1. **Codificar un procedimiento que permita borrar un empleado cuyo número se pasará en la llamada.**

**---------BLOQUE ANÓNIMO-----------**

SET SERVEROUTPUT ON;

BEGIN

EJ16(7369); --Borrar Smith

END;

**---------PROCEDIMIENTO EJ16-------**

CREATE OR REPLACE PROCEDURE EJ16

(

P\_EMPNO IN NUMBER

)

AS

P\_EXISTE VARCHAR2(20); -- Aquí voy a almacenar el nombre para comprobar que existe

BEGIN

EXECUTE IMMEDIATE 'ALTER TABLE offerings DISABLE CONSTRAINT O\_TRAIN\_FK';

--Necesito desactivar esta constraint para que no me dé error.

--Para usar el comando DDL, al parecer debo usar el comando "EXECUTE IMMEDIATE"

/\*

Ejemplo del error que lanzaba:

Informe de error -

ORA-02292: restricción de integridad (CLASE.O\_TRAIN\_FK) violada - registro secundario encontrado

ORA-06512: en "CLASE.EJ16", línea 6

ORA-06512: en línea 2

02292. 00000 - "integrity constraint (%s.%s) violated - child record found"

\*Cause: attempted to delete a parent key value that had a foreign

dependency.

\*Action: delete dependencies first then parent or disable constraint.

\*/

DELETE FROM employees

WHERE empno=P\_EMPNO;

/\*

--En este punto, el empleado debería estar borrado.

--Para comprobarlo, si intentamos mostrar el nombre del empleado

--descomentando esto, debería dar error al ejecutar.

SELECT ename

INTO P\_EXISTE

FROM EMPLOYEES

WHERE empno=P\_EMPNO;

DBMS\_OUTPUT.PUT\_LINE(P\_EXISTE);

\*/

/\*

Ejemplo del error:

Informe de error -

ORA-01403: No se ha encontrado ningún dato

ORA-06512: en "CLASE.EJ16", línea 14

ORA-06512: en línea 2

01403. 00000 - "no data found"

\*/

--Deshacemos todo con un rollback (y reactivamos la constraint también por si acaso).

EXECUTE IMMEDIATE 'ROLLBACK';

EXECUTE IMMEDIATE 'ALTER TABLE offerings ENABLE CONSTRAINT O\_TRAIN\_FK';

--Mostramos el nombre del empleado. Como hemos deshecho los cambios, no debería dar error.

SELECT ename

INTO P\_EXISTE

FROM EMPLOYEES

WHERE empno=P\_EMPNO;

DBMS\_OUTPUT.PUT\_LINE(P\_EXISTE ||' ha sido borrado (y hemos realizado el rollback)');

END EJ16;

**---------RESULTADO DEL CÓDIGO-------**

bloque anónimo terminado

SMITH ha sido borrado (y hemos realizado el rollback)

1. **Escribir un procedimiento que modifique la localidad de un departamento. El procedimiento recibirá como parámetros el número del departamento y la localidad nueva.**

**---------BLOQUE ANÓNIMO-----------**

SET SERVEROUTPUT ON;

BEGIN

EJ17(10, 'MADRID'); --original era 'NEW YORK'

END;

**--------- PROCEDIMIENTO EJ17-------**

CREATE OR REPLACE PROCEDURE EJ17

(

P\_DEPARTAMENTO IN NUMBER

, P\_LOCALIDAD IN VARCHAR2

)

AS

P\_LOCALIDAD\_SELECT VARCHAR2(20);

BEGIN

SELECT location

INTO P\_LOCALIDAD\_SELECT

FROM departments

WHERE deptno=p\_departamento;

DBMS\_OUTPUT.PUT\_LINE('Localidad actual: '||P\_LOCALIDAD\_SELECT);

UPDATE departments SET location=p\_localidad

WHERE deptno=p\_departamento;

SELECT location

INTO P\_LOCALIDAD\_SELECT

FROM departments

WHERE deptno=p\_departamento;

DBMS\_OUTPUT.PUT\_LINE('Localidad actual: '||P\_LOCALIDAD\_SELECT);

EXECUTE IMMEDIATE 'ROLLBACK';

SELECT location

INTO P\_LOCALIDAD\_SELECT

FROM departments

WHERE deptno=p\_departamento;

DBMS\_OUTPUT.PUT\_LINE('Localidad actual: '||P\_LOCALIDAD\_SELECT);

END EJ17;

**---------RESULTADO DEL CÓDIGO-------**

bloque anónimo terminado

Localidad actual: NEW YORK

Localidad actual: MADRID

Localidad actual: NEW YORK

1. **Crear un procedimiento que reciba como parámetros el número de un empleado y, por referencia, se le pase un parámetro de tipo localidad del departamento del empleado. En la llamada ese parámetro irá sin valor. El procedimiento recogerá la localidad del departamento si la tuviere. En un bloque anónimo, llamaremos al procedimiento y evaluaremos dicha localidad, si tiene valor, lo sacaremos por pantalla, si no tiene nada, diremos que está vacío.**

**---------BLOQUE ANÓNIMO-----------**

SET SERVEROUTPUT ON;

DECLARE

V\_LOCALIDAD VARCHAR2(50);

BEGIN

EJ18(7369, V\_LOCALIDAD);

IF V\_LOCALIDAD IS NOT NULL THEN

DBMS\_OUTPUT.PUT\_LINE(V\_LOCALIDAD);

ELSE

DBMS\_OUTPUT.PUT\_LINE('La localidad del departamento tiene un valor vacío');

END IF;

END;

**---------PROCEDIMIENTO EJ18-------**

CREATE OR REPLACE PROCEDURE EJ18

(

P\_EMPNO IN NUMBER

, P\_LOCALIDAD OUT VARCHAR2

) AS

BEGIN

select d.location

into P\_LOCALIDAD

from departments d

join employees e

on d.deptno=e.deptno

where e.empno=P\_EMPNO;

END EJ18;

**---------RESULTADO DEL CÓDIGO-------**

bloque anónimo terminado

DALLAS

(Muestra la localidad del departamento 20, que es el departamento donde trabaja el empleado 7369, Smith)