
ejla_trigger.sql

```
/* Este ejemplo pretende ilustrar las diferencias entre los diferentes modos de especificar
   los eventos en disparadores (a nivel de sentencia o a nivel de tupla y BEFORE o AFTER) y
   mostrar el orden en que se ejecutan. Para ello se ha implementado un disparador de cada
   tipo que sólo muestra un mensaje al ejecutarse una sentencia de eliminar un suministro.
   S.V. Univ. Zaragoza, 10 mayo 2015
*/
```

```
SET ECHO ON
SET SERVEROUTPUT ON
SET linesize 132
SET pagesize 100
```

```
-- modificar el ancho para columnas de tipo LONG (p.e. el cuerpo de un trigger)
SET LONG 255
```

```
-- mostrar las tuplas de suministrar
SELECT * FROM Suministrar;
```

```
/*-----
   Crear un disparador para ilustrar BEFORE a nivel de sentencia
*/
```

```
CREATE OR REPLACE TRIGGER tr_BefDel_Sum
BEFORE DELETE ON Suministrar
BEGIN
    DBMS_OUTPUT.PUT_LINE('accion BEFORE DELETE en tabla Suministrar');
    -- raise_application_error(-20501, 'error antes de iniciar el borrado en Suministrar');
END tr_BefDel_Sum;
/
```

```
SHOW ERRORS TRIGGER tr_BefDel_Sum
```

```
/*-----
   Crear un disparador para ilustrar BEFORE a nivel de tupla
*/
```

```
CREATE OR REPLACE TRIGGER tr_BefDelRow_Sum
BEFORE DELETE ON Suministrar
FOR EACH ROW
BEGIN
    DBMS_OUTPUT.PUT_LINE('accion BEFORE DELETE el suministro (' || :OLD.clvProv || ', ' || :OLD.clvPieza || ')');
    -- raise_application_error(-20501, 'error antes de borrar tupla en Suministrar');
END tr_BefDelRow_Sum;
/
```

```
SHOW ERRORS TRIGGER tr_BefDelRow_Sum
```

```
/*-----
   Crear un disparador para ilustrar AFTER a nivel de tupla
*/
```

```
CREATE OR REPLACE TRIGGER tr_AftDelRow_Sum
AFTER DELETE ON Suministrar
FOR EACH ROW
BEGIN
    DBMS_OUTPUT.PUT_LINE('accion AFTER DELETE el suministro (' || :OLD.clvProv || ', ' || :OLD.clvPieza || ')');
    -- raise_application_error(-20501, 'error despues de borrar tupla en Suministrar');
END tr_AftDelRow_Sum;
/
```

```
SHOW ERRORS TRIGGER tr_AftDelRow_Sum
```

```
/*-----
   Crear un disparador para ilustrar AFTER a nivel de sentencia
*/
```

```
CREATE OR REPLACE TRIGGER tr_AftDel_Sum
AFTER DELETE ON Suministrar
BEGIN
    DBMS_OUTPUT.PUT_LINE('accion AFTER DELETE en tabla Suministrar');
    -- raise_application_error(-20501, 'error despues de borrar en Suministrar');
END tr_AftDel_Sum;
/
```

```
SHOW ERRORS TRIGGER tr_AftDel_Sum
```

```

column TRIGGER_NAME      heading "nombre"      format a20;
column TRIGGER_TYPE      heading "tipo"        format a20;
column TRIGGERING_EVENT  heading "evento"     format a60;

/* mostrar todos los disparadores del usuario */
-- select * from USER_TRIGGERS;
SELECT TRIGGER_NAME, TRIGGER_TYPE, TRIGGERING_EVENT FROM USER_TRIGGERS;

column OBJECT_NAME      heading "nombre"      format a18;
column DATA_OBJECT_ID  heading "id"         format a2;
column SUBOBJECT_NAME   heading "subobjeto"   format a9;

/* mostrar todos los disparadores del usuario */
SELECT * FROM user_objects WHERE object_type='TRIGGER';

/* mostrar todos los disparadores sobre la tabla Suministrar */
SELECT TRIGGER_NAME, TABLE_OWNER FROM USER_TRIGGERS WHERE TABLE_NAME = 'SUMINISTRAR';

-- insertar algunas tuplas en suministrar
INSERT INTO suministrar VALUES (1, 96);
INSERT INTO suministrar VALUES (1, 97);
INSERT INTO suministrar VALUES (3, 97);

SELECT * FROM Suministrar ORDER BY clvPieza;

-- eliminar una única tupla
DELETE FROM suministrar WHERE clvPieza=96;

-- eliminar dos tuplas en una única sentencia
DELETE FROM suministrar WHERE clvPieza=97;

-- eliminar los disparadores creados en este ejemplo
DROP TRIGGER tr_BefDel_Sum;
DROP TRIGGER tr_BefDelRow_Sum;
DROP TRIGGER tr_AftDel_Sum;
DROP TRIGGER tr_AftDelRow_Sum;

-- eliminar las tuplas añadidas a suministrar (sólo para el caso en que los disparadores no lo hubieran permitido)
DELETE FROM suministrar WHERE clvPieza=96;
DELETE FROM suministrar WHERE clvPieza=97;

```