentasTrans
```

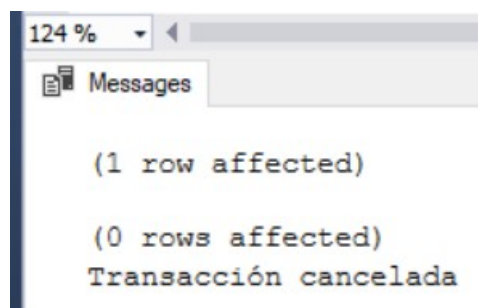
	ID	Nombre	Teléfono	Género	balance
1	00001	Laurence Hills	6524-8596	F	3700.00
2	00002	Carlos Stuart	6894-7582	M	3500.00
3	00003	Marcos Jiménez	6528-2586	M	2195.00
4	00004	Berlinda Sáenz	6728-8425	F	4100.00

5. Ahora veremos parte del uso de la sentencia anterior junto con el Rollback.

Que pasaría si Marcos le pide al banco que devuelva los \$1300 a Laurence y que adicional le de unos \$150.00 más por el favor que le hizo del préstamo, es decir que le de en total unos \$1450.00, pero el banco, sin embargo, por error de dedo en el número de cuenta (id) de Laurence coloca su nombre.

```
WHERE ID = 'Laurence'
```

5.1 Hagan los cambios y ejecuten el código completo, debe darles este mensaje en la ejecución



(1 row affected)

(0 rows affected)

Transacción cancelada

```
SQLQuery1.sql - D...C63UK\userw (154))* X
BEGIN TRY
    BEGIN TRANSACTION;

    -- Restar $1450 del saldo de Marcos
    UPDATE CuentasTrans
    SET balance = balance - 1450
    WHERE ID = '00003';

    -- Verificar si el ID 'Laurence' existe
    IF EXISTS (SELECT 1 FROM CuentasTrans WHERE ID = 'Laurence')
    BEGIN
        -- Se coloca intencionalmente el error en el ID
        UPDATE CuentasTrans
        SET balance = balance + 1450
        WHERE ID = 'Laurence';

        COMMIT TRANSACTION;
        PRINT 'Transacción completada';
    END
    ELSE
    BEGIN
        -- Si el ID no existe, activar una excepción para el bloque CATCH
        THROW 50001, 'El ID especificado no existe en la tabla.', 1;
    END
END TRY

100 %
Messages
(1 row affected)
Error: El ID especificado no existe en la tabla.
Transacción cancelada
```

Nota: la variación en la salida se debe a que se colocó la validación del id para la actualización del balance a Laurence, ya que anteriormente se notó que al no hacerlo el rollback no se llevaba a cabo y la transacción se realizaba a medias, restándole el balance a Marcos.

Comprobacion de que los balances se mantienen:

```
END TRY
BEGIN CATCH
    ROLLBACK TRANSACTION;
    PRINT 'Error: ' + ERROR_MESSAGE(); -- Imprimir el mensaje de error
    PRINT 'Transacción cancelada';
END CATCH;
```

```
select * from CuentasTrans
```

100 %

Results Messages

	ID	Nombre	Teléfono	Género	balance
1	00001	Laurence Hills	6524-8596	F	3700.00
2	00002	Carlos Stuart	6894-7582	M	3500.00
3	00003	Marcos Jiménez	6528-2586	M	2195.00
4	00004	Berlinda Sáenz	6728-8425	F	4100.00