

**max\_2\_ProvPieza\_v2.sql**

```
/* @max_2_ProvPieza_v2.sql */
SET ECHO ON
SET SERVEROUTPUT ON
SET linesize 132
SET pagesize 300
column color      format A32
column numProv    format 999

/* Añadir a la tabla Pieza un atributo para el total de proveedores de la pieza
*/
ALTER TABLE Pieza ADD(numProv number);
ALTER TABLE Pieza ADD CONSTRAINT numProv_NoNeg CHECK(numProv>=0);

-- eliminar el atributo para añadirlo de otro modo
ALTER TABLE Pieza DROP CONSTRAINT numProv_NoNeg;
ALTER TABLE Pieza DROP(numProv);

-- o directamente
ALTER TABLE Pieza ADD(numProv number DEFAULT 0 CONSTRAINT numProv_NoNeg CHECK(numProv>=0));

/* Crear un procedimiento para calcular el total de proveedores de cada pieza
*/
CREATE OR REPLACE PROCEDURE set_numProvPieza_PR
IS
BEGIN
    UPDATE Pieza P SET numProv=(SELECT count(*) FROM Suministrar WHERE clvPieza = P.clvPieza);
END set_numProvPieza_PR;
/
SHOW ERRORS PROCEDURE set_numProvPieza_PR

column OBJECT_NAME      heading "nombre"      format a20;
column DATA_OBJECT_ID  heading "id"          format a2;
column SUBOBJECT_NAME   heading "subobjeto"   format a9;

/* mostrar todos los procedimientos del usuario */
SELECT * FROM user_objects WHERE object_type='PROCEDURE';

/* asignar valor al total de proveedores de cada pieza
*/
EXECUTE set_numProvPieza_PR

/* Crear un disparador para actualizar el nº de Proveedores por pieza
*/
CREATE OR REPLACE TRIGGER updateNumProvPieza_TR
AFTER INSERT OR DELETE OR UPDATE ON Suministrar
FOR EACH ROW
BEGIN
    IF INSERTING THEN
        UPDATE Pieza SET numProv=numProv+1 WHERE clvPieza = :NEW.clvPieza;
    ELSIF DELETING THEN
        UPDATE Pieza SET numProv=numProv-1 WHERE clvPieza = :OLD.clvPieza;
    ELSIF UPDATING THEN
        UPDATE Pieza SET numProv=numProv+1 WHERE clvPieza = :NEW.clvPieza;
        UPDATE Pieza SET numProv=numProv-1 WHERE clvPieza = :OLD.clvPieza;
    END IF;
END updateNumProvPieza_TR;
/
SHOW ERRORS TRIGGER updateNumProvPieza_TR
```

```
/* Crear un disparador para impedir que el nº de Proveedores por pieza sea > 2
*/
```

```
CREATE OR REPLACE TRIGGER max2ProvPieza_TR
BEFORE INSERT OR UPDATE ON Suministrar
FOR EACH ROW
DECLARE
    totProv number;
BEGIN
    SELECT numProv INTO totProv
    FROM Pieza
    WHERE :NEW.clvPieza = clvPieza;
    DBMS_OUTPUT.PUT_LINE('la pieza ya tiene '||totProv||' proveedores');

    IF totProv > 1
    THEN
        raise_application_error( -20501, 'la pieza tiene el máximo de proveedores');
    END IF;
END max2ProvPieza_TR;
```

```
/
SHOW ERRORS TRIGGER max2ProvPieza_TR
```

```
SELECT * FROM Pieza;
SELECT * FROM Suministrar ORDER BY clvProv, clvPieza;
```

```
-- Mostrar todos los disparadores y su estado
SELECT TRIGGER_NAME, STATUS FROM USER_TRIGGERS;
```

```
-- no se producirá error, pues suministrar ya no es mutante
INSERT INTO suministrar VALUES (2, 95);
INSERT INTO suministrar VALUES (2, 91);
INSERT INTO suministrar VALUES (3, 92);
INSERT INTO suministrar VALUES (3, 91);
```

```
DELETE FROM suministrar WHERE clvProv = 3;
```

```
BEGIN
    INSERT INTO suministrar SELECT 3, clvPieza FROM Suministrar WHERE clvProv=2;
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE = -20501 THEN DBMS_OUTPUT.PUT_LINE('ERROR: max. 2 Prov/Pieza');
        END IF;
-- DBMS_OUTPUT.PUT_LINE('ERROR '||SQLERRM);
END;
```

```
/
```

```
-----
-- FINAL DEL TEST. Eliminar de la BD los elementos añadidos.
```

```
-- eliminar los disparadores y procedures creados en este ejemplo
DROP TRIGGER max2ProvPieza_TR;
DROP TRIGGER updateNumProvPieza_TR;
DROP PROCEDURE set_numProvPieza_PR;
```

```
-- eliminar las tuplas añadidas
DELETE FROM suministrar WHERE (clvProv, clvPieza) IN ((2,91), (2,95));
DELETE FROM suministrar WHERE clvProv = 3;
```

```
-- eliminar el atributo numProv añadido a la tabla Pieza
ALTER TABLE Pieza DROP COLUMN numProv;
```