
trigger_saldo.sql

```
SET echo ON
SELECT sysdate FROM dual;
```

```
CREATE TABLE Cuentas(
  nCC number(3) PRIMARY KEY,
  saldo number(6) NOT NULL
);
```

```
/* disparador que impide la disminución del saldo en cuentas con saldo negativo V1
*/
```

```
CREATE OR REPLACE TRIGGER controlSaldo
BEFORE UPDATE OF saldo ON Cuentas
FOR EACH ROW
BEGIN
  IF :old.saldo < 0 AND :new.saldo < :old.saldo THEN
    raise_application_error (-20900, 'decremento de saldo no permitido');
  END IF;
END controlSaldo;
/
```

```
SHOW ERRORS TRIGGER controlSaldo
```

```
INSERT INTO Cuentas VALUES (1, 25);
INSERT INTO Cuentas VALUES (2, -3);
```

```
/* dará error al intentar modificar el saldo de la cuenta 2 y se deshará la
transacción; es decir, no se modificará ninguna cuenta */
```

```
UPDATE Cuentas SET saldo = saldo - 5;

SELECT * FROM Cuentas;
UPDATE Cuentas SET saldo = saldo - 5 WHERE nCC = 1;
UPDATE Cuentas SET saldo = saldo - 5 WHERE nCC = 2;
SELECT * FROM Cuentas;

UPDATE Cuentas SET saldo = saldo + 5 WHERE nCC = 2;
UPDATE Cuentas SET saldo = saldo - 7 WHERE nCC = 2;
```

```
/* disparador que impide la disminución del saldo en cuentas con saldo negativo V2
*/
```

```
CREATE OR REPLACE TRIGGER controlSaldo
BEFORE UPDATE OF saldo ON Cuentas
FOR EACH ROW
WHEN (old.saldo < 0 AND new.saldo < old.saldo)
BEGIN
  raise_application_error (-20900, 'decremento de saldo no permitido');
END controlSaldo;
/
```

```
SHOW ERRORS TRIGGER controlSaldo
```

```
/* disparador que impide la disminución del saldo en cuentas con saldo negativo V3
*/
```

```
CREATE OR REPLACE TRIGGER controlSaldo
AFTER UPDATE OF saldo ON Cuentas
FOR EACH ROW
WHEN (old.saldo < 0 AND new.saldo < old.saldo)
BEGIN
  raise_application_error (-20900, 'decremento de saldo no permitido');
END controlSaldo;
/
```

```
SHOW ERRORS TRIGGER controlSaldo
```

```
/* disparador que impide la disminución del saldo en cuentas con saldo negativo V4
*/
```

```
CREATE OR REPLACE TRIGGER controlSaldo
AFTER UPDATE OF saldo ON Cuentas
FOR EACH ROW
BEGIN
  IF :old.saldo < 0 AND :new.saldo < :old.saldo THEN
    raise_application_error (-20900, 'decremento de saldo no permitido');
  END IF;
END controlSaldo;
/
```

```
SHOW ERRORS TRIGGER controlSaldo
```

```
/* disparador que impide la disminución del saldo en cuentas con saldo negativo V5
*/
```

```
CREATE OR REPLACE TRIGGER controlSaldo
BEFORE UPDATE OF saldo ON Cuentas
FOR EACH ROW
BEGIN
  IF :old.saldo < 0 AND :new.saldo < :old.saldo THEN
    /* provocar un error en la operación de actualización */
    :new.nCC := NULL; /* o :new.nCC := NULL; */
  END IF;
END controlSaldo;
/
```

```
SHOW ERRORS TRIGGER controlSaldo
```

```
/* disparador que impide la disminución del saldo en cuentas con saldo negativo V6
*/
```

```
CREATE OR REPLACE TRIGGER controlSaldo
BEFORE UPDATE OF saldo ON Cuentas
FOR EACH ROW
BEGIN
  IF :old.saldo < 0 AND :new.saldo < :old.saldo THEN
    :new.saldo := :old.saldo;
  /* es decir, no cambia el saldo si se intenta decrementar un saldo negativo
  (equivale a que el banco le "perdona" el gasto :-) */
  END IF;
END controlSaldo;
/
```

```
SHOW ERRORS TRIGGER controlSaldo
```

```
/* disparador que impide la disminución del saldo en cuentas con saldo negativo V7
no funciona: error de tabla mutante
*/
```

```
CREATE OR REPLACE TRIGGER controlSaldo
AFTER UPDATE OF saldo ON Cuentas
FOR EACH ROW
BEGIN
  IF :old.saldo < 0 AND :new.saldo < :old.saldo THEN
    UPDATE cuentas SET saldo = :old.saldo WHERE ncc = :new.nCC;
  /* es decir, deshace el cambio si se intenta decrementar un saldo negativo
  (equivale a que el banco le "perdona" el gasto :-) */
  END IF;
END controlSaldo;
/
```

```
SHOW ERRORS TRIGGER controlSaldo
```

```
DROP TRIGGER controlSaldo;
DROP TABLE Cuentas;
```

```
SELECT username, SID, SERIAL# FROM v$session;
ALTER system kill session '26, 64832';
ALTER system kill session '26, 64832' immediate;
```

```
SET LINES 350 PAGES 200;
```

```
COL USERNAME FOR A16;
```

```
COL OSUSER FOR A16;
```

```
COL program FOR A50;
```

```
COL TYPE FOR A12;
```

```
COL MACHINE FOR A35;
```

```
SELECT S.sid, S.SERIAL#, S.USERNAME, S.TYPE, S.OSUSER, S.STATUS, S.MACHINE, S.program
FROM V$SESSION S
```

```
WHERE 1=1
```

```
AND TYPE = 'USER'
```

```
ORDER BY case when s.TYPE = 'USER' THEN 0 ELSE 1 END asc, S.PROGRAM;
```