

Transacciones

SGBD

Jonathan Alexander Garzon Soto 635601

CORPORACIÓN UNIVERSITARIA MINUTO DE DIOS

Tablas Creadas





```
CREATE TABLE cuentas (  
    cuenta VARCHAR2(30),  
    saldo NUMBER  
);  
  
select * from cuentas;  
  
CREATE TABLE movimientos(  
    cuenta_Origen VARCHAR2(30),  
    cuenta_Destino VARCHAR2(30),  
    importe NUMBER,  
    f_Movimiento TIMESTAMP(0),  
    se_realizo VARCHAR2(2)  
);
```

Cuentas utilizadas

```
select * from cuentas;
```

Salida de Script x

Resultado de la Consulta x

 | Todas las Filas Recuperadas: 2 en 0,008 segundos

	CUENTA	SALDO
1	1234567	1130000
2	89012345	1870000

Tabla de transacciones

Select * from movimientos;

INSERT INTO cuentas

Resultado de la Consulta x

Todas las Filas Recuperadas: 14 en 0,023 segundos

	CUENTA_ORIGEN	CUENTA_DESTINO	IMPORTE	F_MOVIMIENTO	SE_REALIZO
1	89012345	1234567	-100000	25/03/20 07:08:59,000000000	PM SI
2	1234567	89012345	100000	25/03/20 07:09:00,000000000	PM SI
3	89012345	1234567	-100000	25/03/20 07:26:15,000000000	PM SI
4	1234567	89012345	100000	25/03/20 07:26:15,000000000	PM SI
5	89012345	1234567	-100000	25/03/20 07:29:23,000000000	PM SI
6	1234567	89012345	100000	25/03/20 07:29:23,000000000	PM SI
7	89012345	1234567	-100000	25/03/20 07:29:23,000000000	PM SI
8	1234567	89012345	100000	25/03/20 07:29:23,000000000	PM SI
9	1234567	89012345	-100000	07/04/20 06:21:13,000000000	PM SI
10	89012345	1234567	100000	07/04/20 06:21:13,000000000	PM SI

Procedimiento creado

```
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 count(*) INTO cuentaO FROM cuentas
    WHERE cuenta = ctaOrigen;
    /*Uso del segundo cursos implicito para verificar cuenta origen*/
    SELECT count(*) 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 = 1) 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 = 1) 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('Saldo 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;

```

```

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;

```

Ejecución proceso

Proceso exitoso

```

/*Ejecucion de procedimiento anteriormente creado*/
EXECUTE transaccion(1700000,'89012345','1234567');

EXECUTE transaccion(170000,'1234567','89012345');

EXECUTE transaccion(1700000,'1234567','89012345');

select * from cuentas;

```

Salida de Script x

Tarea terminada en 0,117 segundos

Transaccion Realizada con exito

Procedimiento PL/SQL terminado correctamente.

Estado cuentas

```
select * from cuentas;
```

	CUENTA	SALDO
1	1234567	2830000
2	89012345	170000

Ejecución proceso

Transacción exitosa

```
EXECUTE transaccion(170000,'1234567','89012345');  
  
EXECUTE transaccion(1700000,'1234567','89012345');  
  
select * from cuentas;
```

Salida de Script x Resultado de la Consulta x

Tarea terminada en 0,099 segundos

Transaccion Realizada con exito

Procedimiento PL/SQL terminado correctamente.

Estado cuentas

```
select * from cuentas;
```

	CUENTA	SALDO
1	1234567	2490000
2	89012345	510000

Fondos insuficientes

```
EXECUTE transaccion(1700000,'1234567','89012345');

select * from cuentas;
```

Salida de Script x Resultado de la Consulta x

Tarea terminada en 0,114 segundos

Saldos insuficiente para esta transacción

Procedimiento PL/SQL terminado correctamente.

Error con la cuenta

```
EXECUTE transaccion(1700000,'12345679','89012345');

select * from cuentas;
```

Salida de Script x Resultado de la Consulta x

Tarea terminada en 0,135 segundos

La cuenta origen no existe

Procedimiento PL/SQL terminado correctamente.

```
EXECUTE transaccion(1700000,'1234567','890123456');

select * from cuentas;
```

Salida de Script x Resultado de la Consulta x

Tarea terminada en 0,133 segundos

La cuenta destino no existe

Procedimiento PL/SQL terminado correctamente.