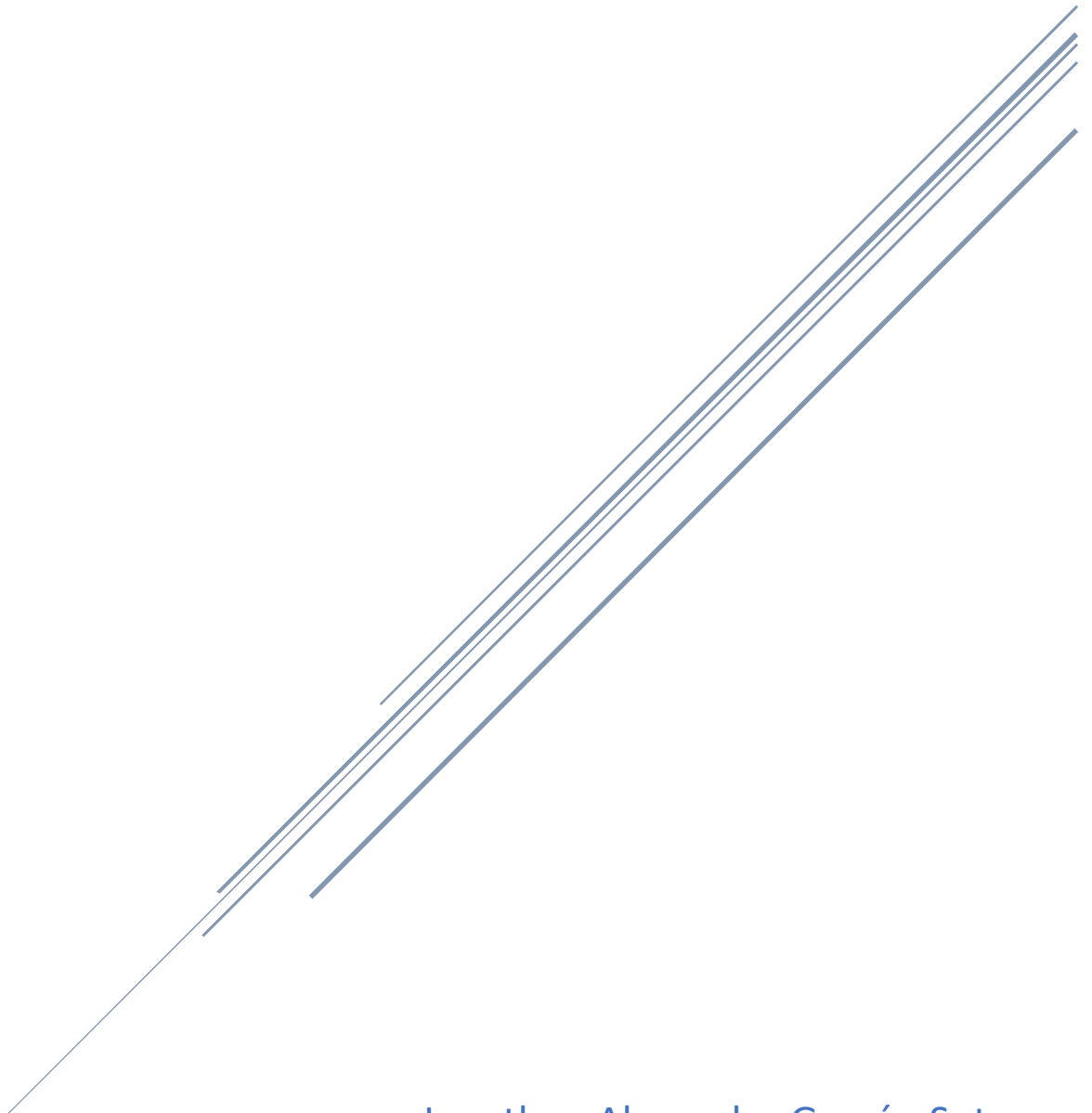


TRANSACCIÓN

PL-SQL



Jonathan Alexander Garzón Soto
635601

```

CREATE TABLE cuentas (
    cuenta VARCHAR2(30),
    saldo NUMBER
);

CREATE TABLE movimientos(

    cuenta_Origen VARCHAR2(30),
    cuenta_Destino VARCHAR2(30),
    importe NUMBER,
    f_Movimiento TIMESTAMP(0),
    se_realizo VARCHAR2(2)
);

INSERT INTO cuentas
(cuenta, saldo)
VALUES
('1234567', 1000000);

INSERT INTO cuentas
(cuenta, saldo)
VALUES
('89012345', 2000000);

```

Creación de tablas para registrar las cuentas y los movimientos que se lleguen a realizar, también se realiza la inserción de datos en la tabla cuentas con algunos datos de ejemplo para realizar el ejercicio.

```

/*Creacion de procedimineto almacenado para manejo de las transacciones*/
CREATE OR REPLACE PROCEDURE Transaccion(importe NUMBER,
    ctaOrigen VARCHAR2,
    ctaDestino VARCHAR2)
AS
/*Declaracion de variables para incluirlos en los cursores implicitos*/
    saldoO NUMBER;
    saldoD NUMBER;
    cuentaO VARCHAR2(30);
    cuentaD VARCHAR2(30);
/*Begin inicio de lo que va a realizar el procedimiento almacenado*/
BEGIN
    /*Uso del primer cursos implicito para verificar cuenta origen*/
    SELECT cuenta INTO cuentaO FROM cuentas
    WHERE cuenta = ctaOrigen;
    /*Uso del segundo cursos implicito para verificar cuenta origen*/
    SELECT cuenta INTO cuentaD FROM cuentas
    WHERE cuenta = ctaDestino;
    /*evalua si la cuenta origen ingresada como parametro del procedimiento
    es igual a lo encontrado en el cursor*/
    IF (cuentaO = ctaOrigen) THEN
    /*Uso del tercer cursos implicito para verificar cuenta el saldo de la cuenta
    origen*/
        SELECT saldo INTO saldoO FROM cuentas
        WHERE cuenta = ctaOrigen;
        /*evalua si la cuenta destino ingresada como parametro del procedimiento
        es igual a lo encontrado en el cursor*/

```

```

IF (cuentaD = ctaDestino) THEN
/*Uso del cuarto curso implicito para verificar cuenta el saldo de la cuenta
origen*/
SELECT saldo INTO saldoD FROM cuentas
WHERE cuenta = ctaDestino;
/*verifica si el importe es menor que el saldo de la cuenta origen*/
IF (saldoO>importe) THEN
/*Si la condicion anterior se cumple realiza la transaccion*/
UPDATE cuentas SET saldo = saldo - importe
WHERE cuenta = ctaOrigen;

UPDATE cuentas SET saldo = saldo + importe
WHERE cuenta = ctaDestino;
/*Aqui realiza un registro de la operacion*/
INSERT INTO MOVIMIENTOS
(cuenta_Origen, cuenta_Destino, importe, f_Movimiento, se_realizo)
VALUES
(ctaOrigen, ctaDestino, importe*(-1), SYSTIMESTAMP, 'SI');

INSERT INTO MOVIMIENTOS
(cuenta_Origen, cuenta_Destino, importe, f_Movimiento, se_realizo)
VALUES
(ctaDestino, ctaOrigen, importe, SYSTIMESTAMP, 'SI');
/*Si se realiza esta correctamente la transaccion arroja el
siguiente mensaje*/
DBMS_OUTPUT.PUT_LINE('Transaccion Realizada con exito');

```

```

ELSE
/*Si la operacion no es exitosa arroja el siguiente mensaje*/
DBMS_OUTPUT.PUT_LINE('Saldos insuficiente para esta transacción');
/*Ingresa el movimiento que se intento realizar e indicara que no
fue posible realizarlo*/
INSERT INTO MOVIMIENTOS
(cuenta_Origen, cuenta_Destino, importe, f_Movimiento, se_realizo)
VALUES
(ctaOrigen, ctaDestino, importe*(-1), SYSTIMESTAMP, 'NO');

INSERT INTO MOVIMIENTOS
(cuenta_Origen, cuenta_Destino, importe, f_Movimiento, se_realizo)
VALUES
(ctaDestino, ctaOrigen, importe, SYSTIMESTAMP, 'NO');
END IF;
ELSE
/*Si la cuenta destino no existe entonces arroja el siguiente mensaje */
DBMS_OUTPUT.PUT_LINE('La cuenta destino no existe');
/*Ingresa el movimiento que se intento realizar*/
INSERT INTO MOVIMIENTOS
(cuenta_Origen, cuenta_Destino, importe, f_Movimiento, se_realizo)
VALUES
(ctaOrigen, ctaDestino, importe*(-1), SYSTIMESTAMP, 'NO');

INSERT INTO MOVIMIENTOS
(cuenta_Origen, cuenta_Destino, importe, f_Movimiento, se_realizo)
VALUES
(ctaDestino, ctaOrigen, importe, SYSTIMESTAMP, 'NO');
END IF;

```

```

ELSE
/*Si la cuenta origen no existe arroja el siguiente mensaje*/
  DBMS_OUTPUT.PUT_LINE('La cuenta origen no existe');
/*Ingresa el movimiento que se intento realizar*/
  INSERT INTO MOVIMIENTOS
    (cuenta_Origen, cuenta_Destino, importe, f_Movimiento, se_realizo)
  VALUES
    (ctaOrigen, ctaDestino, importe*(-1), SYSTIMESTAMP, 'NO');

  INSERT INTO MOVIMIENTOS
    (cuenta_Origen, cuenta_Destino, importe, f_Movimiento, se_realizo)
  VALUES
    (ctaDestino, ctaOrigen, importe, SYSTIMESTAMP, 'NO');

END IF;

EXCEPTION

/*Si algun error inesperado sucede arroja el siguiente mensaje*/
WHEN OTHERS THEN

  dbms_output.put_line('Error en la transaccion: '||SQLERRM);

END Transaccion;
/*Fina de procedimiento almacenado*/

```

Compilación del proceso almacenado

```

/*Si algun error inesperado sucede arroja el siguiente mensaje */
WHEN OTHERS THEN

  dbms_output.put_line('Error en la transaccion: '||SQLERRM);

END Transaccion;
/*Fina de procedimiento almacenado*/

/*Permite ver los mensajes por consola*/
set SERVEROUTPUT on;
/*Ejecucion de procedimiento anteriormente creado*/
EXECUTE transaccion(10000000,'89012345','1234567');

```

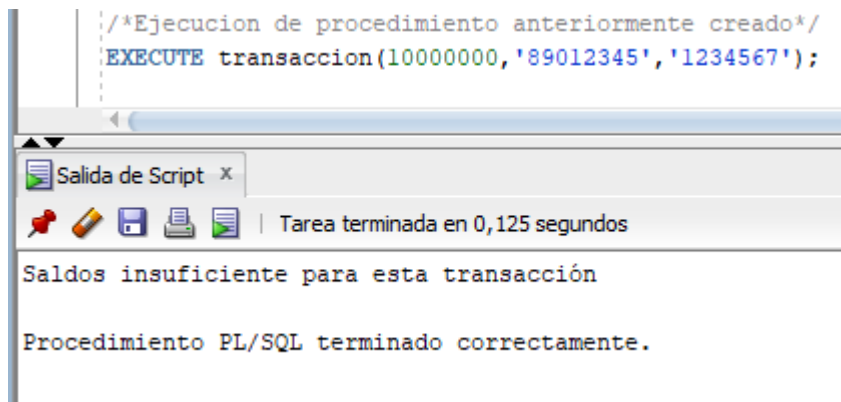
Tarea terminada en 2,375 segundos

Procedure TRANSACCION compilado

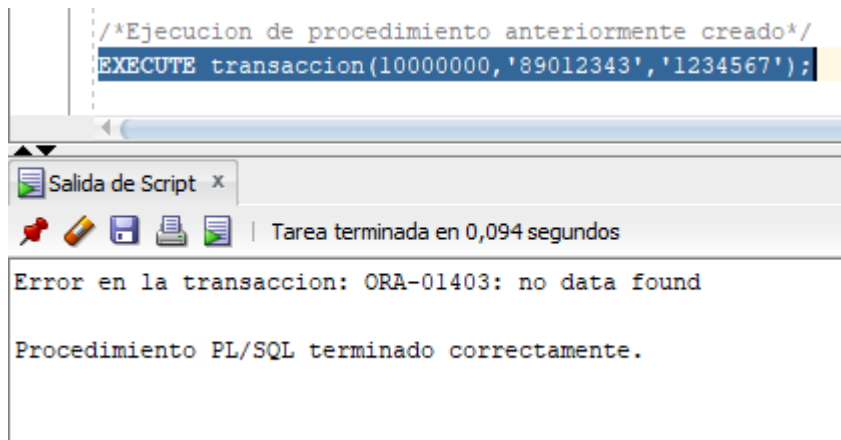
```

/*Permite ver los mensajes por consola*/
set SERVEROUTPUT on;
/*Ejecucion de procedimiento anteriormente creado*/
EXECUTE transaccion(10000000,'89012345','1234567');

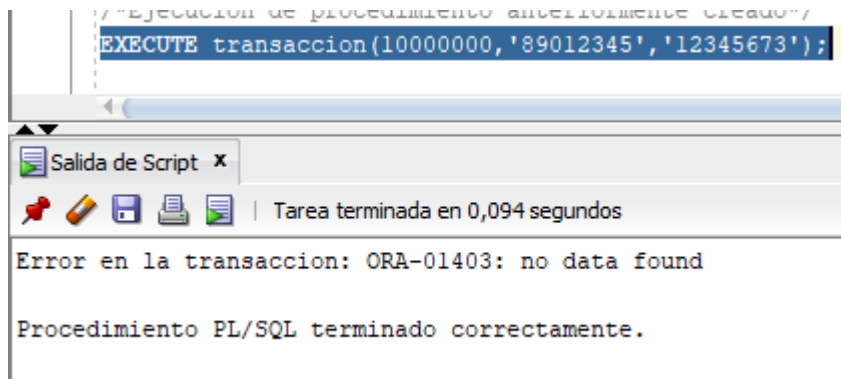
```



Indica mensaje ya que el valor del importe es mayor a el saldo de la cuenta origen



No encuentra la cuenta origen



No encuentra la cuenta destino

```
/*Ejecucion de procedimiento anteriormente creado*/  
EXECUTE transaccion(100000,'89012345','1234567');
```

Salida de Script x

Tarea terminada en 0,093 segundos

Transaccion Realizada con exito

Procedimiento PL/SQL terminado correctamente.

Transacción se realiza correctamente

```
select * from cuentas  
  
select * from movimientos
```

Salida de Script x Resultado de la Consulta x

Todas las Filas Recuperadas: 2 en 0,016 segun

	CUENTA	SALDO
1	1234567	1100000
2	89012345	1900000

Prueba de la realización de la transacción

select * from movimientos					
Salida de Script x Resultado de la Consulta x					
SQL Todas las Filas Recuperadas: 12 en 0,031 segundos					
	CUENTA_ORIGEN	CUENTA_DESTINO	IMPORTE	F_MOVIMIENTO	SE_REALIZO
5	89012345	1234567	-10000000	08/10/19 09:22:01,000000000	AM NO
6	1234567	89012345	10000000	08/10/19 09:22:01,000000000	AM NO
7	89012345	1234567	-10000000	08/10/19 09:25:12,000000000	AM NO
8	1234567	89012345	10000000	08/10/19 09:25:12,000000000	AM NO
9	89012345	1234567	-10000000	08/10/19 09:25:20,000000000	AM NO
10	1234567	89012345	10000000	08/10/19 09:25:20,000000000	AM NO
11	89012345	1234567	-100000	08/10/19 09:29:55,000000000	AM SI
12	1234567	89012345	100000	08/10/19 09:29:55,000000000	AM SI

Movimientos realizados con el numero de cuenta ingresado, el importe requerido, el momento con fecha y hora y si se realizo o no el proceso.